УЧЕБНИК ДЛЯ ВУЗОВ

В. Г. Олифер, Н. А. Олифер

КОМПЬЮТЕРНЫЕ СЕТИ

3-е издание

ПРИНЦИПЫ, ТЕХНОЛОГИИ, ПРОТОКОЛЫ

- для студентов, аспирантов и технических специалистов, работающих в области сетевых технологий
- подробный обзор современных сетевых технологий и оборудования

ПИТЕР®

СЕРИЯ

УЧЕБНИК ДЛЯ ВУЗОВ

ПИТЕР®

УЧЕБНИК ДЛЯ ВУЗОВ

В. Г. Олифер, Н. А. Олифер

КОМПЬЮТЕРНЫЕ СЕТИ

ПРИНЦИПЫ, ТЕХНОЛОГИИ, ПРОТОКОЛЫ

3-Е ИЗДАНИЕ

Рекомендовано Министерством образования Российской Федерации
в качестве учебного пособия для студентов высших учебных заведений,
обучающихся по направлению 552800 — «Информатика
и вычислительная техника» и по специальностям
220100 — «Вычислительные машины, комплексы, системы и сети»,
220200 — «Автоматизированные системы обработки информации
и управления» и 220400 — «Программное обеспечение
вычислительной техники и автоматизированных систем»

ПИТЕР®

Москва · Санкт-Петербург · Нижний Новгород · Воронеж
Ростов-на-Дону · Екатеринбург · Самара · Новосибирск
Киев · Харьков · Минск
2006

ББК 32.973.202я22
УДК 004.72(075)
О-54

Рецензенты:

Кафедра «Вычислительная техника» факультета «Вычислительные машины и системы» Московского государственного института радиотехники, электроники и автоматики (Технического университета)

Григорьев Ю. А., доктор технических наук, профессор, кафедра «Системы обработки информации и управления» Московский государственный технический университет им. Н. Э. Баумана

Прижуков Б. Ф., кандидат технических наук, заместитель начальника ИВЦ ОАО Московский междугородный и международный телефон

Олифер В. Г., Олифер Н. А.

О-54 Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 3-е изд. — СПб.: Питер, 2006. — 958 с.: ил.

ISBN 5-469-00504-6

За время, прошедшее после выхода первых двух изданий, сетевой мир не стоял на месте, появлялись новые и совершенствовались существующие протоколы, типы оборудования. Все эти изменения потребовали радикального пересмотра некоторых разделов книги, хотя большая ее часть по-прежнему посвящена традиционным принципам организации сетей, фундаментальным понятиям и базовым, устоявшимся сетевым технологиям. Книга предназначена для студентов, аспирантов и технических специалистов, которые хотели бы получить базовые знания о принципах построения компьютерных сетей, понять особенности традиционных и перспективных технологий локальных и глобальных сетей, изучить способы создания крупных составных сетей и управления такими сетями.

Рекомендовано Министерством образования Российской Федерации в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлению «Информатика и вычислительная техника» и по специальностям «Вычислительные машины, комплексы, системы и сети», «Автоматизированные машины, комплексы, системы и сети», «Программное обеспечение вычислительной техники и автоматизированных систем».

ББК 32.973.202я22
УДК 004.72(075)

ISBN 5-469-00504-6

Краткое содержание

Содержание

Часть II. Технологии физического уровня

Часть III. Локальные сети

Посвящаем нашей дочери Анне

Благодарности

Прежде всего мы хотим поблагодарить наших читателей за их многочисленные пожелания, вопросы и замечания. Этот мощный поток читательских отзывов убеждает нас в востребованности книги и вдохновляет на продолжение работы над ней.

Мы признательны сотрудникам издательства «Питер», особенно ее главному редактору и одновременно самому главному редактору нашей книги Екатерине Строгановой, ее заместителю Евгении Журавлевой. Огромное спасибо нашему неизменному литературному редактору Алексею Жданову, который, помимо традиционного литературного редактирования, постоянно решал труднейшую проблему поиска русских эквивалентов для англоязычных сетевых терминов.

Мы также очень благодарны Ольге Кокоревой, с которой мы на протяжении последних двух лет работали над переводом этой книги на английский язык (в этом году учебник выходит в Великобритании в издательстве Wiley). Тщательность и скрупулезность, с которой Ольга работала над переводом, в значительной степени помогли улучшить и русский текст.

Мы благодарны рецензентам книги: сотрудникам кафедры «Вычислительная техника» МИРЭА, а также д. т. н., профессору МГТУ им. Н. Э. Баумана Григорьеву Ю. А. и к. т. н., заместителю начальника ИВЦ ОАО Московский ММТ Прижукову Б. Ф. за ценные советы и замечания, позволившие улучшить качество этой книги.

Виктор и Наталья Олифер

От авторов

Эта книга является результатом многолетнего опыта преподавания авторами курсов сетевой тематики в аудиториях государственных вузов, коммерческих учебных центров, а также учебных центров предприятий и корпораций.

Основу книги составили материалы курсов «Проблемы построения корпоративных сетей», «Основы сетевых технологий», «Организация удаленного доступа», «Сети TCP/IP», «Стратегическое планирование сетей масштаба предприятия» и ряда других. Эти материалы прошли успешную проверку в бескомпромиссной и сложной аудитории, состоящей из слушателей с существенно разным уровнем подготовки и кругом профессиональных интересов. Среди них были студенты и аспиранты вузов, сетевые администраторы и интеграторы, начальники отделов автоматизации и преподаватели. Учитывая специфику аудитории, курсы лекций строились так, чтобы начинающий получил основу для дальнейшего изучения, а специалист систематизировал и актуализировал свои знания. В соответствии с такими же принципами написана и эта книга — она является фундаментальным курсом по компьютерным сетям, который сочетает широту охвата основных областей, проблем и технологий этой быстро развивающейся области знаний с основательным рассмотрением деталей каждой технологии и особенностей оборудования.

Для кого эта книга

Книга предназначена для студентов, аспирантов и технических специалистов, которые хотят получить базовые знания о принципах построения компьютерных сетей, понять особенности традиционных и перспективных технологий локальных и глобальных сетей, изучить способы создания крупных составных сетей и управления такими сетями.

Книга будет полезна начинающим специалистам в области сетевых технологий, имеющим только общие представления о работе сетей из опыта общения с персональными компьютерами и Интернетом, но желающим получить фундаментальные знания, позволяющие продолжить изучение сетей самостоятельно.

Сложившимся сетевым специалистам книга может помочь в знакомстве с теми технологиями, с которыми им не приходилось сталкиваться в практической работе, систематизировать имеющиеся знания, стать справочным средством, позволяющим найти описание конкретного протокола, формата кадра и т. п. Книга

также дает необходимую теоретическую основу для подготовки к сертификационным экзаменам таких компаний, как Cisco: CCNA, CCNP, CCDP и CCIP.

Студенты высших учебных заведений, обучающиеся по направлению «220000. Информатика и вычислительная техника» и по специальностям «Вычислительные машины, комплексы, системы и сети», «Автоматизированные машины, комплексы, системы и сети», «Программное обеспечение вычислительной техники и автоматизированных систем» могут использовать книгу в качестве рекомендованного Министерством образования Российской Федерации учебного пособия.

Изменения в третьем издании

В третье издание книги были включены следующие новые темы:

Беспроводные технологии и сети. Отсутствие этой темы было достаточно серьезным недостатком первого и второго изданий книги, который теперь ликвидирован. Рассматриваются как общие принципы беспроводной связи, так и конкретные технологии этого класса, такие как IEEE 802.11, Bluetooth, Wireless Local Loop.

Существенно расширено описание технологии TCP/IP и смежных технологий: IPSec, IPv6. Эти изменения соответствуют новой роли IP в современном телекоммуникационном мире.

Изменен подход к описанию QoS — теперь эта тема пронизывает все книгу, так что общие принципы обеспечения качества обслуживания в пакетных сетях, описанные в отдельной главе первой части, рассматриваются более конкретно при изучении основных транспортных технологий, таких как коммутируемый Ethernet, IP, Frame Relay, ATM и MPLS.

Заново, более подробно, написаны разделы, посвященные VPN, в частности детально описана быстро набирающая популярность технология MPLS VPN.

Добавлено много примеров, рисунков и задач.

Как видим, формально содержание учебника не претерпело кардинальных изменений. Фактически же большая часть книги переписана заново. За 6 лет, прошедших с момента выхода в свет первого издания книги, изменились не только технологии компьютерных сетей, но и взгляды авторов на многие фундаментальные проблемы телекоммуникаций. Результатом этого переосмысления и стало третье издание учебника.

И, наконец, было исправлено много мелких ошибок и опечаток в тексте и рисунках, замеченных читателями и самими авторами.

Структура книги

Книга состоит из 24 глав, объединенных в 5 частей.

- Первую часть, «Основы сетей передачи данных», можно представить «первым витком» изучения компьютерных сетей. Известно, что процесс познания

всегда развивается по спирали. Мы не можем сразу понять и осознать сложное явление, мы должны рассматривать его с разных точек зрения, в целом и по частям, изолированно и во взаимодействии с другими явлениями, накапливая знания постепенно, время от времени возвращаясь к уже казалось бы понятому и с каждым новым витком все больше проникая в суть явления. В первой части, состоящей из 7 глав, описаны основные принципы и архитектурные решения, которые лежат в основе всех современных сетевых технологий, рассматриваемых в последующих частях книги. Следуя процессу конвергенции сетей, мы рассматривали принципы коммутации, мультиплексирования, маршрутизации, адресации и архитектуры сетей с общих позиций, сравнивая принципы организации компьютерных сетей с аналогичными принципами других телекоммуникационных сетей — телефонных, первичных, радио и телевизионных. Завершает часть глава, посвященная проблемам качества обслуживания в пакетных сетях. Таким образом, концепции качества обслуживания, которые достаточно долго рассматривались как нетривиальное направление сетевой отрасли, вошли в число базовых принципов построения компьютерных сетей.

- Вторая часть, «Технологии физического уровня», состоит из четырех глав: «Линии связи», «Кодирование и мультиплексирование данных», «Беспроводная передача данных» и «Первичные сети», из которых первые две носят вспомогательный характер. В них описываются различные типы линий связи, детально излагаются современные методы передачи дискретной информации в сетях. Наличие этого материала в учебнике дает возможность читателю, не тратя время на просмотр большого количества литературы, получить необходимый минимум знаний в таких областях, как теория информации, спектральный анализ, физическое и логическое кодирование данных, обнаружение и коррекция ошибок. В главе 11 изучаются технологии PDH, SDH/SONET, DWDM, создающие инфраструктуру физических каналов для глобальных телекоммуникационных сетей. На основе каналов, образованных первичными сетями, работают наложенные компьютерные или телефонные сети. Глава 10 посвящена беспроводной передаче данных, которая приобретает все большую популярность. Высокий уровень помех и сложные пути распространения волн требуют применения в беспроводных каналах особых способов кодирования и передачи сигналов.

- В третьей части, «Локальные сети», даны подробные описания практически всех основных технологий локальных сетей, как традиционных, таких как Ethernet, Token Ring и FDDI, так и новых высокоскоростных. Современные локальные сети существуют в условиях одной победившей технологии — Ethernet, точнее — семейства технологий Ethernet. Естественно, что этой технологии в книге уделяется значительно большее внимание, чем остальным. В главе 12 рассматривается классическая технология Ethernet 10 Мбит/с на разделяемой среде, а в главе 13 — высокоскоростные версии Ethernet на разделяемой среде — Fast Ethernet и Gigabit Ethernet. В главе 14 описаны другие технологии LAN, также использующие разделяемую среду, — Token Ring, FDDI, а также две беспроводные технологии — IEEE 802.11 LAN и Bluetooth PAN. Последние две главы этой части (15 и 16) посвящены коммутируемым

локальным сетям. В первой из них рассматриваются основные принципы работы таких сетей: алгоритм работы коммутатора локальной сети, дуплексные версии LAN-протоколов, особенности реализации LAN-коммутаторов. В главе 16 изучаются расширенные возможности сетей этого типа: резервные связи на основе алгоритма покрывающего дерева, агрегирование каналов, техника виртуальных локальных сетей (VLAN).

- Следуя логике, диктуемой моделью OSI, вслед за частями, в которых были рассмотрены технологии физического и канального уровней, четвертую часть мы посвящаем средствам сетевого уровня, то есть средствам, которые обеспечивают возможность объединения множества сетей в единую сеть. Учитывая, что бесспорным лидером среди протоколов сетевого уровня является протокол IP, ему в книге уделяется основное внимание. В главе 17 описываются различные аспекты IP-адресации: способы отображения локальных, сетевых и символьных адресов, использование масок и современных методов агрегирования IP-адресов, а также способы автоматического конфигурирования IP-узлов. В главе 18 детально рассмотрена работа протокола IP по продвижению и фрагментации пакетов, изучается общий формат таблицы маршрутизации и примеры ее частных реализаций в программных и аппаратных маршрутизаторах различных типов. При обсуждении особенностей новой версии IPv6 подробно рассматривается схема модернизации адресации, а также изменение формата IP-заголовка. Глава 19 начинается с изучения протоколов TCP и UDP, исполняющих посредническую роль между приложениями и транспортной инфраструктурой сети. Далее подробно описываются протоколы маршрутизации RIP, OSPF и BGP, анализируются области применения этих протоколов и возможности их комбинирования. Завершает главу рассмотрение протокола ICMP, являющегося средством оповещения отправителя о причинах недоставки его пакетов адресату. В главе 20 описываются типы и основные характеристики маршрутизаторов, варианты их внутренней организации, методы комбинирования функций коммутации и маршрутизации в одном устройстве — коммутаторе 3-го уровня. Всестороннее изучение в четвертой части протоколов стека TCP/IP придает ей самостоятельное значение введения в сети IP.

- Пятая часть, «Технологии глобальных сетей», состоит из 4 глав. Технология IP, которая рассматривалась в предыдущей части книги, позволяет строить составные сети различного типа, как локальные, так и глобальные. Существуют также технологии, основанные на технике виртуальных каналов, которые были разработаны специально для глобальных сетей. Этим технологиям, реализованным в активно применяемых сегодня сетях Frame Relay и ATM, посвящена глава 21. Техника виртуальных каналов является альтернативой дейтаграммному способу продвижения пакетов, на котором основаны сети Ethernet и IP. Конкуренция между этими двумя базовыми принципами передачи данных существует давно, практически с момента появления сетей с коммутацией пакетов. Вопросы использования технологии IP для построения глобальных сетей рассматриваются в главе 22. Новым словом в области интеграции IP с технологиями виртуальных каналов является технология MPLS. Эта технология занимает промежуточное место между уровнем IP и уровнем таких техноло-

гий, как ATM, FR или Ethernet, объединяя их в единую эффективную транспортную систему. Глава завершается описанием систем сетевого управления, базирующихся на протоколе SNMP, который широко используется для управления не только IP-маршрутизаторами (для чего этот протокол был создан), но и телекоммуникационными устройствами разных типов. В главе 23 изучаются различные подходы к организации высокоскоростного доступа пользователей к сетевой магистрали. Наиболее эффективными являются технологии, опирающиеся на существующую кабельную инфраструктуру (например, линии ADSL, работающие на абонентских окончаниях телефонной сети) или на кабельные модемы, использующиеся в системах кабельного телевидения. Альтернативным решением является беспроводной доступ, как мобильный, так и фиксированный. Часть, а вместе с ней и книга, завершается главой 24, которая посвящена безопасности транспортной системы сети. Здесь рассматриваются различные типы виртуальных частных сетей (VPN), в частности, подробно описываются сети, построенные на основе защищенного протокола IP (IPSec), а также одна из наиболее популярных современных технологий виртуальных частных сетей — технология MPLS VPN.

Авторы стремились сделать работу читателя с книгой максимально эффективной. Подробный индексный указатель позволяет быстро найти интересующий материал по одному из многочисленных терминов, используемых в сетевой индустрии. Каждая глава завершается выводами, которые призваны сконцентрировать внимание читателя на основных идеях, темах и результатах главы, помогая ему не упустить из виду главное за обилием, хотя и полезных, но частных фактов и деталей. В конце каждой главы помещены вопросы и упражнения для проверки степени усвоения основных концепций, а в отдельных случаях и для углубления понимания некоторых идей.

Веб-сайт

Дополнительную информацию читатели могут найти на сайте авторов www.olifer.co.uk. Авторы намерены сделать этот сайт дополнением к данному учебнику, который, как мы надеемся, окажется полезным и для студентов, и для преподавателей, и для специалистов. Понятно, что работа над сайтом является непрерывной. Первоначально авторы планируют разместить на нем следующие материалы:

- Все иллюстрации из книги.
- Презентации в форматах Power Point и HTML последовательно по всем главам книги.
- Ответы на вопросы, помещенные в конце каждой главы учебника.
- Путеводитель по книге, который призван помочь преподавателю при создании учебных курсов на базе этой книги, таких, например, как «Беспроводные системы», «Введение в IP», «Качество обслуживания», «Удаленный доступ» и т. п. В этом путеводителе авторы перечисляют последовательность глав

(маршрут), в которых содержится соответствующий материал, и при необходимости, дают методические советы.

- Дополнительные примеры, которые могут быть использованы как темы для курсовых проектов.
- Информационные ресурсы Интернета, связанные с темами книги.
- И, наконец, мнения, замечания и вопросы читателей, замеченные опечатки и ошибки.

Мы с благодарностью примем ваши отзывы по адресам

victor@olifer.co.uk и natalia@olifer.co.uk.

Виктор Олифер, к. т. н., ССІР
Наталья Олифер, к. т. н., доцент

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу электронной почты comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства http://www.piter.com вы найдете подробную информацию о наших книгах.

Часть I

Основы сетей передачи данных

Процесс познания всегда развивается по спирали. Мы не можем сразу понять и осознать сложное явление, мы должны рассматривать его с разных точек зрения, в целом и по частям, изолированно и во взаимодействии с другими явлениями, накапливая знания постепенно, время от времени возвращаясь к уже казалось бы понятому и с каждым новым витком все больше проникая в суть явления. Хорошим подходом является первоначальное изучение общих принципов некоторой области знаний с последующим детальным рассмотрением реализации этих принципов в конкретных методах, технологиях или конструкциях.

Первая часть книги является таким «первым витком» изучения компьютерных сетей. В этой части, состоящей из семи глав, описаны основные принципы и архитектурные решения, которые лежат в основе всех современных сетевых технологий, рассматриваемых в последующих частях книги. Следуя процессу конвергенции сетей, мы рассматривали принципы коммутации, мультиплексирования, маршрутизации, адресации и архитектуры сетей с наиболее общих позиций, сравнивая принципы организации компьютерных сетей с аналогичными принципами других телекоммуникационных сетей — телефонных, первичных, радио и телевизионных.

Завершает часть глава, посвященная проблемам качества обслуживания в пакетных сетях. Новая роль компьютерных сетей как основы для создания следующего поколения публичных сетей, предоставляющих все виды информационных услуг и переносящих данные, а также аудио- и видеотрафик, привела к проникновению методов качества обслуживания практически во все коммуникационные технологии. Таким образом, концепции качества обслуживания, которые достаточно долго рассматривались как нетривиальное направление сетевой отрасли, вошли в число базовых принципов построения компьютерных сетей.

- Глава 1. Эволюция компьютерных сетей
- Глава 2. Общие принципы построения сетей
- Глава 3. Коммутация пакетов и каналов
- Глава 4. Архитектура и стандартизация сетей
- Глава 5. Примеры сетей
- Глава 6. Сетевые характеристики
- Глава 7. Методы обеспечения качества обслуживания

ГЛАВА 1 Эволюция компьютерных сетей

История любой отрасли науки и техники позволяет не только удовлетворить естественное любопытство, но и глубже понять сущность основных достижений в этой отрасли, осознать существующие тенденции и правильно оценить перспективность тех или иных направлений развития. Компьютерные сети появились сравнительно недавно, в конце 60-х годов (правда, уточнение «прошлого столетия» прибавляет им вес и даже делает старше своих «тридцати с чем-то» лет). Естественно, что компьютерные сети унаследовали много полезных свойств от других, более старых и распространенных телекоммуникационных сетей, а именно телефонных. В этом нет ничего удивительного, так как компьютер, как и телефон, является универсальным инструментом в руках своего хозяина и помогает ему общаться с друзьями, приобретать новых знакомых, удовлетворять любознательность и любопытство, делать покупки и т. д., и т. п.

В то же время компьютерные сети привнесли в телекоммуникационный мир нечто совершенно новое — неисчерпаемые запасы информации, созданные цивилизацией за несколько тысячелетий своего существования и продолжающие пополняться с растущей скоростью в наши дни. Этот эффект особенно проявился в середине 90-х, во время интернет-революции, когда стало ясно, что возможности свободного и анонимного доступа к информации и быстрому, хотя и письменному общению очень ценятся людьми.

Результатом влияния компьютерных сетей на остальные типы телекоммуникационных сетей стал процесс их конвергенции. Этот процесс начался достаточно давно, одним из первых признаков сближения стала передача телефонными сетями голоса в цифровой форме. Компьютерные сети также активно идут навстречу телекоммуникационным сетям, разрабатывая новые сервисы, которые ранее были прерогативой телефонных, радио и телевизионных сетей — сервисы IP-телефонии, радио- и видеовещания, ряд других. Процесс конвергенции продолжается, и о том, каким будет его конечный результат, с уверенностью пока говорить рано. Однако понимание истории развития сетей, описываемой в данной главе, делает более ясными основные проблемы, стоящие перед разработчиками компьютерных сетей.

Два корня компьютерных сетей

Список ключевых слов: компьютерная сеть, или сеть передачи данных, система пакетной обработки, многотерминальная система разделения времени, закон Гроша.

Вычислительная и телекоммуникационная технологии

Компьютерные сети, которым посвящена данная книга, отнюдь не являются единственным видом сетей, созданным человеческой цивилизацией. Даже водопроводы Древнего Рима можно рассматривать как один из наиболее древних примеров сетей, покрывающих большие территории и обслуживающих многочисленных клиентов. Другой, менее экзотический пример — электрические сети. В них легко можно найти аналоги компонентов любой территориальной компьютерной сети: источникам информационных ресурсов соответствуют электростанции, магистралям — высоковольтные линии электропередач, сетям доступа — трансформаторные подстанции, клиентским терминалам — осветительные и бытовые электроприборы.

Компьютерные сети, называемые также **сетями передачи данных**, являются логическим результатом эволюции двух важнейших научно-технических отраслей современной цивилизации — компьютерных и телекоммуникационных технологий.

С одной стороны, сети представляют собой частный случай распределенных вычислительных систем, в которых группа компьютеров согласованно выполняет набор взаимосвязанных задач, обмениваясь данными в автоматическом режиме. С другой стороны, компьютерные сети могут рассматриваться как средство передачи информации на большие расстояния, для чего в них применяются методы кодирования и мультиплексирования данных, получившие развитие в различных телекоммуникационных системах (рис. 1.1).

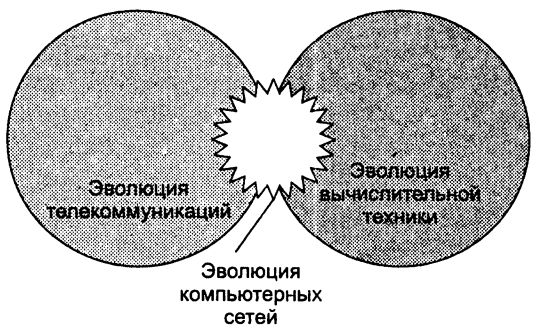


Рис. 1.1. Эволюция компьютерных сетей на стыке вычислительной техники и телекоммуникационных технологий

Системы пакетной обработки

Обратимся сначала к компьютерному корню вычислительных сетей. Первые компьютеры 50-х годов — большие, громоздкие и дорогие — предназначались для очень небольшого числа избранных пользователей. Часто эти монстры занимали целые здания. Такие компьютеры не были предназначены для интерактивной работы пользователя, а использовались в режиме пакетной обработки.

Системы пакетной обработки, как правило, строились на базе мэйнфрейма — мощного и надежного компьютера универсального назначения. Пользователи подготавливали перфокарты, содержащие данные и команды программ, и передавали их в вычислительный центр (рис. 1.2). Операторы вводили эти карты в компьютер, а распечатанные результаты пользователи получали обычно только на следующий день. Таким образом, одна неверно набитая карта означала как минимум суточную задержку. Конечно, для пользователей интерактивный режим работы, при котором можно с терминала оперативно руководить процессом обработки своих данных, был бы удобней. Но интересами пользователей на первых этапах развития вычислительных систем в значительной степени пренебрегали. Во главу угла ставилась эффективность работы самого дорогого устройства вычислительной машины — процессора, даже в ущерб эффективности работы использующих его специалистов.

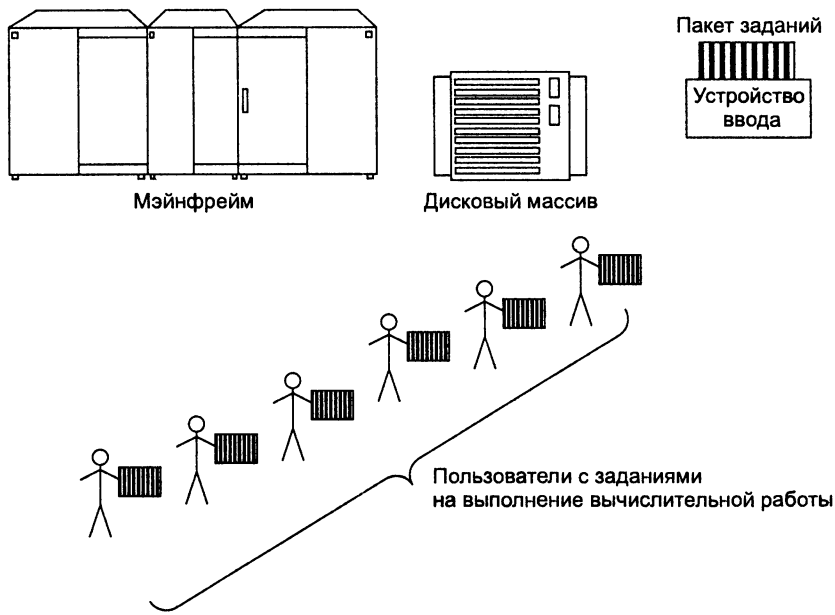


Рис. 1.2. Централизованная система на базе мэйнфрейма

Многотерминальные системы — прообраз сети

По мере удешевления процессоров в начале 60-х годов появились новые способы организации вычислительного процесса, которые позволили учесть интересы

пользователей. Начали развиваться интерактивные **многотерминальные системы разделения времени** (рис. 1.3). В таких системах каждый пользователь получал собственный терминал, с помощью которого он мог вести диалог с компьютером. Количество одновременно работающих с компьютером пользователей зависело от его мощности так, чтобы время реакции вычислительной системы было достаточно мало, и пользователю была не слишком заметна параллельная работа с компьютером других пользователей.

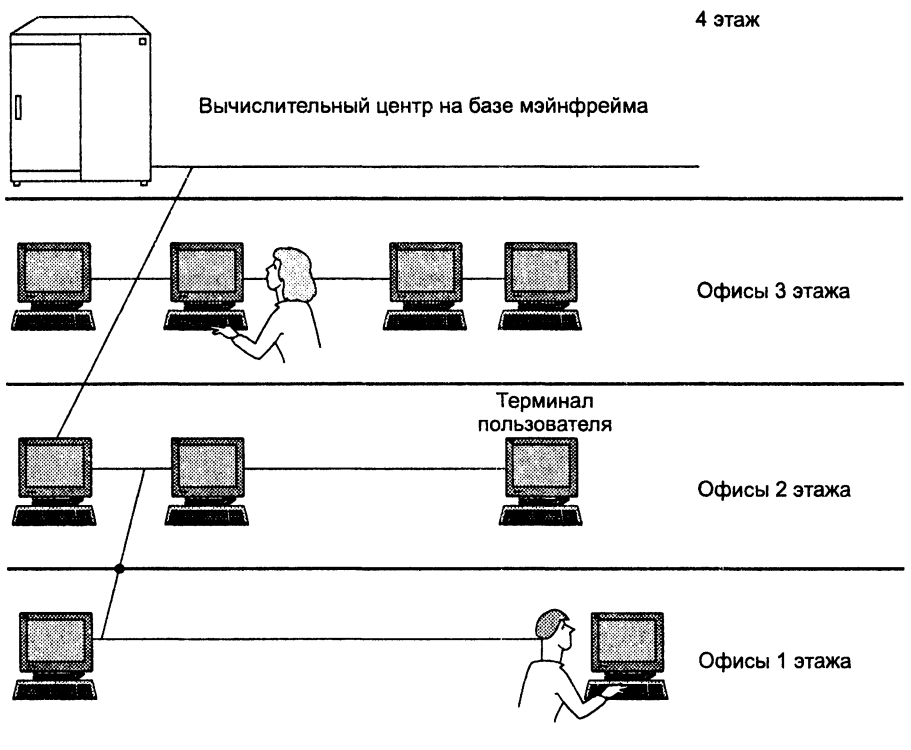


Рис. 1.3. Многотерминальная система — прообраз вычислительной сети

Терминалы, выйдя за пределы вычислительного центра, рассредоточились по всему предприятию. И хотя вычислительная мощность оставалась полностью централизованной, некоторые функции — такие как ввод и вывод данных — стали распределенными. Подобные многотерминальные централизованные системы внешне уже были очень похожи на локальные вычислительные сети. Действительно, рядовой пользователь работу за терминалом мэйнфрейма воспринимал примерно так же, как сейчас он воспринимает работу за подключенным к сети персональным компьютером. Пользователь мог получить доступ к общим файлам и периферийным устройствам, при этом у него поддерживалась полная иллюзия единоличного владения компьютером, так как он мог запустить нужную ему программу в любой момент и почти сразу же получить результат. (Некото-

рые, далекие от вычислительной техники пользователи даже были уверены, что все вычисления выполняются внутри их дисплея.)

Многотерминальные системы, работающие в режиме разделения времени, стали первым шагом на пути создания локальных вычислительных сетей.

Однако до появления локальных сетей нужно было пройти еще большой путь, так как многотерминальные системы, хотя и имели внешние черты распределенных систем, все еще поддерживали централизованную обработку данных.

С другой стороны, и потребность предприятий в создании локальных сетей в это время еще не созрела — в одном здании просто нечего было объединять в сеть, так как из-за высокой стоимости вычислительной техники предприятия не могли себе позволить роскошь приобретения нескольких компьютеров. В этот период был справедлив так называемый **закон Гроша**, который эмпирически отражал уровень технологии того времени. В соответствии с этим законом производительность компьютера была пропорциональна квадрату его стоимости, отсюда следовало, что за одну и ту же сумму было выгоднее купить одну мощную машину, чем две менее мощных — их суммарная мощность оказывалась намного ниже мощности дорогой машины.

Первые компьютерные сети

Список ключевых слов: глобальная сеть, сетевая операционная система, первичная, или опорная, сеть, наложенная сеть, большая интегральная схема, локальная сеть, сетевая технология, стандартная сетевая технология, персональный компьютер.

Первые глобальные сети

А вот потребность в соединении компьютеров, находящихся на большом расстоянии друг от друга, к этому времени вполне назрела. Началось все с решения более простой задачи — доступа к компьютеру с терминалов, удаленных от него на многие сотни, а то и тысячи километров. Терминалы соединялись с компьютерами через телефонные сети с помощью модемов. Такие сети позволяли многочисленным пользователям получать удаленный доступ к разделяемым ресурсам нескольких мощных компьютеров класса суперЭВМ. Затем появились системы, в которых наряду с удаленными соединениями типа *терминал–компьютер* были реализованы и удаленные связи типа *компьютер–компьютер*.

Компьютеры получили возможность обмениваться данными в автоматическом режиме, что, собственно, и является базовым признаком любой вычислительной сети. На основе подобного механизма в первых сетях были реализованы службы обмена файлами, синхронизации баз данных, электронной почты и другие, ставшие теперь традиционными сетевые службы.

Итак, хронологически первыми появились **глобальные сети** (Wide Area Networks, WAN), то есть сети, объединяющие территориально рассредоточенные компьютеры, возможно находящиеся в различных городах и странах.

Именно при построении глобальных сетей были впервые предложены и отработаны многие основные идеи, лежащие в основе современных вычислительных сетей. Такие, например, как многоуровневое построение коммуникационных протоколов, концепции коммутации и маршрутизации пакетов.

Глобальные компьютерные сети очень многое унаследовали от других, гораздо более старых и распространенных глобальных сетей — *телефонных*. Главное технологическое новшество, которое привнесли с собой первые глобальные компьютерные сети, состояло в отказе от принципа коммутации каналов, на протяжении многих десятков лет успешно использовавшегося в телефонных сетях.

Выделяемый на все время сеанса связи составной телефонный канал, передающий информацию с постоянной скоростью, не мог эффективно использоваться пульсирующим трафиком компьютерных данных, у которого периоды интенсивного обмена чередуются с продолжительными паузами. Натурные эксперименты и математическое моделирование показали, что пульсирующий и в значительной степени не чувствительный к задержкам компьютерный трафик гораздо эффективней передается сетями, работающими по принципу коммутации пакетов, когда данные разделяются на небольшие порции — пакеты, — которые самостоятельно перемещаются по сети благодаря наличию адреса конечного узла в заголовке пакета.

Так как прокладка высококачественных линий связи на большие расстояния обходится очень дорого, то в первых глобальных сетях часто использовались уже существующие каналы связи, изначально предназначенные совсем для других целей. Например, в течение многих лет глобальные сети строились на основе телефонных каналов тональной частоты, способных в каждый момент времени вести передачу только одного разговора в аналоговой форме. Поскольку скорость передачи дискретных компьютерных данных по таким каналам была очень низкой (десятки килобит в секунду), набор предоставляемых услуг в глобальных сетях такого типа обычно ограничивался передачей файлов, преимущественно в фоновом режиме, и электронной почтой. Помимо низкой скорости такие каналы имеют и другой недостаток — они вносят значительные искажения в передаваемые сигналы. Поэтому протоколы глобальных сетей, построенных с использованием каналов связи низкого качества, отличаются сложными процедурами контроля и восстановления данных. Типичным примером таких сетей являются сети X.25, разработанные еще в начале 70-х, когда низкоскоростные аналоговые каналы, арендуемые у телефонных компаний, были преобладающим типом каналов, соединяющих компьютеры и коммутаторы глобальной вычислительной сети.

В 1969 году министерство обороны США инициировало работы по объединению в единую сеть суперкомпьютеров оборонных и научно-исследовательских центров. Эта сеть, получившая название ARPANET, стала отправной точкой для создания первой и самой известной ныне глобальной сети — Internet.

Сеть ARPANET объединяла компьютеры разных типов, работавшие под управлением различных ОС с дополнительными модулями, реализующими коммуникационные протоколы, общие для всех компьютеров сети. ОС этих компьютеров можно считать *первыми* **сетевыми операционными системами**.

Истинно сетевые ОС в отличие от многотерминальных ОС позволяли не только рассредоточить пользователей, но и организовать распределенные хранение и обработку данных между несколькими компьютерами, связанными электрическими связями. Любая сетевая операционная система, с одной стороны, выполняет все функции локальной операционной системы, а с другой стороны, обладает некоторыми дополнительными средствами, позволяющими ей взаимодействовать через сеть с операционными системами других компьютеров. Программные модули, реализующие сетевые функции, появлялись в операционных системах постепенно, по мере развития сетевых технологий, аппаратной базы компьютеров и возникновения новых задач, требующих сетевой обработки.

Прогресс глобальных компьютерных сетей во многом определялся прогрессом телефонных сетей.

С конца 60-х годов в телефонных сетях все чаще стала применяться передача голоса в цифровой форме.

Это привело к появлению высокоскоростных цифровых каналов, соединяющих автоматические телефонные станции (АТС) и позволяющих одновременно передавать десятки и сотни разговоров. Была разработана специальная технология для создания так называемых **первичных**, или **опорных**, **сетей**. Такие сети не предоставляют услуг конечным пользователям, они являются фундаментом, на котором строятся скоростные цифровые каналы «точка-точка», соединяющие оборудование других, так называемых **наложенных сетей**, которые уже работают на конечного пользователя.

Сначала технология первичных сетей была исключительно внутренней технологией телефонных компаний. Однако со временем эти компании стали сдавать часть своих цифровых каналов, образованных в первичных сетях, в аренду предприятиям, которые использовали их для создания собственных телефонных и глобальных компьютерных сетей. Сегодня первичные сети обеспечивают скорости передачи данных до сотен гигабит (а в некоторых случаях до нескольких терабит) в секунду и густо покрывают территории всех развитых стран.

К настоящему времени глобальные сети по разнообразию и качеству предоставляемых услуг догнали локальные сети, которые долгое время лидировали в этом отношении, хотя и появились на свет значительно позже.

Первые локальные сети

Важное событие, повлиявшее на эволюцию компьютерных сетей, произошло в начале 70-х годов. В результате технологического прорыва в области производства компьютерных компонентов появились **большие интегральные схемы** (БИС). Их сравнительно невысокая стоимость и хорошие функциональные воз-

можности привели к созданию мини-компьютеров, которые стали реальными конкурентами мэйнфреймов. Эмпирический закон Гроша перестал соответствовать действительности, так как десяток мини-компьютеров, имея ту же стоимость, что и мэйнфрейм, решали некоторые задачи (как правило, хорошо распараллеливаемые) быстрее.

Даже небольшие подразделения предприятий получили возможность иметь собственные компьютеры. Мини-компьютеры решали задачи управления технологическим оборудованием, складом и другие задачи уровня отдела предприятия. Таким образом, появилась концепция распределения компьютерных ресурсов по всему предприятию. Однако при этом все компьютеры одной организации попрежнему продолжали работать *автономно* (рис. 1.4).

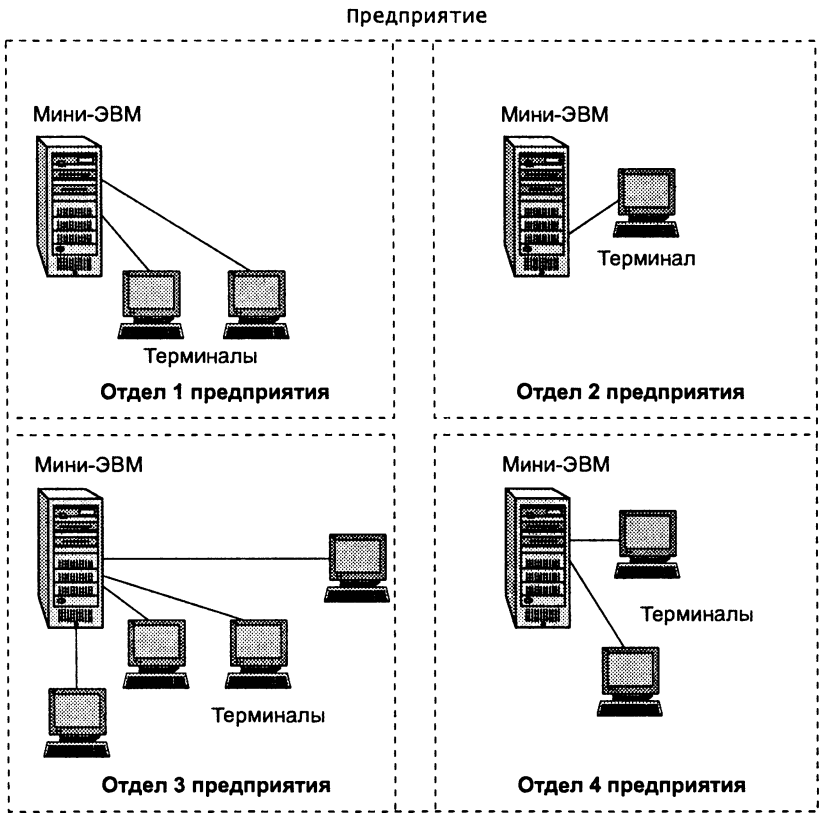


Рис. 1.4. Автономное использование нескольких мини-компьютеров на одном предприятии

Шло время, и потребности пользователей вычислительной техники росли. Их уже не удовлетворяла изолированная работа на собственном компьютере, им хотелось в автоматическом режиме обмениваться компьютерными данными с пользователями других подразделений. Ответом на эту потребность стало появление первых локальных вычислительных сетей (рис. 1.5).

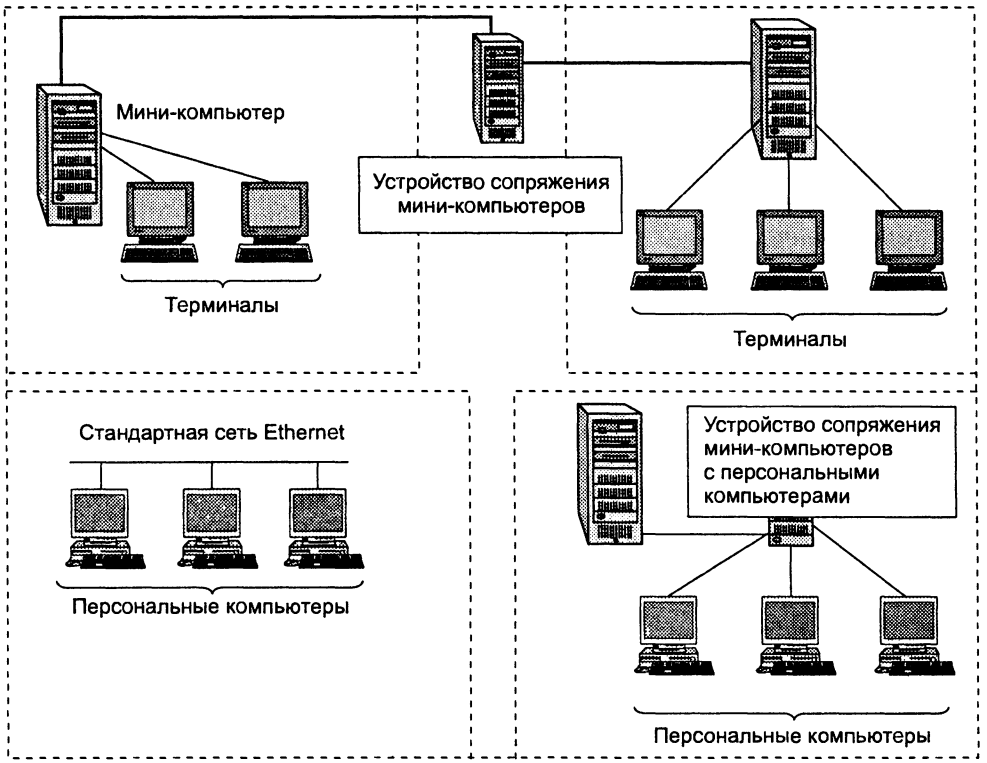


Рис. 1.5. Различные типы связей в первых локальных сетях

Локальные сети (Local Area Networks, LAN) — это объединения компьютеров, сосредоточенных на небольшой территории, обычно в радиусе не более 1–2 км, хотя в отдельных случаях локальная сеть может иметь и более протяженные размеры, например, несколько десятков километров. В общем случае локальная сеть представляет собой коммуникационную систему, принадлежащую одной организации.

На первых порах для соединения компьютеров друг с другом использовались нестандартные сетевые технологии.

Сетевая технология — это согласованный набор программных и аппаратных средств (например, драйверов, сетевых адаптеров, кабелей и разъемов) и механизмов передачи данных по линиям связи, достаточный для построения вычислительной сети.

Разнообразные устройства сопряжения, использующие собственные способы представления данных на линиях связи, свои типы кабелей и т. п., могли соединять только те конкретные модели компьютеров, для которых были разработаны, например, мини-компьютеры PDP-11 с мэйнфреймом IBM 360 или мини-компьютеры HP с микрокомпьютерами LSI-11. Такая ситуация создала большой простор для творчества студентов — названия многих курсовых и дипломных проектов начинались тогда со слов «Устройство сопряжения...».

В середине 80-х годов положение дел в локальных сетях кардинально изменилось. Утвердились **стандартные сетевые технологии** объединения компьютеров в сеть — Ethernet, ArcNet, Token Ring, Token Bus, несколько позже — FDDI.

Мощным стимулом для их появления послужили **персональные компьютеры**. Эти массовые продукты стали идеальными элементами для построения сетей — с одной стороны, они были достаточно мощными, чтобы обеспечивать работу сетевого программного обеспечения, а с другой — явно нуждались в объединении своей вычислительной мощности для решения сложных задач, а также разделения дорогих периферийных устройств и дисковых массивов. Поэтому персональные компьютеры стали преобладать в локальных сетях, причем не только в качестве клиентских компьютеров, но и в качестве центров хранения и обработки данных, то есть сетевых серверов, потеснив с этих привычных ролей мини-компьютеры и мэйнфреймы.

Все стандартные технологии локальных сетей опирались на тот же принцип коммутации, который был с успехом опробован и доказал свои преимущества при передаче трафика данных в глобальных компьютерных сетях, — принцип коммутации пакетов.

Стандартные сетевые технологии превратили процесс построения локальной сети из искусства в рутинную работу. Для создания сети достаточно было приобрести стандартный кабель, сетевые адаптеры соответствующего стандарта, например Ethernet, вставить адаптеры в компьютеры, присоединить их к кабелю стандартными разъемами и установить на компьютеры одну из популярных сетевых операционных систем, например Novell NetWare.

Разработчики локальных сетей привнесли много нового в организацию работы пользователей. Так, стало намного проще и удобнее, чем в глобальных сетях, получать доступ к общим сетевым ресурсам — в отличие от глобальной в локальной сети пользователь освобождается от запоминания сложных идентификаторов разделяемых ресурсов. Для этих целей система предоставляет ему список ресурсов в удобной для восприятия форме, например в виде древовидной иерархической структуры («дерева» ресурсов). Еще один прием, рационализирующий работу пользователя в локальной сети, состоит в том, что после соединения с удаленным ресурсом пользователь получает возможность обращаться к нему с помощью тех же команд, что и для работы с локальными ресурсами. Последствием и одновременно движущей силой такого прогресса стало появление огромного числа непрофессиональных пользователей, освобожденных от необходимости изучать специальные (и достаточно сложные) команды для сетевой работы.

Может возникнуть вопрос — почему все эти удобства пользователи получили только с появлением локальных сетей? Главным образом, это связано с использованием в локальных сетях качественных кабельных линий связи, на которых даже сетевые адаптеры первого поколения обеспечивали скорость передачи данных до 10 Мбит/с. При небольшой протяженности, свойственной локальным сетям, стоимость таких линий связи была вполне приемлемой. Поэтому экономное расходование пропускной способности каналов, одна из основных задач технологий первых глобальных сетей, никогда не выходило на первый план при разра-

ботке протоколов локальных сетей. В таких условиях основным механизмом прозрачного доступа к ресурсам локальных сетей стали периодические широковещательные объявления серверов о своих ресурсах и услугах. На основании таких объявлений клиентские компьютеры составляли списки имеющихся в сети ресурсов и предоставляли их пользователю.

Конец 90-х выявил явного лидера среди технологий локальных сетей — семейство Ethernet, в которое вошли классическая технология Ethernet 10 Мбит/с, а также Fast Ethernet 100 Мбит/с и Gigabit Ethernet 1000 Мбит/с.

Простые алгоритмы работы предопределили низкую стоимость оборудования Ethernet. Широкий диапазон иерархии скоростей позволяет рационально строить локальную сеть, выбирая ту технологию семейства, которая в наибольшей степени отвечает задачам предприятия и потребностям пользователей. Важно также, что все технологии Ethernet очень близки друг к другу по принципам работы, что упрощает обслуживание и интеграцию этих сетей.

Хронологическую последовательность важнейших событий, ставших историческими вехами на пути появления первых компьютерных сетей, иллюстрирует табл. 1.1.

Таблица 1.1. Хронология важнейших событий на пути появления первых компьютерных сетей

Этап	Время
Первые глобальные связи компьютеров, первые эксперименты с пакетными сетями	Конец 60-х
Начало передач по телефонным сетям голоса в цифровой форме	Конец 60-х
Появление больших интегральных схем, первые миникомпьютеры. Первые нестандартные локальные сети	Начало 70-х
Создание сетевой архитектуры IBM SNA	1974
Стандартизация технологии X.25	1974
Появление персональных компьютеров, создание Интернета в современном виде, установка на всех узлах стека TCP/IP	Начало 80-х
Появление стандартных технологий локальных сетей (Ethernet — 1980 г., Token Ring — 1985 г., FDDI — 1985 г.)	Середина 80-х
Начало коммерческого использования Интернета	Конец 80-х
Изобретение Web	1991

Конвергенция сетей

Список ключевых слов: intranet-технологии, городская сеть, или сеть мегаполиса, мультисервисная сеть, сеть передачи данных, цифровая сеть с интегрированным обслуживанием, мультисервисная сеть нового поколения, интеллектуальная сеть, качество обслуживания, инфокоммуникационная сеть.

Сближение локальных и глобальных сетей

В конце 80-х годов отличия между локальными и глобальными сетями проявлялись весьма отчетливо.

- *Протяженность и качество линий связи.* Локальные компьютерные сети по определению отличаются от глобальных сетей небольшими расстояниями между узлами сети. Это в принципе делает возможным использование в локальных сетях более качественных линий связи.
- *Сложность методов передачи данных.* В условиях низкой надежности физических каналов в глобальных сетях требуются более сложные, чем в локальных сетях, методы передачи данных и соответствующее оборудование.
- *Скорость обмена данными* в локальных сетях (10, 16 и 100 Мбит/с) в то время была существенно выше, чем в глобальных (от 2,4 Кбит/с до 2 Мбит/с).
- *Разнообразие услуг.* Высокие скорости обмена данными позволили предоставлять в локальных сетях широкий спектр услуг — это, прежде всего, разнообразные механизмы использования файлов, хранящихся на дисках других компьютеров сети, совместное использование устройств печати, модемов, факсов, доступ к единой базе данных, электронная почта и др. В то же время глобальные сети в основном ограничивались почтовыми и файловыми услугами в их простейшем (не самом удобном для пользователя) виде.

Постепенно различия между локальными и глобальными типами сетевых технологий стали сглаживаться. Изолированные ранее локальные сети начали объединять друг с другом, при этом в качестве связующей среды использовались глобальные сети. Тесная интеграция локальных и глобальных сетей привела к значительному взаимопроникновению соответствующих технологий.

Сближение в методах передачи данных происходит на платформе цифровой (немодулированной) передачи данных по волоконно-оптическим линиям связи. Эту среду передачи данных используют практически все технологии локальных сетей для скоростного обмена информацией на расстояниях свыше 100 метров, на ней же построены современные магистрали первичных сетей SDH и DWDM, предоставляющих свои цифровые каналы для объединения оборудования глобальных компьютерных сетей.

Высокое качество цифровых каналов изменило требования к протоколам глобальных компьютерных сетей. На первый план вместо процедур обеспечения надежности вышли процедуры обеспечения гарантированной средней скорости доставки информации пользователям, а также механизмы приоритетной обра-

ботки пакетов особенно чувствительного к задержкам трафика, например, голосового. Эти изменения нашли отражение в новых технологиях глобальных сетей, таких как Frame Relay и ATM. В этих сетях предполагается, что искажение битов происходит настолько редко, что ошибочный пакет выгоднее просто уничтожить, а все проблемы, связанные с его потерей, перепоручить программному обеспечению более высокого уровня, которое непосредственно не входит в состав сетей Frame Relay и ATM.

Большой вклад в сближение локальных и глобальных сетей внесло доминирование протокола IP. Этот протокол сегодня работает поверх любых технологий локальных и глобальных сетей (Ethernet, Token Ring, ATM, Frame Relay), объединяя различные подсети в единую составную сеть.

Начиная с 90-х годов, компьютерные глобальные сети, работающие на основе скоростных цифровых каналов, существенно расширили спектр предоставляемых услуг и догнали в этом отношении локальные сети. Стало возможным создание служб, работа которых связана с доставкой пользователю больших объемов информации в реальном времени — изображений, видеофильмов, голоса, в общем, всего того, что получило название мультимедийной информации. Наиболее яркий пример — гипертекстовая информационная служба World Wide Web, ставшая основным поставщиком информации в Интернете. Ее интерактивные возможности превзошли возможности многих аналогичных служб локальных сетей, так что разработчикам локальных сетей пришлось просто позаимствовать эту службу у глобальных сетей. Процесс переноса технологий из глобальной сети Интернет в локальные приобрел такой массовый характер, что появился даже специальный термин — **intranet-технологии** (intra — внутренний).

В локальных сетях в последнее время уделяется такое же большое внимание методам обеспечения защиты информации от несанкционированного доступа, как и в глобальных. Это обусловлено тем, что локальные сети перестали быть изолированными, чаще всего они имеют выход в «большой мир» через глобальные связи.

И, наконец, появляются новые технологии, изначально предназначенные для обоих видов сетей. Ярким представителем нового поколения технологий является технология ATM, которая может служить основой как глобальных, так и локальных сетей, эффективно объединяя все существующие типы трафика в одной транспортной сети. Другим примером является семейство технологий Ethernet, имеющее явные «локальные» корни. Новый стандарт Ethernet 10G, позволяющий передавать данные со скоростью 10 Гбит/с, предназначен для магистралей как глобальных, так и крупных локальных сетей.

Еще одним признаком сближения локальных и глобальных сетей является появление сетей, занимающих промежуточное положение между локальными и глобальными сетями. **Городские сети**, или **сети мегаполисов** (Metropolitan Area Networks, MAN), предназначены для обслуживания территории крупного города.

Эти сети используют цифровые линии связи, часто оптоволоконные, со скоростями на магистрали от 155 Мбит/с и выше. Они обеспечивают экономичное соединение локальных сетей между собой, а также выход в глобальные сети. Сети

MAN первоначально были разработаны только для передачи данных, но сейчас перечень предоставляемых ими услуг расширился, в частности они поддерживают видеоконференции и интегральную передачу голоса и текста. Современные сети MAN отличаются разнообразием предоставляемых услуг, позволяя своим клиентам объединять коммуникационное оборудование различного типа, в том числе и офисные АТС.

Конвергенция компьютерных и телекоммуникационных сетей

С каждым годом усиливается тенденция сближения компьютерных и телекоммуникационных сетей разных видов. Предпринимаются попытки создания универсальной, так называемой **мультисервисной сети**, способной предоставлять услуги как компьютерных, так и телекоммуникационных сетей.

К телекоммуникационным сетям относятся телефонные сети, радиосети и телевизионные сети. Главное, что объединяет их с компьютерными сетями, — то, что в качестве ресурса, предоставляемого клиентам, выступает информация. Однако эти сети, как правило, представляют информацию в разном виде. Так, изначально компьютерные сети разрабатывались для передачи алфавитно-цифровой информации, которую часто называют просто данными, в результате у компьютерных сетей имеется и другое название — **сети передачи данных**, в то время как телефонные сети и радиосети были созданы для передачи только голосовой информации, а телевизионные сети передают и голос, и изображение.

Несмотря на это, конвергенция телекоммуникационных и компьютерных сетей идет по нескольким направлениям.

Прежде всего, наблюдается *сближение видов услуг*, предоставляемых клиентам. Первая и не очень успешная попытка создания мультисервисной сети, способной оказывать различные услуги, в том числе услуги телефонии и передачи данных, привела к появлению технологии **цифровых сетей с интегрированным обслуживанием** (Integrated Services Digital Network, ISDN). Однако на практике ISDN предоставляет сегодня в основном телефонные услуги. Сегодня на роль глобальной **мультисервисной сети нового поколения**, часто называемой в англоязычной литературе Next Generation Network (NGN), или New Public Network (NPN), претендует Интернет. Наибольшую привлекательность сейчас представляют собой новые виды комбинированных услуг, в которых сочетаются несколько традиционных услуг, например, услуга универсальной службы сообщений, объединяющей электронную почту, телефонию, факсимильную службу и пейджинговую связь. Наибольших успехов на практическом поприще достигла IP-телефония, услугами которой прямо или косвенно сегодня пользуются миллионы людей. Однако для того, чтобы стать сетью NGN, Интернету еще предстоит пройти большой путь.

Технологическое сближение сетей происходит сегодня на основе цифровой передачи информации различного типа, метода коммутации пакетов и программирования услуг. Телефония уже давно сделала ряд шагов навстречу компьютерным сетям, прежде всего, за счет представления голоса в цифровой форме, что делает принципиально возможным передачу телефонного и компьютерного трафика по

одним и тем же цифровым каналам (телевидение также может сегодня передавать изображение в цифровой форме). Телефонные сети широко используют комбинацию методов коммутации каналов и пакетов. Так, для передачи служебных сообщений (называемых сообщениями сигнализации) применяются протоколы коммутации пакетов, аналогичные протоколам компьютерных сетей, а для передачи собственно голоса между абонентами коммутируется традиционный составной канал.

Дополнительные услуги телефонных сетей, такие как переадресация вызова, конференц-связь, телеголосование и другие, могут создаваться с помощью так называемой **интеллектуальной сети** (Intelligent Network, IN), по своей сути являющейся компьютерной сетью с серверами, на которых программируется логика услуг.

Сегодня пакетные методы коммутации постепенно теснят традиционные для телефонных сетей методы коммутации каналов даже при передаче голоса. У этой тенденции есть достаточно очевидная причина — на основе метода коммутации пакетов можно более эффективно использовать пропускную способность каналов связи и коммутационного оборудования. Например, паузы в телефонном разговоре могут составлять до 40 % общего времени соединения, однако только пакетная коммутация позволяет «вырезать» паузы и использовать высвободившуюся пропускную способность канала для передачи трафика других абонентов. Другой веской причиной перехода к коммутации пакетов является популярность Интернета — сети, построенной на основе данной технологии.

Использование коммутации пакетов для одновременной передачи через пакетные сети разнородного трафика — голоса, видео и текста — сделало актуальным разработку новых методов обеспечения требуемого **качества обслуживания** (Quality of Service, QoS). Методы QoS призваны минимизировать уровень задержек для чувствительного к ним трафика, например голосового, и одновременно гарантировать среднюю скорость и динамичную передачу пульсаций для трафика данных.

Однако неверно было бы говорить, что методы коммутации каналов морально устарели и у них нет будущего. На новом витке спирали развития они находят свое применение, но уже в новых технологиях.

Компьютерные сети тоже многое позаимствовали у телефонных и телевизионных сетей. В частности они берут на вооружение методы обеспечения отказоустойчивости телефонных сетей, за счет которых последние демонстрируют высокую степень надежности, так недостающую порой Интернету и корпоративным сетям.

Сегодня становится все более очевидным, что мультисервисная сеть нового поколения не может быть создана в результате «победы» какой-нибудь одной технологии или подхода. Ее может породить только процесс конвергенции, когда от каждой технологии будет взято все самое лучшее и соединено в некоторый новый сплав, который и даст требуемое качество для поддержки существующих и создания новых услуг. Появился новый термин — **инфокоммуникационная сеть**, который прямо говорит о двух составляющих современной сети — информационной (компьютерной) и телекоммуникационной. Учитывая, что новый термин еще не приобрел достаточной популярности, мы будем использовать устоявший-

ся термин «телекоммуникационная сеть» в расширенном значении — то есть включать в него и компьютерные сети.

Выводы

Компьютерные сети стали логическим результатом эволюции компьютерных и телекоммуникационных технологий. С одной стороны, они являются частным случаем распределенных компьютерных систем, а с другой стороны, могут рассматриваться как средство передачи информации на большие расстояния, для чего в них применяются методы кодирования и мультиплексирования данных, получившие развитие в различных телекоммуникационных системах.

Классифицируя сети по территориальному признаку, различают глобальные (WAN), локальные (LAN) и городские (MAN) сети.

Хронологически первыми появились сети WAN. Они объединяют компьютеры, рассредоточенные на расстоянии сотен и тысяч километров. Первые глобальные компьютерные сети очень многое унаследовали от телефонных сетей. В них часто использовались уже существующие и не очень качественные линии связи, что приводило к низким скоростям передачи данных и ограничивало набор предоставляемых услуг передачей файлов в фоновом режиме и электронной почтой.

Сети LAN ограничены расстояниями в несколько километров; они строятся с использованием высококачественных линий связи, которые позволяют, применяя более простые методы передачи данных, чем в глобальных сетях, достигать высоких скоростей обмена данными до нескольких гигабитов в секунду. Услуги предоставляются в режиме подключения и отличаются разнообразием.

Сети MAN предназначены для обслуживания территории крупного города. При достаточно больших расстояниях между узлами (десятки километров) они обладают качественными линиями связи и поддерживают высокие скорости обмена. Сети MAN обеспечивают экономичное соединение локальных сетей между собой, а также доступ к глобальным сетям.

Важнейший этап в развитии сетей — появление стандартных сетевых технологий: Ethernet, FDDI, Token Ring, позволяющих быстро и эффективно объединять компьютеры различных типов.

В конце 80-х годов локальные и глобальные сети имели существенные отличия по протяженности и качеству линий связи, сложности методов передачи данных, скорости обмена данными, разнообразию предоставляемых услуг и масштабируемости. В дальнейшем в результате тесной интеграции LAN, WAN и MAN произошло взаимопроникновение соответствующих технологий.

Тенденция сближения различных типов сетей характерна не только для локальных и глобальных компьютерных сетей, но и для телекоммуникационных сетей других типов: телефонных сетей, радиосетей, телевизионных сетей. В настоящее время ведутся активные работы по созданию универсальных мультисервисных сетей, способных одинаково эффективно передавать информацию любого типа: данные, голос и видео.

Вопросы и задания

1. Какие свойства многотерминальной системы отличают ее от компьютерной сети?
2. Когда впервые были получены значимые практические результаты по объединению компьютеров с помощью глобальных связей?

3. Что такое ARPANET?
4. Какое из этих событий произошло позже других:
   - изобретение Web;
   - появление стандартных технологий LAN;
   - начало передачи голоса в цифровой форме по телефонным сетям.
5. Какое событие послужило стимулом к активизации работ по созданию LAN?
6. Когда была стандартизована технология Ethernet? Token Ring? FDDI?
7. По каким направлениям идет сближение компьютерных и телекоммуникационных сетей.
8. Поясните термины «мультисервисная сеть», «инфокоммуникационная сеть», «интеллектуальная сеть».
9. Поясните, почему сети WAN появились раньше, чем сети LAN.
10. Найдите исторические связи между технологией X.25 и сетью ARPANET, пользуясь источниками информации в Интернете.
11. Считаете ли вы, что история компьютерных сетей может быть сведена к истории Интернета? Обоснуйте свое мнение.

ГЛАВА 2 Общие принципы построения сетей

Когда вы приступаете к изучению конкретных технологий для сетей LAN, WAN и MAN, таких как Ethernet, IP или ATM, то очень скоро начинаете понимать, что у этих технологий есть много общего. При этом они не являются тождественными, в каждой технологии и протоколе есть свои особенности, так что нельзя механически перенести знания из одной технологии в другую. Решением, повышающим эффективность учебного процесса, является предварительное изучение общих принципов построения компьютерных сетей, определяющих выбор топологии сети, а также способов маршрутизации, коммутации и мультиплексирования информационных потоков. Известное высказывание «Знание нескольких принципов освобождает от запоминания множества фактов» не стоит понимать буквально. Хороший специалист должен знать множество деталей. Однако знание принципов помогает осознать и более эффективно использовать отдельные факты и детали, так как при этом они связываются друг с другом в стройную систему.

Система принципов построения сетей передачи данных появилась в результате решения ряда ключевых проблем, многие из которых являются общими для телекоммуникационных сетей любого типа.

Одной из основных, если не сказать главных, проблем построения сетей является коммутация. Каждый узел, выполняющий транзитную передачу трафика, должен уметь его коммутировать, то есть обеспечить взаимодействие пользователей сети.

На технологию коммутации непосредственно влияет принцип выбора маршрута передачи информационных потоков через сеть. Маршрут, то есть последовательность транзитных узлов сети, которые должны пройти данные, чтобы попасть к получателю, должен выбираться так, чтобы одновременно достигались две цели. Во-первых, данные каждого пользователя должны передаваться как можно быстрее, с минимальными задержками в пути. Во-вторых, ресурсы сети должны использоваться максимально эффективно, так чтобы сеть в единицу времени передавала как можно больше данных, поступающих от всех пользователей сети. Задача состоит в том, чтобы добиться совмещения этих целей (эгоистичной цели отдельного пользователя и коллективной цели сети как единой системы). Компьютерные сети традиционно решали эту проблему неэффективно, в пользу индивидуальных потоков, и только в последнее время появились более совершенные методы маршрутизации.

В этой главе мы также изучим принципы мультиплексирования различных информационных потоков и разделения передающей среды, проблему адресации и выбора топологии, логическую и физическую структуризацию.

Совместное использование ресурсов компьютеров

Список ключевых слов: физический интерфейс, или порт, логический интерфейс, интерфейсная карта, драйвер, контроллер, стартовый и стоповый сигналы, сообщение, протокол, клиент, сервер, удаленный и локальный ресурсы, перенаправление, редиректор, сетевая операционная система, сервис, сетевая служба, распределенная программа, пользовательское приложение, сетевое приложение.

Одним из очевидных удобств, получаемых пользователем, компьютер которого подключается к сети, является возможность использования периферийных устройств «чужих» компьютеров, таких как диски, принтеры, плоттеры. Как и при автономной работе, компьютер, включенный в сеть, способен непосредственно управлять только теми периферийными устройствами, которые к нему физически подсоединены. Чтобы обеспечить пользователей разных компьютеров возможностью совместного использования периферийных устройств, сеть необходимо оснастить некими дополнительными средствами. Давайте выясним, что они из себя могут представлять, на простейшем примере, когда сеть образована только двумя компьютерами (рис. 2.1).

Рис. 2.1. Совместное использование принтера

Однако для начала рассмотрим, как взаимодействуют друг с другом компьютер и периферийное устройство (ПУ).

Связь компьютера с периферийными устройствами

Для организации связи между компьютером и периферийным устройством (ПУ) в обоих этих устройствах предусмотрены внешние физические интерфейсы.

Интерфейс — в широком смысле — формально определенная логическая и физическая границы между взаимодействующими независимыми объектами. Интерфейс задает параметры, процедуры и характеристики взаимодействия объектов.

Физический интерфейс (называемый также **портом**) — определяется набором электрических связей и характеристиками сигналов. Обычно он представляет собой разъем с набором контактов, каждый из которых имеет определенное назначение, например, это может быть группа контактов для передачи данных, контакт

синхронизации данных и т. п. Пара разъемов соединяется кабелем, который состоит из набора проводов, каждый из которых соединяет соответствующие контакты (рис. 2.2).

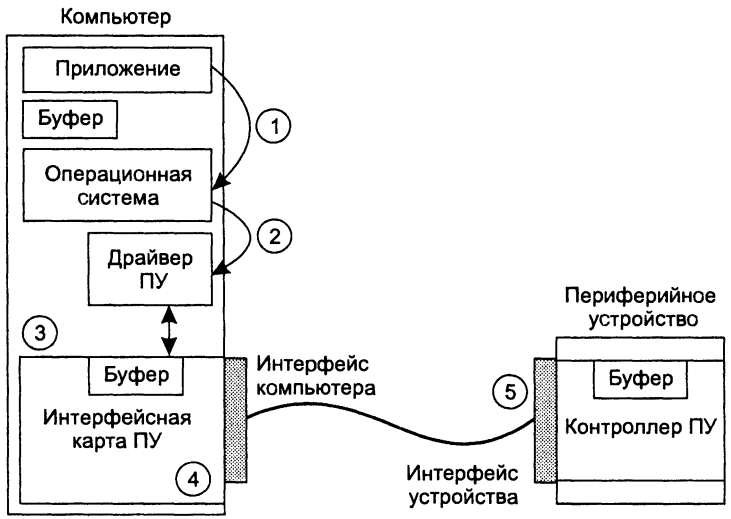


Рис. 2.2. Связь компьютера с периферийным устройством

Логический интерфейс — это набор информационных сообщений определенного формата, которыми обмениваются два устройства или две программы (в данном случае компьютер и периферийное устройство), а также набор правил, определяющих логику обмена этими сообщениями.

Примерами стандартных интерфейсов, используемых в компьютерах, являются параллельный (передающий данные байтами) интерфейс Centronics, предназначенный, как правило, для подключения принтеров, и последовательный интерфейс (передающий данные битами) RS-232C (известный также как COM-порт), который имеет более универсальное назначение — он поддерживается не только принтерами, но и графопостроителями, манипуляторами типа «мышь» и многими другими устройствами. Существуют также специализированные интерфейсы, которые предназначены для подключения уникальных периферийных устройств, например сложной физической экспериментальной установки.

В компьютере операции интерфейса реализуется совокупностью аппаратных и программных средств: **интерфейсной картой** (аппаратное устройство) и специальной программой, управляющей этим контроллером, которую часто называют **драйвером** соответствующего периферийного устройства.

В ПУ интерфейс чаще всего полностью реализуется аппаратным устройством — **контроллером**, хотя встречаются и программно-управляемые контроллеры для управления современными принтерами, обладающими более сложной логикой.

Периферийные устройства могут принимать от компьютера как данные, например байты информации, которую нужно распечатать на бумаге, так и команды управления, в ответ на которые контроллер ПУ может выполнять специальные действия (перевести головку диска на требуемую дорожку, вытолкнуть лист бумаги из принтера и т. д.). Контроллер принтера, например, поддерживает некоторый набор достаточно простых команд, таких как «Печать символа», «Перевод строки», «Возврат каретки» и т. п., которые он получает от компьютера по интерфейсу и отрабатывает, управляя электромеханическими частями принтера.

Итак, рассмотрим порядок действий, в результате которых приложение распечатывает данные на принтере.

1. Приложение обращается с запросом на выполнение операции ввода-вывода к операционной системе. В запросе указываются адрес данных в оперативной памяти, идентифицирующая информация о периферийном устройстве и операция, которую надо выполнить.
2. Получив запрос, операционная система запускает драйвер принтера. Дальнейшие действия по выполнению операции ввода-вывода со стороны компьютера реализуются интерфейсной картой, работающей под управлением драйвера.
3. Драйвер принтера оперирует командами, понятными контроллеру принтера, то есть командами «Печать символа», «Перевод строки», «Возврат каретки». Драйвер в определенной последовательности помещает коды этих команд в регистр интерфейсной карты, которая побайтно передает их по сети контроллеру периферийного устройства. Для одного и того же контроллера можно разработать различные драйверы, которые с помощью одного набора команд будут реализовывать разные алгоритмы управления ПУ.
4. Интерфейсная карта выполняет низкоуровневую работу, она не вдается в смысл данных и команд, передаваемых ей драйвером, считая их однородным потоком байтов. После получения от драйвера очередного байта интерфейсная карта просто последовательно передает биты в линию связи, представляя каждый бит электрическим сигналом. Чтобы контроллеру ПУ стало понятно, что начинается передача байта, перед передачей первого бита информационная карта формирует **стартовый сигнал** специфической формы, а после передачи последнего информационного бита — **стоповый сигнал**. Эти сигналы *синхронизируют* передачу байта. Контроллер, опознав стартовый бит, начинает принимать информационные биты, формируя из них байт в своем приемном буфере.

   Помимо информационных битов карта может передавать бит контроля четности для повышения достоверности обмена. При корректно выполненной передаче в регистре контроллера устанавливается соответствующий признак.
5. Получив очередной байт, контроллер интерпретирует его и запускает заданную операцию принтера. Закончив работу по печати всех символов документа, драйвер принтера сообщает операционной системе о выполнении запроса, а та, в свою очередь, сигнализирует об этом событии приложению.

Простейший случай взаимодействия двух компьютеров

Вернемся к исходному вопросу: как пользователю, работающему с некоторым приложением на компьютере А, распечатать текст на принтере компьютера В (рис. 2.3).

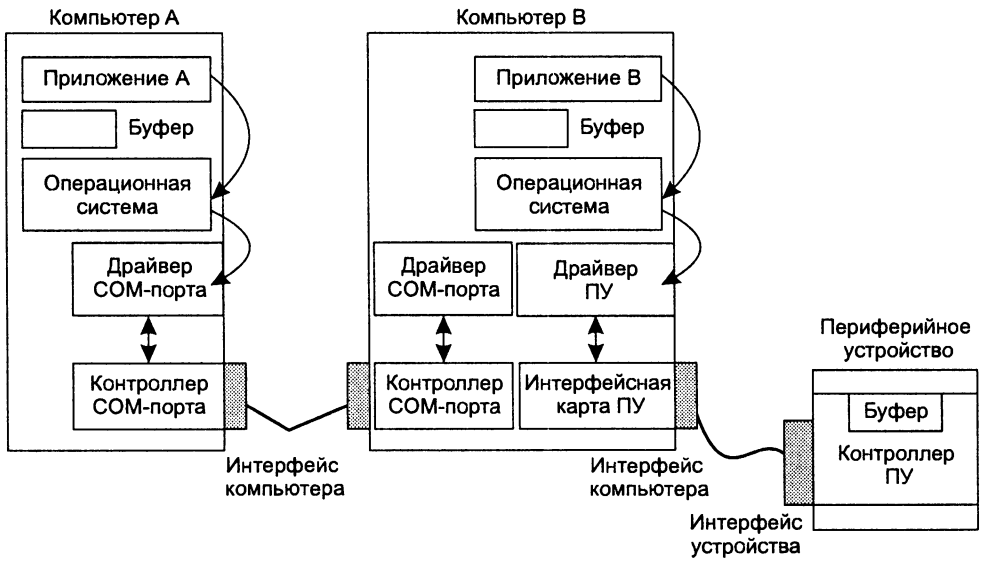


Рис. 2.3. Совместное использование принтера в компьютерной сети

Приложение А не может получить непосредственный доступ к ресурсам компьютера В — его дискам, файлам, принтеру. Оно может только «попросить» об этом другую программу, выполняемую на том компьютере, которому принадлежат эти ресурсы. Эти «просьбы» выражаются в виде **сообщений**, передаваемых по каналам связи между компьютерами. Сообщения могут содержать как команды на выполнение некоторых действий («открыть файл»), так и собственно информационные данные (содержимое некоторого файла).

Механизмы взаимодействия компьютеров в сети многое позаимствовали у схемы взаимодействия компьютера с периферийными устройствами. В самом простом случае связь компьютеров может быть реализована с помощью тех же самых средств, которые используются для связи компьютера с периферией. Пусть для определенности связь между компьютерами будет осуществляться через последовательный интерфейс — COM-порт. С каждой стороны COM-порт работает под управлением драйвера COM-порта. Вместе они обеспечивают передачу по кабелю между компьютерами одного байта информации.

ПРИМЕЧАНИЕ

В «настоящих» локальных сетях подобные функции передачи данных в линию связи выполняются сетевыми интерфейсными картами (Network Interface Card, NIC), называемыми также сетевыми адаптерами, и их драйверами.

Итак, механизм обмена байтами между двумя компьютерами определен. Однако этого еще недостаточно для решения поставленной задачи — распечатки текста на «чужом» принтере. В частности, необходимо, чтобы компьютер В «понял», какую операцию он должен выполнить с передаваемыми данными, на каком из имеющихся в его распоряжении устройств, в каком виде должен быть распечатан текст и т. п. Обо всем этом должны договориться приложения А и В путем обмена сообщениями.

Чтобы приложения могли «понимать» получаемую друг от друга информацию, программисты, разрабатывавшие приложения А и В, должны *строго оговорить* форматы сообщений, которыми будут обмениваться приложения, и их семантику. Например, они могут договориться о том, что любое выполнение удаленной операции печати начинается с передачи сообщения, запрашивающего информацию о готовности приложения В; что в следующем сообщении идут идентификаторы компьютера и пользователя, сделавшего запрос; что признаком срочного завершения печати является определенная кодовая комбинация и т. п. Тем самым определяется **протокол** взаимодействия приложений.

Рассмотрим взаимодействие всех элементов этой небольшой сети, которые позволят приложению на компьютере А распечатать текст на принтере компьютера В.

1. Приложение А формирует сообщение-запрос для приложения В на печать текста и помещает его в свой буфер. Чтобы передать данный запрос компьютеру В, приложение А обращается к локальной ОС, которая запускает драйвер COM-порта компьютера и сообщает ему адрес буфера, где хранится запрос. Затем по только что описанной схеме драйвер и контроллер COM-порта компьютера А, взаимодействуя с драйвером и контроллером COM-порта компьютера В, передают сообщение байт за байтом в компьютер В.
2. Драйвер COM-порта компьютера В постоянно находится в режиме ожидания прихода информации из внешнего мира. В некоторых случаях драйвер вызывается асинхронно, по прерываниям от контроллера. Получив очередной байт и убедившись в его корректности, драйвер помещает его в буфер приложения В.
3. Приложение В принимает сообщение, интерпретирует его и формирует запрос к локальной ОС на выполнение тех или иных действий с принтером. В ходе печати могут возникнуть ситуации, о которых необходимо сообщить приложению А. В этом случае используется симметричная схема: теперь запрос на передачу сообщения поступает от приложения В к локальной ОС компьютера В. Драйверы и контроллеры COM-портов обоих компьютеров организуют побайтную передачу сообщения, которое затем помещается в буфер приложения А.

Потребность в доступе к удаленным файлам может возникать у пользователей многих других приложений: текстового редактора, графического редактора, системы управления базой данных (СУБД). Очевидно, нерационально включать рассмотренные универсальные функции по организации ввода-вывода в состав каждого приложения. Более эффективно решают задачу пара специализированных программных модулей.

Клиент — модуль, предназначенный для формирования сообщений-запросов к удаленной машине от разных приложений, а затем приема результатов и передачи их соответствующим приложениям.

Сервер — модуль, который постоянно ожидает прихода из сети запросов от клиентов и, приняв запрос, пытается его выполнить, возможно с участием локальной ОС; один сервер может выполнять запросы сразу нескольких клиентов (последовательно или одновременно).

Очень удобной и полезной функцией клиентской программы является способность отличить запрос к **удаленному** ресурсу от запроса к **локальному** ресурсу. Если клиентская программа умеет это делать, то приложения не должны заботиться, например, о том, с каким принтером они работают (локальным или удаленным), клиентская программа сама распознает и **перенаправляет** (redirect) запрос к удаленной машине. Отсюда и название, часто используемое для клиентского модуля, — **редиректор**. Иногда функции распознавания выделяются в отдельный программный блок, в этом случае редиректором называют не весь клиентский модуль, а только этот блок.

Клиент и сервер выполняют системные функции по обслуживанию запросов всех приложений компьютера А на удаленный доступ к ресурсу (принтеру, файлам, факсу) компьютера В. Чтобы приложения компьютера В могли пользоваться ресурсами компьютера А, описанную схему нужно симметрично дополнить клиентом для компьютера В и сервером для компьютера А.

Схема взаимодействия клиента и сервера с приложениями и локальной операционной системой приведена на рис. 2.4.

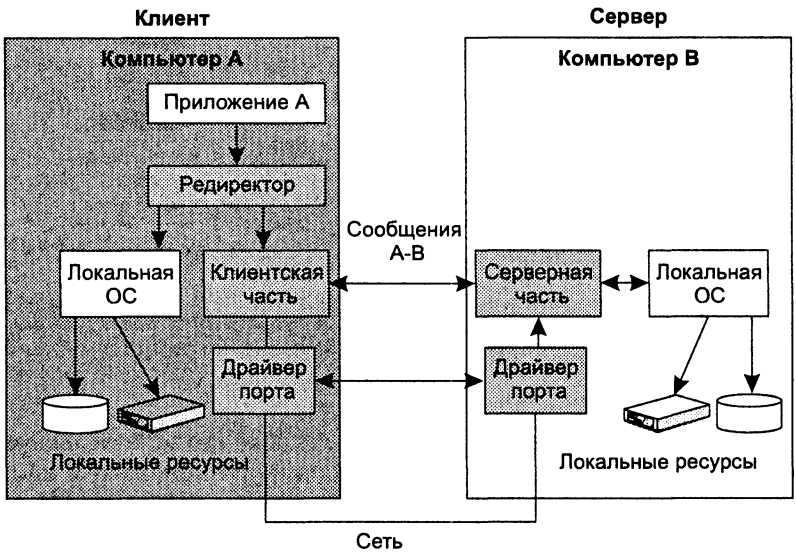


Рис. 2.4. Взаимодействие программных компонентов при связи двух компьютеров

Несмотря на то что мы рассмотрели очень простую схему связи только двух компьютеров, функции программ, обеспечивающих удаленный доступ к принтеру, во многом совпадают с функциями **сетевой операционной системы**, работающей в сети с более сложными аппаратными связями компьютеров.

ПРИМЕЧАНИЕ

Термины «клиент» и «сервер» используются для обозначения не только программных модулей, но и компьютеров, подключенных к сети. Если компьютер преимущественно предоставляет свои ресурсы другим компьютерам сети, то он называется сервером, а если он их потребляет — клиентом. Иногда один и тот же компьютер может одновременно играть роли и сервера, и клиента.

Сетевые службы и приложения

Предоставление пользователям совместного доступа к определенному типу ресурсов, например к файлам, называют также предоставлением **сервиса** (в данном случае файлового сервиса). Обычно сетевая операционная система поддерживает несколько видов сетевых сервисов для своих пользователей — файловый сервис, сервис печати, сервис электронной почты, сервис удаленного доступа и т. п. Программы, реализующие сетевые сервисы, относятся к классу распределенных программ.

Распределенная программа — это программа, которая состоит из нескольких взаимодействующих частей (в приведенном на рис. 2.5 примере — из двух), причем каждая часть может выполняться и, как правило, выполняется на отдельном компьютере сети.

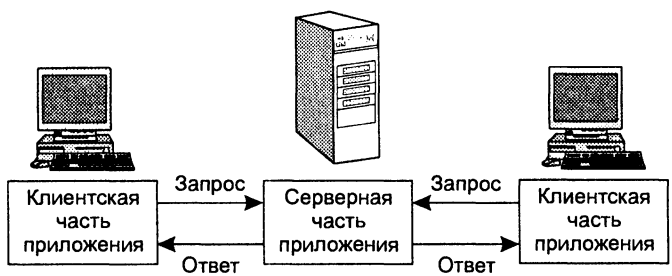


Рис. 2.5. Взаимодействие частей распределенного приложения

Сетевые службы — это *системные* распределенные программы, реализующие сетевые сервисы. Они часто представляют собой пару «клиент-сервер» и являются неотъемлемыми компонентами ОС.

Однако в сети могут выполняться и распределенные **пользовательские приложения**. Распределенное приложение также состоит из нескольких частей, каждая из которых выполняет какую-то определенную законченную работу по решению прикладной задачи. Например, одна часть приложения, выполняющаяся на компьютере пользователя, может поддерживать специализированный графический интерфейс, вторая — работать на мощном выделенном компьютере и за-

ниматься статистической обработкой введенных пользователем данных, третья — заносить полученные результаты в базу данных на компьютере с установленной стандартной СУБД. Распределенные приложения в полной мере используют потенциальные возможности распределенной обработки, предоставляемые вычислительной сетью, и поэтому часто называются **сетевыми приложениями**.

ПРИМЕЧАНИЕ

Не всякое приложение, выполняемое в сети, является распределенным. Значительная часть истории локальных сетей связана как раз с использованием таких нераспределенных приложений. Рассмотрим, например, как происходила работа пользователя с известной в свое время СУБД dBase. Файлы базы данных, с которыми работали все пользователи сети, располагались на файловом сервере. Сама же СУБД хранилась на каждом клиентском компьютере в виде единого программного модуля. Программа dBase была рассчитана только на обработку данных, расположенных на том же компьютере, что и сама программа. Пользователь запускал dBase на своем компьютере, и программа искала данные на локальном диске, совершенно не принимая во внимание существование сети. Чтобы обрабатывать с помощью dBase данные, расположенные на удаленном компьютере, пользователь обращался к услугам файловой службы, которая доставляла данные с сервера на клиентский компьютер и создавала для СУБД эффект их локального хранения.

Большинство приложений, используемых в локальных сетях в середине 80-х годов, являлись обычными нераспределенными приложениями. И это понятно — они были написаны для автономных компьютеров, а потом просто были перенесены в сетевую среду. Создание же распределенных приложений, хотя и сулило много преимуществ (снижение сетевого трафика, специализация компьютеров), оказалось делом совсем не простым. Нужно было решать множество дополнительных проблем: на сколько частей разбить приложение, какие функции возложить на каждую часть, как организовать взаимодействие этих частей, чтобы в случае сбоев и отказов оставшиеся части корректно завершали работу и т. д., и т. п. Поэтому до сих пор только небольшая часть приложений являются распределенными, хотя очевидно, что именно за этим классом приложений будущее, так как они в полной мере могут использовать потенциальные возможности сетей по распараллеливанию вычислений.

Физическая передача данных по линиям связи

Список ключевых слов: двоичный код, кодирование, потенциальный и импульсный способы кодирования, линия связи, модуляция, синхронизация, контрольная сумма, квитанция, предложенная нагрузка, скорость передачи данных, емкость, или пропускная способность, канала связи, битовая скорость передатчика, полоса пропускания, дуплексный, полудуплексный и симплексный каналы.

Даже при рассмотрении простейшей сети, состоящей всего из двух машин, можно выявить многие проблемы, связанные с физической передачей сигналов по линиям связи.

Кодирование

В вычислительной технике для представления данных используется **двоичный код**. Внутри компьютера единицам и нулям данных соответствуют дискретные электрические сигналы.

Представление данных в виде электрических или оптических сигналов называется **кодированием**.

Существуют различные способы кодирования двоичных цифр, например **потенциальный способ**, при котором единице соответствует один уровень напряжения, а нулю — другой, или **импульсный способ**, когда для представления цифр используются импульсы различной полярности.

Аналогичные подходы применимы для кодирования данных и при передаче их между двумя компьютерами **по линиям связи**. Однако эти линии связи отличаются по своим характеристикам от линий внутри компьютера. Главное отличие внешних линий связи от внутренних состоит в их гораздо большей протяженности, а также в том, что они проходят вне экранированного корпуса по пространствам, зачастую подверженным воздействию сильных электромагнитных помех. Все это приводит к существенно бóльшим искажениям прямоугольных импульсов (например, «заваливанию» фронтов), чем внутри компьютера. Поэтому для надежного распознавания импульсов на приемном конце линии связи при передаче данных внутри и вне компьютера не всегда можно использовать одни и те же скорости и способы кодирования. Например, медленное нарастание фронта импульса из-за высокой емкостной нагрузки линии требует, чтобы импульсы передавались с меньшей скоростью (чтобы передний и задний фронты соседних импульсов не перекрывались, и импульс успел «дорасти» до требуемого уровня).

В вычислительных сетях применяют как потенциальное, так и импульсное кодирование дискретных данных, а также специфический способ представления данных, который никогда не используется внутри компьютера, — **модуляцию** (рис. 2.6). При модуляции дискретная информация представляется синусоидальным сигналом той частоты, которую хорошо передает имеющаяся линия связи.

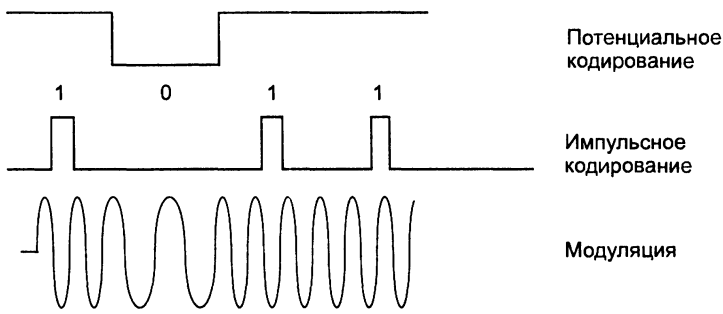


Рис. 2.6. Примеры представления дискретной информации

Потенциальное, или импульсное, кодирование применяется на каналах *высокого качества*, а модуляция на основе синусоидальных сигналов предпочтительнее в том случае, когда канал вносит сильные искажения в передаваемые сигналы. Например, модуляция используется в глобальных сетях при передаче данных через аналоговые телефонные каналы связи, которые были разработаны для передачи голоса в аналоговой форме и поэтому плохо подходят для непосредственной передачи импульсов.

На способ передачи сигналов влияет и *количество проводов* в линиях связи между компьютерами. Для снижения стоимости линий связи в сетях обычно стремятся к сокращению количества проводов и из-за этого используют не параллельную передачу всех битов одного байта или даже нескольких байтов, как это делается внутри компьютера, а последовательную побитную передачу, требующую всего одной пары проводов.

Еще одной проблемой, которую нужно решать при передаче сигналов, является проблема взаимной **синхронизации** передатчика одного компьютера с приемником другого. При организации взаимодействия модулей внутри компьютера эта проблема решается очень просто, так как в этом случае все модули синхронизируются от общего тактового генератора. Проблема синхронизации при связи компьютеров может решаться разными способами, как путем обмена специальными тактовыми синхроимпульсами по отдельной линии, так и путем периодической синхронизации заранее обусловленными кодами или импульсами характерной формы, отличающейся от формы импульсов данных.

Несмотря на предпринимаемые меры (выбор соответствующей скорости обмена данными, линий связи с определенными характеристиками, способа синхронизации приемника и передатчика), существует вероятность искажения некоторых битов передаваемых данных. Для повышения надежности передачи данных между компьютерами часто используется стандартный прием — подсчет **контрольной суммы** и передача ее по линиям связи после каждого байта или после некоторого блока байтов. Часто в протокол обмена данными включается как обязательный элемент **сигнал-квитанция**, который подтверждает правильность приема данных и посылается от получателя отправителю.

Характеристики физических каналов

Существует большое количество характеристик, связанных с передачей трафика через физические каналы. С теми из них, которые будут необходимы нам уже в ближайшее время, мы познакомимся сейчас, а некоторые изучим позже, в главе 6.

- **Предложенная нагрузка** — это поток данных, поступающий от пользователя на вход сети. Предложенную нагрузку можно характеризовать скоростью поступления данных в сеть — в битах в секунду (или килобитах, мегабитах и т. д.).
- **Скорость передачи данных** (information rate или throughput, оба английских термина используются равноправно) — это *фактическая* скорость потока данных, прошедшего через сеть. Эта скорость может быть меньше, чем ско-

рость предложенной нагрузки, так как данные в сети могут искажаться или теряться.

- **Емкость канала связи** (capacity), называемая также **пропускной способностью**, представляет собой *максимально возможную* скорость передачи информации по каналу.

  Спецификой этой характеристики является то, что она отражает не только параметры *физической среды передачи*, но и особенности *выбранного способа передачи* дискретной информации по этой среде. Например, емкость канала связи в сети Ethernet на оптическом волокне равна 10 Мбит/с. Эта скорость является предельно возможной для сочетания технологии Ethernet и оптического волокна. Однако для того же самого оптического волокна можно разработать и другую технологию передачи данных, отличающуюся способом кодирования данных, тактовой частотой и другими параметрами, которая будет иметь другую емкость. Так, технология Fast Ethernet обеспечивает передачу данных по тому же оптическому волокну с максимальной скоростью 100 Мбит/с, а технология Gigabit Ethernet – 1000 Мбит/с. Передатчик коммуникационного устройства должен работать со скоростью, равной пропускной способности канала. Эта скорость иногда называется **битовой скоростью передатчика** (bit rate of transmitter).

- **Полоса пропускания** (bandwidth) — этот термин может ввести в заблуждение, потому что он используется в двух разных значениях. Во-первых, с его помощью могут характеризовать *среду передачи*. В этом случае он означает ширину полосы частот, которую линия передает без существенных искажений. Из этого определения понятно происхождение термина. Во-вторых, термин «полоса пропускания» используется как *синоним термина «емкость канала связи»*. В первом случае полоса пропускания измеряется в герцах (Гц), во втором — в битах в секунду. Различать значения этого термина нужно по контексту, хотя иногда это достаточно трудно. Конечно, лучше было бы использовать разные термины для различных характеристик, но существуют традиции, которые изменить трудно. Такое двойное использование термина «полоса пропускания» уже вошло во многие стандарты и книги, поэтому и в данной книге мы будем следовать сложившемуся подходу. Нужно также учитывать, что этот термин в его втором значении является даже более распространенным, чем емкость, поэтому из этих двух синонимов мы будем использовать полосу пропускания.

Еще одна группа характеристик канала связи связана с возможностью передачи информации по каналу в одну или обе стороны.

При взаимодействии двух компьютеров обычно требуется передавать информацию в обоих направлениях, от компьютера А к компьютеру В и обратно. Даже в том случае, когда пользователю кажется, что он только получает информацию (например, загружает музыкальный файл из Интернета) или передает (отправляет электронное письмо), обмен информации идет в двух направлениях. Просто существует основной поток данных, которые интересуют пользователя, и вспомогательный поток противоположного направления, который образуют квитанции о получении этих данных.

Физические каналы связи делятся на несколько типов в зависимости от того, могут они передавать информацию в обоих направлениях или нет.

- **Дуплексный канал** обеспечивает одновременную передачу информации в обоих направлениях. Дуплексный канал может состоять их двух физических сред, каждая их которых используется для передачи информации только в одном направлении. Возможен вариант, когда одна среда служит для одновременной передачи встречных потоков, в этом случае применяют дополнительные методы выделения каждого потока из суммарного сигнала.
- **Полудуплексный канал** также обеспечивает передачу информации в обоих направлениях, но не одновременно, а по очереди. То есть в течение определенного периода времени информация передается в одном направлении, а в течении следующего периода — в обратном.
- **Симплексный канал** позволяет передавать информацию только в одном направлении. Часто дуплексный канал состоит из двух симплексных каналов.

Подробно вопросы физической передачи дискретных данных обсуждаются в части II.

Проблемы связи нескольких компьютеров

Список ключевых слов: полносвязная и неполносвязная топологии сети, ячеистая топология, кольцевая топология, звездообразная топология, концентратор, иерархическая звезда, или дерево, общая шина, смешанная топология, уникальный, групповой и широковещательный адреса, адрес произвольной рассылки, числовой и символьный адреса, адресное пространство, MAC-адрес, аппаратный адрес, протокол разрешения адресов, коммутация, маршрут.

До сих пор мы рассматривали вырожденную сеть, состоящую всего из двух машин. При объединении в сеть большего числа компьютеров возникает целый комплекс новых проблем.

Топология физических связей

Объединяя в сеть несколько (больше двух) компьютеров, необходимо решить, каким образом соединить их друг с другом, другими словами, выбрать конфигурацию физических связей, или топологию.

Под **топологией сети** понимается конфигурация графа, вершинам которого соответствуют конечные узлы сети (например, компьютеры) и коммуникационное оборудование (например, маршрутизаторы), а ребрам — физические или информационные связи между вершинами.

Число возможных вариантов конфигураций резко возрастает при увеличении числа связываемых устройств. Так, если три компьютера мы можем связать двумя способами (рис. 2.7, *а*), то для четырех можно предложить уже шесть тополо-

гически разных конфигураций (при условии неразличимости компьютеров), что и иллюстрирует рис. 2.7, *б*.

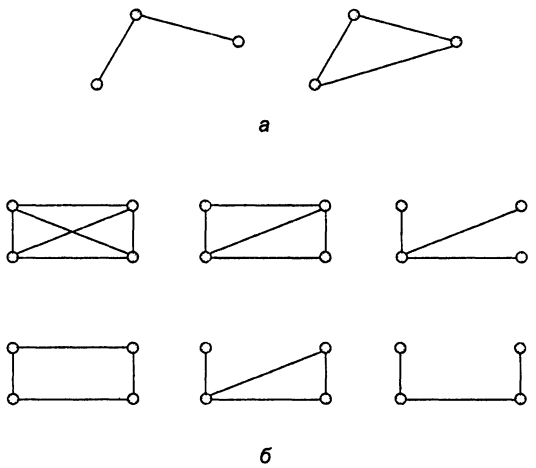


Рис. 2.7. Варианты связи компьютеров

Мы можем соединять каждый компьютер с каждым или же связывать их последовательно, предполагая, что они будут общаться, передавая сообщения друг другу «транзитом». Транзитные узлы должны быть оснащены специальными средствами, позволяющими им выполнять эту специфическую посредническую операцию. В качестве транзитного узла может выступать как универсальный компьютер, так и специализированное устройство.

От выбора топологии связей существенно зависят характеристики сети. Например, наличие между узлами нескольких путей повышает надежность сети и делает возможным балансировку загрузки отдельных каналов. Простота присоединения новых узлов, свойственная некоторым топологиям, делает сеть легко *расширяемой*. Экономические соображения часто приводят к выбору топологий, для которых характерна минимальная суммарная длина линий связи.

Среди множества возможных конфигураций различают полносвязные и неполносвязные.

Полносвязная топология (рис. 2.8, *а*) соответствует сети, в которой каждый компьютер непосредственно связан со всеми остальными. Несмотря на логическую простоту, этот вариант оказывается громоздким и неэффективным. Действительно, в таком случае каждый компьютер в сети должен иметь большое количество коммуникационных портов, достаточное для связи с каждым из остальных компьютеров сети. Для каждой пары компьютеров должна быть выделена отдельная физическая линия связи. (В некоторых случаях даже две, если невозможно использование этой линии для двусторонней передачи.) Полносвязные топологии в крупных сетях применяются редко, так как для связи N узлов требуется $N(N-1)/2$ физических дуплексных линий связей, то есть имеет место квадратичная зависимость от числа узлов. Чаще этот вид топологии используется

в многомашинных комплексах или в сетях, объединяющих небольшое количество компьютеров.

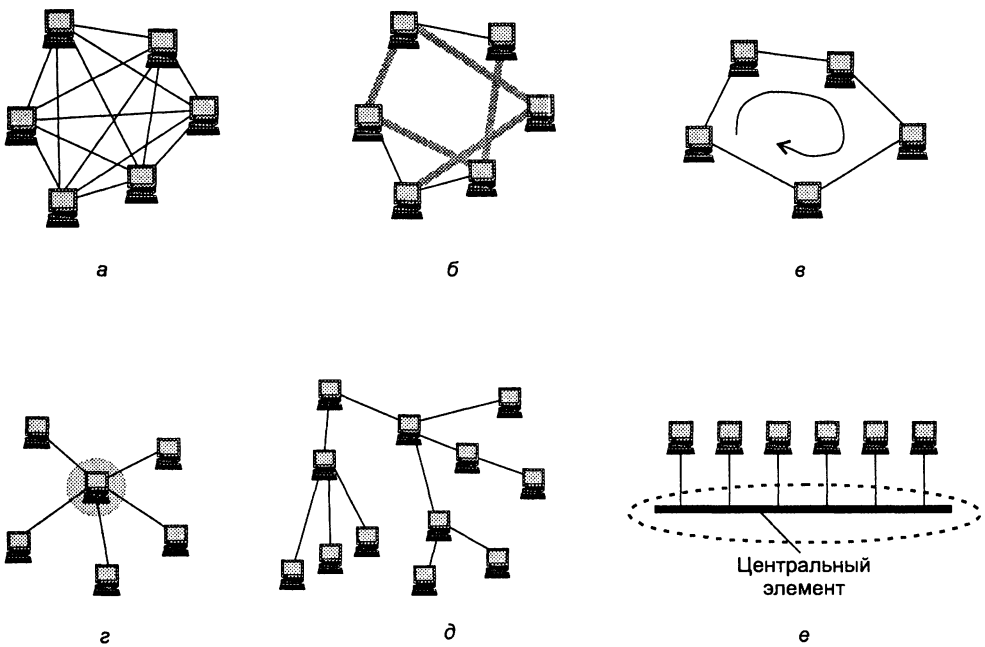


Рис. 2.8. Типовые топологии сетей

Все другие варианты основаны на **неполносвязных топологиях**, когда для обмена данными между двумя компьютерами может потребоваться транзитная передача данных через другие узлы сети.

Ячеистая топология[1] получается из полносвязной путем удаления некоторых связей (рис. 2.8, *б*). Ячеистая топология допускает соединение большого количества компьютеров и характерна, как правило, для крупных сетей.

В сетях с **кольцевой топологией** (рис. 2.8, *в*) данные передаются по кольцу от одного компьютера к другому. Главным достоинством кольца является то, что оно по своей природе обеспечивает резервирование связей. Действительно, любая пара узлов соединена здесь двумя путями — по часовой стрелке и против нее. Кольцо представляет собой очень удобную конфигурацию и для организации обратной связи — данные, сделав полный оборот, возвращаются к узлу-источнику. Поэтому источник может контролировать процесс доставки данных адресату. Часто это свойство кольца используется для тестирования связности сети и поиска узла, работающего некорректно. В то же время в сетях с кольцевой топологией необходимо принимать специальные меры, чтобы в случае выхода из строя или отключения какого-либо компьютера не прерывался канал связи между остальными узлами кольца.

[1] Иногда ячеистой называют полносвязную или близкую к полносвязной топологию.

Звездообразная топология (рис. 2.8, *г*) образуется в случае, когда каждый компьютер подключается непосредственно к общему центральному устройству, называемому **концентратором**[1]. В функции концентратора входит направление передаваемой компьютером информации одному или всем остальным компьютерам сети. В качестве концентратора может выступать как универсальный компьютер, так и специализированное устройство. К недостаткам топологии типа звезда относится более высокая стоимость сетевого оборудования из-за необходимости приобретения специализированного центрального устройства. Кроме того, возможности по наращиванию количества узлов в сети ограничиваются количеством портов концентратора. Иногда имеет смысл строить сеть с использованием нескольких концентраторов, иерархически соединенных между собой связями типа звезда (рис. 2.8, *д*). Получаемую в результате структуру называют **иерархической звездой**, а также **деревом**. В настоящее время дерево является самой распространенной топологией связей как в локальных, так и глобальных сетях.

Особым частным случаем звезды является конфигурация **общая шина** (рис. 2.8, *е*). Здесь в качестве центрального элемента выступает пассивный кабель, к которому по схеме «монтажного ИЛИ» подключается несколько компьютеров (такую же топологию имеют многие сети, использующие беспроводную связь — роль общей шины здесь играет общая радиосреда). Передаваемая информация распространяется по кабелю и доступна одновременно всем компьютерам, присоединенным к этому кабелю. Основными преимуществами такой схемы являются ее дешевизна и простота присоединения новых узлов к сети, а недостатками — низкая надежность (любой дефект кабеля полностью парализует всю сеть) и невысокая производительность (в каждый момент времени только один компьютер может передавать данные по сети, поэтому пропускная способность делится здесь между всеми узлами сети).

В то время как небольшие сети, как правило, имеют типовую топологию — звезда, кольцо или общая шина, для крупных сетей характерно наличие произвольных связей между компьютерами. В таких сетях можно выделить отдельные произвольно связанные фрагменты (подсети), имеющие типовую топологию, поэтому их называют сетями со **смешанной топологией** (рис. 2.9).

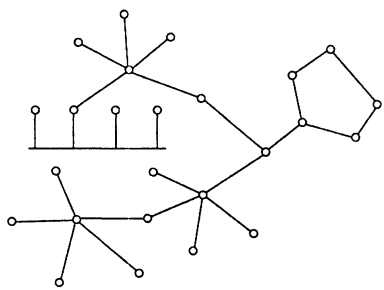


Рис. 2.9. Смешанная топология

[1] В данном случае термин «концентратор» используется в широком смысле, обозначая любое многовходовое устройство, способное служить центральным элементом, например коммутатор или маршрутизатор.

Адресация узлов сети

Еще одной новой проблемой, которую нужно учитывать при объединении трех и более компьютеров, является проблема их адресации, точнее адресации их сетевых интерфейсов[1]. Один компьютер может иметь несколько сетевых интерфейсов. Например, для создания полносвязной структуры из N компьютеров необходимо, чтобы у каждого из них имелся $N-1$ интерфейс.

По количеству адресуемых интерфейсов адреса можно классифицировать следующим образом:

- **уникальный адрес** (unicast) используется для идентификации отдельных интерфейсов;
- **групповой адрес** (multicast) идентифицирует сразу несколько интерфейсов, поэтому данные, помеченные групповым адресом, доставляются каждому из узлов, входящих в группу;
- данные, направленные по **широковещательному адресу** (broadcast), должны быть доставлены всем узлам сети;
- в новой версии протокола IPv6 определен **адрес произвольной рассылки** (anycast), который, так же как и групповой адрес, задает группу адресов, однако данные, посланные по этому адресу, должны быть доставлены не всем адресам данной группы, а любому из них.

Адреса могут быть **числовыми** (например, 129.26.255.255 или 81.1a.ff.ff) и **символьными** (site.domen.ru, willi-winki).

Символьные адреса (имена) предназначены для запоминания людьми и поэтому обычно несут смысловую нагрузку. Для работы в больших сетях символьное имя может иметь иерархическую структуру, например ftp-arch1.ucl.ac.uk. Этот адрес говорит о том, что данный компьютер поддерживает ftp-архив в сети одного из колледжей Лондонского университета (University College London — ucl) и эта сеть относится к академической ветви (ac) Интернета Великобритании (United Kingdom — uk). При работе в пределах сети Лондонского университета такое длинное символьное имя явно избыточно и вместо него можно пользоваться кратким символьным именем ftp-arch1. Хотя символьные имена удобны для людей, из-за переменного формата и потенциально большой длины их передача по сети не очень экономична.

Множество всех адресов, которые являются допустимыми в рамках некоторой схемы адресации, называется **адресным пространством**.

Адресное пространство может иметь плоскую (линейную) организацию (рис. 2.10) или иерархическую организацию (рис. 2.11).

При **плоской организации** множество адресов никак не структурировано. Примером плоского числового адреса является **MAC-адрес**, предназначенный для однозначной идентификации сетевых интерфейсов в локальных сетях. Такой

[1] Иногда вместо точного выражения «адрес сетевого интерфейса» мы будем использовать упрощенное — «адрес узла сети».

адрес обычно используется только аппаратурой, поэтому его стараются сделать по возможности компактным и записывают в виде двоичного или шестнадцатеричного числа, например 0081005e24a8. При задании MAC-адресов не требуется выполнение ручной работы, так как они обычно встраиваются в аппаратуру компанией-изготовителем, поэтому их называют также **аппаратными адресами** (hardware addresses). Использование плоских адресов является жестким решением — при замене аппаратуры, например, сетевого адаптера, изменяется и адрес сетевого интерфейса компьютера.

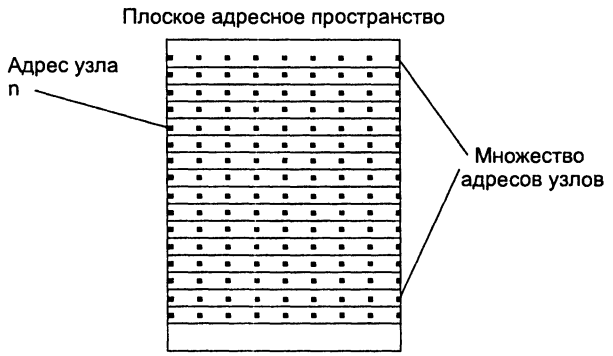


Рис. 2.10. Плоская организация адресного пространства

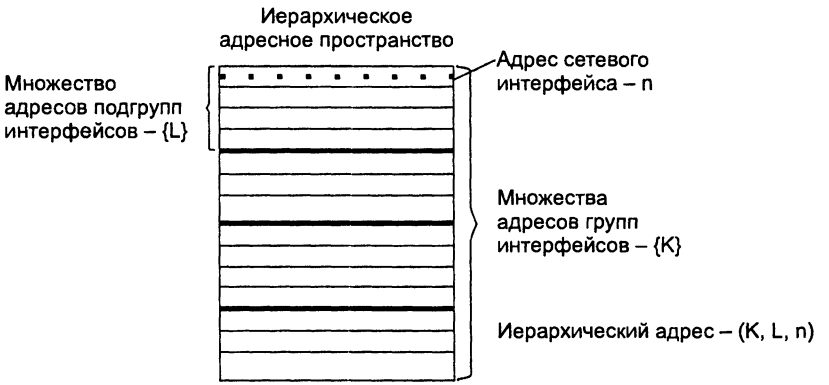


Рис. 2.11. Иерархическая организация адресного пространства

При **иерархической** организации адресное пространство организовано в виде вложенных друг в друга подгрупп, которые, последовательно сужая адресуемую область, в конце концов, определяют отдельный сетевой интерфейс.

В показанной на рис. 2.11 трехуровневой структуре адресного пространства адрес конечного узла задается тремя составляющими: идентификатором группы (K), в которую входит данный узел, идентификатором подгруппы (L) и, наконец, идентификатором узла (n), однозначно определяющим его в подгруппе. Иерархическая адресация во многих случаях оказывается более рациональной, чем плоская. В больших сетях, состоящих из многих тысяч узлов, использование

плоских адресов приводит к большим издержкам — конечным узлам и коммуникационному оборудованию приходится оперировать таблицами адресов, состоящими из тысяч записей. В противоположность этому иерархическая система адресации позволяет при перемещении данных до определенного момента пользоваться только старшей составляющей адреса (например, идентификатором группы K), затем для дальнейшей локализации адресата задействовать следующую по старшинству часть (L) и в конечном счете — младшую часть (n).

Типичными представителями иерархических числовых адресов являются сетевые IP- и IPX-адреса. В них поддерживается двухуровневая иерархия, адрес делится на старшую часть — номер сети и младшую — номер узла. Такое деление позволяет передавать сообщения между сетями только на основании номера сети, а номер узла используется после доставки сообщения в нужную сеть; точно так же, как название улицы используется почтальоном только после того, как письмо доставлено в нужный город.

На практике обычно применяют сразу несколько схем адресации, так что сетевой интерфейс компьютера может одновременно иметь несколько адресов-имен. Каждый адрес задействуется в той ситуации, когда соответствующий вид адресации наиболее удобен. А для преобразования адресов из одного вида в другой используются специальные вспомогательные протоколы, которые называют **протоколами разрешения адресов**.

Пользователи адресуют компьютеры иерархическими символьными именами, которые автоматически заменяются в сообщениях, передаваемых по сети, иерархическими числовыми адресами. С помощью этих числовых адресов сообщения передаются из одной сети в другую, а после доставки сообщения в сеть назначения вместо иерархического числового адреса используется плоский аппаратный адрес компьютера. Проблема установления соответствия между адресами различных типов может решаться как централизованными, так и распределенными средствами.

При *централизованном подходе* в сети выделяется один или несколько компьютеров (серверов имен), в которых хранится таблица соответствия имен различных типов, например символьных имен и числовых адресов. Все остальные компьютеры обращаются к серверу имен с запросами, чтобы по символьному имени найти числовой номер необходимого компьютера.

При *распределенном подходе* каждый компьютер сам хранит все назначенные ему адреса разного типа. Тогда компьютер, которому необходимо определить по известному иерархическому числовому адресу некоторого компьютера его плоский аппаратный адрес, посылает в сеть широковещательный запрос. Все компьютеры сети сравнивают содержащийся в запросе адрес с собственным. Тот компьютер, у которого обнаружилось совпадение, посылает ответ, содержащий искомый аппаратный адрес. Такая схема использована в **протоколе разрешения адресов** (Address Resolution Protocol, ARP) стека TCP/IP.

Достоинство распределенного подхода в том, что в этом случае не нужно выделять специальный компьютер, который к тому же часто требует ручного задания таблицы соответствия адресов. Недостатком его является необходимость широ-

ковещательных сообщений, перегружающих сеть. Именно поэтому распределенный подход используется в небольших сетях, а централизованный — в больших.

До сих пор мы говорили об адресах сетевых интерфейсов, компьютеров и коммуникационных устройств, однако конечной целью данных, пересылаемых по сети, являются не сетевые интерфейсы или компьютеры, а выполняемые на этих устройствах программы — процессы. Поэтому в адресе назначения наряду с информацией, идентифицирующей интерфейс устройства, должен указываться адрес процесса, которому предназначены посылаемые по сети данные. Очевидно, что достаточно обеспечить уникальность адреса процесса в пределах компьютера. Примером адресов процессов являются *номера портов TCP и UDP*, используемые в стеке TCP/IP.

Коммутация

Итак, пусть компьютеры физически связаны между собой в соответствии с некоторой топологией и выбрана система адресации. Остается нерешенной самая важная проблема: каким способом передавать данные между конечными узлами? Особую сложность приобретает эта задача, когда топология сети неполносвязная. В таком случае обмен данными между произвольной парой конечных узлов (пользователей) должен идти в общем случае через транзитные узлы.

Соединение конечных узлов через сеть транзитных узлов называют **коммутацией**. Последовательность узлов, лежащих на пути от отправителя к получателю, образует **маршрут**.

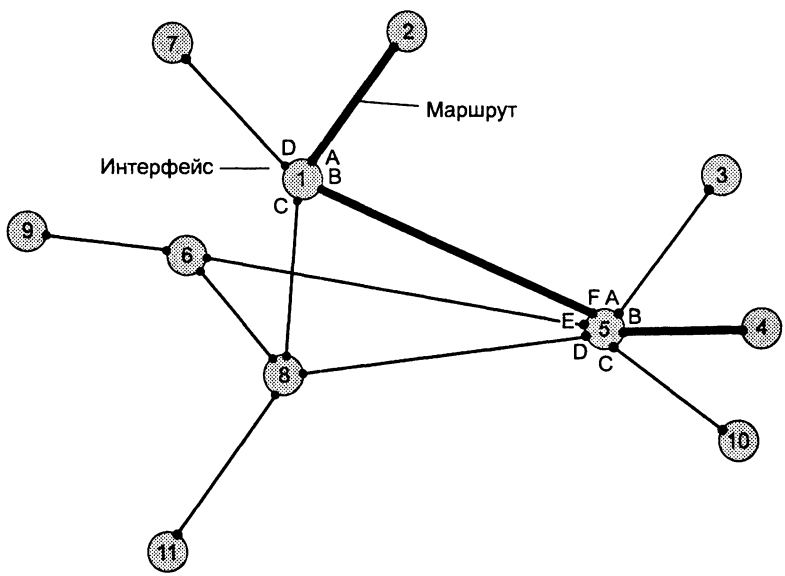


Рис. 2.12. Коммутация абонентов через сеть транзитных узлов

Например, в сети, показанной на рис. 2.12, узлы 2 и 4, непосредственно между собой не связанные, вынуждены передавать данные через транзитные узлы, в ка-

честве которых могут выступить, например, узлы 1 и 5. Узел 1 должен выполнить передачу данных между своими интерфейсами A и B, а узел 5 — между интерфейсами F и B. В данном случае маршрутом является последовательность: 2-1-5-4, где 2 — узел-отправитель, 1 и 5 — транзитные узлы, 4 — узел-получатель.

Обобщенная задача коммутации

Список ключевых слов: информационный поток, подпоток, адрес назначения, глобальная и локальные метки потока, метрика, таблица коммутации, коммутация интерфейсов, коммутатор, коммутационная сеть, мультиплексирование и демультиплексирование, разделение времени, частотное разделение, мультиплексор и демультиплексор, разделяемая среда передачи данных, арбитр, коммутация каналов и пакетов.

В самом общем виде задача коммутации может быть представлена в виде следующих взаимосвязанных частных задач.

1. Определение информационных потоков, для которых требуется прокладывать маршруты.
2. Маршрутизация потоков.
3. Продвижение потоков, то есть распознавание потоков и их локальная коммутация на каждом транзитном узле.
4. Мультиплексирование и демультиплексирование потоков.

Определение информационных потоков

Понятно, что через один транзитный узел может проходить несколько маршрутов, например, через узел 5 (см. рис. 2.12) проходят как минимум все данные, направляемые узлом 4 каждому из остальных узлов, а также все данные, поступающие в узлы 3, 4 и 10. Транзитный узел должен уметь *распознавать* поступающие на него потоки данных, для того чтобы обеспечивать передачу каждого из них именно на тот свой интерфейс, который ведет к нужному узлу.

Информационным потоком, или потоком данных, называют непрерывную последовательность данных, объединенных набором общих признаков, выделяющих их из общего сетевого трафика.

Например, как поток можно определить все данные, поступающие от одного компьютера; объединяющим признаком в данном случае служит адрес источника. Эти же данные можно представить как совокупность нескольких **подпотоков**, каждый из которых в качестве дифференцирующего признака имеет адрес назначения. Наконец, каждый из этих подпотоков, в свою очередь, можно разделить на более мелкие подпотоки, порожденные разными сетевыми приложениями — электронной почтой, программой копирования файлов, веб-сервером. Данные, образующие поток, могут быть представлены в виде различных информационных единиц данных — пакетов, кадров или ячеек.

ПРИМЕЧАНИЕ

В англоязычной литературе для потоков данных, передающихся с равномерной и неравномерной скоростью, обычно используют разные термины — соответственно «data stream» и «data flow». Например, при передаче веб-страницы через Интернет предложенная нагрузка представляет собой неравномерный поток данных, а при вещании музыки интернет-станцией — равномерный. Для сетей передачи данных характерна неравномерная скорость передачи, поэтому далее в большинстве ситуаций под термином «поток данных» мы будем понимать именно неравномерный поток данных и указывать на равномерный характер этого процесса только тогда, когда это нужно подчеркнуть.

Очевидно, что при коммутации в качестве обязательного признака выступает адрес назначения данных. На основании этого признака весь поток входящих в транзитный узел данных разделяется на подпотоки, каждый из которых передается на интерфейс, соответствующий маршруту продвижения данных.

Адрес источника и адрес назначения определяют поток для пары соответствующих конечных узлов. Однако часто бывает полезно представить этот поток в виде нескольких подпотоков, причем для каждого из них может быть проложен свой особый маршрут. Рассмотрим пример, когда на одной и той же паре конечных узлов выполняется несколько взаимодействующих по сети приложений, каждое из которых предъявляет к сети свои особые требования. В таком случае выбор маршрута должен осуществляться с учетом характера передаваемых данных, например, для файлового сервера важно, чтобы передаваемые им большие объемы данных направлялись по каналам, обладающим высокой пропускной способностью, а для программной системы управления, которая посылает в сеть короткие сообщения, требующие обязательной и немедленной отработки, при выборе маршрута более важна надежность линии связи и минимальный уровень задержек на маршруте. Кроме того, даже для данных, предъявляющих к сети одинаковые требования, может прокладываться несколько маршрутов, чтобы за счет распараллеливания ускорить передачу данных.

Признаки потока могут иметь *глобальное* или *локальное* значение — в первом случае они однозначно определяют поток в пределах всей сети, а во втором — в пределах одного транзитного узла. Пара адресов конечных узлов для идентификации потока — это пример глобального признака. Примером признака, локально определяющего поток в пределах устройства, может служить номер (идентификатор) интерфейса данного устройства, на который поступили данные. Например, возвращаясь к рис. 2.12, узел 1 может быть настроен передавать все данные, поступившие с интерфейса A, на интерфейс B, а данные, поступившие с интерфейса D, на интерфейс C. Такое правило позволяет отделить поток данных узла 2 от потока данных узла 7 и направлять их для транзитной передачи через разные узлы сети, в данном случае поток узла 2 — через узел 5, а поток узла 7 — через узел 8.

Метка потока — это особый тип признака. Она представляет собой некоторое число, которое несут все данные потока. **Глобальная метка** назначается данным потока и не меняет своего значения на всем протяжении его пути следования от узла источника до узла назначения, таким образом она уникально определяет поток в пределах сети. В некоторых технологиях используются **локальные мет-**

ки потока, динамически меняющие свое значение при передаче данных от одного узла к другому.

Таким образом, распознавание потоков во время коммутации происходит на основании признаков, в качестве которых, помимо обязательного адреса назначения данных, могут выступать и другие признаки, такие, например, как идентификаторы приложений.

Маршрутизация

Задача маршрутизации, в свою очередь, включает в себя две подзадачи:

- определение маршрута;
- оповещение сети о выбранном маршруте.

Определить маршрут — это значит выбрать последовательность транзитных узлов и их интерфейсов, через которые надо передавать данные, чтобы доставить их адресату. Определение маршрута — сложная задача, особенно когда конфигурация сети такова, что между парой взаимодействующих сетевых интерфейсов существует множество путей. Чаще всего выбор останавливают на одном *оптимальном*[1] по некоторому критерию маршруте. В качестве критериев оптимальности могут выступать, например, номинальная пропускная способность и загруженность каналов связи; задержки, вносимые каналами; количество промежуточных транзитных узлов; надежность каналов и транзитных узлов.

Но даже в том случае, когда между конечными узлами существует только *один* путь, при сложной топологии сети его нахождение может представлять собой нетривиальную задачу.

Маршрут может определяться эмпирически («вручную») администратором сети на основании различных, часто не формализуемых соображений. Среди побудительных мотивов выбора пути могут быть: особые требования к сети со стороны различных типов приложений, решение передавать трафик через сеть определенного поставщика услуг, предположения о пиковых нагрузках на некоторые каналы сети, соображения безопасности.

Однако эмпирический подход к определению маршрутов мало пригоден для большой сети со сложной топологией. В этом случае используются автоматические методы определения маршрутов. Для этого конечные узлы и другие устройства сети оснащаются специальными программными средствами, которые организуют взаимный обмен служебными сообщениями, позволяющий каждому узлу составить свое «представление» о сети. Затем на основе собранных данных программными методами определяются рациональные маршруты.

При выборе маршрута часто ограничиваются только информацией о топологии сети. Этот подход иллюстрирует рис. 2.13. Для передачи трафика между конечными узлами A и C существуют два альтернативных маршрута: A-1-2-3-C и A-1-3-C. Если мы учитываем только топологию, то выбор очевиден — маршрут A-1-3-C, который имеет меньше транзитных узлов.

---

[1] На практике для снижения объема вычислений ограничиваются поиском не оптимального в математическом смысле, а рационального, то есть близкого к оптимальному, маршрута.

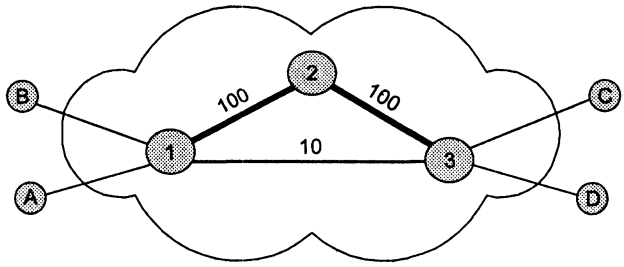


Рис. 2.13. Выбор маршрута

Решение было найдено путем минимизации критерия, в качестве которого в данном примере выступала длина маршрута, измеренная количеством транзитных узлов. Однако, возможно, наш выбор был не самым лучшим. На рисунке показано, что каналы 1-2 и 2-3 обладают пропускной способностью 100 Мбит/с, а канал 1-3 — только 10 Мбит/с. Если мы хотим, чтобы наша информация передавалась по сети с максимально возможной скоростью, то нам следовало бы выбрать маршрут A-1-2-3-C, хотя он и проходит через большее количество промежуточных узлов. То есть можно сказать, что маршрут A-1-2-3-C в данном случае оказывается «более коротким».

Абстрактный способ измерения степени близости между двумя объектами называется **метрикой**. Так, для измерения длины маршрута могут быть использованы разные метрики — количество транзитных узлов, как в предыдущем примере, линейная протяженность маршрута и даже его стоимость в денежном выражении. Для построения метрики, учитывающей пропускную способность, часто используют следующий прием: длину каждого канала-участка характеризуют величиной, обратной его пропускной способности. Чтобы оперировать целыми числами, выбирают некоторую константу, заведомо большую, чем пропускные способности каналов в сети. Например, если мы в качестве такой константы выберем 100 Мбит/с, то метрика каждого из каналов 1-2 и 2-3 равна 1, а метрика канала 1-3 равна 10. Метрика маршрута равна сумме метрик составляющих его каналов, поэтому часть пути 1-2-3 обладает метрикой 2, а альтернативная часть пути1-3 — метрикой 10. Мы выбираем более «короткий» путь, то есть путь A-1-2-3-C.

Описанные подходы к выбору маршрутов не учитывают текущую степень загруженности каналов трафиком[1]. Используя аналогию с автомобильным трафиком, можно сказать, что мы выбирали маршрут по карте, учитывая количество промежуточных городов и ширину дороги (аналог пропускной способности канала), отдавая предпочтение скоростным магистралям. Но мы не стали слушать радио или телевизионную программу, которая сообщает о текущих заторах на дорогах. Так что наше решение могло оказаться отнюдь не лучшим, если по маршруту A-1-2-3-C уже передается большое количество потоков, а маршрут A-1-3-C практически свободен.

После того как маршрут определен (вручную или автоматически), надо *оповестить* о нем все устройства сети. Сообщение о маршруте должно нести каждо-

[1] Такие методы, в которых используется информация о текущей загруженности каналов связи, позволяют определять более рациональные маршруты, однако требуют интенсивного обмена служебной информацией между узлами сети.

му транзитному устройству примерно такую информацию: «каждый раз, когда в устройство поступят данные, относящиеся к потоку n, их следует передать для дальнейшего продвижения на интерфейс F». Каждое подобное сообщение о маршруте обрабатывается устройством, в результате создается новая запись в **таблице коммутации**. В этой таблице локальному или глобальному признаку (признакам) потока (например, метке, номеру входного интерфейса или адресу назначения) ставится в соответствие номер интерфейса, на который устройство должно передавать данные, относящиеся к этому потоку.

Таблица 2.1 является фрагментом таблицы коммутации, содержащий запись, сделанную на основании сообщения о необходимости передачи потока n на интерфейс F.

Таблица 2.1. Фрагмент таблицы коммутации

Признаки потока	Направление передачи данных (номер интерфейса и/или адрес следующего узла)
...	...
n	F
...	...

Конечно, детальное описание структуры сообщения о маршруте и содержимого таблицы коммутации зависит от конкретной технологии, однако эти особенности не меняют сущности рассматриваемых процессов.

Передача информации транзитным устройствам о выбранных маршрутах, так же как и определение маршрута, может осуществляться и вручную, и автоматически. Администратор сети может зафиксировать маршрут, выполнив в ручном режиме конфигурирование устройства, например, жестко скоммутировав на длительное время определенные пары входных и выходных интерфейсов (как работали «телефонные барышни» на первых коммутаторах). Он может также по собственной инициативе внести запись о маршруте в таблицу коммутации.

Однако поскольку топология и состав информационных потоков может меняться (отказы узлов или появление новых промежуточных узлов, изменение адресов или определение новых потоков), гибкое решение задач определения и задания маршрутов предполагает постоянный анализ состояния сети и обновление маршрутов и таблиц коммутации. В таких случаях задачи прокладки маршрутов, как правило, не могут быть решены без достаточно сложных программных и аппаратных средств.

Продвижение данных

Итак, пусть маршруты определены, записи о них сделаны в таблицах всех транзитных узлов, все готово к выполнению основной операции — передаче данных между абонентами (коммутации абонентов).

Для каждой пары абонентов эта операция может быть представлена несколькими (по числу транзитных узлов) *локальными* операциями коммутации. Прежде всего, отправитель должен выставить данные на тот свой интерфейс, с которого начинается найденный маршрут, а все транзитные узлы должны соответствующим образом выполнить «переброску» данных с одного своего интерфейса на другой, другими словами, выполнить **коммутацию интерфейсов**. Устройство, функциональным назначением которого является коммутация, называется **коммутатором** (рис. 2.14).

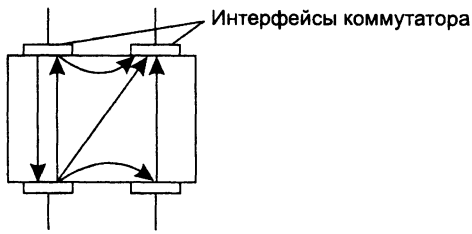


Рис. 2.14. Коммутатор

Однако прежде чем выполнить коммутацию, коммутатор должен распознать поток. Для этого поступившие данные анализируются на предмет наличия в них признаков какого-либо из потоков, заданных в таблице коммутации. Если произошло совпадение, то эти данные направляются на интерфейс, определенный для них в маршруте.

ВНИМАНИЕ

Термины «коммутация», «таблица коммутации» и «коммутатор» в телекоммуникационных сетях могут трактоваться неоднозначно. Мы уже определили коммутацию как процесс соединения абонентов сети через транзитные узлы. Этим же термином мы обозначаем и соединение интерфейсов в пределах отдельного транзитного узла. Коммутатором в широком смысле называется устройство любого типа, способное выполнять операции переключения потока данных с одного интерфейса на другой. Операция коммутации может быть выполнена в соответствии с различными правилами и алгоритмами. Некоторые способы коммутации и соответствующие им таблицы и устройства получили специальные названия. Например, в технологиях сетевого уровня, таких как IP и IPX, для обозначения аналогичных понятий используются термины «маршрутизация», «таблица маршрутизации», «маршрутизатор». В то же время за другими специальными типами коммутации и соответствующими устройствами закрепились те же самые названия «коммутация», «таблица коммутации» и «коммутатор», используемые в узком смысле, например, как коммутация и коммутатор локальной сети. Для телефонных сетей, которые появились намного раньше компьютерных, также характерна аналогичная терминология, коммутатор является здесь синонимом телефонной станции. Из-за солидного возраста и гораздо большей (пока) распространенности телефонных сетей чаще всего в телекоммуникациях под термином «коммутатор» понимают именно телефонный коммутатор.

Коммутатором может быть как специализированное устройство, так и универсальный компьютер со встроенным программным механизмом коммутации, в этом случае коммутатор называется программным. Компьютер может совмещать функции коммутации данных с выполнением своих обычных функций как конечного узла. Однако во многих случаях более рациональным является решение, в соот-

ветствии с которым некоторые узлы в сети выделяются *специально* для коммутации. Эти узлы образуют **коммутационную сеть**, к которой подключаются все остальные. На рис. 2.15 показана коммутационная сеть, образованная из узлов 1, 5, 6 и 8, к которой подключаются конечные узлы 2, 3, 4, 7, 9 и 10.

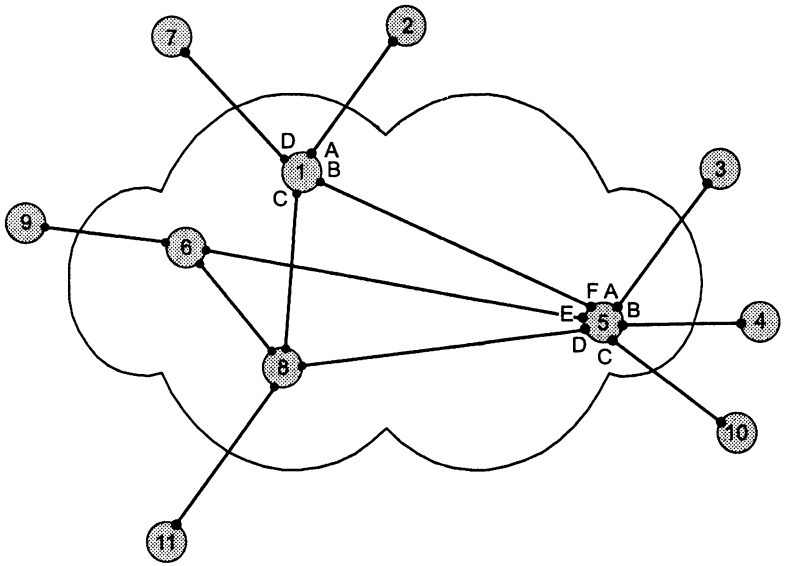


Рис. 2.15. Коммутационная сеть

Мультиплексирование и демультиплексирование

Чтобы определить, на какой интерфейс следует передать поступившие данные, коммутатор должен определить, к какому потоку они относятся. Эта задача должна решаться независимо от того, поступает на вход коммутатора только один «чистый» поток или «смешанный» поток, являющийся результатом агрегирования нескольких потоков. В последнем случае к задаче распознавания потоков добавляется задача **демультиплексирования**, то есть разделения суммарного агрегированного потока на несколько составляющих его потоков.

Как правило, операцию коммутации сопровождает также обратная операция — **мультиплексирование**. При мультиплексировании из нескольких отдельных потоков образуется общий агрегированный поток, который можно передавать по одному физическому каналу связи.

Операции мультиплексирования/демультиплексирования имеют такое же важное значение в любой сети, как и операции коммутации, потому что без них пришлось бы для каждого потока предусматривать отдельный канал, что привело бы к большому количеству параллельных связей в сети и свело бы «на нет» все преимущества неполносвязной сети.

На рис. 2.16 показан фрагмент сети, состоящий из трех коммутаторов. Коммутатор 1 имеет пять сетевых интерфейсов. Рассмотрим, что происходит на интер-

фейсе Инт. 1. Сюда поступают данные с трех интерфейсов — Инт. 3, Инт. 4 и Инт. 5. Все их надо передать в общий физический канал, то есть выполнить операцию мультиплексирования. Мультиплексирование является способом разделения имеющегося одного физического канала между несколькими одновременно протекающими сеансами связи между абонентами сети.

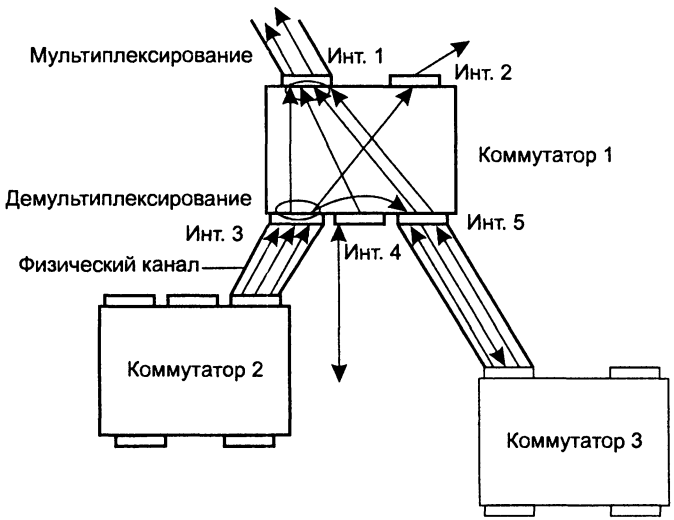


Рис. 2.16. Операции мультиплексирования и демультиплексирования потоков при коммутации

Одним из основных способов мультиплексирования потоков является **разделение времени**. При этом способе каждый поток время от времени (с фиксированным или случайным периодом) получает физический канал в полное свое распоряжение и передает по нему свои данные. Распространено также **частотное разделение** канала, когда каждый поток передает данные в выделенном ему частотном диапазоне.

Технология мультиплексирования должна позволять получателю такого суммарного потока выполнять обратную операцию — разделение (демультиплексирование) данных на слагаемые потоки. На интерфейсе Инт. 3 коммутатор выполняет демультиплексирование потока на три составляющих его подпотока. Один из них он передает на интерфейс Инт. 1, другой — на Инт. 2, а третий — на Инт. 5. А вот на интерфейсе Инт. 2 нет необходимости выполнять мультиплексирование или демультиплексирование — этот интерфейс выделен одному потоку в монопольное использование. Вообще говоря, на каждом интерфейсе могут одновременно выполняться обе функции — мультиплексирования и демультиплексирования.

Частный случай коммутатора, у которого все входящие информационные потоки коммутируются на один выходной интерфейс, где они мультиплексируются в один агрегированный поток, называется **мультиплексором** (рис. 2.17, *а*). Коммутатор, который имеет один входной интерфейс и несколько выходных, называется **демультиплексором** (рис. 2.17, *б*).

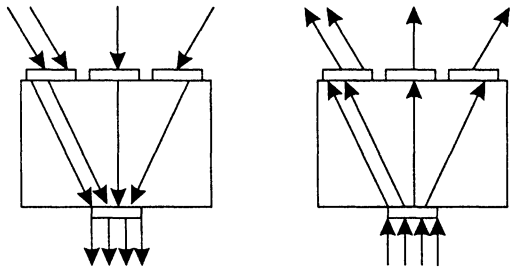


Рис. 2.17. Мультиплексор и демультиплексор

Разделяемая среда передачи данных

Еще одним параметром разделяемого канала связи является *количество подключенных к нему узлов*. В приведенных выше примерах к каждому каналу связи подключались только два взаимодействующих узла, точнее — два интерфейса (рис. 2.18, *а* и *б*). В телекоммуникационных сетях используется и другой вид подключения, когда к одному каналу подключается несколько интерфейсов (рис. 2.18, *в*). Такое множественное подключение интерфейсов порождает уже рассматривавшуюся выше топологию «общая шина», иногда называемую также шлейфовым подключением. Во всех этих случаях возникает проблема организации совместного использования канала несколькими интерфейсами.

Возможны различные варианты разделения каналов связи между интерфейсами. На рис. 2.18, *а* коммутаторы К1 и К2 связаны двумя однонаправленными физическими каналами, то есть такими, по которым информация может передаваться только в одном направлении. В этом случае передающий интерфейс является активным, и физическая среда передачи находится под его управлением. Пассивный интерфейс только принимает данные. *Проблема разделения канала между интерфейсами здесь отсутствует.* (Заметим, однако, что задача мультиплексирования потоков данных в канале при этом сохраняется.) На практике два однонаправленных канала, реализующие в целом дуплексную связь между двумя устройствами, обычно рассматриваются как один дуплексный канал, а пара интерфейсов одного устройства — как передающая и принимающая части одного и того же интерфейса.

На рис. 2.18, *б* коммутаторы К1 и К2 связаны каналом, который может передавать данные в обе стороны, но только попеременно. При этом *возникает необходимость в механизме синхронизации доступа* интерфейсов К1 и К2 к такому каналу. Обобщением этого варианта является случай, показанный на рис. 2.18, *в*, когда к каналу связи подключаются несколько (больше двух) интерфейсов, образуя общую шину.

Совместно используемый несколькими интерфейсами физический канал называют **разделяемым** (shared). Часто применяют также термин **разделяемая среда передачи данных**[1]. Разделяемые каналы связи требуются не только для связей

[1] Нужно подчеркнуть, что термин «разделяемая среда» традиционно относят именно к случаю разделения канала между интерфейсами и практически никогда — к случаю разделения канала между потоками.

типа коммутатор–коммутатор, но и для связей компьютер-коммутатор и компьютер–компьютер.

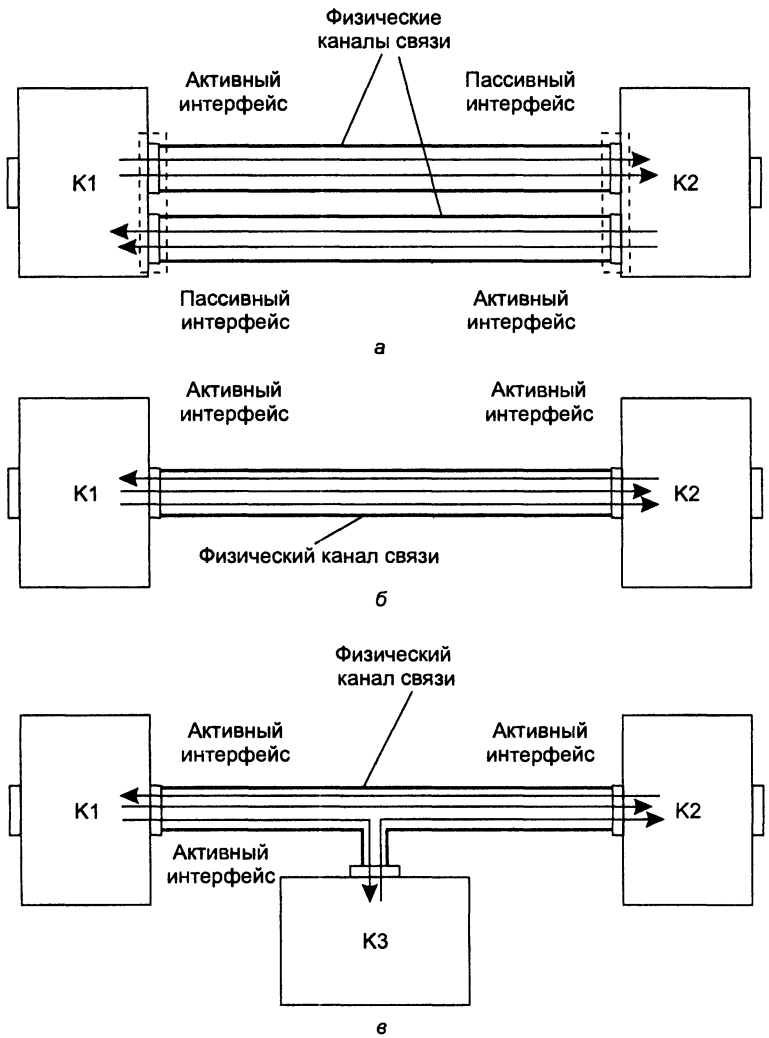


Рис. 2.18. Совместное использование канала связи

Существуют различные способы решения задачи организации совместного доступа к разделяемым линиям связи. Одни из них подразумевают централизованный подход, когда доступом управляет специальное устройство — **арбитр**, другие — децентрализованный. Внутри компьютера проблемы разделения линий связи между различными модулями также существуют — примером является доступ к системной шине, которым управляет либо процессор, либо специальный арбитр шины. В сетях организация совместного доступа к линиям связи имеет свою специфику из-за существенно большего времени распространения сигналов по линиям связи, поэтому процедуры согласования доступа к линии связи могут за-

нимать слишком большой промежуток времени и приводить к значительным потерям производительности сети. Именно по этой причине разделяемые между интерфейсами среды практически не используются в глобальных сетях.

В локальных же сетях разделяемые среды используются достаточно часто благодаря простоте и экономичности их реализации. Этот подход, в частности, применяется в доминирующей сегодня в локальных сетях технологии Ethernet, а также в популярных в прошлом технологиях Token Ring и FDDI.

Однако в последние годы стала преобладать другая тенденция — отказ от разделяемых сред передачи данных и в локальных сетях. Это связано с тем, что за достигаемое таким образом удешевление сети приходится расплачиваться производительностью.

ВНИМАНИЕ

Сеть с разделяемой средой при большом количестве узлов будет работать всегда медленнее, чем аналогичная сеть с индивидуальными двухточечными линиями связями, так как пропускная способность линии связи при ее совместном использовании делится между несколькими компьютерами сети.

И тем не менее не только в классических, но и в некоторых совсем новых технологиях, разработанных для локальных сетей, сохраняется режим разделяемых линий связи. Например, разработчики технологии Gigabit Ethernet, принятой в 1998 году в качестве нового стандарта, включили режим разделения среды в свои спецификации наряду с режимом работы по индивидуальным линиям связи.

Типы коммутации

Комплекс технических решений обобщенной задачи коммутации в своей совокупности составляет основу любой сетевой технологии. Как уже отмечалось, к этим частным задачам относятся:

- определение потоков и соответствующих маршрутов;
- фиксация маршрутов в конфигурационных параметрах и таблицах сетевых устройств;
- распознавание потоков и передача данных между интерфейсами одного устройства;
- мультиплексирование/демультиплексирование потоков;
- разделение среды передачи.

Среди множества возможных подходов к решению задачи коммутации абонентов в сетях выделяют два основополагающих, к которым относят **коммутацию каналов** и **коммутацию пакетов**.

Сети с коммутацией каналов имеют более богатую историю, они происходят от первых телефонных сетей. Сети с коммутацией пакетов сравнительно молоды, они появились в конце 60-х годов как результат экспериментов с первыми глобальными компьютерными сетями. Каждая из этих схем имеет свои достоинства

и недостатки, но по долгосрочным прогнозам многих специалистов будущее принадлежит технологии коммутации пакетов, как более гибкой и универсальной.

Пример

Поясним достаточно абстрактное описание обобщенной модели коммутации на примере работы традиционной почтовой службы. Почта также работает с потоками, которые в данном случае составляют почтовые отправления. Основным признаком почтового потока является адрес получателя. Для упрощения будем рассматривать в качестве адреса только страну, например, Индия, Норвегия, Россия, Бразилия и т. д. Дополнительным признаком потока может служить особое требование к надежности или скорости доставки. Например, пометка «Avia» на почтовых отправлениях в Бразилию выделит из общего потока почты в Бразилию подпоток, который будет доставляться самолетом.

Для каждого потока почтовая служба должна определить маршрут, который будет проходить через последовательность почтовых отделений, являющихся аналогами коммутаторов. В результате многолетней работы почтовой службы уже определены маршруты для большинства адресов назначения. Иногда возникают новые маршруты, связанные с появлением новых возможностей — политических, транспортных, экономических. После выбора нового маршрута нужно оповестить о нем сеть почтовых отделений. Как видно, эти действия очень напоминают работу телекоммуникационной сети. Информация о выбранных маршрутах следования почты представлена в каждом почтовом отделении в виде таблицы, в которой задано соответствие между страной назначения и следующим почтовым отделением. Например, в почтовом отделении города Саратова все письма, адресованные в Индию, направляются в почтовое отделение Ашхабада, а письма, адресованные в Норвегию, — в почтовое отделение Санкт-Петербурга. Такая таблица направлений доставки почты является прямой аналогией таблицы коммутации коммуникационной сети.

Каждое почтовое отделение работает подобно коммутатору. Все поступающие от абонентов и других почтовых отделений почтовые отправления сортируются, то есть происходит распознавание потоков. После этого почтовые отправления, принадлежащие одному «потоку», упаковываются в мешок, для которого в соответствии с таблицей направлений определяется следующее по маршруту почтовое отделение.

Выводы

Для того чтобы пользователь сети получил возможность использовать ресурсы «чужих» компьютеров, таких как диски, принтеры, плоттеры, необходимо дополнить все компьютеры сети специальными средствами. В каждом компьютере функции передачи данных в линию связи выполняют совместно аппаратный модуль, называемый сетевым адаптером, или сетевой интерфейсной картой, и управляющая программа — драйвер. Задачи более высокого уровня — формирование запросов к ресурсам и их выполнение — решают соответственно клиентские и серверные модули ОС.

Даже в простейшей сети, состоящей из двух компьютеров, возникают проблемы физической передачи сигналов по линиям связи: кодирование и модуляция, синхронизация передающего и принимающего устройств, контроль корректности переданных данных.

Важными характеристиками, связанными с передачей трафика через физические каналы, являются: предложенная нагрузка, скорость передачи данных, пропускная способность, емкость канала связи, полоса пропускания.

При связывании в сеть более двух компьютеров возникают проблемы выбора топологии (полносвязной, звезды, кольца, общей шины, иерархического дерева, произвольной); способа адресации (плоского или иерархического, числового или символьного); способа разделения линий связи и механизма коммутации.

В неполносвязных сетях соединение пользователей осуществляется путем коммутации через сеть транзитных узлов. При этом должны быть решены следующие задачи: определение потоков данных и маршрутов для них, продвижение данных в каждом транзитном узле, мультиплексирование и демультиплексирование потоков.

Среди множества возможных подходов к решению задачи коммутации выделяют два основополагающих — коммутацию каналов и пакетов.

Вопросы и задания

1. Какая информация передается по каналу, связывающему внешние интерфейсы компьютера и периферийного устройства?
2. Какие компоненты включает интерфейс устройства?
3. Какие задачи решает ОС при обмене с периферийным устройством?
4. Какие функции возлагаются на драйвер периферийного устройства?
5. Дайте определение понятия «топология».
6. К какому типу топологии можно отнести структуру, образованную тремя связанными друг с другом узлами (в виде треугольника)?
7. К какому типу топологии можно отнести структуру, образованную четырьмя связанными друг с другом узлами (в виде квадрата)?
8. К какому типу топологии можно отнести структуру, образованную тремя последовательно соединенными друг с другом узлами (последний не связан с первым)?
9. Частным случаем какой топологии является общая шина:
   - полносвязная;
   - кольцо;
   - звезда.
10. Какая из известных топологий обладает повышенной надежностью?
11. Какой тип топологии наиболее распространен сегодня в локальных сетях?
12. Какие требования предъявляются к системе адресации?
13. К какому типу можно отнести следующие адреса:
    - www.olifer.net;
    - 20-34-a2-00-c2-27;
    - 128.145.23.170.
14. Чем неравномерный поток данных отличается от равномерного?
15. Какие параметры передаваемых данных могут служить признаком потока?

16. Какие из утверждений о маршруте, на ваш взгляд, не всегда верны:
    - маршрут — это последовательность промежуточных узлов (интерфейсов), которые проходят данные по пути от отправителя к получателю;
    - при определении маршрута всегда выбирается один из нескольких возможных путей;
    - каждый маршрут назначается для определенного потока данных;
    - из нескольких возможных маршрутов всегда выбирается оптимальный.
17. Опишите основные подходы и критерии, используемые при выборе маршрута.
18. Какие из этих утверждений могут быть в некоторых случаях верными:
    - маршруты фиксируются в коммутаторах путем жесткого соединения пар интерфейсов;
    - маршруты определяются администратором и заносятся вручную в специальную таблицу;
    - таблица маршрутов строится автоматически сетевым программно-аппаратным обеспечением;
    - для каждого коммутатора строится своя таблица маршрутов, которая на нем и хранится.
19. Какое из этих устройств можно назвать коммутатором:
    - электрический выключатель;
    - автоматическая телефонная станция;
    - маршрутизатор;
    - мост;
    - мультиплексор;
    - ни одно из названных.
20. Какие методы используются при мультиплексировании?
21. Объясните различия между разделением среды передачи и мультиплексированием.
22. Опишите, какие основные задачи нужно решить, чтобы обеспечить информационное взаимодействие любой пары абонентов в коммуникационной сети любого типа.
23. Как представление общего городского трафика в виде нескольких различных потоков позволяет рационализировать управление городским транспортом?
24. Пусть в сети существует несколько маршрутов между двумя конечными узлами A и B. Перечислите достоинства и недостатки следующих вариантов передачи данных между этими узлами:
    - использовать все имеющиеся маршруты для параллельной передачи данных;
    - передавать все данные по одному оптимальному по некоторому критерию маршруту;
    - использовать несколько маршрутов из набора всех возможных маршрутов и разделять между ними передаваемые данные.

    Какое правило можно применить для определения маршрута передачи очередного пакета в последнем из перечисленных случаев?

ГЛАВА 3 Коммутация пакетов и каналов

В этой главе продолжается исследование общих принципов коммутации в телекоммуникационных сетях. Мы детально изучим и сравним два основных типа коммутации — коммутацию каналов и коммутацию пакетов.

Исторически коммутация каналов появилась намного раньше коммутации пакетов и ведет свое происхождение от первых телефонных сетей. Невозможность динамического перераспределения пропускной способности физического канала является принципиальным ограничением сети с коммутацией каналов.

Принцип коммутации пакетов был изобретен разработчиками компьютерных сетей. Коммутация пакетов учитывает особенности компьютерного трафика и является более эффективным способом коммутации для компьютерных сетей по сравнению с традиционным методом коммутации каналов, применяющимся в телефонных сетях.

Однако достоинства и недостатки любой сетевой технологии — относительны. Наличие буферной памяти в коммутаторах пакетных сетей позволяет эффективно использовать пропускную способность каналов при передаче пульсирующего трафика, но приводит к случайным задержкам в доставке пакетов, что является недостатком для трафика реального времени, который традиционно передается с помощью техники коммутации каналов.

В главе рассматриваются три метода продвижения пакетов, используемые в сетях с коммутацией пакетов: дейтаграммная передача, передача с установлением логического соединения и техника виртуальных каналов.

Заканчивается глава изучением принципа разделяемой среды, широко используемого в локальных сетях.

Коммутация каналов

Список ключевых слов: линия связи, установление соединения, запрос, трафик реального времени, подканал, коэффициент пульсации трафика.

Чтобы объяснить основную идею коммутации каналов, рассмотрим ее в максимально упрощенном виде. Как показано на рис. 3.1, коммутационная сеть состоит из коммутаторов (S1, S2, ..., S5), связанных между собой **линиями связи**. Каждая линия имеет одну и ту же пропускную способность.

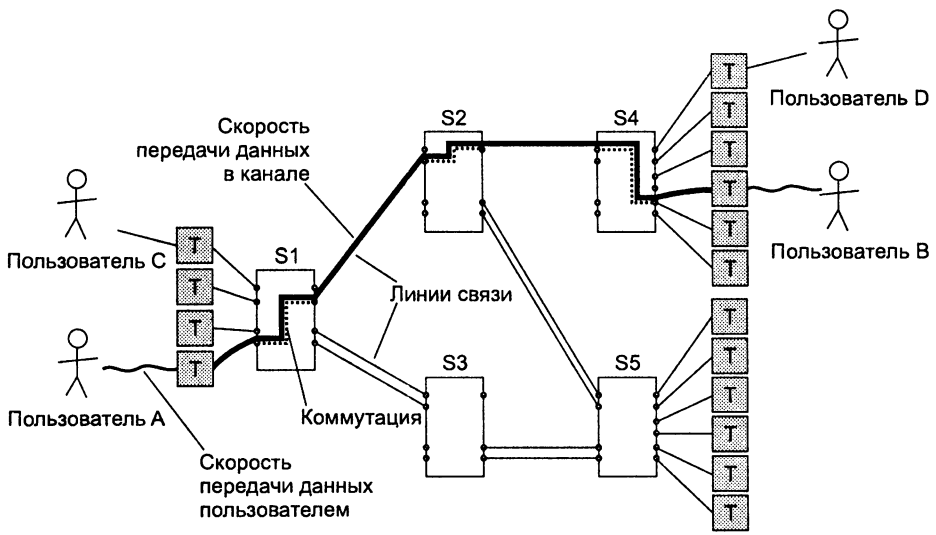


Рис. 3.1. Коммутация каналов без мультиплексирования

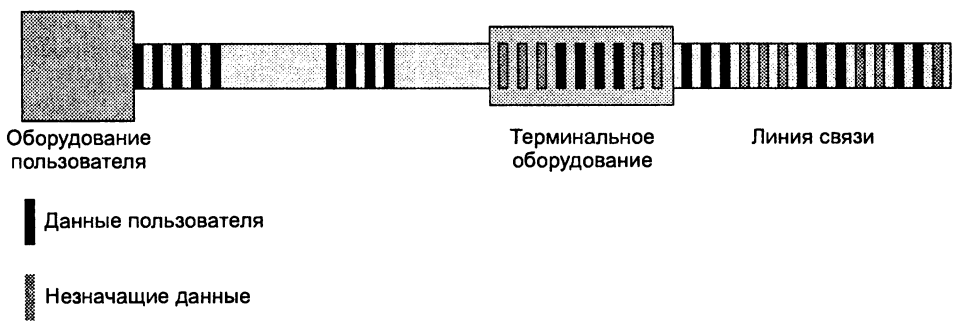


Рис. 3.2. Дополнение потока до пропускной способности линии

Каждый абонент подключается к сети с помощью терминального устройства (T), которое посылает данные в сеть с постоянной скоростью, причем эта скорость в точности равна пропускной способности линий. Если в какие-то периоды времени у абонента скорость информации, которую он хочет передать в сеть (предложенная нагрузка), оказывается меньше пропускной способности линии, то

терминальное устройство продолжает питать сеть постоянным потоком данных, дополняя полезную информацию пользователя «пустыми» (незначащими) данными (рис. 3.2). О том, что часть потока битов несет полезную информацию, а часть является «заполнителем», знает терминальное устройство получателя, которое должно отбрасывать незначащую информацию и передавать пользователю только те данные, которые посылал в сеть отправитель.

Так как все мы — многолетние пользователи телефонной сети, которая является наиболее распространенным представителем сетей с коммутацией каналов, то мы будем сопровождать наше объяснение примерами из области *телефонии*.

Установление соединения

Обмен данными начинается с предварительного **установления соединения**.

Пусть два абонента A и B хотят передать друг другу некоторые данные (см. рис. 3.1). Прежде чем отправить данные в сеть (начать разговор), абонент A посылает **запрос** в коммутационную сеть, в котором указывается адрес (телефонный номер) абонента B. Цель посылки запроса — установить соединение абонентов A и B информационным каналом, свойства которого подобны свойствам непрерывной линии связи: на всем своем протяжении он передает данные с одной и той же скоростью. Это означает, что в транзитных коммутаторах *нет необходимости буферизовать данные* пользователей.

Для создания такого канала запрос должен пройти через последовательность коммутаторов, лежащих на пути от A к B, и удостовериться, что все необходимые отрезки пути (линии связи) в данный момент свободны. Кроме того, для успешного соединения необходимо, чтобы конечный узел B не был занят в другом соединении. Чтобы зафиксировать соединение, в каждом из коммутаторов вдоль пути от A до B запоминается информация о том, что соответствующая линия связи выделена соединению абонентов A и B (зарезервирована). В каждом коммутаторе выполняется внутреннее соединение интерфейсов, соответствующих маршруту прохождения данных.

Отказ в установлении соединения

Важной особенностью технологии коммутации каналов является возможность *отказа в установлении соединения*. Если какая-либо другая пара абонентов обратится в сеть с запросом на установление соединения, для создания которого необходима хотя бы одна из уже назначенных для A и B линий связи, сеть ответит отказом. Например (см. рис. 3.1), если абонент C пошлет вызов в сеть на установление соединения с абонентом D, то получит отказ, потому что единственная линия, соединяющая коммутаторы S2 и S4, уже выделена соединению абонентов A и B. При отказе в установлении соединения сеть информирует вызывающего абонента об этом неприятном событии. Телефонная сеть в этом случае передает короткие гудки — сигнал «занято». Некоторые телефонные сети различают события «сеть занята» и «абонент занят», передавая гудки с разной частотой или используя разные тона.

Гарантированная пропускная способность

Итак, соединение абонентов A и B установлено. Теперь в их (и только в их) распоряжении имеется канал, обладающий фиксированной пропускной способностью. Это означает, что в течение всего времени соединения они должны посылать в сеть данные с этой скоростью и сеть гарантированно будет доставлять эти данные вызываемому абоненту без потерь и с той же скоростью вне зависимости от того, существуют ли в это время в сети другие соединения или нет. Абонент не может передавать данные в сеть со скоростью, превышающей пропускную способность линии, но и сеть не может снижать скорость передачи пользовательских данных.

Нагрузка сети будет влиять только на вероятность отказа в установлении соединения — чем больше соединений в данный момент поддерживает сеть, тем больше вероятность того, что запрос на установление нового соединения столкнется с отказом.

Приятным обстоятельством является то, что сеть будет доставлять данные с низким и постоянным уровнем задержки, что позволяет качественно передавать данные, чувствительные к задержкам (называемые также **трафиком реального времени**) — голос, видео.

Мультиплексирование

Описанная сеть с коммутацией каналов, в которой каждая *физическая линия всегда передает данные с одной и той же скоростью*, работает неэффективно.

Во-первых, пользователи в такой сети в общем случае получают не то, что они хотят. Действительно, их заставляют стать неким универсальным стандартным пользователем, который всегда передает в сеть информацию с постоянной единственно разрешенной скоростью. Сегодня трудно представить такого пользователя, учитывая оснащенность его различными терминальными устройствами — мобильными телефонами, компьютерами и офисными АТС (корпоративный пользователь). Поэтому скорость пользовательского трафика в общем случае не совпадает с фиксированной пропускной способностью физических каналов. Последняя может быть как намного выше, так и намного ниже потребности пользователя. В первом случае пользователь недоиспользует канал, а во втором должен либо ограничить свои потребности, либо задействовать несколько физических линий связи.

Во-вторых, и сама сеть в таком случае неэффективно использует собственные ресурсы. Очевидно, что в сети, изображенной на рис. 3.1, линий связи между коммутаторами недостаточно. Такая структура сети была выбрана только для того, чтобы более наглядно пояснить причину отказов в удовлетворении запросов пользователей. Для того чтобы вероятность отказа была достаточно низкой и приемлемой для пользователя, между коммутаторами нужно проложить большое количество параллельных физических линий, а это очень затратный вариант.

Чтобы повысить эффективность в сетях с коммутацией каналов, стали использовать мультиплексирование; это позволило одновременно передавать через каждый физический канал трафик нескольких логических соединений. Мультиплексирование в сетях с коммутацией каналов имеет свои особенности. Так, пропускная способность каждой линии связи делится *на равные части*, образуя одинаковое число так называемых **подканалов** (для простоты их часто называют просто каналами). Обычно линия, подключающая пользователя к сети, поддерживает меньшее число подканалов, чем линии, соединяющие коммутаторы, — в этом случае вероятность отказа уменьшается. Например, пользовательская линия может состоять из 2, 24 или 30 подканалов, а линия между коммутаторами — из 480, 1920. Наиболее распространенной скоростью цифрового подканала является сегодня скорость 64 Кбит/с, которая обеспечивает качественную цифровую передачу голоса.

После того как сеть с коммутацией каналов была дополнена механизмом мультиплексирования, схема ее работы претерпела некоторые изменения. Запрос на установление логического соединения, переданный абонентом перед началом передачи, резервирует не целиком линии связи, а только их подканалы. Таким образом, соединение устанавливается не на уровне линий связи, а на уровне подканалов. Несколько подканалов используется в том случае, если пропускной способности одного канала оказывается недостаточно. Это дает возможность пользователю резервировать подканал (или подканалы), скорость передачи данных которого является максимально близкой к реально необходимой. Кроме того, мультиплексирование позволяет более экономно строить связи между коммутаторами. Действительно, для снижения вероятности отказов вместо нескольких линий связи теперь достаточно использовать одну физическую линию с большим количеством логических подканалов.

ВНИМАНИЕ

В случае мультиплексирования в сети с коммутацией каналов установленный при соединении составной канал по-прежнему состоит из линий связи с одинаковой пропускной способностью, только роль линий связи играют подканалы.

Очевидно, что при мультиплексировании усложняется процедура обработки трафика коммутаторами — вместо простой и понятной процедуры коммутации соответствующих интерфейсов теперь необходимо передать данные в нужный подканал. При временно́м мультиплексировании это требует высокой степени синхронизации двух информационных потоков, а при частотном — преобразования частот.

На рис. 3.3 показана сеть с коммутацией каналов и мультиплексированием. В сети установлены два соединения A-B и C-D, одно из них использует по одному подканалу в каждой линии связи, а второе — по два. Таким образом, несмотря на то что данная сеть имеет ту же физическую структуру, что и сеть, представленная на рис. 3.1, блокирования второго вызова (C-D) здесь не происходит, так как коммутаторы поддерживают мультиплексирование.

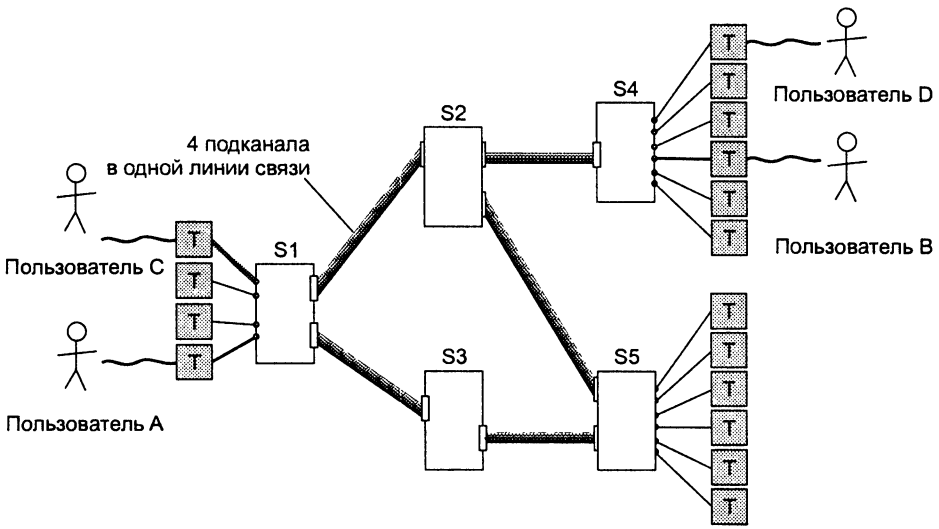


Рис. 3.3. Коммутация каналов с мультиплексированием

Неэффективность передачи пульсирующего трафика

Имеется еще одна причина неэффективной работы сети с коммутацией каналов, а именно: *фиксированная пропускная способность составного канала выделяется на все время соединения.*

Мы отметили, что мультиплексирование повышает эффективность работы сети, потому что пользователь может точнее подобрать скорость соединения в соответствии со своими реальными потребностями. Но это относится только к пользователям, скорость информационных потоков которых постоянна. А что делать пользователям, информационные потоки которых имеют пульсирующий характер?

Если присмотреться к пользовательскому трафику, то окажется, что практически все пользователи телекоммуникационных сетей относятся к этой категории. Так, кажущееся постоянство скорости информационных потоков пользователей телефонной сети достигается за счет обработки неравномерных пользовательских информационных потоков терминальными устройствами телефонной сети — телефонами. Например, цифровой телефон действительно постоянно передает в сеть информационный поток со скоростью 64 Кбит/с независимо от того, говорит в это время его пользователь или нет. Наверно, более эффективно телефон работал бы, если бы он «вырезал» паузы из разговора и передавал в сеть только полезную информацию.

Существует и другая категория пользователей, потребности которой в передаче информации с различной скоростью еще более очевидны, — это пользователи компьютеров. Действительно, работа пользователя, который сканирует веб-ресурсы Интернета, очевидным образом порождает неравномерный трафик. При загрузке очередной страницы в компьютер скорость резко возрастает, а после окончания загрузки падает практически до нуля.

Коэффициент пульсации трафика отдельного пользователя сети определяется как отношение пиковой скорости на каком-либо небольшом интервале времени к средней скорости обмена данными на длительном интервале времени и может достигать значений 100:1. Если для описанного сеанса организовать коммутацию канала между компьютером пользователя и сервером, то большую часть времени канал будет простаивать. В то же время часть производительности сети останется закрепленной за данной парой абонентов и недоступной другим пользователям сети. Сеть в такие периоды похожа на пустой эскалатор метро, который движется, но полезную работу не выполняет, другими словами, «перевозит воздух».

Наиболее эффективно пользовательский трафик передается сетью с коммутацией каналов в том случае, когда скорость его постоянна в течение всего сеанса связи и соответствует пропускной способности физических линий связи сети.

Достоинства и недостатки любой сетевой технологии — относительны. В определенных ситуациях на первый план выходят достоинства, а недостатки становятся несущественными. Так, техника коммутации каналов хорошо работает в тех случаях, когда нужно передавать трафик телефонных разговоров, а с невозможностью «вырезать» паузы из разговора и нерациональным использованием магистральных физических каналов между коммутаторами можно мириться. Однако при передаче очень неравномерного компьютерного трафика на первый план выходит именно эта нерациональность.

Коммутация пакетов

Список ключевых слов: заголовок пакета, концевик пакета, контрольная сумма, входной буфер, сохранение с продвижением, входная очередь, переполнение буфера, перегрузка сети, дейтаграмма, идентификатор пакета, информация о состоянии соединения, виртуальный канал, адрес назначения, таблица коммутации, баланс нагрузки, доставка с максимальными усилиями, метка пакета, пакет установления соединения.

Техника коммутации пакетов была специально разработана для эффективной передачи компьютерного трафика. При коммутации пакетов все передаваемые пользователем сети данные разбиваются в исходном узле на сравнительно небольшие части, называемые пакетами, кадрами, или ячейками, — в данном контексте различия в значении этих терминов не существенны (рис. 3.4). Каждый пакет снабжается **заголовком**, в котором указывается адрес, необходимый для доставки пакета узлу назначения. Наличие адреса в каждом пакете является одним из важнейших свойств техники коммутации пакетов, так как каждый пакет *может*[1] быть обработан коммутатором независимо от других пакетов информационного потока. Помимо заголовка у пакета имеется еще одно дополнительное

[1] Слово «может» в данном контексте очень важно, так как в некоторых вариантах технологии коммутации пакетов полная независимость обработки пакетов не обеспечивается (например, в технологии виртуальных каналов).

поле, которое обычно размещается в конце пакета и поэтому называется **концевиком**. В концевике помещается **контрольная сумма**, которая позволяет проверить, была ли искажена информация при передаче через сеть или нет.

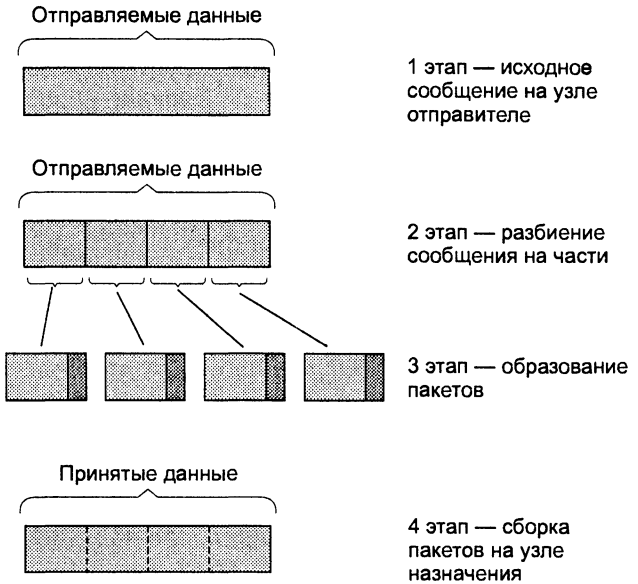


Рис. 3.4. Разбиение потока данных на пакеты

Пакеты поступают в сеть *без предварительного резервирования линий связи* и *не с фиксированной наперед заданной скоростью*, как это делается в сетях с коммутацией каналов, а в том темпе, в котором их генерирует источник. Предполагается, что сеть с коммутацией пакетов в отличие от сети с коммутацией каналов всегда готова принять пакет от конечного узла.

ПРИМЕЧАНИЕ

Процедура резервирования пропускной способности может применяться и в пакетных сетях. Однако основная идея такого резервирования принципиально отличается от идеи резервирования пропускной способности в сетях с коммутацией каналов. Разница заключается в том, что пропускная способность канала сети с коммутацией пакетов может динамически перераспределяться между информационными потоками в зависимости от текущих потребностей каждого потока, чего не может обеспечить техника коммутации каналов. С деталями такого резервирования вы познакомитесь позже, в главе 7.

Буферы и очереди

Сеть с коммутацией пакетов, так же как и сеть с коммутацией каналов, состоит из коммутаторов, связанных физическими линиями связи. Однако коммутаторы функционируют в этих сетях по-разному. Главное отличие состоит в том, что па-

кетные коммутаторы *имеют внутреннюю буферную память* для временного хранения пакетов. Действительно, пакетный коммутатор не может принять решения о продвижении пакета, не имея в своей памяти всего пакета. Коммутатор проверяет контрольную сумму, и только если она говорит о том, что данные пакета не искажены, начинает обрабатывать пакет и по адресу назначения определяет следующий коммутатор. Поэтому *каждый* пакет последовательно бит за битом помещается во **входной буфер**. Имея в виду это свойство, говорят, что сети с коммутацией пакетов используют технику **сохранения с продвижением** (store-and-forward). Заметим, что для этой цели достаточно иметь буфер размером в один пакет.

Буферизация необходима пакетному коммутатору также *для согласования скорости поступления пакетов со скоростью их коммутации*. Если коммутирующий блок не успевает обрабатывать пакеты, то на интерфейсах коммутатора возникают **входные очереди**. Очевидно, что для хранения входной очереди объем буфера должен превышать размер одного пакета. Существуют различные подходы к построению коммутирующего блока. Традиционный способ основан на одном центральном процессоре, который обслуживает все входные очереди коммутатора. Такой способ построения может приводить к большим очередям, так как производительность процессора разделяется между несколькими очередями. Современные способы построения коммутирующего блока основаны на многопроцессорном подходе, когда каждый интерфейс имеет свой встроенный процессор для обработки пакетов. Кроме того, существует также центральный процессор, координирующий работу интерфейсных процессоров. Использование интерфейсных процессоров повышает производительность коммутатора и уменьшает очереди во входных интерфейсах. Однако такие очереди все равно могут возникать, так как центральный процессор по-прежнему остается «узким местом». Более подробно вопросы внутреннего устройства коммутаторов обсуждаются в главе 15.

Наконец, буферы нужны *для согласования скоростей передачи данных в каналах*, подключенных к пакетному коммутатору. Действительно, если скорость поступления пакетов из одного канала в течение некоторого периода превышает пропускную способность того канала, в который эти пакеты должны быть направлены, то во избежание потерь пакетов на целевом интерфейсе необходимо организовать выходную очередь (рис. 3.5).

В сети с коммутацией пакетов пульсации трафика отдельных абонентов в соответствии с законом больших чисел распределяются во времени так, что их пики чаще всего не совпадают. Поэтому коммутаторы постоянно и достаточно равномерно загружены работой, если число обслуживаемых ими абонентов действительно велико. На рис. 3.6 показано, что трафик, поступающий от каждого из конечных узлов на коммутаторы, очень неравномерно распределен во времени. Однако коммутаторы более высокого уровня иерархии, которые обслуживают соединения между коммутаторами нижнего уровня, загружены более равномерно, и магистральные каналы, соединяющие коммутаторы верхнего уровня, имеют близкие к максимальным коэффициенты использования. Буферизация сглаживает пульсации, поэтому коэффициент пульсации на магистральных каналах гораздо ниже, чем на каналах абонентского доступа.

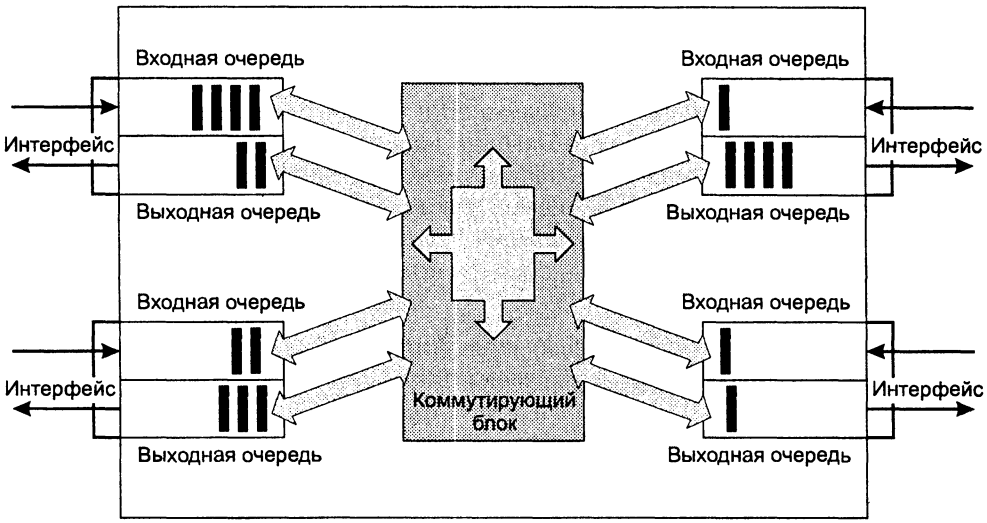


Рис. 3.5. Очереди в пакетном коммутаторе

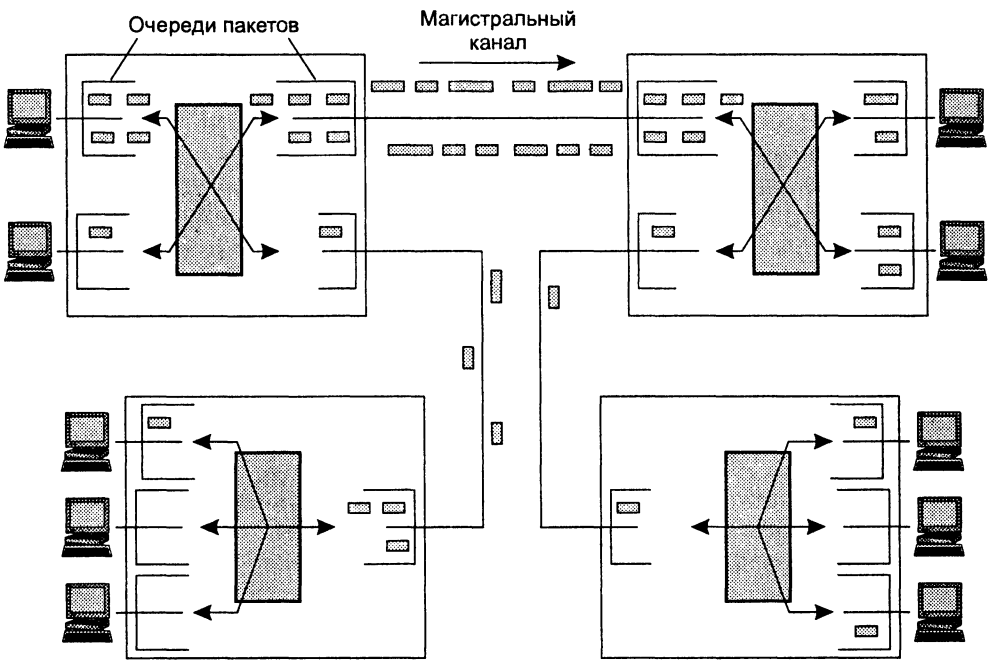


Рис. 3.6. Сглаживание пульсаций трафика в сети с коммутацией пакетов

Поскольку объем буферов в коммутаторах ограничен, иногда происходит потеря пакетов из-за **переполнения** буферов при временной **перегрузке** части сети, когда совпадают периоды пульсации нескольких информационных потоков. Так как потеря пакетов является неотъемлемым свойством сети с коммутацией паке-

тов, то для нормальной работы таких сетей разработан ряд механизмов, которые компенсируют этот эффект. Эти механизмы, называемые методами обеспечения качества обслуживания и инжиниринга трафика, рассматриваются в главе 7.

Методы продвижения пакетов

Решение о том, на какой интерфейс передать пришедший пакет, принимается на основании одного из трех методов продвижения пакетов:

- При *дейтаграммной передаче* соединение не устанавливается, и все передаваемые пакеты *продвигаются* (передаются от одного узла сети другому) *независимо* друг от друга на основании одних и тех же правил. Процедура обработки пакета определяется только значениями параметров, которые он несет в себе, и текущим состоянием сети (например, в зависимости от ее нагрузки пакет может стоять в очереди на обслуживание большее или меньшее время). Однако никакая информация об уже переданных пакетах сетью не хранится и в ходе обработки очередного пакета во внимание не принимается. То есть каждый отдельный пакет рассматривается сетью как совершенно независимая единица передачи — **дейтаграмма**.
- *Передача с установлением логического соединения* распадается на так называемые сеансы, или логические соединения. Процедура обработки определяется не для отдельного пакета, а для всего множества пакетов, передаваемых в рамках каждого соединения. Для того чтобы реализовать дифференцированное обслуживание пакетов, принадлежащих разным соединениям, сеть должна, во-первых, присвоить каждому соединению **идентификатор**, во-вторых, запомнить параметры соединения, то есть значения, определяющие процедуру обработки пакетов в рамках данного соединения. Эта информация называется **информацией о состоянии соединения**. Фиксированный маршрут не является обязательным параметром соединения. Пакеты, принадлежащие одному и тому же соединению, даже имеющие одни и те же адреса отправления и назначения, могут перемещаться по разным независимым друг от друга маршрутам.
- *Передача с установлением виртуального канала.* Если в число параметров соединения *входит* маршрут, то все пакеты, передаваемые в рамках данного соединения, должны проходить по указанному пути. Такой единственный заранее проложенный фиксированный маршрут, соединяющий конечные узлы в сети с коммутацией пакетов, называют **виртуальным каналом** (virtual circuit, или virtual channel).

Классификация методов коммутации приведена на рис. 3.7.

В одной и той же сетевой технологии могут быть задействованы разные способы обмена данными. Так, дейтаграммный протокол IP используется для передачи данных между отдельными сетями, составляющими Интернет. В то же время обеспечением надежной доставки данных между конечными узлами этой сети занимается протокол TCP, устанавливающий логические соединения без фиксации маршрута. И наконец, Интернет является примером сети, использующей технику виртуальных каналов, так как в состав Интернета входит немало сетей ATM и Frame Relay, поддерживающих виртуальные каналы.

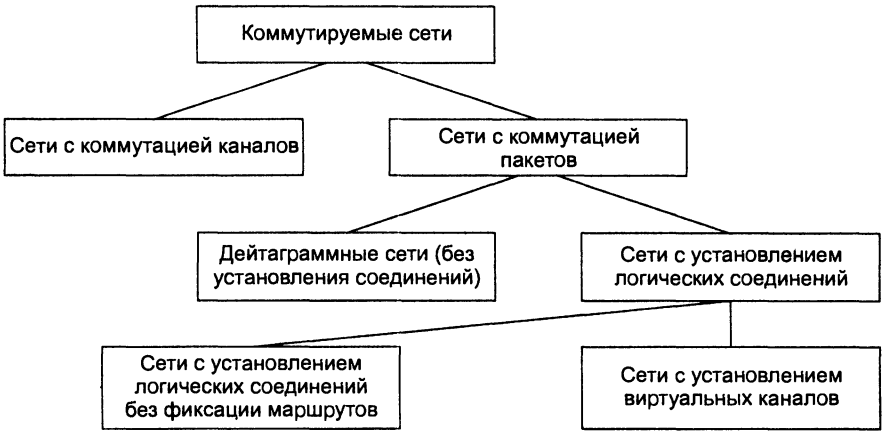


Рис. 3.7. Таксономия коммутируемых сетей

Дейтаграммная передача

Итак, как было сказано, дейтаграммный способ передачи данных основан на том, что все передаваемые пакеты обрабатываются независимо друг от друга. Выбор интерфейса, на который надо передать поступивший пакет, происходит только на основании **адреса назначения**, содержащегося в заголовке пакета. Принадлежность пакета к определенному информационному потоку никак не учитывается.

Решение о продвижении пакета принимается на основе **таблицы коммутации**, содержащей набор адресов назначения и адресную информацию, однозначно определяющую следующий по маршруту (транзитный или конечный) узел. Напомним, что в разных технологиях для обозначения таблиц, имеющих указанное выше функциональное назначение, могут использоваться другие термины (таблица маршрутизации, таблица продвижения и др.). Далее для простоты будем пользоваться термином «таблица коммутации» в качестве обобщенного названия таблиц такого рода, применяемых для дейтаграммной передачи на основании только адреса назначения конечного узла.

Таблица коммутации дейтаграммной сети должна содержать записи обо всех адресах, куда могут быть направлены пакеты, поступающие на интерфейсы коммутатора. А они в общем случае могут быть адресованы любому узлу сети. На практике используются приемы, уменьшающие число записей в таблице, например, иерархическая адресация. В этом случае таблица коммутации может содержать только старшие части адресов, которые соответствуют не отдельным узлам, а некоторой группе узлов (для их обозначения часто применяют термин «подсеть»). Если обратиться к аналогии с почтовыми адресами, то такими старшими частями адреса являются названия стран и городов, число которых, естественно, несоизмеримо меньше, чем названий улиц, домов и имен отдельных людей.

Несмотря на применение иерархической адресации в некоторых крупных сетях (например, в Интернете), коммутаторы могут иметь таблицы с числом входов, превышающем несколько тысяч. На рис. 3.8 показано, как могла бы выглядеть таблица коммутации в дейтаграммной сети.

Таблица коммутации коммутатора S1

Адрес назначения	Адрес следующего коммутатора
N1	Пакет не требуется передавать через сеть
N2	S2
N3	S3
N4	S3
N5	S6
N6	S6

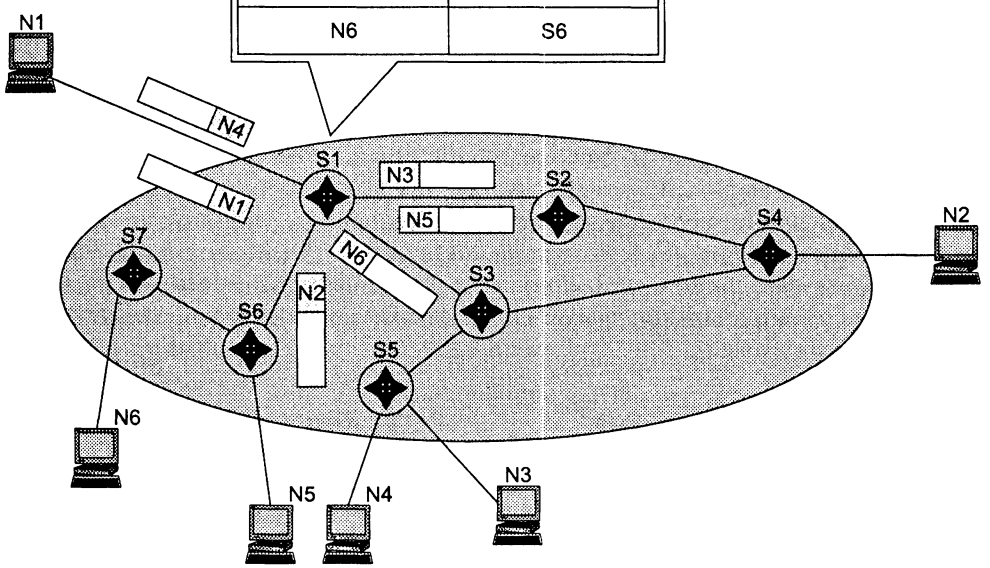


Рис. 3.8. Иллюстрация дейтаграммного принципа передачи пакетов

В таблице коммутации для одного и того же адреса назначения может содержаться несколько записей, указывающих соответственно на различные адреса следующего коммутатора. Такой подход называется **балансом нагрузки** и используется для повышения производительности и надежности сети. В примере, показанном на рис. 3.8, пакеты, поступающие в коммутатор S1 для узла назначения с адресом N2 в целях баланса нагрузки распределяются между двумя следующими коммутаторами — S2 и S3, что снижает нагрузку на каждый из них, а значит, уменьшает очереди и ускоряет доставку. Некоторая «размытость» путей следования пакетов с одним и тем же адресом назначения через сеть является прямым следствием принципа независимой обработки каждого пакета, присущего дейтаграммному методу. Пакеты, следующие по одному и тому же адресу назначения, могут добираться до него разными путями также вследствие изменения состояния сети, например отказа промежуточных коммутаторов.

Дейтаграммный метод работает быстро, так как никаких предварительных действий перед отправкой данных проводить не требуется. Однако при таком методе трудно проверить факт доставки пакета узлу назначения. Этот метод не гарантирует доставку пакета, он делает это по мере возможности — для описания такого свойства используется термин **доставка с максимальными усилиями** (best effort).

Логическое соединение

Передача с установлением логического соединения основывается на знании «предыстории» обмена. Это позволяет более рационально по сравнению с дейтаграммным способом обрабатывать пакеты. Например, при потере нескольких предыдущих пакетов может быть снижена скорость отправки последующих. Или благодаря нумерации пакетов и отслеживанию номеров отправленных и принятых пакетов можно повысить надежность путем отбрасывания дубликатов, упорядочивания поступивших и повторения передачи потерянных пакетов.

Параметры соединения могут быть как постоянными в течение всего соединения (например, максимальный размер пакета), так и переменными, динамически отражающими текущее состояние соединения (например, упомянутые выше последовательные номера пакетов). Когда отправитель и получатель *фиксируют* начало нового соединения, они, прежде всего, «договариваются» о начальных значениях параметров процедуры обмена и только после этого начинают передачу собственно данных.

Рис. 3.9. Передача без установления соединения (*а*) и с установлением соединения (*б*)

Передача с установлением соединения более надежна, но требует больше времени для передачи данных и вычислительных затрат от конечных узлов, что иллюстрирует рис. 3.9.

При передаче с установлением соединения узлу-получателю отправляется служебный кадр специального формата с предложением установить соединение, как показано на рис. 3.9, *б*. Если узел-получатель согласен с этим, то он посылает в ответ другой служебный кадр, подтверждающий установление соединения и предлагающий некоторые параметры, которые будут использоваться в рамках данного логического соединения. Это могут быть, например, идентификатор соединения, максимальное значение длины поля данных кадров, количество кадров, которые можно отправить без получения подтверждения, и т. п. Узел-инициатор соединения может закончить процесс установления соединения отправкой третьего служебного кадра, в котором сообщит, что предложенные параметры ему подходят. На этом логическое соединение считается установленным. Логическое соединение может быть рассчитано на передачу данных как в одном направлении — от инициатора соединения, так и в обоих направлениях. После передачи некоторого законченного набора данных, например определенного файла, узел-отправитель инициирует разрыв данного логического соединения, посылая соответствующий служебный кадр.

Заметим, что, в отличие от передачи дейтаграммного типа, в которой поддерживается только один тип кадра — информационный, передача с установлением соединения должна поддерживать как минимум два типа кадров — информационные, переносящие собственно пользовательские данные, и служебные, предназначенные для установления (разрыва) соединения.

Виртуальный канал

Виртуальные каналы (virtual circuit, или virtual channel) — это устойчивые пути следования трафика, создаваемые в сети с коммутацией пакетов. Виртуальные каналы являются базовой концепцией технологий X.25, Frame Relay и ATM.

Техника виртуальных каналов учитывает существование в сети потоков данных. Для того чтобы выделить поток данных из общего трафика, каждый пакет этого потока помечается **меткой**. Так же как в сетях с установлением логических соединений, прокладка виртуального канала начинается с отправки из узла-источника запроса, называемого также **пакетом установления соединения**. В запросе указывается адрес назначения и метка потока, для которого прокладывается этот виртуальный канал. Запрос, проходя по сети, формирует новую запись в каждом из коммутаторов, расположенных на пути от отправителя до получателя. Запись говорит о том, каким образом коммутатор должен обслуживать пакет, имеющий заданную метку. Образованный виртуальный канал идентифицируется той же меткой[1].

[1] Эта метка в различных технологиях называется по-разному: номером логического канала (Logical Channel number, LCN) в технологии X.25, идентификатором соединения уровня канала данных (Data Link Connection Identifier, DLCI) в технологии Frame Relay, идентификатором виртуального канала (Virual Channel Identifier, VCI) в технологии ATM.

После прокладки виртуального канала сеть может передавать по нему соответствующий поток данных. Во всех пакетах, которые переносят пользовательские данные, адрес назначения уже не указывается, его роль играет метка виртуального канала. При поступлении пакета на входной интерфейс коммутатор читает значение метки из заголовка пришедшего пакета и просматривает свою таблицу коммутации, по которой определяет, на какой выходной порт передать пришедший пакет.

Таблица коммутации в сетях, использующих виртуальные каналы, отличается от таблицы коммутации в дейтаграммных сетях. Она содержит записи *только о проходящих через коммутатор виртуальных каналах*, а не обо всех возможных адресах назначения, как это имеет место в сетях с дейтаграммным алгоритмом продвижения. Обычно в крупной сети количество проложенных через узел виртуальных каналов существенно меньше общего количества узлов, поэтому и таблицы коммутации в этом случае намного короче, а, следовательно, анализ такой таблицы занимает у коммутатора меньше времени. По этой же причине метка короче адреса конечного узла, и заголовок пакета в сетях с виртуальными каналами переносит по сети вместо длинного адреса компактный идентификатор потока.

ПРИМЕЧАНИЕ

Использование в сетях техники виртуальных каналов не делает их сетями с коммутацией каналов. Хотя в подобных сетях применяется процедура предварительного установления канала, этот канал является виртуальным, то есть по нему передаются отдельные пакеты, а не потоки информации с постоянной скоростью, как в сетях с коммутацией каналов.

На рис. 3.10 показан фрагмент сети, в которой проложены два виртуальных канала — VC1 и VC2.

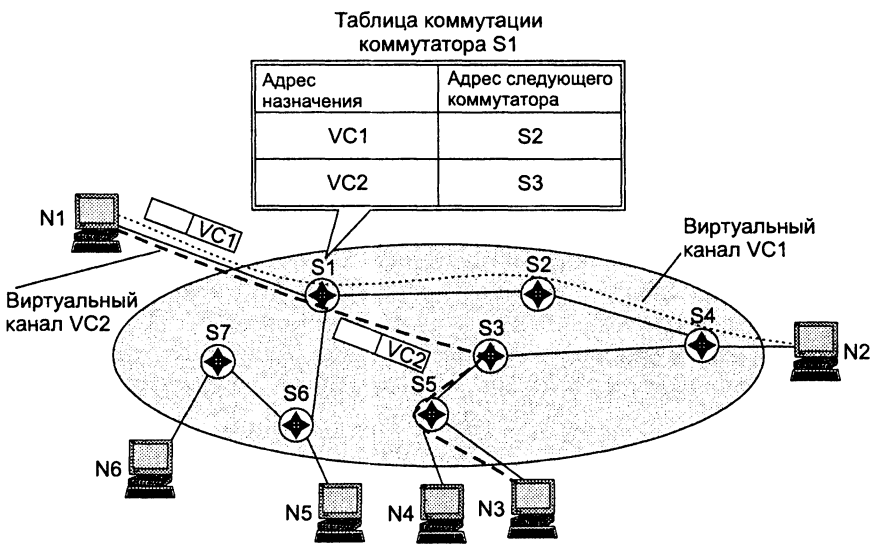


Рис. 3.10. Иллюстрация принципа работы виртуального канала

Первый проходит от конечного узла с адресом N1, до конечного узла с адресом N2 через промежуточные коммутаторы S1, S2 и S4. Второй обеспечивает продвижение данных по пути N3 — S5 — S3 — S1 — N1. В принципе между двумя конечными узлами может быть проложено несколько виртуальных каналов, например, еще один виртуальный канал между узлами N1 и N2 мог бы проходить через промежуточные коммутаторы S1, S3 и S4.

Сравнение сетей с коммутацией пакетов и каналов

Список ключевых слов: мультипрограммная операционная система, время буферизации, коммутация пакетов, коммутация каналов, время распространения сигнала в физической среде, время пакетизации, конвейерная передача.

Прежде чем проводить техническое сравнение сетей с коммутацией пакетов и сетей с коммутацией каналов, проведем их неформальное сравнение на основе, как нам кажется, весьма продуктивной транспортной аналогии.

Транспортная аналогия для сетей с коммутацией пакетов и каналов

Для начала убедимся, что движение на дорогах имеет много общего с перемещением пакетов в сети *с коммутацией пакетов*.

Пусть автомобили в этой аналогии соответствуют пакетам, дороги — каналам связи, а перекрестки — коммутаторам. Подобно пакетам, автомобили перемещаются независимо друг от друга, разделяя пропускную способность дорог и создавая препятствия друг другу. Слишком интенсивный трафик, не соответствующий пропускной способности дороги, приводит к перегруженности дорог, в результате автомобили стоят в пробках, что соответствует очередям пакетов в коммутаторах.

На перекрестках происходит «коммутация» потоков автомобилей, каждый из автомобилей выбирает подходящее направление перекрестка, чтобы попасть в пункт назначения. Конечно, перекресток играет намного более пассивную роль по сравнению с коммутатором пакетов. Его активное участие в обработке трафика можно заметить только на регулируемых перекрестках, где светофор определяет очередность пересечения перекрестка потоками автомобилей. Еще активней, естественно, поведение регулировщика трафика, который может выбрать для продвижения не только поток автомобилей в целом, но и отдельный автомобиль.

Как и в сетях с коммутацией пакетов, к образованию заторов на дорогах приводит неравномерность движения автомобилей. Так, даже кратковременное снижение скорости одного автомобиля на узкой дороге может создать большую проб-

ку, которой бы не было, если бы все автомобили всегда двигались с одной и той же скоростью и равными интервалами.

А теперь попробуем найти общее в автомобильном движении и в сетях с *коммутацией каналов*.

Иногда на дороге возникает ситуация, когда нужно обеспечить особые условия для движения колонны автомобилей. Например, представим, что очень длинная колонна автобусов перевозит детей из города в летний лагерь по многополосному шоссе. Для того чтобы колонна двигалась без препятствий, заранее для ее движения разрабатывается маршрут.

Затем на протяжении всего этого маршрута, который пересекает несколько перекрестков, для колонны выделяется отдельная полоса на всех отрезках шоссе. При этом полоса освобождается от другого трафика еще за некоторое время до начала движения колонны, и это резервирование отменяется только после того, как колонна достигает пункта назначения.

Во время движения все автомобили колонны едут с одинаковой скоростью и приблизительно равными интервалами между собой, не создавая препятствий друг другу. Очевидно, что для колонны автомобилей создаются наиболее благоприятные условия для движения, но при этом автомобили теряют свою самостоятельность, превращаясь в поток, из которого нельзя «свернуть» в сторону. Дорога при такой организации движения используется не рационально, так как полоса простаивает значительную часть времени, как и полоса пропускания в сетях с коммутацией каналов.

Количественное сравнение задержек

Вернемся от автомобилей к сетевому трафику. Пусть пользователю сети необходимо передать достаточно неравномерный трафик, состоящий из периодов активности и пауз. Представим также, что он может выбрать, через какую сеть, с коммутацией каналов или пакетов, передавать ему свой трафик, причем в обеих сетях производительность каналов связи одинакова. Очевидно, что более эффективной с точки зрения временны́х затрат для нашего пользователя была бы работа в сети с коммутацией каналов, где ему в единоличное владение предоставляется зарезервированный канал связи. При этом способе все данные поступали бы адресату без задержки. Тот факт, что значительную часть времени зарезервированный канал будет простаивать (во время пауз), нашего пользователя не волнует — ему важно быстро решить собственную задачу.

Если бы пользователь обратился к услугам сети с коммутацией пакетов, то процесс передачи данных оказался бы более медленным, так как его пакеты вероятно не раз задерживались бы в очередях, ожидая освобождения необходимых сетевых ресурсов наравне с пакетами других абонентов.

Давайте рассмотрим более детально механизм возникновения задержек при передаче данных в сетях обоих типов. Пусть от конечного узла N1 отправляется сообщение к конечному узлу N2 (рис. 3.11). На пути передачи данных расположены два коммутатора.

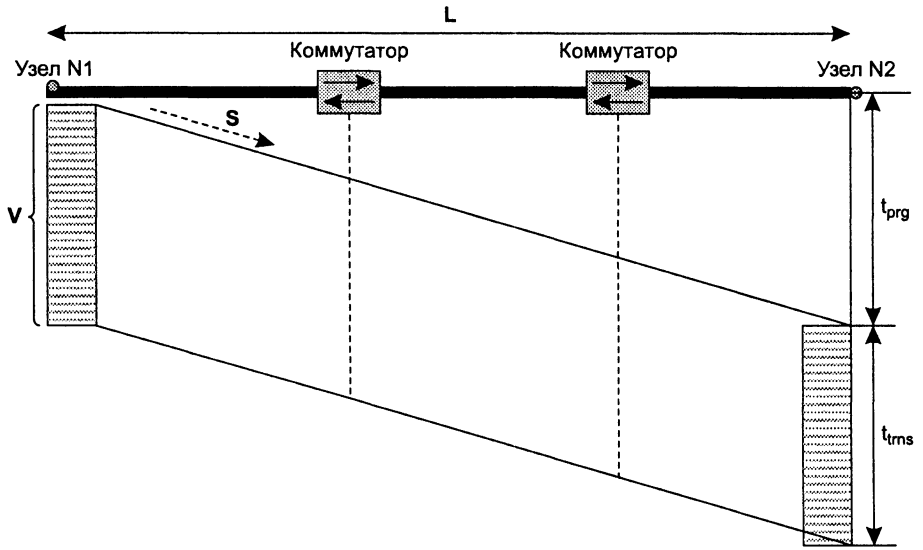


Рис. 3.11. Временна́я диаграмма передачи сообщения в сети с коммутацией каналов

В сети с коммутацией каналов данные после задержки, связанной с установлением канала, начинают передаваться на стандартной для канала скорости. Время доставки данных T адресату равно сумме *времени распространения сигнала в канале* t_{prg} и *времени передачи сообщения в канал* t_{trns}. Наличие коммутаторов в сети с коммутацией каналов никак не влияет на суммарное время прохождения данных через сеть.

ПРИМЕЧАНИЕ

Заметим, что время передачи сообщения в канал в точности совпадает с временем приема сообщения из канала в буфер узла назначения, в этом случае оно называется временем буферизации.

Время распространения сигнала зависит от расстояния между абонентами L и скорости S распространения электромагнитных волн в конкретной физической среде, которая колеблется от 0,6 до 0,9 скорости света в вакууме:

$$t_{prg} = L/S.$$

Время передачи сообщения в канал (а значит, и время буферизации в узле назначения) равно отношению объема сообщения V в битах к пропускной способности канала C в битах в секунду:

$$t_{trna} = V/C.$$

В сети с коммутацией пакетов передача данных не требует обязательного установления соединения. Предположим, что в сеть, показанную на рис. 3.12, передается сообщение того же объема V, что и в предыдущем случае (см. рис. 3.11), однако оно разделено на пакеты, каждый из которых снабжен заголовком. Пакеты передаются от узла N1 в узел N2, между которыми расположены два коммутатора.

На каждом коммутаторе каждый пакет изображен дважды: в момент прихода на входной интерфейс и в момент передачи в сеть с выходного интерфейса. Из рисунка видно, что коммутатор задерживает пакет на некоторое время. Здесь T_1 — время доставки адресату первого пакета сообщения, а T_{ps} — всего сообщения.

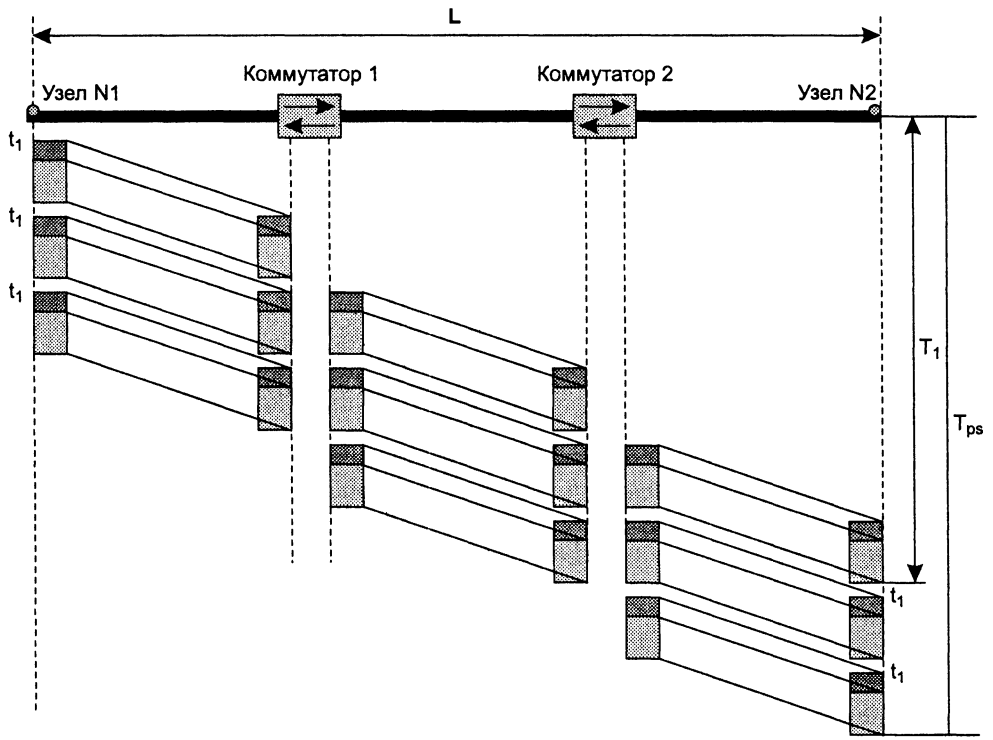


Рис. 3.12. Временная диаграмма передачи сообщения, разделенного на пакеты, в сети с коммутацией пакетов

Сравнивая временные диаграммы передачи данных в сетях с коммутацией каналов и пакетов, отметим два факта:

- значения времени распространения сигнала (t_{prg}) в одинаковой физической среде на одно и то же расстояние одинаковы;
- учитывая, что значения пропускной способности каналов в обеих сетях одинаковы, значения времени передачи сообщения в канал (t_{trns}) будут *также равны*.

Однако разбиение передаваемого сообщения на пакеты с последующей их передачей по сети с коммутацией пакетов приводит к дополнительным задержкам. Проследим путь первого пакета и отметим, из каких составляющих складывается время его передачи в узел назначения и какие из них специфичны для сети с коммутацией пакетов (рис. 3.13).

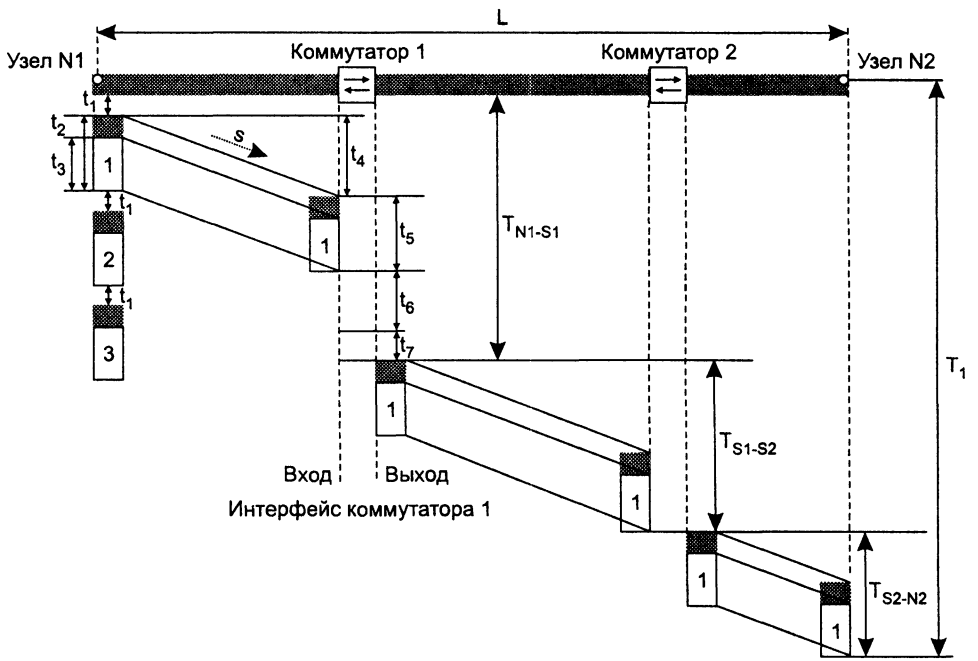


Рис. 3.13. Временная диаграмма передачи одного пакета в сети с коммутацией пакетов

Время передачи одного пакета от узла N1 до коммутатора 1 можно представить в виде суммы нескольких слагаемых.

- Во-первых, время тратится в узле-отправителе N1:
  - t_1 — время формирования пакета, также называемое временем пакетизации (значение этой задержки зависит от различных параметров работы программного и аппаратного обеспечения узла-отправителя и не зависит от параметров сети);
  - t_2 — время передачи в канал заголовка;
  - t_3 — время передачи в канал поля данных пакета.
- Во-вторых, дополнительное время тратится на распространение сигналов по каналам связи. Обозначим через t_4 время распространения сигнала, представляющего один бит информации, от узла N1 до коммутатора 1.
- В-третьих, дополнительное время тратится в промежуточном коммутаторе:
  - t_5 —время приема пакета с его заголовком из канала во входной буфер коммутатора; как уже было отмечено, это время равно ($t_2 + t_3$), то есть времени передачи пакета с заголовком в канал из узла источника;
  - t_6 — время ожидания пакета в очереди колеблется в очень широких пределах и заранее неизвестно, так как зависит от текущей загрузки сети;
  - t_7 — время коммутации пакета при его передаче в выходной порт фиксировано для конкретной модели и обычно невелико (от нескольких микросекунд до нескольких миллисекунд).

Обозначим через $T_{N1\text{-}S1}$ время передачи пакета из узла N1 на выходной интерфейс коммутатора 1. Это время складывается из следующих составляющих:

$$T_{N1\text{-}S1} = t_1 + t_4 + t_5 + t_6 + t_7.$$

Обратите внимание, что среди слагаемых отсутствуют составляющие t_2 и t_3. Из рис. 3.13 видно, что передача битов из передатчика в канал совмещается по времени с передачей битов по каналу связи.

Время, затрачиваемое на оставшиеся два отрезка пути, обозначим соответственно $T_{S1\text{-}S2}$ и $T_{S2\text{-}N2}$. Эти величины имеют такую же структуру, что и $T_{N1\text{-}S1}$, за исключением того, что в них не входит время пакетизации, и, кроме того, $T_{S2\text{-}N2}$ не включает время коммутации (так как отрезок заканчивается конечным узлом). Итак, полное *время передачи одного пакета* по сети составляет:

$$T_1 = T_{N1\text{-}S1} + T_{S1\text{-}S2} + T_{S2\text{-}N2}.$$

А чему же будет равно время передачи сообщения, состоящего из нескольких пакетов? Сумме времен передачи каждого пакета? Конечно, нет! Ведь сеть с коммутацией пакетов работает как конвейер (см. рис. 3.12): пакет обрабатывается за несколько этапов, и все устройства сети выполняют эти этапы параллельно. Поэтому время передачи такого сообщения будет значительно меньше, чем сумма значений времени передачи каждого пакета сообщения. Точно рассчитать это время сложно из-за неопределенности состояния сети и, вследствие этого, неопределенности значений времени ожидания пакетов в очередях коммутаторов. Однако если предположить, что пакеты стоят в очереди примерно одинаковое время, то общее *время передачи сообщения, состоящего из n пакетов*, можно оценить следующим образом

$$T_{PS} = T_1 + (n - 1)(t_1 + t_5).$$

Пример

Сравним задержки передачи данных в сетях с коммутацией пакетов с задержками в сетях с коммутацией каналов, основываясь на рис. 3.13. Пусть тестовое сообщение, которое нужно передать в обоих видах сетей, составляет 200 000 байт. Отправитель находится от получателя на расстоянии 5000 км. Пропускная способность линий связи составляет 2 Мбит/с. Время передачи данных по сети с коммутацией каналов складывается из времени распространения сигнала, которое для расстояния 5000 км можно оценить примерно в 25 мс, и времени передачи сообщения в канал, которое при пропускной способности 2 Мбит/с и размере сообщения 200 000 байт равно примерно 800 мс, то есть всего передача данных абоненту занимает 825 мс. Оценим дополнительное время, которое требуется для передачи этого сообщения по сети с коммутацией пакетов. Будем считать, что путь от отправителя до получателя пролегает через 10 коммутаторов. Также предположим, что сеть работает в недогруженном режиме и очереди в коммутаторах отсутствуют. Исходное сообщение разбивается на пакеты по 1000 байт, всего 200 пакетов.

Если принять интервал между отправкой пакетов равным 1 мс, тогда время передачи сообщения увеличится на дополнительные 200 мс. Время передачи сообщения в канал также увеличится из-за необходимости передавать заголовки пакетов. Предположим, что доля служебной информации, размещенной в заголовках пакетов, по отношению к общему объему сообщения составляет 10 %. Следовательно, дополнительная задерж-

ка, связанная с передачей заголовков пакетов, составляет 10 % от времени передачи исходного сообщения, то есть 80 мс. При прохождении пакетов через каждый коммутатор возникает задержка буферизации пакета. Эта задержка при величине пакета 1000 байт, заголовке 100 байт и пропускной способности линии 2 Мбит/с составляет 4,4 мс в одном коммутаторе. Плюс задержка коммутации 2 мс. В результате прохождения 10 коммутаторов пакет придет с суммарной задержкой 64 мс, потраченной на буферизацию и коммутацию. В результате дополнительная задержка, созданная сетью с коммутацией пакетов, составляет 344 мс.

Учитывая, что вся передача данных по сети с коммутацией каналов занимает 825 мс, эту дополнительную задержку можно считать существенной. Хотя приведенный расчет носит очень приблизительный характер, он делает более понятными те причины, по которым для отдельного абонента процесс передачи данных по сети с коммутацией пакетов является более медленным, чем по сети с коммутацией каналов.

Что же следует из приведенного примера? Можно ли считать, что сеть с коммутацией каналов более эффективна, чем сеть с коммутацией пакетов? Попробуем ответить на этот вопрос.

При рассмотрении сети в целом логично использовать в качестве критерия эффективности сети не скорость передачи трафика отдельного пользователя, а более интегральный критерий, например общий объем передаваемых сетью данных в единицу времени. В этом случае эффективность сетей с коммутацией пакетов по сравнению с сетями с коммутацией каналов (при равной пропускной способности каналов связи) оказывается выше. Такой результат был доказан в 60-е годы как экспериментально, так и аналитически с помощью теории массового обслуживания.

Пример

Используем для сравнения эффективности сетей с коммутацией каналов и пакетов еще один пример (рис. 3.14). Два коммутатора объединены каналом связи с пропускной способностью 100 Мбит/с. Пользователи сети подключаются к сети с помощью каналов доступа (access links) с пропускной способностью 10 Мбит/с. Предположим, что все пользователи создают одинаковый пульсирующий трафик со средней скоростью 1 Мбит/с. При этом в течение непродолжительных периодов времени скорость данной предложенной нагрузки возрастает до максимальной скорости канала доступа, то есть до 10 Мбит/с. Такие периоды длятся не более одной секунды. Предположим также, что все пользователи, подключенные к коммутатору S1, передают информацию только пользователям, подключенным к коммутатору S2.

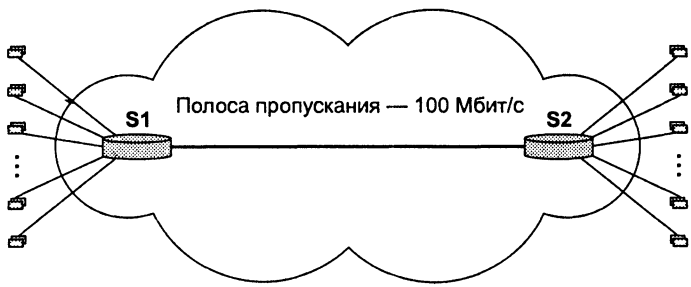


Рис. 3.14. Сравнение эффективности сетей с коммутацией пакетов и каналов

Пусть представленная на рисунке сеть является сетью с коммутацией каналов. Поскольку пики пользовательского трафика достигают 10 Мбит/с, каждому из пользователей необходимо установить соединение с пропускной способностью 10 Мбит/с. Таким образом, одновременно через сеть смогут передавать данные только 10 пользователей. Суммарная средняя скорость передачи данных через сеть будет равна только 10 Мбит/с (10 пользователей передают данные со средней скоростью 1 Мбит/с). Следовательно, линия связи между коммутаторами, хотя и имеет общую пропускную способность 100 Мбит/с, используется только на 10 %.

Теперь рассмотрим вариант, когда та же сеть работает на основе техники коммутации пакетов. При средней скорости пользовательских потоков 1 Мбит/с сеть может передавать одновременно до 100/1 = 100 (!) информационных потоков пользователей, полностью расходуя пропускную способность канала между коммутаторами. Однако это справедливо, если емкости буферов коммутаторов достаточно для хранения пакетов на периодах перегрузки, когда суммарная скорость потока данных превышает 100 Мбит/с. Оценим необходимый объем буфера коммутатора S1. За период перегрузки в коммутатор S1 от каждого потока поступит 10 Мбит/с × 1 с = 10 Мбит, а от 100 потоков — 1000 Мбит. Из этих данных за 1 с коммутатор успеет передать в выходной канал только 100 Мбит. Значит, чтобы ни один пакет не был потерян во время перегрузки сети, общий объем входных буферов коммутатора должен быть не меньше 1000 – 100 = = 900 Мбит, или более 100 Мбайт. Сегодняшние коммутаторы обычно имеют меньшие объемы буферов (1–10 Мбайт). Однако не нужно забывать, что вероятность совпадения периодов пиковой нагрузки у всех потоков, поступающих на входы коммутатора, очень мала. Так что даже если коммутатор имеет меньший объем буферной памяти, в подавляющем большинстве случаев он будет справляться с предложенной нагрузкой.

При сравнении сетей с коммутацией каналов и пакетов уместна аналогия с **мультипрограммными операционными системами**. Каждая отдельная программа в такой системе выполняется дольше, чем в однопрограммной системе, когда программе выделяется все процессорное время, пока она не завершит свое выполнение. Однако общее число программ, выполняемых в единицу времени, в мультипрограммной системе больше, чем в однопрограммной. Аналогично однопрограммной системе, в которой время от времени простаивает процессор или периферийные устройства, в сетях с коммутацией каналов при передаче пульсирующего трафика значительная часть зарезервированной пропускной способности каналов часто не используется.

Неопределенная пропускная способность сети с коммутацией пакетов — это плата за ее общую эффективность при некотором ущемлении интересов отдельных абонентов. Аналогично, в мультипрограммной операционной системе время выполнения приложения предсказать заранее невозможно, так как оно зависит от количества других приложений, с которыми делит процессор данное приложение.

В заключение этого раздела приведем табл. 3.1, в которой сведены свойства обоих видов сетей. На основании этих данных можно аргументированно утверждать, в каких случаях рациональнее использовать сети с коммутацией каналов, а в каких — с коммутацией пакетов.

Таблица 3.1. Сравнение сетей с коммутацией каналов и пакетов

Коммутация каналов	Коммутация пакетов
Необходимо предварительно устанавливать соединение	Отсутствует этап установления соединения (дейтаграммный способ)
Адрес требуется только на этапе установления соединения	Адрес и другая служебная информация передается с каждым пакетом
Сеть может отказать абоненту в установлении соединения	Сеть всегда готова принять данные от абонента
Гарантированная пропускная способность (полоса пропускания) для взаимодействующих абонентов	Пропускная способность сети для абонентов неизвестна, задержки передачи носят случайный характер
Трафик реального времени передается без задержек	Ресурсы сети используются эффективно при передаче пульсирующего трафика
Высокая надежность передачи	Возможные потери данных из-за переполнения буферов
Нерациональное использование пропускной способности каналов, снижающее общую эффективность сети	Автоматическое динамическое распределение пропускной способности физических каналов в соответствии с фактической интенсивностью трафика абонентов

Разделение среды

Список ключевых слов: разделение среды передачи, случайный метод доступа, сетевая интерфейсная карта, или сетевой адаптер, коллизия, детерминированный метод доступа, маркер, или токен, доступа, коммуникационное оборудование, топология физических связей, топология логических связей, повторитель, концентратор, или хаб, мост, логический сегмент, коммутатор, маршрутизатор, контрольная сумма.

Ранее уже шла речь о принципах совместного использования канала несколькими интерфейсами, или, другими словами, о разделении передающей среды. Рассмотрим, как эти принципы работают в локальных сетях с коммутацией пакетов.

Разделение среды передачи долгое время было самой популярной концепцией построения локальных сетей, на ее основе разработаны такие известные технологии, как Ethernet, FDDI, Token Ring. Однако можно согласиться и с теми, кто считает, что сети на разделяемых средах уже пережили пик своей популярности и сейчас в локальных сетях доминирует коммутируемый стандарт Ethernet. Но сетевой мир так быстро меняется, что уже снова возникли признаки возрождения интереса к разделяемой среде. Новыми сферами применения разделяемой среды передачи данных являются, например, домашние проводные сети Home PNA (Home Phoneline Networking Alliance), которые используют в качестве разделяемой среды силовую проводку в доме. На принципе разделения среды осно-

ваны и персональные радиосети новой технологии *Bluetooth*, предназначенные для объединения всех «компьютеризованных» устройств личного пользования (помимо собственно компьютера это могут быть устройства PDA[1], телевизор, мобильный телефон и даже холодильник). Кроме того, в последнее время стали популярны локальные сети Radio Ethernet, применяемые для подключения пользователей к Интернету в аэропортах, вокзалах и других местах скопления мобильных пользователей.

Новое — это хорошо забытое старое — классическая сеть Ethernet тоже начиналась с радиосети Aloha Гавайского университета, где впервые была опробована разделяемая среда. Просто долгое время радиоэфир не фигурировал в качестве передающей среды в стандартах Ethernet, хотя всегда использовался в нестандартных продуктах отдельных компаний. С появлением стандартов Radio Ethernet в конце 90-х годов историческая справедливость была восстановлена.

Принципы разделения среды

Разделяемой средой называется физическая среда передачи данных, к которой непосредственно подключено несколько конечных узлов сети. Причем в каждый момент времени только один из конечных узлов получает доступ к разделяемой среде и задействует ее для передачи пакета (или нескольких пакетов) другому узлу, подключенному к этой же среде.

В качестве разделяемой среды может использоваться коаксиальный кабель, витая пара, оптическое волокно или радиоволны.

Одним из возможных подходов к разделению среды является **метод случайного доступа**[2]. Управление доступом к среде в данном случае осуществляется децентрализовано: в этом процессе участвуют все сетевые интерфейсы, непосредственно подключенные к разделяемой среде. В частности, в компьютерах эту функцию выполняют **сетевые интерфейсные карты**, или **сетевые адаптеры**.

Суть случайного метода доступа состоит в следующем.

1. Компьютер может передавать данные по сети, только если сеть свободна, то есть если никакой другой компьютер в данный момент не занимается обменом и электрические (или оптические) сигналы в среде отсутствуют.
2. После того как компьютер убеждается, что среда свободна, он начинает передачу, «захватывая» среду. Время монопольного использования разделяемой среды одним узлом ограничивается временем передачи одного кадра.
3. При попадании кадра в разделяемую среду все сетевые адаптеры одновременно начинают принимать этот кадр. Каждый из них анализирует адрес назначения, располагающийся в одном из начальных полей кадра.
4. Если этот адрес совпадает с их собственным адресом, кадр помещается во внутренний буфер сетевого адаптера. Таким образом компьютер-адресат получает предназначенные ему данные.

[1] PDA (Personal Digital Assistant) — миниатюрное вычислительное устройство, по размеру меньшее переносного компьютера. Предоставляет доступ к электронной почте, Интернету, а также основные офисные средства.

[2] Этот принцип положен в основу Ethernet.

При методе случайного доступа может возникать ситуация, когда одновременно два или более компьютеров решают, что сеть свободна, и начинают передавать информацию. Такая ситуация, называемая **коллизией**, препятствует правильной передаче данных по сети. Сигналы нескольких передатчиков накладываются друг на друга, и суммарный сигнал становится искаженным. Во всех сетевых технологиях, построенных на разделяемых средах, предусмотрен алгоритм обнаружения и корректной обработки коллизий. Вероятность возникновения коллизии зависит от интенсивности сетевого трафика. После обнаружения коллизии сетевые адаптеры, которые пытались передать свои кадры, прекращают передачу и после паузы случайной длительности пытаются снова получить доступ к среде, чтобы передать кадр, вызвавший коллизию.

Другим распространенным способом доступа к разделяемой среде является **метод детерминированного доступа**. Этот метод основан на использовании кадра специального формата, который обычно называют **маркером**, или **токеном**, доступа. Компьютер имеет право пользоваться разделяемой средой только тогда, когда он владеет токеном. Время владения токеном ограничено, так что после истечения этого срока компьютер обязан передать токен другому компьютеру. Правило, определяющее порядок передачи токена, должно гарантировать каждому компьютеру доступ к разделяемой среде в течение некоторого фиксированного времени.

Метод детерминированного доступа может быть реализован на основе как распределенного, так и централизованного подходов. В первом случае в сети нет узла, определяющего очередность владения распределенной средой, во втором — такой узел существует и называется арбитром доступа.

Причины структуризации локальных сетей

Первые локальные сети с небольшим (10–30) количеством компьютеров использовали только одну общую для всех подключенных к сети устройств разделяемую среду. При этом в соответствии с ограничениями технологий сети имели типовые топологии — общая шина (звезда) для Ethernet, кольцо для FDDI и Token Ring. Все перечисленные топологии обладают свойством однородности, то есть все компьютеры в такой сети неразличимы на уровне физических связей. Такая однородность структуры делает простой процедуру наращивания числа компьютеров, облегчает обслуживание и эксплуатацию сети.

Однако при построении больших сетей однородная структура связей превращается из достоинства в недостаток. В таких сетях использование типовых структур порождает различные ограничения, важнейшими из которых являются ограничения:

- на длину связи между узлами;
- на количество узлов в сети;
- на интенсивность трафика, порождаемого узлами сети.

Например, технология Ethernet на тонком коаксиальном кабеле позволяла использовать кабель длиной не более 185 метров, к которому можно было подключить не более 30 компьютеров. Однако если компьютеры начинали интенсивно обмениваться информацией между собой, тогда приходилось снижать число

подключенных к кабелю компьютеров до 20, а то и до 10, чтобы каждому компьютеру доставалась приемлемая доля общей пропускной способности сети.

Для снятия этих ограничений стали использовать структуризацию сети на основе специального структурообразующего **коммуникационного оборудования**, в том числе повторителей, концентраторов, мостов, коммутаторов.

Физическая структуризация локальной сети

Различают топологию физических связей (физическую структуру сети) и топологию логических связей сети (логическую структуру сети).

Конфигурация **физических связей** определяется электрическими соединениями компьютеров и может быть представлена в виде графа, узлами которого являются компьютеры и коммуникационное оборудование, а ребра соответствуют отрезкам кабеля, связывающим пары узлов.

Логические связи представляют собой пути прохождения информационных потоков по сети; они образуются путем соответствующей настройки коммуникационного оборудования.

В некоторых случаях физическая и логическая топологии сети совпадают. Например, сеть, представленная на рис. 3.15, *а*, имеет физическую кольцевую топологию. Пусть компьютеры этой сети используют метод детерминированного доступа. Причем токен всегда передается последовательно от компьютера к компьютеру в том же порядке, в котором компьютеры образуют физическое кольцо: то есть компьютер А передает токен компьютеру В, компьютер В — компьютеру С и т. д. В этом случае логическая топология сети также является кольцом.

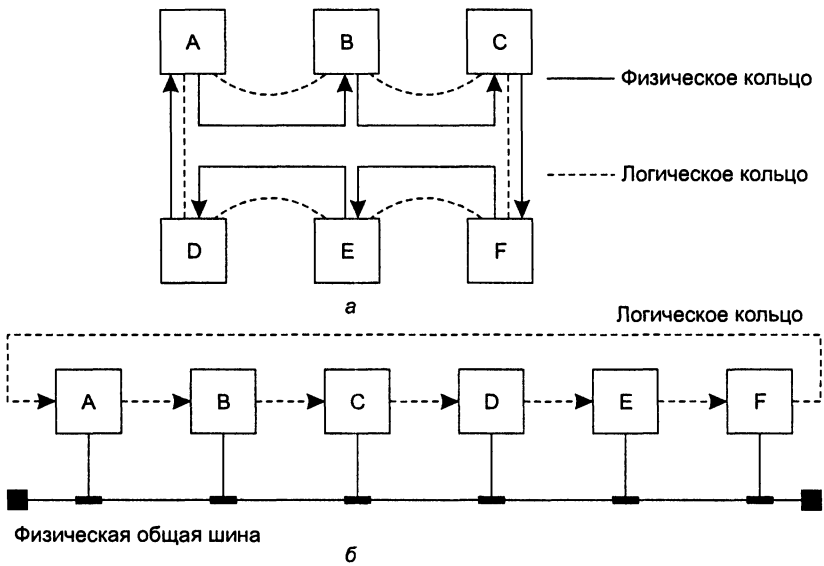


Рис. 3.15. Логическая и физическая топологии сети

Сеть, показанная на рис. 3.15, *б*, являет собой пример несовпадения физической и логической топологий. Физически компьютеры соединены по топологии общая шина (звезда). Доступ же к шине происходит не по алгоритму случайного доступа, а путем передачи токена в кольцевом порядке: от компьютера А — компьютеру В, от компьютера В — компьютеру С и т. д. Здесь порядок передачи токена уже не повторяет физические связи, а определяется логическим конфигурированием драйверов сетевых адаптеров. Ничто не мешает настроить сетевые адаптеры и их драйверы так, чтобы компьютеры образовали кольцо в другом порядке, например: В, А, С... При этом физическая структура сети никак не меняется.

Физическая структуризация единой разделяемой среды была первым шагом на пути построения более качественных локальных сетей. Цель физической структуризации — обеспечить построение сети не из одного, а из нескольких физических отрезков кабеля. Причем эти различные в физическом отношении отрезки должны были по-прежнему работать как единая разделяемая среда.

Основными средствами физической структуризации локальных сетей являются **повторители** (repeater) и **концентраторы** (concentrator), или **хабы** (hub).

Простейшее из коммуникационных устройств — повторитель — используется для физического соединения различных сегментов кабеля локальной сети с целью увеличения общей длины сети. Повторитель повторяет сигналы, приходящие из одного сегмента сети в другие ее сегменты (рис. 3.16), улучшая их физические характеристики — мощность и форму сигналов, а также синхронность следования (исправляет неравномерность интервалов между импульсами). За счет этого повторитель позволяет преодолеть ограничения на длину линий связи. Так как поток сигналов, передаваемых узлом в сеть, распространяется по всем отрезкам сети, такая сеть остается сетью с единой разделяемой средой.

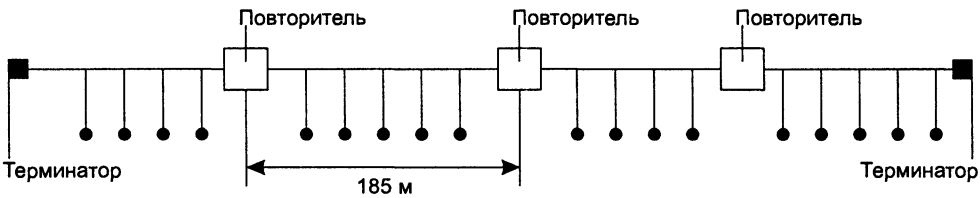


Рис. 3.16. Повторители позволяют увеличить длину сети

Повторитель, который имеет несколько портов и соединяет несколько физических сегментов, часто называют концентратором, или хабом. Эти названия отражают тот факт, что в данном устройстве сосредоточиваются все связи между сегментами сети.

ВНИМАНИЕ

Добавление в сеть повторителя всегда изменяет ее физическую топологию, но при этом оставляет без изменения логическую топологию.

Концентраторы являются необходимыми устройствами практически во всех базовых технологиях локальных сетей — Ethernet, ArcNet, Token Ring, FDDI, Fast Ethernet, Gigabit Ethernet, 100VG-AnyLAN[1]. В работе концентраторов любых технологий много общего — они повторяют сигналы, пришедшие с одного из своих портов, на других своих портах. Разница состоит в том, на каких именно портах повторяются входные сигналы. Так, концентратор Ethernet повторяет входной сигнал на *всех* своих портах, кроме того, с которого этот сигнал поступил (рис. 3.17, *а*). А концентратор Token Ring (рис. 3.17, *б*) повторяет входной сигнал только на *одном, соседнем* порту.

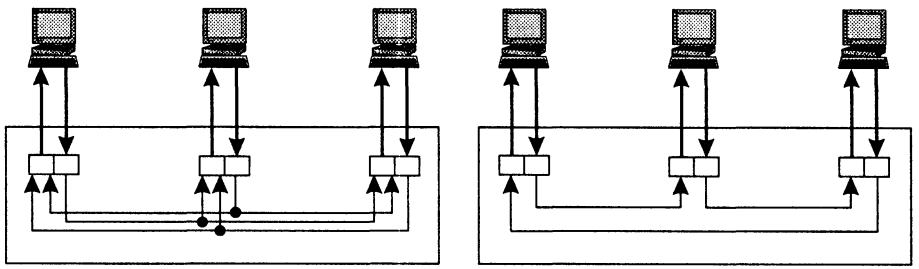


Рис. 3.17. Концентраторы различных технологий

Логическая структуризация сети на разделяемой среде

Физическая структуризация сети не позволяет справиться с такими важными проблемами, как дефицит пропускной способности, невозможность использования в разных частях сети линий связи разной пропускной способности. В таком случае может помочь логическая структуризация сети.

Типовые физические топологии сети (шина, кольцо, звезда), которые ограничивают все сетевые устройства, предоставляя им для обмена данными только одну разделяемую среду, оказываются неадекватными структуре информационных потоков в большой сети. Например, в сети с общей шиной взаимодействие любой пары компьютеров занимает ее на все время обмена, поэтому при увеличении числа компьютеров в сети шина становится узким местом.

Пример

Пусть на предприятии имелась простейшая односегментная сеть Ethernet (рис. 3.18, *а*). К коаксиальному кабелю были подключены все имеющиеся на предприятии компьютеры. Шло время, количество пользователей увеличивалось, сеть все чаще оказывалась занятой, пользователям приходилось все дольше ждать ответа сетевых приложений. Кроме того, начали сказываться ограничения на длину связей между компьютерами, так оказалось невозможным разместить компьютеры в помещении, выделенном для новой рабочей группы. Было принято решение применить концентраторы. На рис. 3.18, *б*

[1] Не все из перечисленных технологий актуальны — технологии ArcNet и 100VG-AnyLAN могут рассматриваться сегодня только в качестве примеров оригинальных технических решений.

показана сеть, полученная в результате физической структуризации. Появилась возможность разнести компьютеры пользователей на большие расстояния, и физическая структура сети стала соответствовать административному устройству предприятия. Однако проблемы, связанные с производительностью, остались нерешенными. Например, каждый раз, когда пользователь компьютера А посылал данные своему соседу, работающему за компьютером В, оказывалась блокированной вся сеть. Ничего удивительного — в соответствии с логикой работы концентратора кадр, посылаемый компьютером А компьютеру В, повторялся на всех интерфейсах всех узлов сети. И до тех пор, пока компьютер В не получал адресованный ему кадр, ни один из компьютеров сети не мог получить доступ к разделяемой среде передачи. Такая ситуация возникла из-за того, что использование концентраторов изменило только физическую структуру сети, оставив без изменения ее логическую структуру.

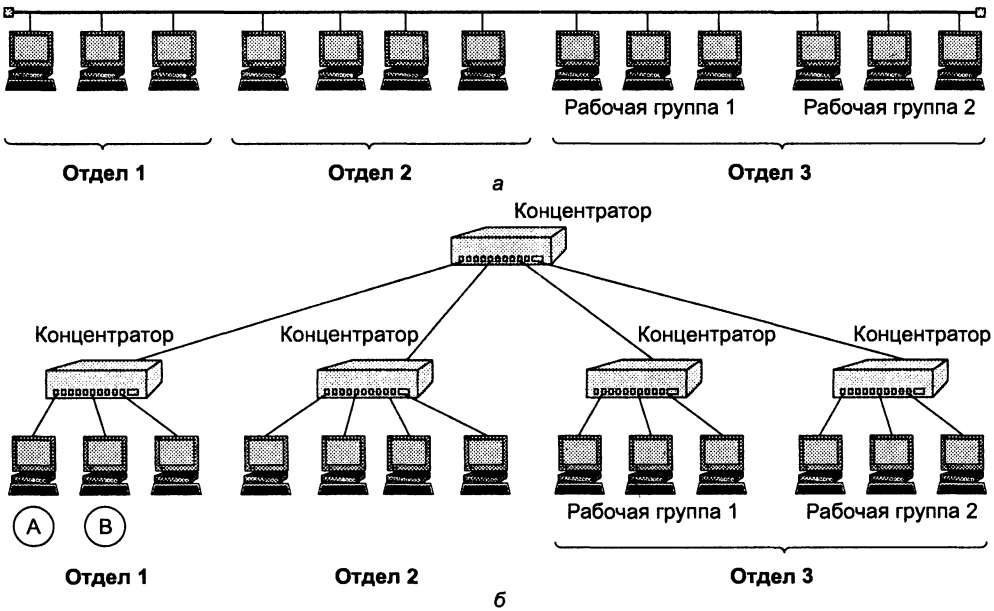


Рис. 3.18. Изменение физической структуры сети не сказывается на производительности сети

Решение представленной в примере проблемы состоит в отказе от идеи одной общей для всех узлов разделяемой среды. Например, в данном случае желательно было бы сделать так, чтобы кадры, которые передают компьютеры отдела 1, выходили бы за пределы этой части сети в том и только в том случае, если они направлялись компьютеру какого-либо из других отделов. С другой стороны, в сеть каждого из отделов должны попадать те и только те кадры, которые адресованы узлам этой сети. Таким образом, в пределах каждого отдела используется отдельная «собственная» разделяемая среда.

Распространение трафика, предназначенного для компьютеров некоторого сегмента сети, только в пределах этого сегмента называется **локализацией трафика**. **Логическая структуризация сети** — это процесс разбиения сети на сегменты с локализованным трафиком.

При правильно проведенной логической структуризации производительность сети может существенно повыситься, так как компьютеры одного отдела не будут простаивать в то время, когда обмениваются данными компьютеры других отделов. Кроме того, логическая структуризация позволяет дифференцировать доступную пропускную способность в разных частях сети.

Логическая структуризация сети проводится путем использования мостов, коммутаторов, маршрутизаторов и шлюзов.

Мост (bridge) делит единую среду передачи на части (часто называемые **логическими сегментами**), передавая информацию из одного сегмента в другой только в том случае, если такая передача действительно необходима, то есть если адрес компьютера назначения принадлежит другому сегменту (рис. 3.19). Тем самым мост изолирует трафик одного сегмента от трафика другого, повышая общую производительность сети. Локализация трафика не только экономит пропускную способность, но и снижает возможность несанкционированного доступа к данным, так как кадры не выходят за пределы своего сегмента и их сложнее перехватить злоумышленнику.

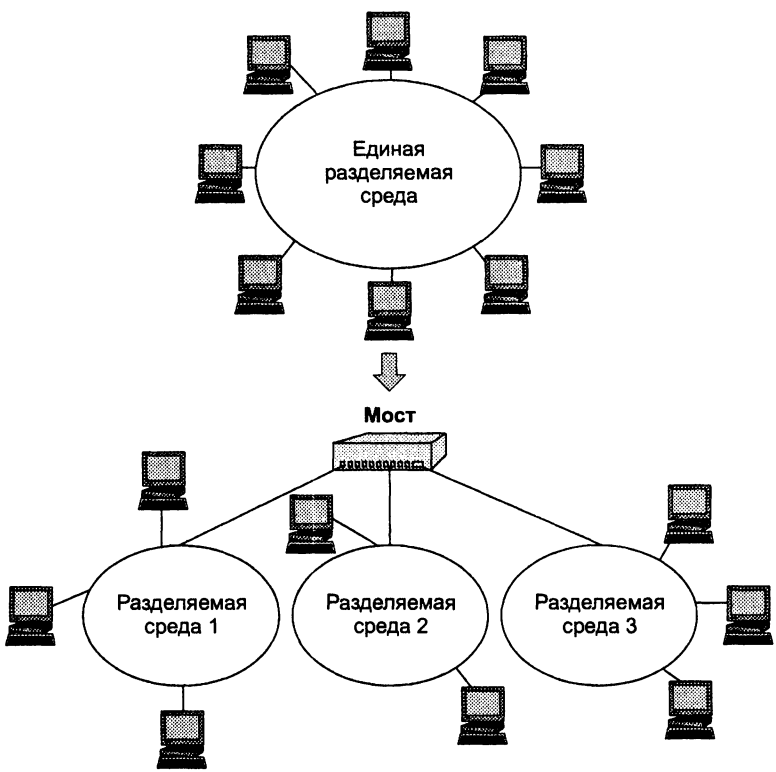


Рис. 3.19. Мост делит единую среду передачи

На рис. 3.20 показана сеть, которая была получена из сети с центральным концентратором (см. рис. 3.18, *б*) путем его замены мостом. Сети отделов 1 и 2 со-

стоят из отдельных логических сегментов, а сеть отдела 3 — из двух логических сегментов. Каждый логический сегмент построен на базе концентратора и имеет простейшую физическую структуру, образованную отрезками кабеля, связывающими компьютеры с портами концентратора. Если пользователь компьютера A пошлет данные пользователю компьютера B, находящемуся в одном с ним сегменте, то эти данные будут повторены только на сетевых интерфейсах их общего сегмента.

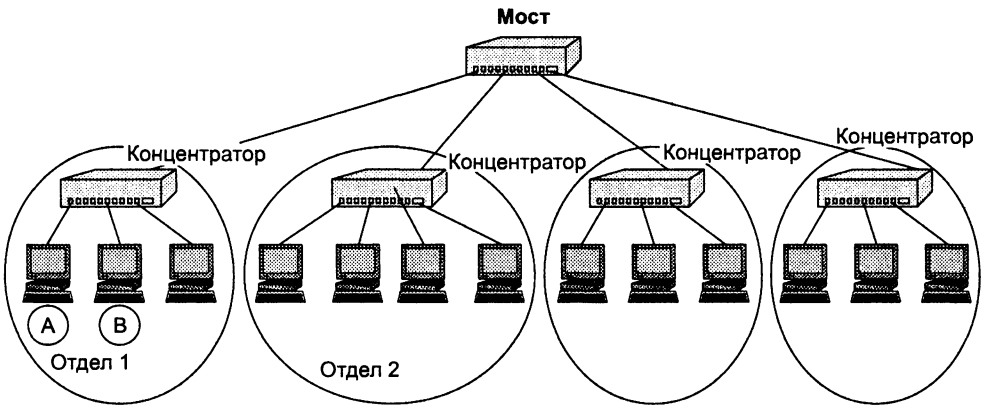


Рис. 3.20. Единая разделяемая среда с помощью моста преобразована в четыре разделяемых среды

Мосты используют для локализации трафика аппаратные адреса компьютеров. У читателя может возникнуть вопрос, каким образом мост узнает интерфейс, на который надо передать кадр, ведь аппаратный адрес не содержит никакой информации о принадлежности компьютера с данным адресом тому или иному сегменту. Конечно, такая информация может быть предоставлена мосту администратором в результате ручного конфигурирования. Однако такой способ мало пригоден для больших сетей. Мост решает эту задачу автоматически, используя простой обучающий алгоритм. Все кадры, приходящие на определенный интерфейс моста, сгенерированы компьютерами, относящимися к сегменту, подключенному к этому интерфейсу. Мост извлекает из поступающих кадров адреса отправителей и помещает их в таблицу, где делает отметку о том, на какой его интерфейс поступил каждый из кадров. Так мост определяет, какие компьютеры подключены к каждому из его интерфейсов. В дальнейшем мост использует эту информацию для передачи кадра именно на тот интерфейс, через который идет путь к компьютеру назначения. Поскольку, точная топология связей между логическими сегментами мосту неизвестна, он может правильно работать только в тех сетях, в которых межсегментные связи не образуют замкнутых контуров (петель).

Коммутатор[1] (switch) функционально подобен мосту и отличается от моста в основном более высокой производительностью. Каждый интерфейс коммутатора

[1] В данном контексте термин «коммутатор» используется в узком смысле, обозначая конкретный тип устройства.

оснащен специализированным процессором, который обрабатывает кадры по алгоритму моста независимо от процессоров других портов. За счет этого общая производительность коммутатора обычно намного выше производительности традиционного моста, имеющего один процессорный блок. Можно сказать, что коммутаторы — это усовершенствованные мосты, которые обрабатывают кадры в параллельном режиме. Когда стало экономически оправданно использовать отдельные специализированные процессоры на каждом порту коммуникационного устройства, коммутаторы локальных сетей полностью вытеснили мосты.

Ограничения, связанные с применением мостов и коммутаторов, — по топологии связей, а также ряд других, — привели к тому, что в перечне коммуникационных устройств появилось еще одно устройство — **маршрутизатор** (router). Маршрутизаторы более надежно и более эффективно, чем мосты, изолируют трафик отдельных частей сети друг от друга. Помимо локализации трафика маршрутизаторы выполняют еще много других полезных функций. Так, маршрутизаторы могут работать в сети с замкнутыми контурами, при этом они обеспечивают выбор наиболее рациональных маршрутов. Другой важной функцией маршрутизаторов является их способность связывать в единую сеть сети, построенные на базе разных сетевых технологий, например Ethernet и ATM.

Помимо перечисленных устройств отдельные части сети может соединять шлюз (gateway). Шлюз позволяет объединять сети, построенные на существенно разных программных и аппаратных платформах. Например, шлюз может позволить пользователям, работающим в сети Unix, взаимодействовать с пользователями сети Windows. Традиционно в Интернете термины «шлюз» и «маршрутизатор» используются как синонимы.

Ethernet — пример стандартной технологии

Рассмотрим, каким образом описанные выше концепции воплощены в одной из первых стандартных сетевых технологий — технологии Ethernet на разделяемой среде. В этом разделе мы коснемся только самых общих принципов функционирования одного из вариантов Ethernet. Детальное описание всех вариантов Ethernet, в том числе и коммутируемой сети Ethernet, вы найдете в части III.

- *Топология.* В стандарте Ethernet строго зафиксирована топология физических связей — общая шина. На рис. 3.21 показан простейший вариант топологии, состоящий из одного сегмента — все компьютеры сети подключены к общей разделяемой среде.

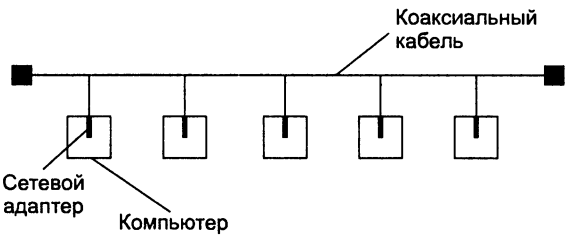


Рис. 3.21. Сеть Ethernet

- *Способ коммутации.* В технологии Ethernet используется дейтаграммная коммутация пакетов. Единицы данных, которыми обмениваются компьютеры в сети Ethernet, называются кадрами. Кадр имеет фиксированный формат и наряду с полем данных содержит различную служебную информацию. А где же в односегментной сети Ethernet на разделяемой среде происходит коммутация? Где хотя бы один коммутатор, который, как мы сказали, является главным элементом любой сети с коммутацией пакетов? Или же Ethernet представляет собой особый вид коммутации? Оказывается, коммутатор в односегментной сети Ethernet существует, но его не так просто разглядеть, потому что его функции распределены по всей сети. «Коммутатор» Ethernet состоит из сетевых адаптеров и разделяемой среды. Сетевые адаптеры представляют собой интерфейсы такого виртуального коммутатора, а разделяемая среда — коммутационный блок, который передает кадры между интерфейсами. Часть функций коммутационного блока выполняют и адаптеры, так как они решают, какой кадр адресован их компьютеру, а какой — нет.
- *Адресация.* Каждый компьютер, а точнее каждый сетевой адаптер, имеет уникальный аппаратный адрес (так называемый MAC-адрес, вы уже встречали этот акроним в главе 2). Адрес Ethernet является плоским числовым адресом, иерархия здесь не используется. Поддерживаются адреса для выборочной, широковещательной и групповой рассылки.
- *Разделение среды и мультиплексирование.* Конечные узлы для обмена данными используют единственную разделяемую среду, применяя метод случайного доступа. Информационные потоки, поступающие от конечных узлов сети Ethernet, мультиплексируются в единственном передающем канале в режиме разделения времени. То есть кадрам разных потоков поочередно предоставляется канал. Чтобы подчеркнуть не всегда очевидную разницу между понятиями мультиплексирования и разделения среды, рассмотрим ситуацию, когда из всех компьютеров сети Ethernet только один имеет потребность передавать данные, причем данные от нескольких приложений. В этом случае проблема разделения среды между сетевыми интерфейсами не возникает, в то время как задача передачи нескольких информационных потоков по общей линии связи (то есть мультиплексирование) остается.
- *Кодирование.* Адаптеры Ethernet работают с тактовой частотой 20 МГц, передавая в среду прямоугольные импульсы, соответствующие единицам и нулям данных компьютера. Когда начинается передача кадра, то все его биты передаются в сеть с постоянной скоростью 10 Мбит/с (каждый бит передается за два такта). Это скорость определяется пропускной способностью линии связи в сети Ethernet.
- *Надежность.* Для повышения надежности передачи данных Ethernet используется стандартный прием — подсчет **контрольной суммы** и передача ее в концевике кадра. Если принимающий адаптер путем повторного подсчета контрольной суммы обнаруживает ошибку в данных кадра, то такой кадр отбрасывается. Повторная передача кадра протоколом Ethernet не выполняется, эта задача должна решаться другими технологиями, например протоколом TCP в сетях TCP/IP.

- *Полудуплексный способ передачи.* Разделяемая среда Ethernet представляет собой полудуплексный канал передачи. Сетевой адаптер выполняет операции передачи данных и их приема попеременно.
- *Очереди.* На первый взгляд может показаться, что в Ethernet на разделяемой среде нет очередей, свойственных сетям с коммутацией пакетов. Однако отсутствие коммутатора с буферной памятью в сети Ethernet не означает, что очереди в ней отсутствуют. Просто здесь очереди переместились в буферную память сетевого адаптера. В те периоды времени, когда среда занята передачей кадров других сетевых адаптеров, данные (предложенная нагрузка) по-прежнему поступают в сетевой адаптер. Так как они не могут быть переданы в это время в сеть, они начинают накапливаться во внутреннем буфере адаптера Ethernet, образуя очередь. Поэтому в сети Ethernet существуют переменные задержки доставки кадров, как и во всех сетях с коммутацией пакетов.

Выводы

В сетях с коммутацией каналов по запросу пользователя создается непрерывный информационный канал, который образуется путем резервирования «цепочки» линий связи, соединяющих абонентов на время передачи данных. На всем своем протяжении канал передает данные с одной и той же скоростью. Это означает, что через сеть с коммутацией каналов можно качественно передавать данные, чувствительные к задержкам (голос, видео). Однако невозможность динамического перераспределения пропускной способности физического канала является принципиальным недостатком сети с коммутацией каналов, который делает ее неэффективной для передачи пульсирующего компьютерного трафика.

При коммутации пакетов передаваемые данные разбиваются в исходном узле на небольшие части — пакеты. Пакет снабжается заголовком, в котором указывается адрес назначения, поэтому он может быть обработан коммутатором независимо от остальных данных. Способ коммутации пакетов повышает производительность сети при передаче пульсирующего трафика, так как при обслуживании большого числа независимых потоков периоды их активности не всегда совпадают во времени. Пакеты поступают в сеть без предварительного резервирования ресурсов в том темпе, в котором их генерирует источник. Однако этот способ коммутации имеет и отрицательные стороны: задержки передачи носят случайный характер, поэтому возникают проблемы при передаче трафика реального времени.

В сетях с коммутацией пакетов может использоваться один из трех алгоритмов продвижения пакетов: дейтаграммная передача, передача с установлением логического соединения и передача с установлением виртуального канала.

Разделяемой средой называется физическая среда передачи данных (коаксиальный кабель, витая пара, оптическое волокно, радиоволны), к которой непосредственно подключено несколько конечных узлов сети и которой они могут пользоваться только по очереди. В основе таких хорошо известных технологий, как Ethernet, FDDI, Token Ring, лежит принцип разделяемой среды. Хотя, казалось бы, сети на разделяемых средах уже пережили пик своей популярности, сегодня существуют явные признаки возрождения интереса к этой технологии, о чем свидетельствуют домашние проводные сети, персональные и локальные беспроводные сети, RadioEthernet — во всех этих современных технологиях используется принцип разделения среды.

Вопросы и задания

1. Какие типы мультиплексирования и коммутации используются в телефонных сетях?
2. Какие свойства сетей с коммутацией каналов свидетельствую об их недостатках?
3. Какие свойства сетей с коммутацией пакетов негативно сказываются на передаче мультимедийной информации?
4. Используется ли буферизация в сетях с коммутацией каналов?
5. Какой элемент сети с коммутацией каналов может отказать узлу в запросе на установление составного канала:
6. Какие концепции характерны для сетевой технологии Ethernet?
7. Учитывается ли в дейтаграммных сетях существование потоков данных?
8. Дайте определение логического соединения.
9. Можно ли организовать надежную передачу данных между двумя конечными узлами без установления логического соединения?
10. Какое логическое соединение может быть названо виртуальным каналом?
11. В каких сетях используется технология виртуальных каналов?
12. Укажите, какие из перечисленных устройств являются функционально подобными:
    - ❍ хаб;
    - ❍ коммутатор;
    - ❍ концентратор;
    - ❍ повторитель;
    - ❍ маршрутизатор;
    - ❍ мост.
13. Чем отличается мост от коммутатора?
14. Верно ли следующее утверждение: «Сеть Ethernet, имеющая звездообразную топологию с концентратором в центре, надежнее, чем та же сеть на коаксиальном кабеле, имеющая топологию общей шины»?
15. Как можно повысить пропускную способность, приходящуюся на компьютер каждого конечного пользователя, в сети, построенной на основе концентраторов?
16. Определите, на сколько увеличится время передачи данных в сети с коммутацией пакетов по сравнению с сетью коммутации каналов, если известно:
    - ❍ общий объем передаваемых данных — 200 Кбайт;
    - ❍ суммарная длина канала — 5000 км;
    - ❍ скорость передачи сигнала — 0,66 скорости света;
    - ❍ пропускная способность канала — 2 Мбит/с;
    - ❍ размер пакета без учета заголовка — 4 Кбайт;

- размер заголовка — 40 байт;
- интервал между пакетами — 1 мс;
- количество промежуточных коммутаторов — 10;
- время коммутации на каждом коммутаторе — 2 мс.

Считайте, что сеть работает в недогруженном режиме, так что очереди в коммутаторах отсутствуют.

17. Если все коммуникационные устройства в приведенном на рис. 3.22 фрагменте сети являются концентраторами, то на каких портах появится кадр, если его отправил компьютер A компьютеру B?

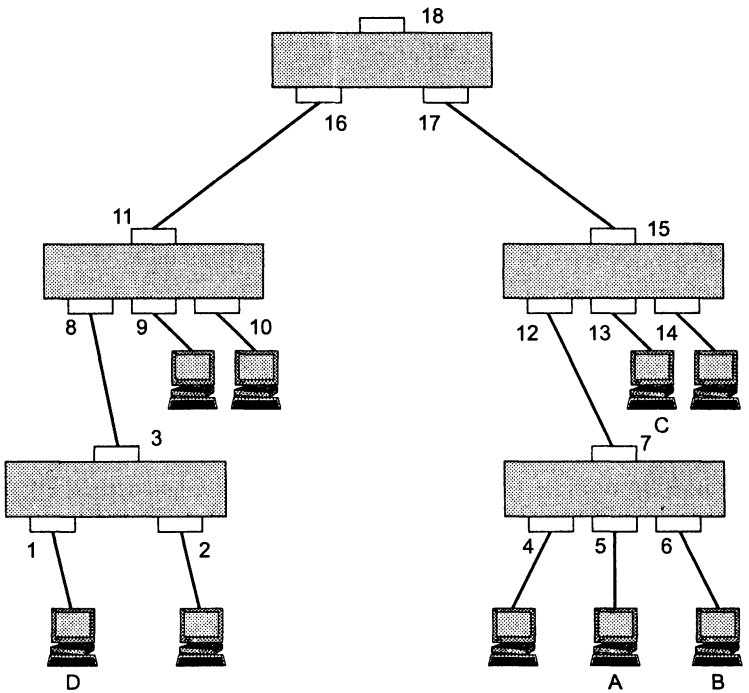


Рис. 3.22. Фрагмент сети

18. Если все коммуникационные устройства в приведенном на рис. 3.22 фрагменте сети являются коммутаторами, то на каких портах появится кадр, если его отправил компьютер A компьютеру B?
19. Если все коммуникационные устройства в приведенном на рис. 3.22 фрагменте сети являются коммутаторами, кроме одного концентратора, к которому подключены компьютеры A и B, то на каких портах появится кадр, если его отправил компьютер A компьютеру D?
20. В дейтаграммной сети между узлами A и B существует три потока и три альтернативных маршрута. Можно ли направить каждый поток по отдельному маршруту?

21. В сети, поддерживающей технику виртуальных каналов, между узлами A и B существует три потока и три альтернативных маршрута. Можно ли направить каждый поток по отдельному маршруту?
22. Сеть построена на разделяемой среде с пропускной способностью 10 Мбит/с и состоит из 100 узлов. С какой максимальной скоростью могут обмениваться данными два узла в сети?
23. Сеть может передавать данные в двух режимах: с помощью дейтаграмм и по виртуальным каналам. Какие соображения вы бы приняли во внимание при выборе того или иного режима для передачи ваших данных, если главным критерием выбора для вас является скорость и надежность доставки?
24. Считаете ли вы, что сети с коммутацией каналов в ближайшем будущем будут заменены сетями с коммутацией пакетов? Или же напротив, сети с коммутацией пакетов будут вытеснены сетями с коммутацией каналов? Или же эти технологии будут сосуществовать? Приведите аргументы в пользу вашего мнения. Рассмотрите разные области использования этих технологий.

ГЛАВА 4 Архитектура и стандартизация сетей

Архитектура подразумевает представление сети в виде системы элементов, каждый из которых выполняет определенную частную функцию, при этом все элементы вместе согласованно решают общую задачу взаимодействия компьютеров. Другими словами, архитектура сети отражает декомпозицию общей задачи взаимодействия компьютеров на отдельные подзадачи, которые должны решаться отдельными элементами сети. Одним из важных элементов архитектуры сети является коммуникационный протокол — формализованный набор правил взаимодействия узлов сети.

Прорывом в стандартизации архитектуры компьютерной сети стала разработка модели взаимодействия открытых систем (Open System Interconnection, OSI), которая в начале 80-х годов обобщила накопленный к тому времени опыт. Модель OSI является международным стандартом и определяет способ декомпозиции задачи взаимодействия «по вертикали», поручая эту задачу коммуникационным протоколам семи уровней. Уровни образуют иерархию, известную как стек протоколов, где каждый вышестоящий уровень использует нижестоящий в качестве удобного инструмента для решения своих задач.

Существующие сегодня (или существовавшие еще недавно) стеки протоколов в целом отражают архитектуру модели OSI. Однако в каждом стеке протоколов имеются свои особенности и отличия от архитектуры OSI. Так, наиболее популярный стек TCP/IP состоит из четырех уровней. Стандартная архитектура компьютерной сети определяет также распределение протоколов между элементами сети — конечными узлами (компьютерами) и промежуточными узлами (коммутаторами и маршрутизаторами). Промежуточные узлы выполняют только транспортные функции стека протоколов, передавая трафик между конечными узлами. Конечные узлы поддерживают весь стек протоколов, предоставляя информационные услуги, например веб-сервис. Такое распределение функций означает смещение «интеллекта» сети на ее периферию.

Декомпозиция задачи сетевого взаимодействия

Список ключевых слов: декомпозиция, модуль, спецификация, многоуровневый подход, иерархия, межуровневый интерфейс, интерфейс услуг, протокол, стек протоколов, файловая система, протокольные сущности, сообщение, заголовок сообщения, поля данных сообщения.

Организация взаимодействия между устройствами сети является сложной задачей. Для решения сложных задач используется известный универсальный прием — *декомпозиция*, то есть разбиение одной сложной задачи на несколько более простых задач-модулей. Декомпозиция состоит в четком определении функций каждого модуля, а также порядка их взаимодействия (то есть межмодульных интерфейсов). При таком подходе каждый модуль можно рассматривать как «черный ящик», абстрагируясь от его внутренних механизмов и концентрируя внимание на способе взаимодействия этих модулей. В результате такого логического упрощения задачи появляется возможность независимого тестирования, разработки и модификации модулей. Так, любой из показанных на рис. 4.1 модулей может быть переписан заново. Пусть, например, это будет модуль A, и если при этом разработчики сохранят без изменения межмодульные связи (в данном случае интерфейсы A-B и A-C), то это не потребует никаких изменений в остальных модулях.

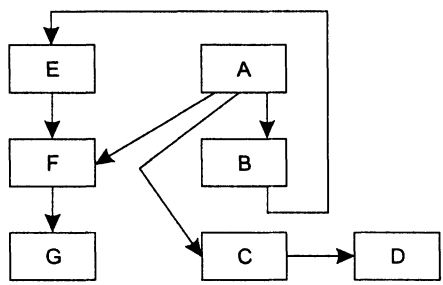


Рис. 4.1. Пример декомпозиции задачи

Многоуровневый подход

Еще более эффективной концепцией, развивающей идею декомпозиции, является *многоуровневый подход*. После представления исходной задачи в виде множества модулей эти модули группируют и упорядочивают по уровням, образующим иерархию. В соответствии с принципом иерархии для каждого промежуточного уровня можно указать непосредственно примыкающие к нему соседние вышележащий и нижележащий уровни (рис. 4.2).

Группа модулей, составляющих каждый уровень, для решения своих задач должна обращаться с запросами только к модулям соседнего нижележащего уровня. С другой стороны, результаты работы каждого из модулей, отнесенных к неко-

торому уровню, могут быть переданы только модулям соседнего вышележащего уровня. Такая иерархическая декомпозиция задачи предполагает четкое определение функций и интерфейсов не только отдельных модулей, но и каждого уровня.

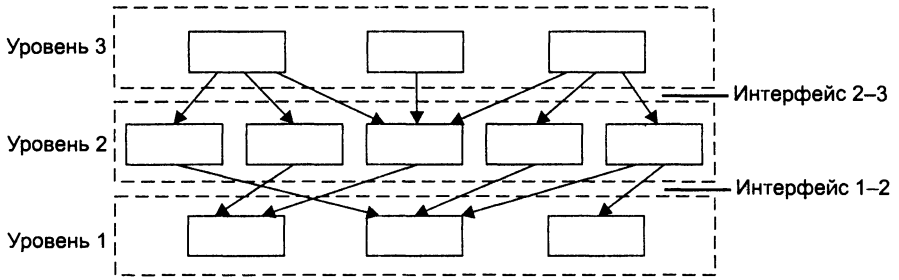


Рис. 4.2. Многоуровневый подход — создание иерархии задач

Межуровневый интерфейс, называемый также **интерфейсом услуг**, определяет набор функций, которые нижележащий уровень предоставляет вышележащему (рис. 4.3).

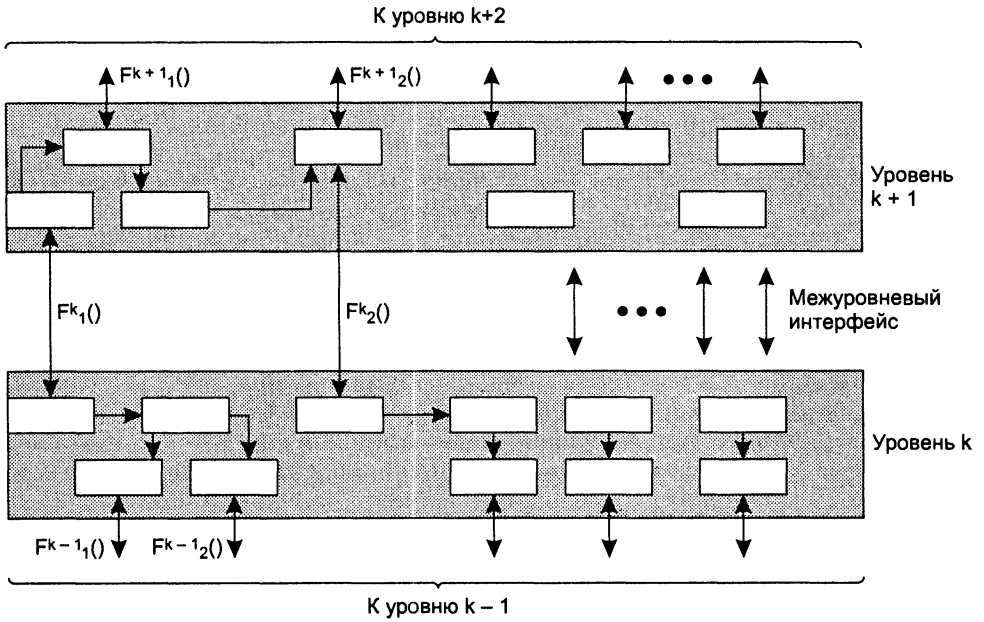


Рис. 4.3. Концепция многоуровневого взаимодействия

Такой подход дает возможность проводить разработку, тестирование и модификацию отдельного уровня независимо от других уровней. Иерархическая декомпозиция позволяет, двигаясь от более низкого уровня к более высокому, переходить ко все более и более абстрактному, а значит, и более простому представлению исходной задачи.

Пример

Рассмотрим задачу считывания логической записи из файла, расположенного на локальном диске. Ее (очень упрощенно) можно представить в виде следующей иерархии частных задач.

1. *Поиск по символьному имени файла его характеристик, необходимых для доступа к данным: информации о физическом расположении файла на диске, размере и др.* Поскольку функции этого уровня связаны только с просмотром каталогов, представление о файловой системе на этом уровне чрезвычайно абстрактно: файловая система имеет вид графа, в узлах которого находятся каталоги, а листьями являются файлы. Никакие детали физической и логической организации данных на диске данный уровень не интересуют.
2. *Определение считываемой части файла.* Для решения этой задачи необходимо снизить степень абстракции файловой системы. Функции данного уровня оперируют с файлом как с совокупностью определенным образом связанных физических блоков диска.
3. *Считывание данных с диска.* После определения номера физического блока файловая система обращается к системе ввода-вывода для выполнения операции чтения. На этом уровне уже фигурируют такие детали устройства файловой системы, как номера цилиндров, дорожек, секторов.

Среди функций, которые может запросить прикладная программа, обращаясь к верхнему уровню файловой системы, может быть, например, такая:

```
ПРОЧИТАТЬ 22 ЛОГИЧЕСКУЮ ЗАПИСЬ ФАЙЛА DIR1/MY/FILE.TXT
```

Верхний уровень не может выполнить этот запрос «только своими силами», определив по символьному имени DIR1/MY/FILE.TXT физический адрес файла, он обращается с запросом к нижележащему уровню:

```
ПРОЧИТАТЬ 22 ЛОГИЧЕСКУЮ ЗАПИСЬ ИЗ ФАЙЛА,
ИМЕЮЩЕГО ФИЗИЧЕСКИЙ АДРЕС 174 И РАЗМЕР 235
```

В ответ на запрос второй уровень определяет, что файл с адресом 174 занимает на диске пять несмежных областей, а искомая запись находится в четвертой области в физическом блоке 345. После этого он обращается к драйверу с запросом о чтении требуемой логической записи.

В соответствии с этой упрощенной схемой взаимодействие уровней файловой системы было однонаправленным — сверху вниз. Однако реальная картина существенно сложнее. Действительно, чтобы определить характеристики файла, верхний уровень должен «раскрутить» его символьное имя, то есть последовательно прочитать всю цепочку каталогов, указанную в имени файла. А это значит, что для решения свой задачи он несколько раз обратится к нижележащему уровню, который в свою очередь несколько раз «попросит» драйвер считать данные каталогов с диска. И каждый раз результаты будут передаваться снизу вверх.

Задача организации взаимодействия компьютеров в сети тоже может быть представлена в виде иерархически организованного множества модулей. Например, модулям нижнего уровня можно поручить вопросы, связанные с надежной передачей информации между двумя соседними узлами, а модулям следующего, более высокого уровня — транспортировку сообщений в пределах всей сети. Очевидно, что последняя задача — организация связи двух любых, не обязательно соседних узлов — является более общей и поэтому ее решение может быть полу-

чено путем многократных обращений к нижележащему уровню. Так, организация взаимодействия узлов A и B (рис. 4.4) может быть сведена к поочередному взаимодействию пар промежуточных смежных узлов.

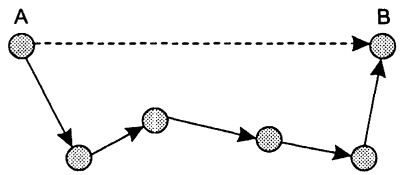


Рис. 4.4. Взаимодействие произвольной пары узлов

Протокол и стек протоколов

Многоуровневое представление средств сетевого взаимодействия имеет свою специфику, связанную с тем, что в процессе обмена сообщениями участвуют, по меньшей мере, *две стороны*, то есть в данном случае необходимо организовать согласованную работу двух иерархий аппаратных и программных средств, работающих на разных компьютерах. Оба участника сетевого обмена должны принять множество соглашений. Например, они должны согласовать уровни и форму электрических сигналов, способ определения размера сообщений, договориться о методах контроля достоверности и т. п. Другими словами, соглашения должны быть приняты на всех уровнях, начиная от самого низкого — уровня передачи битов и заканчивая самым высоким, реализующим обслуживание пользователей сети.

На рис. 4.5 показана модель взаимодействия двух узлов.

С каждой стороны средства взаимодействия представлены четырьмя уровнями. Каждый уровень поддерживает интерфейсы двух типов. Во-первых, это интерфейсы услуг с выше- и нижележащим уровнями «своей» иерархии средств. Во-вторых, это интерфейс со средствами взаимодействия другой стороны, расположенными на том же уровне иерархии. Этот тип интерфейса называют **протоколом**. Таким образом, протокол всегда является одноранговым интерфейсом.

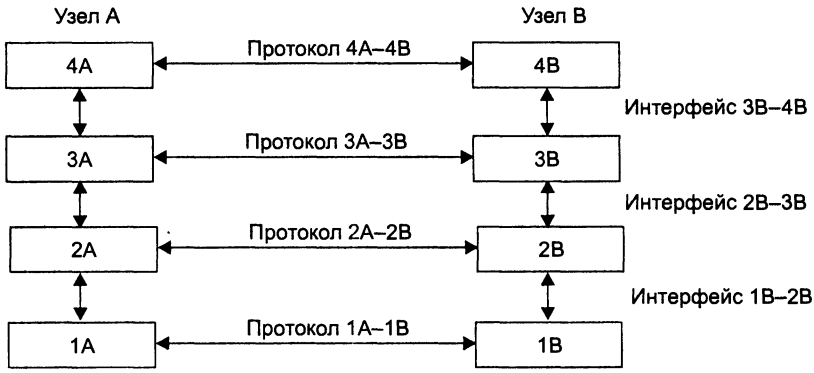


Рис. 4.5. Взаимодействие двух узлов

ПРИМЕЧАНИЕ

В сущности, термины «протокол» и «интерфейс» выражают одно и то же понятие — формализованное описание процедуры взаимодействия двух объектов, но традиционно в сетях за ними закрепили разные области действия: протоколы определяют правила взаимодействия модулей одного уровня в разных узлах, а интерфейсы — правила взаимодействия модулей соседних уровней в одном узле.

Иерархически организованный набор протоколов, достаточный для организации взаимодействия узлов в сети, называется **стеком протоколов**.

Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней, как правило, программными средствами.

Программный модуль, реализующий некоторый протокол, называют протокольной сущностью, или, для краткости, тоже протоколом. Понятно, что один и тот же протокол может быть реализован с разной степенью эффективности. Именно поэтому при сравнении протоколов следует учитывать не только логику их работы, но и качество программной реализации. Более того, на эффективность взаимодействия устройств в сети влияет качество всей совокупности протоколов, составляющих стек, в частности то, *насколько рационально распределены функции между протоколами* разных уровней и насколько хорошо определены интерфейсы между ними.

Протокольные сущности одного уровня двух взаимодействующих сторон обмениваются сообщениями в соответствии с определенным для них протоколом. Сообщения состоят из заголовка и поля данных (иногда оно может отсутствовать). Обмен сообщениями является своеобразным языком общения, с помощью которого каждая из сторон «объясняет» другой стороне, что необходимо сделать на каждом этапе взаимодействия. Работа каждого протокольного модуля состоит в интерпретации заголовков поступающих к нему сообщений и выполнении связанных с этим действий. Заголовки сообщений разных протоколов имеют разную структуру, что соответствует различиям в их функциональности. Понятно, что чем сложнее структура заголовка сообщения, тем более сложные функции возложены на соответствующий протокол.

Модель OSI

Список ключевых слов: взаимодействие открытых систем, прикладной программный интерфейс, протокольная единица данных, сообщение, кадр, пакет, дейтаграмма, сегмент, физический уровень, канальный уровень, контрольная последовательность кадра, управление доступом к среде, сетевой уровень, составная сеть, или Интернет, межсетевое взаимодействие, сетевой, или глобальный, адрес, таблица маршрутизации, маршрутизируемый протокол, маршрутизирующий протокол, или протокол маршрутизации, транспортный уровень, классы транспортного сервиса, сеансовый уровень, уровень представления, прикладной уровень, коммутация пакетов.

Из того, что протокол является соглашением, принятым двумя взаимодействующими узлами сети, совсем не следует, что он обязательно является стандартным. Но на практике при реализации сетей стремятся использовать стандартные протоколы. Это могут быть фирменные, национальные или международные стандарты.

В начале 80-х годов ряд международных организаций по стандартизации, в частности International Organization for Standardization (ISO), часто называемая также International Standards Organization, а также International Telecommunications Union (ITU) и некоторые другие, — разработали стандартную модель **взаимодействия открытых систем** (Open System Interconnection, OSI). Эта модель сыграла значительную роль в развитии компьютерных сетей.

Общая характеристика модели OSI

К концу 70-х годов в мире уже существовало большое количество фирменных стеков коммуникационных протоколов, среди которых можно назвать, например, такие популярные стеки, как DECnet, TCP/IP и SNA. Такое разнообразие средств межсетевого взаимодействия вывело на первый план проблему несовместимости устройств, использующих разные протоколы. Одним из путей разрешения этой проблемы в то время виделся всеобщий переход на единый, общий для всех систем стек протоколов, созданный с учетом недостатков уже существующих стеков. Такой академический подход к созданию нового стека начался с разработки модели OSI и занял семь лет (с 1977 по 1984 год). Назначение модели OSI состоит в обобщенном представлении средств сетевого взаимодействия. Она разрабатывалась в качестве своего рода универсального языка сетевых специалистов, именно поэтому ее называют справочной моделью.

ВНИМАНИЕ

Модель OSI определяет, во-первых, уровни взаимодействия систем в сетях с коммутацией пакетов, во-вторых, стандартные названия уровней, в-третьих, функции, которые должен выполнять каждый уровень. Модель OSI не содержит описаний реализаций конкретного набора протоколов.

В модели OSI (рис. 4.6) средства взаимодействия делятся на *семь* уровней: прикладной, представления, сеансовый, транспортный, сетевой, канальный и физический. Каждый уровень имеет дело с совершенно определенным аспектом взаимодействия сетевых устройств.

ВНИМАНИЕ

Модель OSI описывает только системные средства взаимодействия, реализуемые операционной системой, системными утилитами, системными аппаратными средствами. Модель не включает средства взаимодействия приложений конечных пользователей. Важно различать уровень взаимодействия приложений и прикладной уровень семиуровневой модели.

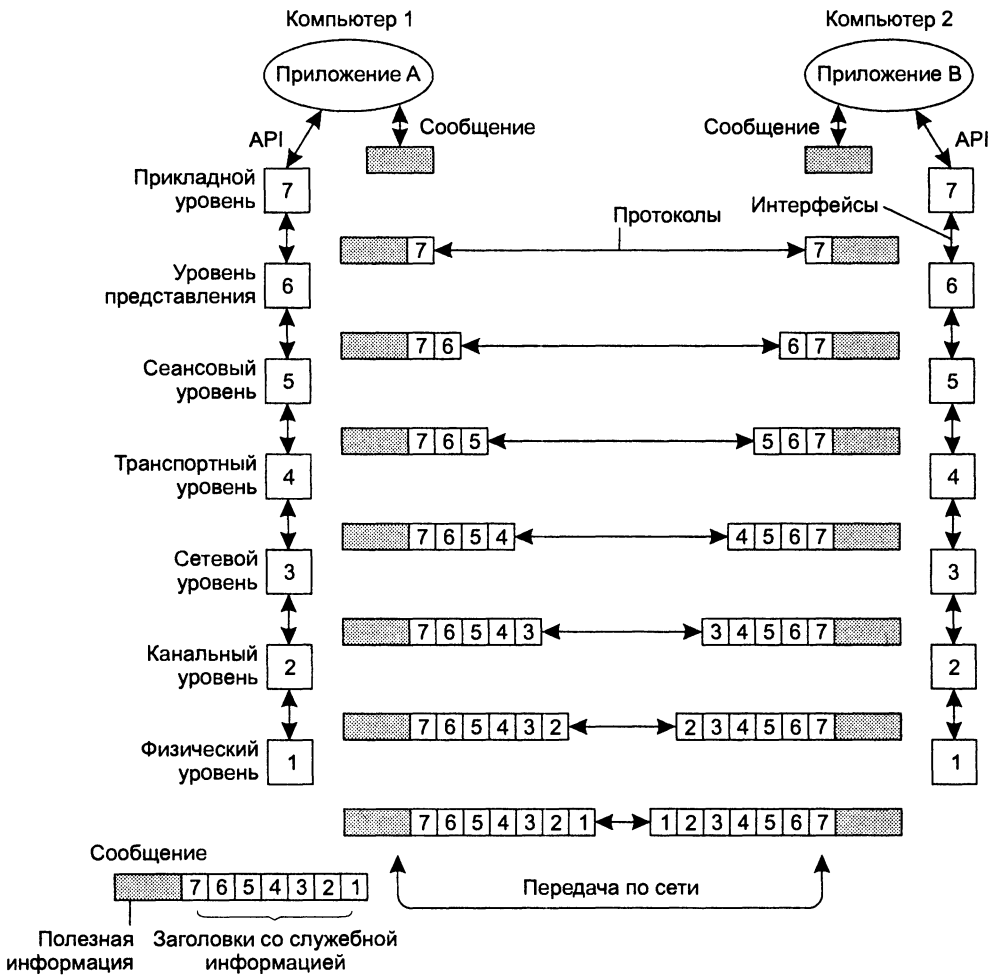


Рис. 4.6. Модель взаимодействия открытых систем ISO/OSI

Приложения могут реализовывать собственные протоколы взаимодействия, используя для этих целей многоуровневую совокупность системных средств. Именно для этого в распоряжение программистов предоставляется **прикладной программный интерфейс** (Application Program Interface, API). В соответствии с идеальной схемой модели OSI приложение может обращаться с запросами к самому верхнему уровню — уровню приложений, однако на практике многие стеки коммуникационных протоколов предоставляют возможность программистам напрямую обращаться к сервисам, или службам, расположенным ниже уровней.

Например, некоторые СУБД имеют встроенные средства удаленного доступа к файлам. В этом случае приложение, выполняя доступ к удаленным ресурсам, не использует системную файловую службу; оно обходит верхние уровни модели OSI и обращается непосредственно к ответственным за транспортировку

сообщений по сети системным средствам, которые располагаются на нижних уровнях модели OSI.

Итак, пусть приложение узла А хочет взаимодействовать с приложением узла В. Для этого приложение А обращается с запросом к прикладному уровню, например к файловой службе. На основании этого запроса программное обеспечение прикладного уровня формирует сообщение стандартного формата. Но для того чтобы доставить эту информацию по назначению, предстоит решить еще много задач, ответственность за которые несут нижележащие уровни.

После формирования сообщения прикладной уровень направляет его вниз по стеку уровню представления. Протокол уровня представления на основании информации, полученной из заголовка сообщения прикладного уровня, выполняет требуемые действия и добавляет к сообщению собственную служебную информацию — заголовок уровня представления, в котором содержатся указания для протокола уровня представления машины-адресата. Полученное в результате сообщение передается вниз сеансовому уровню, который в свою очередь добавляет свой заголовок и т. д. (Некоторые реализации протоколов помещают служебную информацию не только в начале сообщения в виде заголовка, но и в конце в виде так называемого концевика.) Наконец, сообщение достигает нижнего, физического уровня, который, собственно, и передает его по линиям связи машине-адресату. К этому моменту сообщение «обрастает» заголовками всех уровней (рис. 4.7).

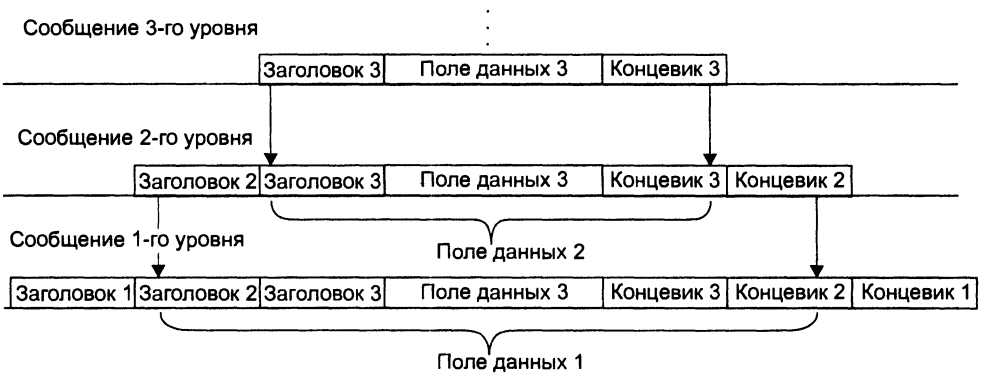


Рис. 4.7. Вложенность сообщений различных уровней

Физический уровень помещает сообщение на физический выходной интерфейс компьютера 1, и оно начинает свое «путешествие» по сети (до этого момента сообщение передавалось от одного уровня другому в пределах компьютера 1).

Когда сообщение по сети поступает на входной интерфейс компьютера 2, оно принимается его физическим уровнем и последовательно перемещается вверх с уровня на уровень. Каждый уровень анализирует и обрабатывает заголовок своего уровня, выполняя соответствующие функции, а затем удаляет этот заголовок и передает сообщение вышележащему уровню.

Как видно из описания, протокольные сущности одного уровня не общаются между собой непосредственно, в этом общении всегда участвуют посредники —

средства протоколов нижележащих уровней. И только физические уровни различных узлов взаимодействуют непосредственно.

В стандартах ISO для обозначения единиц обмена данными, с которыми имеют дело протоколы разных уровней, используется общее название **протокольная единица данных** (Protocol Data Unit, PDU). Для обозначения единиц обмена данными конкретных уровней часто используются *специальные* названия, в частности: **сообщение**, **кадр**, **пакет**, **дейтаграмма**, **сегмент**.

Физический уровень

Физический уровень (physical layer) имеет дело с передачей потока битов по физическим каналам связи, таким как коаксиальный кабель, витая пара, оптоволоконный кабель или цифровой территориальный канал. Мы уже рассмотрели основные особенности этого уровня в разделе «Физическая передача данных по линиям связи» главы 2.

Функции физического уровня реализуются на всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом.

Примером протокола физического уровня может служить спецификация 10Base-T технологии Ethernet, которая определяет в качестве используемого кабеля неэкранированную витую пару категории 3 с волновым сопротивлением 100 Ом, разъем RJ-45, максимальную длину физического сегмента 100 метров, манчестерский код для представления данных в кабеле, а также некоторые другие характеристики среды и электрических сигналов.

Физический уровень не вникает в смысл информации, которую он передает. Для него эта информация представляет однородный поток битов, которые нужно доставить без искажений и в соответствии с заданной тактовой частотой (интервалом между соседними битами).

Канальный уровень

Канальный уровень (data link layer) является первым уровнем (если идти снизу вверх), который работает в режиме коммутации пакетов. На этом уровне PDU обычно носит название **кадр** (frame).

Функции средств канального уровня определяются по-разному для локальных и глобальных сетей.

- *В локальных сетях* канальный уровень должен обеспечивать доставку кадра между *любыми* узлами сети. При этом предполагается, что сеть имеет типовую топологию, например общую шину, кольцо, звезду или дерево (иерархическую звезду). Примерами технологий локальных сетей, применение которых ограничено типовыми топологиями, являются Ethernet, FDDI, Token Ring.
- *В глобальных сетях* канальный уровень должен обеспечивать доставку кадра только между двумя *соседними* узлами, соединенными индивидуальной линией связи. Примерами двухточечных протоколов (как часто называют такие

протоколы) могут служить широко распространенные протоколы PPP и HDLC. На основе двухточечных связей могут быть построены сети произвольной топологии.

Для связи локальных сетей между собой или для доставки сообщений между любыми конечными узлами глобальной сети используются средства более высокого сетевого уровня.

Одной из функций канального уровня является поддержание интерфейсов с нижележащим физическим уровнем и вышележащим сетевым уровнем. Сетевой уровень направляет канальному уровню пакет для передачи в сеть или принимает от него пакет, полученный из сети. Физический уровень используется канальным как инструмент, который принимает и передает в сеть последовательности битов.

Начнем рассмотрение работы канального уровня, начиная с момента, когда сетевой уровень отправителя передает канальному уровню пакет, а также указание, какому узлу его передать. Для решения этой задачи канальный уровень создает кадр, который имеет поле данных и заголовок. Канальный уровень помещает (*инкапсулирует*) пакет в поле данных кадра и заполняет соответствующей служебной информацией заголовок кадра. Важнейшей информацией заголовка кадра является адрес назначения, на основании которого коммутаторы сети будут продвигать пакет.

Одной из задач канального уровня является *обнаружение и коррекция ошибок*. Для этого канальный уровень фиксирует границы кадра, помещая специальную последовательность битов в его начало и конец, а затем добавляет к кадру контрольную сумму, которая называется также **контрольной последовательностью кадра** (Frame Check Sequence, FCS). Контрольная сумма вычисляется по некоторому алгоритму как функция от всех байтов кадра. По значению FCS узел назначения сможет определить, были или нет искажены данные кадра в процессе передачи по сети.

Однако прежде, чем переправить кадр физическому уровню для непосредственной передачи данных в сеть, канальному уровню может потребоваться решить еще одну важную задачу. Если в сети используется разделяемая среда, то прежде чем физический уровень начнет передавать данные, канальный уровень должен *проверить доступность среды*. Функции проверки доступности разделяемой среды иногда выделяют в отдельный подуровень **управления доступом к среде** (Media Access Control, MAC).

Если разделяемая среда освободилась (когда она не используется, то такая проверка, конечно, пропускается), кадр передается средствами физического уровня в сеть, проходит по каналу связи и поступает в виде последовательности битов в распоряжение физического уровня узла назначения. Этот уровень в свою очередь передает полученные биты «наверх» канальному уровню своего узла. Последний группирует биты в кадры, снова вычисляет контрольную сумму полученных данных и сравнивает результат с контрольной суммой, переданной в кадре. Если они совпадают, кадр считается правильным. Если же контрольные суммы не совпадают, фиксируется ошибка. В функции канального уровня входит не

только обнаружение ошибок, но и исправление их за счет повторной передачи поврежденных кадров. Однако эта функция не является обязательной и в некоторых реализациях канального уровня она отсутствует, например в Ethernet, Token Ring, FDDI и Frame Relay.

Протоколы канального уровня реализуются компьютерами, мостами, коммутаторами и маршрутизаторами. В компьютерах функции канального уровня реализуются совместными усилиями сетевых адаптеров и их драйверов.

Протокол канального уровня обычно работает в пределах сети, являющейся одной из составляющих более крупной составной сети, объединенной протоколами сетевого уровня. Адреса, с которыми работает протокол канального уровня, используются для доставки кадров только в пределах этой сети, а для перемещения пакетов между сетями применяются уже адреса следующего, сетевого, уровня.

В локальных сетях канальный уровень поддерживает весьма мощный и законченный набор функций по пересылке сообщений между узлами сети. В некоторых случаях протоколы канального уровня локальных сетей оказываются самодостаточными транспортными средствами и могут допускать работу непосредственно поверх себя протоколов прикладного уровня или приложений без привлечения средств сетевого и транспортного уровней. Тем не менее для качественной передачи сообщений в сетях с произвольной топологией функций канального уровня оказывается недостаточно.

Это утверждение в еще большей степени справедливо для глобальных сетей, в которых протокол канального уровня реализует достаточно простую функцию передачи данных между соседними узлами.

Сетевой уровень

Сетевой уровень (network layer) служит для образования единой транспортной системы, объединяющей несколько сетей, называемой **составной сетью**, или **интернетом**[1].

Технология, позволяющая соединять в единую сеть множество сетей, в общем случае построенных на основе разных технологий, называется технологией **межсетевого взаимодействия** (internetworking).

На рис. 4.8 показаны несколько сетей, каждая из которых использует собственную технологию канального уровня: Ethernet, FDDI, Token Ring, ATM, Frame Relay. На базе этих технологий каждая из указанных сетей может связывать между собой любых пользователей, но только *своей* сети, и не способна обеспечить передачу данных в другую сеть. Причина такого положения вещей очевидна и кроется в существенных отличиях одной технологии от другой. Даже наиболее близкие технологии LAN — Ethernet, FDDI, Token Ring, — имеющие одну и ту же

[1] Не следует путать интернет (со строчной буквы) с Интернетом (с прописной буквы). Интернет — это самая известная и охватывающая весь мир реализация составной сети, построенная на основе технологии TCP/IP.

систему адресации (адреса подуровня MAC, называемые MAC-адресами), отличаются друг от друга форматом используемых кадров и логикой работы протоколов. Еще больше отличий между технологиями LAN и WAN. Во многих технологиях WAN задействована техника предварительно устанавливаемых виртуальных каналов, идентификаторы которых применяются в качестве адресов. Все технологии имеют собственные форматы кадров (в технологии ATM кадр даже называется иначе — ячейкой) и, конечно, собственные стеки протоколов.

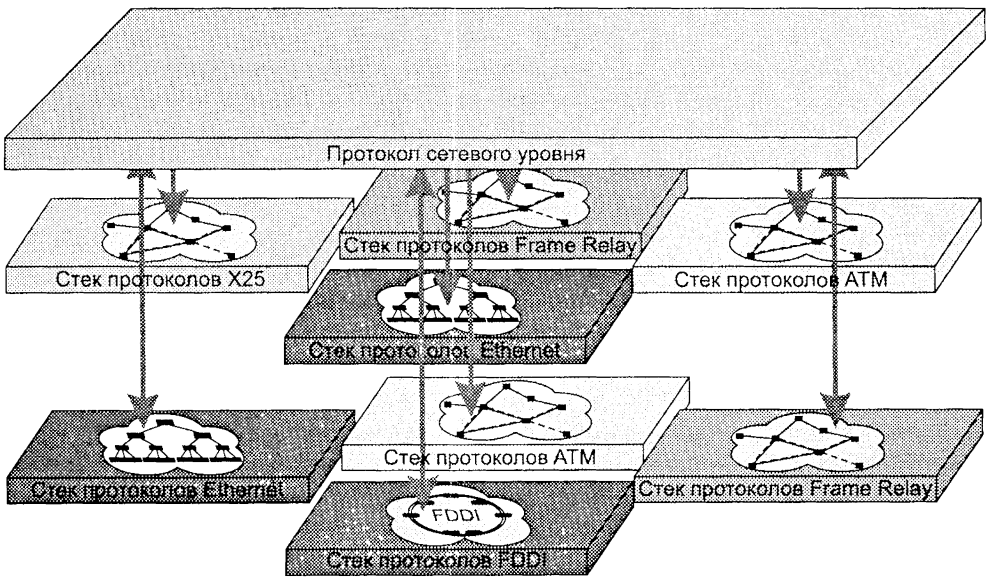


Рис. 4.8. Необходимость сетевого уровня

Чтобы связать между собой сети, построенные на основе столь отличающихся технологий, нужны *дополнительные средства*, и такие средства предоставляет сетевой уровень.

Функции сетевого уровня реализуются:

- группой протоколов;
- специальными устройствами — маршрутизаторами.

Одной из функций маршрутизатора является *физическое соединение сетей*. Маршрутизатор имеет несколько сетевых интерфейсов, подобных интерфейсам компьютера, к каждому из которых может быть подключена одна сеть. Таким образом, все интерфейсы маршрутизатора можно считать узлами разных сетей. Маршрутизатор может быть реализован программно, на базе универсального компьютера (например, типовая конфигурация Unix или Windows включает программный модуль маршрутизатора). Однако чаще маршрутизаторы реализуются на базе специализированных аппаратных платформ. В состав программного обеспечения маршрутизатора входят протокольные модули сетевого уровня.

Итак, чтобы связать сети, показанные на рис. 4.8, необходимо соединить все эти сети маршрутизаторами и установить протокольные модули сетевого уровня на все конечные узлы пользователей, которые хотели бы связываться через составную сеть (рис. 4.9).

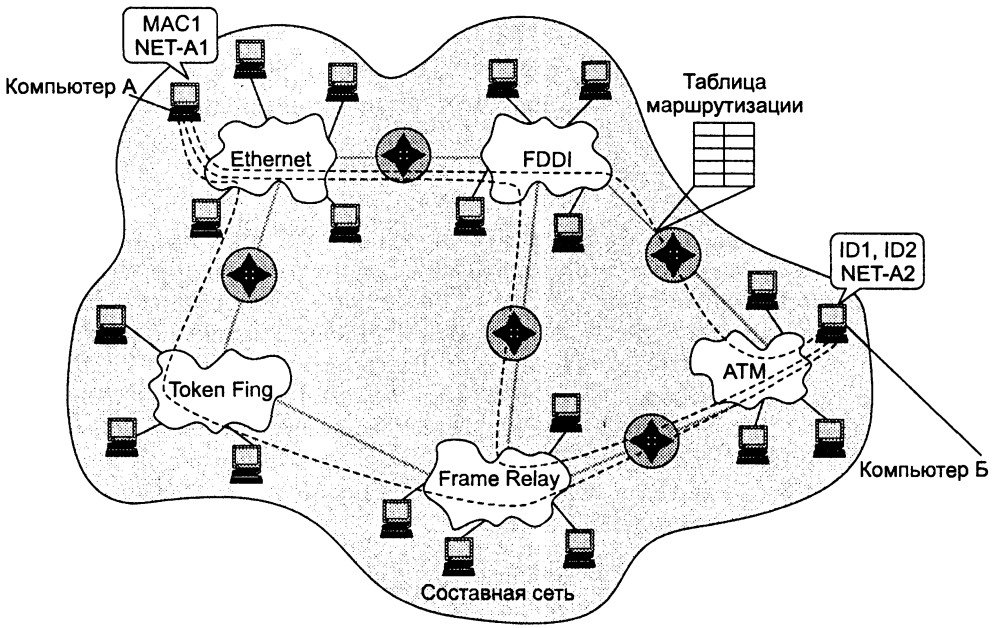


Рис. 4.9. Пример составной сети

Данные, которые необходимо передать через составную сеть, поступают на сетевой уровень от вышележащего транспортного уровня. Эти данные снабжаются заголовком сетевого уровня. Данные вместе с заголовком образуют **пакет** — так называется PDU сетевого уровня. Заголовок пакета сетевого уровня имеет унифицированный формат, не зависящий от форматов кадров канального уровня тех сетей, которые могут входить в составную сеть, и несет наряду с другой служебной информацией данные об адресе назначения этого пакета.

Для того чтобы протоколы сетевого уровня могли доставлять пакеты любому узлу составной сети, эти узлы должны иметь адреса, уникальные в пределах данной составной сети. Такие адреса называются **сетевыми**, или **глобальными**. Каждый узел составной сети, который намерен обмениваться данными с другими узлами составной сети, должен иметь сетевой адрес наряду с адресом, назначенным ему на канальном уровне. Например, на рис. 4.9 компьютер в сети Ethernet, входящей в составную сеть, имеет адрес канального уровня MAC1 и адрес сетевого уровня NET-A1; аналогично в сети ATM узел, адресуемый идентификаторами виртуальных каналов ID1 и ID2, имеет сетевой адрес NET-A2. В пакете в качестве адреса назначения должен быть указан адрес сетевого уровня, на основании которого определяется маршрут пакета. *Определение маршрута* является важ-

ной задачей сетевого уровня. Маршрут описывается последовательностью сетей (или маршрутизаторов), через которые должен пройти пакет, чтобы попасть к адресату. Например, на рис. 4.9 штриховой линией показано 3 маршрута, по которым могут быть переданы данные от компьютера А к компьютеру Б. Маршрутизатор собирает информацию о топологии связей между сетями и на ее основании строит таблицы коммутации, которые в данном случае носят специальное название **таблиц маршрутизации**. Задачу выбора маршрута мы уже коротко обсуждали в разделе «Обобщенная задача коммутации» главы 2.

В соответствии с многоуровневым подходом сетевой уровень для решения своей задачи обращается к нижележащему канальному уровню. Весь путь через составную сеть разбивается на участки от одного маршрутизатора до другого, причем каждый участок соответствует пути через отдельную сеть.

Для того чтобы передать пакет через очередную сеть, сетевой уровень помещает его в поле данных кадра соответствующей канальной технологии, указывая в заголовке кадра канальный адрес интерфейса следующего маршрутизатора. Сеть, используя свою канальную технологию, доставляет кадр с инкапсулированным в него пакетом по заданному адресу. Маршрутизатор извлекает пакет из прибывшего кадра и после необходимой обработки передает пакет для дальнейшей транспортировки в следующую сеть, предварительно упаковав его в новый кадр канального уровня в общем случае другой технологии. Таким образом, сетевой уровень играет роль координатора, организующего совместную работу сетей, построенных на основе разных технологий.

Пример-аналогия

Можно найти аналогию между функционированием сетевого уровня и международной почтовой службы, такой, например, как DHL или TNT (рис. 4.10). Представим, что некоторый груз необходимо доставить из города Абра в город Кадабра, причем эти города расположены на разных континентах. Для доставки груза международная почта использует услуги различных региональных перевозчиков:

1) железную дорогу;
2) морской транспорт;
3) авиаперевозчиков;
4) автомобильный транспорт.

Эти перевозчики могут рассматриваться как аналоги сетей канального уровня, причем каждая «сеть» здесь построена на основе собственной технологии. Из этих региональных служб международная почтовая служба должна организовать единую слаженно работающую сеть. Для этого международная почтовая служба должна, во-первых, продумать маршрут перемещения почты, во-вторых, координировать работу в пунктах смены перевозчиков (например, выгружать почту из вагонов и размещать ее в транспортном отсеке самолета). Каждый же перевозчик ответственен только за перемещение почты по своей части пути и не несет никакой ответственности за состояние почты за его пределами.

В общем случае функции сетевого уровня шире, чем обеспечение обмена в пределах составной сети. Так, сетевой уровень решает задачу создания надежных и гибких барьеров на пути нежелательного трафика между сетями.

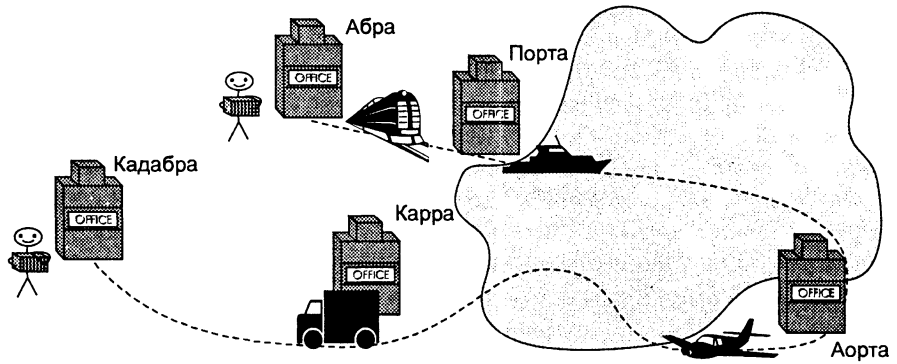


Рис. 4.10. Работа международной почтовой службы

В заключение отметим, что на сетевом уровне определяются два вида протоколов. Первый вид — **маршрутизируемые протоколы** — реализуют продвижение пакетов через сеть. Именно эти протоколы обычно имеют в виду, когда говорят о протоколах сетевого уровня. Однако часто к сетевому уровню относят и другой вид протоколов, называемых **маршрутизирующими протоколами**, или **протоколами маршрутизации**. С помощью этих протоколов маршрутизаторы собирают информацию о топологии межсетевых соединений, на основании которой осуществляется выбор маршрута продвижения пакетов.

Транспортный уровень

На пути от отправителя к получателю пакеты могут быть искажены или утеряны. Хотя некоторые приложения имеют собственные средства обработки ошибок, существуют и такие, которые предпочитают сразу иметь дело с надежным соединением.

Транспортный уровень (transport layer) обеспечивает приложениям или верхним уровням стека — прикладному, представления и сеансовому — передачу данных с той степенью надежности, которая им требуется. Модель OSI определяет пять **классов транспортного сервиса** от низшего класса 0 до высшего класса 4. Эти виды сервиса отличаются качеством предоставляемых услуг: срочностью, возможностью восстановления прерванной связи, наличием средств мультиплексирования нескольких соединений между различными прикладными протоколами через общий транспортный протокол, а главное — способностью к обнаружению и исправлению ошибок передачи, таких как искажение, потеря и дублирование пакетов.

Выбор класса сервиса транспортного уровня определяется, с одной стороны, тем, в какой степени задача обеспечения надежности решается самими приложениями и протоколами более высоких, чем транспортный, уровней. С другой стороны, этот выбор зависит от того, насколько надежной является система транспортировки данных в сети, обеспечиваемая уровнями, расположенными ниже транспортного, — сетевым, канальным и физическим. Так, если качество каналов передачи связи очень высокое и вероятность возникновения ошибок, не обнару-

женных протоколами более низких уровней, невелика, то разумно воспользоваться одним из облегченных сервисов транспортного уровня, не обремененных многочисленными проверками, квитированием и другими приемами повышения надежности. Если же транспортные средства нижних уровней очень ненадежны, то целесообразно обратиться к наиболее развитому сервису транспортного уровня, который работает, используя максимум средств для обнаружения и устранения ошибок, включая предварительное установление логического соединения, контроль доставки сообщений по контрольным суммам и циклической нумерации пакетов, установление тайм-аутов доставки и т. п.

Все протоколы, начиная с транспортного уровня и выше, реализуются программными средствами конечных узлов сети — компонентами их сетевых операционных систем. В качестве примера транспортных протоколов можно привести протоколы TCP и UDP стека TCP/IP и протокол SPX стека Novell.

Протоколы нижних четырех уровней обобщенно называют сетевым транспортом, или транспортной подсистемой, так как они полностью решают задачу транспортировки сообщений с заданным уровнем качества в составных сетях с произвольной топологией и различными технологиями. Оставшиеся три верхних уровня решают задачи предоставления прикладных сервисов, используя нижележащую транспортную подсистему.

Сеансовый уровень

Сеансовый уровень (session layer) обеспечивает управление взаимодействием сторон: фиксирует, какая из сторон является активной в настоящий момент, и предоставляет средства синхронизации сеанса. Эти средства позволяют в ходе длинных передач сохранять информацию о состоянии этих передач в виде контрольных точек, чтобы в случае отказа можно было вернуться назад к последней контрольной точке, а не начинать все с начала. На практике немногие приложения используют сеансовый уровень, и он редко реализуется в виде отдельных протоколов. Функции этого уровня часто объединяют с функциями прикладного уровня и реализуют в одном протоколе.

Уровень представления

Уровень представления (presentation layer), как явствует из его названия, обеспечивает представление передаваемой по сети информации, не меняя при этом ее содержания. За счет уровня представления информация, передаваемая прикладным уровнем одной системы, всегда понятна прикладному уровню другой системы. С помощью средств данного уровня протоколы прикладных уровней могут преодолеть синтаксические различия в представлении данных или же различия в кодах символов, например кодов ASCII и EBCDIC. На этом уровне могут выполняться шифрование и дешифрирование данных, благодаря которым секретность обмена данными обеспечивается сразу для всех прикладных служб. Примером такого протокола является протокол SSL (Secure Socket Layer — слой защищенных сокетов), который обеспечивает секретный обмен сообщениями для протоколов прикладного уровня стека TCP/IP.

Прикладной уровень

Прикладной уровень (application layer) — это в действительности просто набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые веб-страницы, а также организуют свою совместную работу, например, по протоколу электронной почты. Единица данных, которой оперирует прикладной уровень, обычно называется **сообщением**.

Существует очень большое разнообразие протоколов и соответствующих служб прикладного уровня. Приведем в качестве примера несколько наиболее распространенных реализаций сетевых файловых служб: NFS и FTP в стеке TCP/IP, SMB в Microsoft Windows, NCP в операционной системе Novell NetWare.

Модель OSI и сети с коммутацией каналов

Как уже было упомянуто, модель OSI описывает процесс взаимодействия устройств в сети с **коммутацией пакетов**. А как же обстоит дело с сетями коммутации каналов? Существует ли для них собственная справочная модель? Можно ли сопоставить функции технологий коммутации каналов с уровнями модели OSI?

Да, для представления структуры средств межсетевого взаимодействия сетей с коммутацией каналов также используется многоуровневый подход, в соответствии с которым существуют протоколы нескольких уровней, образующих иерархию. Однако общей справочной модели, подобной модели OSI, для сетей с коммутацией каналов не существует. Например, различные типы телефонных сетей используют собственные стеки протоколов, отличающиеся количеством уровней и распределением функций между уровнями. Первичные сети, такие как SDH или DWDM, также обладают собственной иерархией протоколов. Ситуация усложняется еще и тем, что практически все типы современных сетей с коммутацией каналов задействуют эту технику только для передачи пользовательских данных, а для управления процессом установления соединений в сети и общего управления сетью применяют технику коммутации пакетов. Такими сетями являются, например, сети ISDN, SDH, DWDM.

Для сетей с коммутацией пакетов сети с коммутацией каналов предоставляют сервис физического уровня, несмотря на то, что сами они устроены достаточно сложно и поддерживают собственную иерархию протоколов.

Рассмотрим, к примеру, случай, когда несколько локальных пакетных сетей связываются между собой через цифровую телефонную сеть. Очевидно, что функции создания составной сети выполняют протоколы сетевого уровня, поэтому мы устанавливаем в каждой локальной сети маршрутизатор. Маршрутизатор должен быть оснащен интерфейсом, который может установить соединение через телефонную сеть с другой локальной сетью. После того как такое соединение установлено, в телефонной сети образуется поток битов, передаваемых с постоянной скоростью. Это соединение и предоставляет сервис физического уровня для маршрутизаторов. Для того чтобы организовать передачу данных, маршрутизаторы используют поверх этого физического канала какой-либо двухточечный протокол канального уровня.

Стандартизация сетей

Список ключевых слов: открытая спецификация, открытая система, стандарты отдельных фирм, специальных комитетов и объединений, национальные стандарты, международные стандарты, совет по архитектуре Интернета, RFC, стеки TCP/IP, OSI, IPX/SPX, NetBIOS/SMB, DECnet, SNA, поток данных, сегмент, дейтаграмма, кадр, фрейм, хост, шлюз.

Универсальный тезис о пользе стандартизации, справедливый для всех отраслей, в компьютерных сетях приобретает особое значение. Суть сети — это соединение разного оборудования, а значит, проблема совместимости является здесь одной из наиболее острых. Без согласования всеми производителями общепринятых стандартов для оборудования и протоколов прогресс в деле «строительства» сетей был бы невозможен. Поэтому все развитие компьютерной отрасли, в конечном счете, отражено в стандартах — любая новая технология только тогда приобретает «законный» статус, когда ее содержание закрепляется в соответствующем стандарте.

В компьютерных сетях идеологической основой стандартизации является рассмотренная выше модель взаимодействия открытых систем (OSI).

Понятие открытой системы

Что же такое открытая система?

Открытой может быть названа любая система (компьютер, вычислительная сеть, ОС, программный пакет, другие аппаратные и программные продукты), которая построена в соответствии с открытыми спецификациями.

Напомним, что под термином «спецификация» в вычислительной технике понимают формализованное описание аппаратных или программных компонентов, способов их функционирования, взаимодействия с другими компонентами, условий эксплуатации, особых характеристик. Понятно, что не всякая спецификация является стандартом.

Под **открытыми спецификациями** понимаются опубликованные, общедоступные спецификации, соответствующие стандартам и принятые в результате достижения согласия после всестороннего обсуждения всеми заинтересованными сторонами.

Использование при разработке систем открытых спецификаций позволяет третьим сторонам разрабатывать для этих систем различные аппаратные или программные средства расширения и модификации, а также создавать программно-аппаратные комплексы из продуктов разных производителей.

Открытый характер стандартов и спецификаций важен не только для коммуникационных протоколов, но и для всех разнообразных устройств и программ, выпускаемых для построения сети. Нужно отметить, что большинство стандартов, принимаемых сегодня, носят открытый характер. Время закрытых систем, точные спецификации на которые были известны только фирме-производителю, ушло. Все осознали, что возможность взаимодействия с продуктами конкурен-

тов не снижает, а, наоборот, повышает ценность изделия, так как его можно применить в большем количестве работающих сетей, собранных из продуктов разных производителей. Поэтому даже такие фирмы, как IBM, Novell и Microsoft, ранее выпускавшие закрытые системы, сегодня активно участвуют в разработке открытых стандартов и применяют их в своих продуктах.

Для реальных систем полная открытость является недостижимым идеалом. Как правило, даже в системах, называемых открытыми, этому определению соответствуют лишь некоторые части, поддерживающие внешние интерфейсы. Например, открытость семейства операционных систем Unix заключается, помимо всего прочего, в наличии стандартизованного программного интерфейса между ядром и приложениями, что позволяет легко переносить приложения из среды одной версии Unix в среду другой версии.

Модель OSI касается только одного аспекта открытости, а именно открытости средств взаимодействия устройств, связанных в компьютерную сеть. Здесь под открытой системой понимается сетевое устройство, готовое взаимодействовать с другими сетевыми устройствами по стандартным правилам, определяющим формат, содержание и значение принимаемых и отправляемых сообщений.

Если две сети построены с соблюдением принципов открытости, это дает следующие преимущества:

- возможность построения сети из аппаратных и программных средств различных производителей, придерживающихся одного и того же стандарта;
- безболезненная замена отдельных компонентов сети другими, более совершенными, что позволяет сети развиваться с минимальными затратами;
- легкость сопряжения одной сети с другой.

Источники стандартов

Работы по стандартизации вычислительных сетей ведутся большим количеством организаций. В зависимости от статуса организаций различают следующие виды стандартов:

- *стандарты отдельных фирм* (например, стек протоколов SNA компании IBM или графический интерфейс OPEN LOOK для Unix-систем компании Sun);
- *стандарты специальных комитетов и объединений*, создаваемых несколькими компаниями, например стандарты технологии ATM, разрабатываемые специально созданным объединением ATM Forum, насчитывающем около 100 коллективных участников, или стандарты союза Fast Ethernet Alliance по разработке стандартов 100 Мбит Ethernet;
- *национальные стандарты*, например, стандарт FDDI, представляющий один из многочисленных стандартов института ANSI, или стандарты безопасности для операционных систем, разработанные центром NCSC Министерства обороны США;
- *международные стандарты*, например, модель и стек коммуникационных протоколов Международной организации по стандартизации (ISO), многочисленные стандарты Международного союза электросвязи (ITU), в том числе

стандарты на сети с коммутацией пакетов X.25, сети Frame Relay, ISDN, модемы и многие другие.

Некоторые стандарты, непрерывно развиваясь, могут переходить из одной категории в другую. В частности, фирменные стандарты на продукцию, получившую широкое распространение, обычно становятся международными стандартами де-факто, так как вынуждают производителей из разных стран следовать фирменным стандартам, чтобы обеспечить совместимость своих изделий с этими популярными продуктами. Например, из-за феноменального успеха персонального компьютера компании IBM фирменный стандарт на архитектуру IBM PC стал международным стандартом де-факто.

Более того, ввиду широкого распространения некоторые фирменные стандарты становятся основой для национальных и международных стандартов де-юре. Например, стандарт Ethernet, первоначально разработанный компаниями Digital Equipment, Intel и Xerox, через некоторое время и в несколько измененном виде был принят как национальный стандарт IEEE 802.3, а затем организация ISO утвердила его в качестве международного стандарта ISO 8802.3.

Стандартизация Интернета

Ярким примером открытой системы является Интернет. Эта международная сеть развивалась в полном соответствии с требованиями, предъявляемыми к открытым системам. В разработке ее стандартов принимали участие тысячи специалистов-пользователей этой сети из различных университетов, научных организаций и фирм-производителей вычислительной аппаратуры и программного обеспечения, работающих в разных странах. Само название стандартов, определяющих работу Интернета, — **темы для обсуждения** (Request For Comments, RFC) — показывает гласный и открытый характер принимаемых стандартов. В результате Интернет сумел объединить в себе разнообразное оборудование и программное обеспечение огромного числа сетей, разбросанных по всему миру.

Ввиду постоянной растущей популярности Интернета документы RFC становятся международными стандартами де-факто, многие из которых затем приобретают статус официальных международных стандартов в результате их утверждения какой-либо организацией по стандартизации, как правило, ISO и ITU-T.

Существует несколько организационных подразделений, отвечающих за развитие и, в частности, за стандартизацию архитектуры и протоколов Интернета. Основным из них является научно-административное **сообщество Интернета** (Internet Society, ISOC), объединяющее около 100 000 человек, которое занимается социальными, политическими и техническими проблемами эволюции Интернета.

Под управлением ISOC работает **совет по архитектуре Интернета** (Internet Architecture Board, IAB). В IAB входят две основные группы: Internet Research Task Force (IRTF) и Internet Engineering Task Force (IETF). IRTF — координирует долгосрочные исследовательские проекты по протоколам TCP/IP. Вторая группа — IETF — это инженерная группа, которая занимается решением текущих технических проблем Интернета. Именно IETF определяет спецификации, которые затем становятся стандартами Интернета. Процесс разработки и приня-

тия стандарта для протокола Интернета состоит из ряда обязательных этапов, или стадий, включающих обязательную экспериментальную проверку.

В соответствии с принципом открытости Интернета все документы RFC, в отличие, скажем, от стандартов ISO, находятся в свободном доступе. Список RFC можно найти, в частности, на сайте www.rfc-editor.org.

Стандартные стеки коммуникационных протоколов

Важнейшим направлением стандартизации в области вычислительных сетей является стандартизация коммуникационных протоколов. Наиболее известными стеками протоколов являются: OSI, TCP/IP, IPX/SPX, NetBIOS/SMB, DECnet, SNA (не все из них применяются сегодня на практике).

Стек OSI

Важно различать *модель* OSI и *стек протоколов* OSI. В то время как модель OSI является концептуальной схемой взаимодействия открытых систем, стек OSI представляет собой набор спецификаций конкретных протоколов.

В отличие от других стеков протоколов стек OSI (рис. 4.11) полностью соответствует модели OSI, включая спецификации протоколов для всех семи уровней взаимодействия, определенных в этой модели. Это и понятно, разработчики стека OSI использовали модель OSI как прямое руководство к действию.

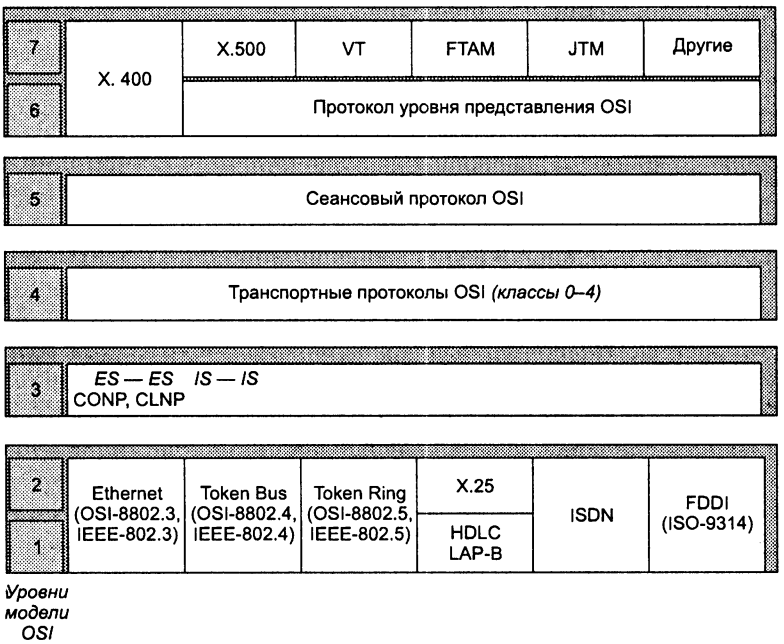


Рис. 4.11. Стек протоколов OSI

Протоколы стека OSI отличает большая сложность и неоднозначность спецификаций. Эти свойства явились результатом общей политики разработчиков стека, стремившихся учесть в своих протоколах все многообразие уже существующих и появляющихся технологий.

На *физическом* и *канальном уровнях* стек OSI поддерживает протоколы Ethernet, Token Ring, FDDI, а также протоколы LLC, X.25 и ISDN, то есть использует все разработанные вне стека популярные протоколы нижних уровней, как и большинство других стеков.

Сетевой уровень включает сравнительно редко используемые протоколы Connection-oriented Network Protocol (CONP) и Connectionless Network Protocol (CLNP). Как следует из названий, первый из них ориентирован на соединение (connection-oriented), второй — нет (connectionless).

Более популярны протоколы маршрутизации стека OSI: ES-IS (End System-Intermediate System) между конечной и промежуточной системами и IS-IS (Intermediate System-Intermediate System) между промежуточными системами.

Транспортный уровень стека OSI в соответствии с функциями, определенными для него в модели OSI, скрывает различия между сетевыми сервисами с установлением соединения и без установления соединения, так что пользователи получают требуемое качество обслуживания независимо от нижележащего сетевого уровня. Чтобы обеспечить это, транспортный уровень требует, чтобы пользователь задал нужное качество обслуживания.

Службы *прикладного уровня* обеспечивают передачу файлов, эмуляцию терминала, службу каталогов и почту. Из них наиболее популярными являются служба каталогов (стандарт X.500), электронная почта (X.400), протокол виртуального терминала (VTP), протокол передачи, доступа и управления файлами (FTAM), протокол пересылки и управления работами (JTM).

Стек IPX/SPX

Стек IPX/SPX является оригинальным стеком протоколов фирмы Novell, разработанным для сетевой операционной системы NetWare еще в начале 80-х годов. Структура стека IPX/SPX и его соответствие модели OSI иллюстрирует рис. 4.12. Протоколы *сетевого* и *транспортного уровней* — Internetwork Packet Exchange (IPX) и Sequenced Packet Exchange (SPX) — дали название стеку. К сетевому уровню этого стека отнесены также протоколы маршрутизации RIP и NLSP. А в качестве представителей трех верхних уровней на рисунке приведены два популярных протокола: протокол удаленного доступа к файлам NetWare Core Protocol (NCP) и протокол объявления о сервисах Service Advertising Protocol (SAP).

ПРИМЕЧАНИЕ

До 1996 года этот стек был бесспорным мировым лидером по числу установленных копий, но затем картина резко изменилась — стек TCP/IP по темпам роста числа установок намного стал опережать другие стеки, а с 1998 года вышел в лидеры и в абсолютном выражении.

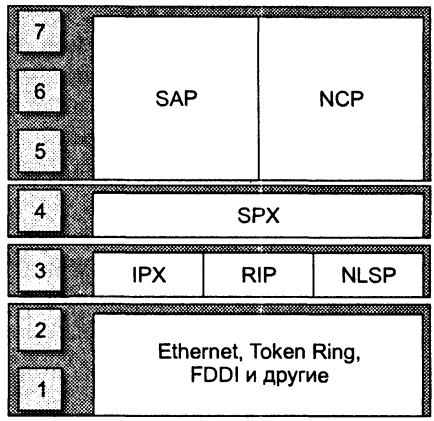


Рис. 4.12. Стек протоколов IPX/SPX

Многие особенности стека IPX/SPX обусловлены ориентацией ранних версий ОС NetWare на работу в локальных сетях небольших размеров, состоящих из персональных компьютеров со скромными ресурсами. Понятно, что для таких компьютеров компании Novell нужны были протоколы, на реализацию которых требовалось бы минимальное количество оперативной памяти (ограниченной в IBM-совместимых компьютерах под управлением MS-DOS объемом 640 Кбайт) и которые бы быстро работали на процессорах небольшой вычислительной мощности. В результате протоколы стека IPX/SPX до недавнего времени отлично справлялись с работой в локальных сетях. Однако в крупных корпоративных сетях они слишком перегружали медленные глобальные связи широковещательными пакетами, интенсивно использующимися несколькими протоколами этого стека, например протоколом SAP. Это обстоятельство, а также тот факт, что стек IPX/SPX является собственностью фирмы Novell и на его реализацию нужно получать лицензию (то есть открытые спецификации не поддерживались), долгое время ограничивали распространенность его только сетями NetWare.

Стек NetBIOS/SMB

Стек NetBIOS/SMB является совместной разработкой компаний IBM и Microsoft (рис. 4.13). На физическом и канальном уровнях этого стека также задействованы уже получившие распространение протоколы, такие как Ethernet, Token Ring, FDDI, а на верхних уровнях — специфические протоколы NetBEUI и SMB.

Протокол Network Basic Input/Output System (NetBIOS) появился в 1984 году как сетевое расширение стандартных функций базовой системы ввода-вывода (BIOS) IBM PC для сетевой программы PC Network фирмы IBM. В дальнейшем этот протокол был заменен так называемым протоколом расширенного пользовательского интерфейса NetBEUI (NetBIOS Extended User Interface). Для обеспечения совместимости приложений в качестве интерфейса к протоколу NetBEUI был сохранен интерфейс NetBIOS. Протокол NetBEUI разрабатывался как

эффективный протокол, потребляющий немного ресурсов и предназначенный для сетей, насчитывающих не более 200 рабочих станций. Этот протокол содержит много полезных сетевых функций, которые можно отнести к транспортному и сеансовому уровням модели OSI, однако с его помощью *невозможна маршрутизация* пакетов. Это ограничивает применение протокола NetBEUI локальными сетями, не разделенными на подсети, и делает невозможным его использование в составных сетях.

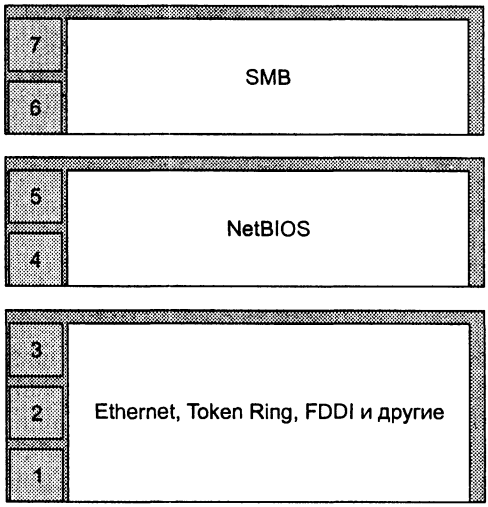


Рис. 4.13. Стек NetBIOS/SMB

Протокол Server Message Block (SMB) поддерживает функции сеансового уровня, уровня представления и прикладного уровня. На основе SMB реализуется файловая служба, а также службы печати и передачи сообщений между приложениями.

Стек TCP/IP

Стек TCP/IP был разработан по инициативе Министерства обороны США (DoD) более 20 лет назад для связи экспериментальной сети ARPANET с другими сетями как набор общих протоколов для разнородной вычислительной среды. Большой вклад в развитие стека TCP/IP, который получил свое название по популярным протоколам IP и TCP, внес университет Беркли, реализовав протоколы стека в своей версии ОС Unix. Популярность этой операционной системы привела к широкому распространению протоколов TCP, IP и других протоколов стека. Сегодня этот стек используется для связи компьютеров в Интернете, а также в огромном числе корпоративных сетей.

Поскольку стек TCP/IP изначально создавался для Интернета, он имеет много особенностей, дающих ему преимущество перед другими протоколами, когда

речь заходит о построении сетей, включающих глобальные связи. В частности, очень полезным свойством, делающим возможным применение этого протокола в больших сетях, является его *способность фрагментировать пакеты*. Действительно, большая составная сеть часто состоит из сетей, построенных на совершенно разных принципах. В каждой из этих сетей может быть собственная величина максимальной длины единицы передаваемых данных (кадра). В таком случае при переходе из одной сети, имеющей большую максимальную длину, в сеть с меньшей максимальной длиной может возникнуть необходимость деления передаваемого кадра на несколько частей. Протокол IP стека TCP/IP эффективно решает эту задачу.

Другой особенностью технологии TCP/IP является *гибкая система адресации*, позволяющая проще, чем другие протоколы аналогичного назначения включать в составную сеть сети разных технологий. Это свойство также способствует применению стека TCP/IP для построения больших гетерогенных сетей.

В стеке TCP/IP очень *экономно используются широковещательные рассылки*. Это свойство совершенно необходимо при работе на медленных каналах связи, характерных для территориальных сетей.

Однако, как и всегда, за получаемые преимущества надо платить, и платой здесь оказываются *высокие требования к ресурсам* и *сложность администрирования* IP-сетей. Мощные функциональные возможности протоколов стека TCP/IP требуют для своей реализации больших вычислительных затрат. Гибкая система адресации и отказ от широковещательных рассылок приводят к наличию в IP-сети разнообразных централизованных служб типа DNS, DHCP и т. п. Каждая из этих служб направлена на облегчение администрирования сети, но в то же время сама требует пристального внимания со стороны администраторов.

Можно приводить и другие доводы за и против стека протоколов Интернета, однако факт остается фактом — сегодня это самый популярный стек протоколов, широко используемый как в глобальных, так и локальных сетях.

На рис. 4.14 приведена структура стека TCP/IP. Так как стек TCP/IP был разработан до появления модели ISO/OSI, то, хотя он также имеет многоуровневую структуру, соответствие уровней стека TCP/IP уровням модели OSI достаточно условно.

В стеке TCP/IP определены 4 уровня.

Прикладной уровень стека TCP/IP соответствует трем верхним уровням модели OSI: прикладному, представления и сеансовому. Он объединяет службы, предоставляемые системой пользовательским приложениям. За долгие годы использования в сетях различных стран и организаций стек TCP/IP накопил большое количество протоколов и служб прикладного уровня. К ним относятся такие распространенные протоколы, как протокол передачи файлов (File Transfer Protocol, FTP), протокол эмуляции терминала (telnet), простой протокол передачи электронной почты (Simple Mail Transfer Protocol, SMTP), протокол передачи гипертекста (HyperText Transfer Protocol, HTTP) и многие другие. Протоколы прикладного уровня развертываются на хостах[1].

[1] В Интернете конечный узел традиционно называют *хостом*, а маршрутизатор — *шлюзом*. В этой главе мы будем также придерживаться такой терминологии.

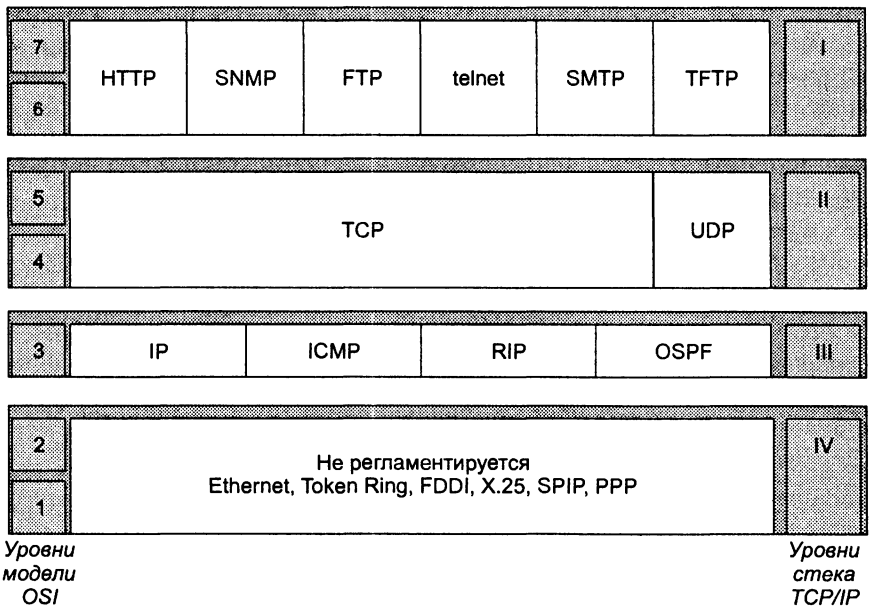


Рис. 4.14. Архитектура стека TCP/IP

Транспортный уровень стека TCP/IP может предоставлять вышележащему уровню два типа сервиса:

- гарантированную доставку обеспечивает **протокол управления передачей** (Transmission Control Protocol, TCP);
- доставку по возможности, или с максимальными усилиями, обеспечивает **протокол пользовательских дейтаграмм** (User Datagram Protocol, UDP).

Для того чтобы обеспечить надежную доставку данных протокол TCP предусматривает установление логического соединения, что позволяет ему нумеровать пакеты, подтверждать их прием квитанциями, в случае потери организовывать повторные передачи, распознавать и уничтожать дубликаты, доставлять прикладному уровню пакеты в том порядке, в котором они были отправлены. Этот протокол позволяет объектам на компьютере-отправителе и компьютере-получателе поддерживать обмен данными в дуплексном режиме. TCP дает возможность без ошибок доставить сформированный на одном из компьютеров поток байтов в любой другой компьютер, входящий в составную сеть. TCP делит поток байтов на фрагменты и передает их нижележащему уровню межсетевого взаимодействия. После того как эти фрагменты будут доставлены средствами уровня межсетевого взаимодействия в пункт назначения, протокол TCP снова соберет их в непрерывный поток байтов.

Второй протокол этого уровня — UDP — является простейшим дейтаграммным протоколом, который используется в том случае, когда задача надежного обмена данными либо вообще не ставится, либо решается средствами более высокого уровня — прикладным уровнем или пользовательскими приложениями.

В функции протоколов транспортного уровня TCP и UDP входит также исполнение роли связующего звена между прилегающими к ним прикладным уровнем и уровнем межсетевого взаимодействия. От прикладного протокола транспортный уровень принимает задание на передачу данных с тем или иным качеством, а после выполнения рапортует ему об этом. Нижележащий уровень межсетевого взаимодействия протоколы TCP и UDP рассматривают как своего рода инструмент, не очень надежный, но способный перемещать пакет в свободном и рискованном путешествии по составной сети.

Программные модули, реализующие протоколы TCP и UDP, подобно модулям протоколов прикладного уровня, устанавливаются на хостах.

Сетевой уровень, называемый также **уровнем интернета**, является стержнем всей архитектуры TCP/IP. Именно этот уровень, функции которого соответствуют сетевому уровню модели OSI, обеспечивает перемещение пакетов в пределах составной сети, образованной объединением множества сетей. Протоколы сетевого уровня поддерживают интерфейс с вышележащим транспортным уровнем, получая от него запросы на передачу данных по составной сети, а также с нижележащим уровнем сетевых интерфейсов, о функциях которого мы расскажем далее.

Основным протоколом сетевого уровня является межсетевой протокол (Internet Protocol, IP). В его задачу входит продвижение пакета между сетями — от одного маршрутизатора до другого до тех пор, пока пакет не попадет в сеть назначения. В отличие от протоколов прикладного и транспортного уровней протокол IP развертывается не только на хостах, но и на всех шлюзах. Протокол IP — это дейтаграммный протокол, работающий без установления соединений по принципу доставки с максимальными усилиями.

К сетевому уровню TCP/IP часто относят протоколы, выполняющие вспомогательные функции по отношению к IP. Это, прежде всего, протоколы маршрутизации RIP и OSPF, занимающиеся изучением топологии сети, определением маршрутов и составлением таблиц маршрутизации, на основании которых протокол IP перемещает пакеты в нужном направлении. По этой же причине к сетевому уровню могут быть отнесены еще два протокола: протокол межсетевых управляющих сообщений (Internet Control Message Protocol, ICMP), предназначенный для передачи маршрутизатором источнику информации об ошибках, возникших при передаче пакета, и протокол групповой адресации (Internet Group Management Protocol, IGMP), использующийся для направления пакета сразу по нескольким адресам.

Идеологическим отличием архитектуры стека TCP/IP от многоуровневой организации других стеков является интерпретация функций самого нижнего уровня — **уровня сетевых интерфейсов**.

Напомним, что нижние уровни модели OSI (канальный и физический) реализуют большое количество функций доступа к среде передачи, формированию кадров и согласованию уровней электрических сигналов, кодированию и синхронизации и некоторые другие. Все эти весьма конкретные функции составляют суть таких протоколов обмена данными, как Ethernet, Token Ring, PPP, HDLC и многих других.

У нижнего уровня стека TCP/IP задача существенно проще — он отвечает только за организацию взаимодействия с технологиями сетей, входящих в составную сеть. TCP/IP рассматривает любую сеть, входящую в составную сеть, как средство транспортировки пакетов до следующего на пути маршрутизатора.

Задачу обеспечения интерфейса между технологией TCP/IP и любой другой технологией промежуточной сети упрощенно можно свести:

- к определению способа упаковки (инкапсуляции) IP-пакета в единицу передаваемых данных промежуточной сети;
- к определению способа преобразования сетевых адресов в адреса технологии данной промежуточной сети.

Такой подход делает составную сеть TCP/IP открытой для включения любой сети, какую бы внутреннюю технологию передачи данных эта сеть не использовала. Для каждой новой технологии должны быть разработаны собственные интерфейсные средства. Следовательно, функции этого уровня нельзя определить раз и навсегда.

Уровень сетевых интерфейсов в стеке TCP/IP не регламентируется. Он поддерживает все популярные технологии; для локальных сетей — это Ethernet, Token Ring, FDDI, Fast Ethernet, Gigabit Ethernet, для глобальных сетей — протоколы двухточечных соединений SLIP и PPP, технологии X.25, Frame Relay, ATM.

Обычно при появлении новой технологии локальных или глобальных сетей она быстро включается в стек TCP/IP путем разработки соответствующего документа RFC, определяющего метод инкапсуляции IP-пакетов в ее кадры (например, спецификация RFC 1577, определяющая работу протокола IP через сети ATM, появилась в 1994 году вскоре после принятия основных стандартов ATM).

ПРИМЕЧАНИЕ

Стек TCP/IP позволяет включать в составную сеть сети независимо от того, каким количеством уровней описывается используемая в них технология. Так, перемещение данных в сети X.25 обеспечивают собственные протоколы физического, канального и сетевого уровней (в терминологии OSI). Тем не менее стек TCP/IP рассматривает сеть X.25 наравне с другими технологиями в качестве средства транспортировки IP-пакетов между двумя пограничными шлюзами. Уровень сетевых интерфейсов обычным образом предоставляет для этой технологии способ инкапсуляции IP-пакета в пакет X.25, а также средства преобразования сетевых IP-адресов в адреса сетевого уровня X.25. Если рассматривать такую организацию сети в строгом соответствии с моделью OSI, то налицо явное противоречие — один сетевой протокол (IP) работает поверх другого сетевого протокола (X.25). Однако для стека TCP/IP это нормальное явление.

Каждый коммуникационный протокол оперирует некоторой единицей передаваемых данных. Названия этих единиц иногда закрепляются стандартом, а чаще просто определяются традицией. В стеке TCP/IP за многие годы его существования образовалась устоявшаяся терминология в этой области (рис. 4.15).

Потоком данных, или просто **потоком**, называют данные, поступающие от приложений на вход протоколов транспортного уровня — TCP и UDP.

Протокол TCP «нарезает» из потока данных **сегменты**.

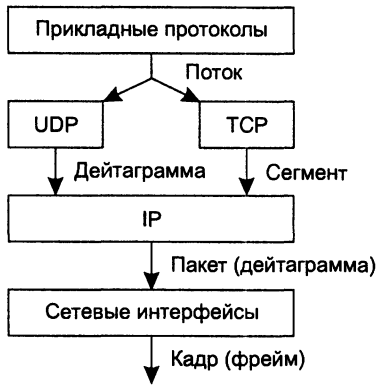


Рис. 4.15. Названия PDU в TCP/IP

Единицу данных протокола UDP часто называют **дейтаграммой**, или **датаграммой**. Дейтаграмма — это общее название для единиц данных, которыми оперируют протоколы без установления соединений. К таким протоколам относится и протокол IP, поэтому его единицу данных также называют дейтаграммой. Однако очень часто используется и другой термин — **пакет**.

В стеке TCP/IP принято называть **кадрами**, или **фреймами**, единицы данных любых технологий, в которые упаковываются IP-пакеты для последующей переноски их через сети составной сети. При этом не имеет значения, какое название используется для этой единицы данных в технологии составляющей сети. Для TCP/IP фреймом является и кадр Ethernet, и ячейка ATM, и пакет X.25, так как все они выступают в качестве контейнера, в котором IP-пакет переносится через составную сеть.

Соответствие популярных стеков протоколов модели OSI

На рис. 4.16 показано, в какой степени популярные стеки протоколов соответствуют рекомендациям модели OSI. Как мы видим, часто это соответствие весьма условно. В большинстве случаев разработчики стеков отдавали предпочтение скорости работы сети в ущерб модульности — ни один стек, кроме стека OSI, не разбит на семь уровней. Чаще всего в стеке явно выделяются 3–4 уровня: уровень сетевых адаптеров, в котором реализуются протоколы физического и канального уровней, сетевой уровень, транспортный уровень и уровень служб, вбирающий в себя функции сеансового уровня, уровня представления и прикладного уровня.

Структура стеков протоколов часто не соответствует рекомендуемому моделью OSI разбиению на уровни и по другим причинам. Давайте вспомним, чем характеризуется идеальная многоуровневая декомпозиция. С одной стороны, необходимо соблюсти принцип иерархии: каждый вышележащий уровень обращается с запросами только к нижележащему, а нижележащий предоставляет свои сервисы только непосредственно соседствующему с ним вышележащему. В стеках протоколов это приводит к тому, что PDU вышележащего уровня всегда инкапсулируется в PDU нижележащего.

<table>
<tr><th>Модель OSI</th><th colspan="3">IBM/Microsoft</th><th colspan="2">TCP/IP</th><th colspan="2">Novell</th><th>Стек OSI</th></tr>
<tr><td>Прикладной</td><td rowspan="2"></td><td rowspan="2">SMB</td><td rowspan="2"></td><td rowspan="2">Telnet,
FTP,
SNMP,
SMTP,
WWW</td><td rowspan="2"></td><td rowspan="3">NCP,
SAP</td><td rowspan="2"></td><td>X.400,
X.500,
FTAM</td></tr>
<tr><td>Представления</td><td>Протокол уровня представления OSI</td></tr>
<tr><td>Сеансовый</td><td rowspan="3"></td><td rowspan="2">NetBIOS</td><td rowspan="2"></td><td rowspan="3">TCP</td><td rowspan="3"></td><td rowspan="2"></td><td>Сеансовый протокол OSI</td></tr>
<tr><td>Транспортный</td><td></td><td>SPX</td><td></td><td>Транспортный протокол OSI</td></tr>
<tr><td>Сетевой</td><td colspan="3"></td><td>IP,
RIP,
OSPF</td><td></td><td>IPX,
RIP,
NLSP</td><td></td><td>ES-ES,
IS-IS</td></tr>
<tr><td>Канальный</td><td></td><td colspan="7">802.3 (Ethernet), 802.5 (Token Ring), FDDI, Fast Ethernet, SLIP, 100VG-AnyLAN, X.25, ATM, LAP-B, LAP-D, PPP</td></tr>
<tr><td>Физический</td><td></td><td colspan="7">Коаксиал, экранированная и неэкранированная витая пара, оптоволокно, радиоволны</td></tr>
</table>

Рис. 4.16. Соответствие популярных стеков протоколов модели OSI

С другой же стороны, идеальная многоуровневая декомпозиция предполагает, что все модули, отнесенные к одному уровню, ответственны за решение общей для всех них задачи. Однако эти требования часто вступают в противоречие. Например, основной функцией протоколов сетевого уровня стека TCP/IP (так же как и сетевого уровня OSI) является обеспечение передачи пакетов через составную сеть. Для решения этой задачи в стеке TCP/IP предусмотрено несколько протоколов: протокол продвижения IP-пакетов и протоколы маршрутизации RIP, OSPF и др. Если считать признаком принадлежности к одному и тому же уровню общность решаемых задач, то, очевидно, протокол IP и протоколы маршрутизации должны быть отнесены к одному уровню. Вместе с тем, если принять во внимание, что сообщения протокола RIP инкапсулируются в дейтаграммы UDP, а сообщения протокола OSPF — в IP-пакеты, то, следуя формально принципу иерархической организации стека, OSPF следовало бы отнести к транспортному, а RIP — к прикладному уровню. На практике же протоколы маршрутизации обычно включают в сетевой уровень.

Информационные и транспортные услуги

Список ключевых слов: транспортные услуги, информационные услуги, базы данных, файловые архивы, веб-сервис, электронная почта, инфокоммуникационная сеть, пользовательский слой, слой управления, слой менеджмента, сетевые приложения, интерфейс API, коммуникационные устройства, концентратор, коммутатор, маршрутизатор.

Услуги компьютерной сети можно разделить на две категории:

- транспортные услуги;
- информационные услуги.

Транспортные услуги состоят в передаче информации между пользователями сети в неизменном виде. При этом сеть принимает информацию от пользователя на одном из своих интерфейсов, передает ее через промежуточные коммутаторы и выдает другому пользователю через другой интерфейс. При оказании транспортных услуг сеть не вносит никаких изменений в передаваемую информацию, передавая ее получателю в том виде, в котором она поступила в сеть от отправителя. Примером транспортной услуги глобальных сетей является объединение локальных сетей клиентов.

Информационные услуги состоят в предоставлении пользователю некоторой новой информации. Информационная услуга всегда связана с операциями по обработке информации: хранению ее в некотором упорядоченном виде (файловая система, база данных), поиску нужной информации и преобразованию информации. Информационные услуги существовали и до появления первых компьютерных сетей, например, справочные услуги телефонной сети. С появлением компьютеров информационные услуги пережили революцию, так как компьютер и был изобретен для автоматической программной обработки информации. Для оказания информационных услуг применяются различные информационные технологии: программирование, базы данных, файловые архивы, веб-сервис, электронная почта.

В телекоммуникационных сетях «докомпьютерной» эры всегда преобладали транспортные услуги. Основной услугой телефонной сети всегда была передача голосового трафика между абонентами, в то время как справочные услуги были дополнительными. В компьютерных сетях одинаково важны обе категории услуг. Эта особенность компьютерных сетей сегодня отражается на названии нового поколения телекоммуникационных сетей, которые появляются в результате конвергенции сетей различных типов. Такие сети все чаще стали называть **инфокоммуникационными**. Это название пока не стало общеупотребительным, но оно хорошо отражает новые тенденции, включая обе составляющие услуг на равных правах.

Деление услуг компьютерных сетей на две категории проявляется во многих аспектах. Существует, например, четкое деление специалистов в области компьютерных сетей на специалистов информационных технологий и сетевых специалистов. К первой категории относятся программисты, разработчики баз данных, администраторы операционных систем, веб-дизайнеры, словом все те, кто имеет дело с разработкой и обслуживанием программного и аппаратного обеспечения компьютеров. Вторая категория специалистов занимается транспортными проблемами сети. Эти специалисты имеют дело с каналами связи и коммуникационным оборудованием, таким как коммутаторы, маршрутизаторы и концентраторы. Они решают проблемы выбора топологии сети, выбора маршрутов для потоков трафика, определения требуемой пропускной способности каналов связи и коммуникационных устройств и другие проблемы, связанные только с передачей трафика через сеть.

Безусловно, каждой категории специалистов необходимо знать проблемы и методы смежной области. Специалисты, занимающиеся разработкой распределенных приложений, должны представлять, какие транспортные услуги они могут получить от сети для организации взаимодействия отдельных частей своих приложений. Например, программист должен понимать, какая из двух предлагаемых стеком TCP/IP транспортных услуг, реализуемых протоколами TCP и UDP, подходит наилучшим образом его приложению. Аналогично, разработчики транспортных средств сети при передаче трафика должны стремиться максимально учитывать требования приложений.

Тем не менее специализация в области информационных технологий сохраняется, отражая двойственное назначение компьютерных сетей. Деление услуг сети на транспортные и информационные сказывается и на организации стека протоколов, а также на распределении протоколов различных уровней по элементам сети.

Распределение протоколов по элементам сети

На рис. 4.17 показаны основные элементы компьютерной сети: конечные узлы — компьютеры и промежуточные узлы — коммутаторы и маршрутизаторы (для примера выбраны протоколы стека TCP/IP, как наиболее распространенного).

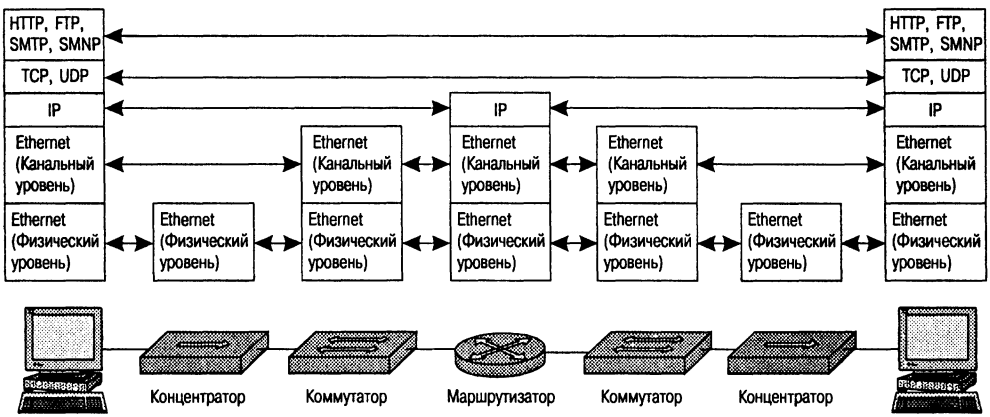


Рис. 4.17. Соответствие функций различных устройств сети уровням модели OSI

Из рисунка видно, что полный стек протоколов реализован только на конечных узлах, а промежуточные узлы поддерживают протоколы всех трех нижних уровней. Это объясняется тем, что коммуникационным устройствам для продвижения пакетов достаточно функциональности нижних трех уровней. Более того, коммуникационное устройство может поддерживать только протоколы двух нижних уровней или даже одного физического уровня — это зависит от типа устройства.

Концентратор — это устройство, которое работает с потоком битов и поэтому ограничивается поддержкой протокола физического уровня.

Коммутаторы локальных сетей поддерживают протоколы двух нижних уровней, физического и канального, что дает им возможность работать в пределах стандартных топологий.

Маршрутизаторы должны поддерживать протоколы всех трех уровней, так как сетевой уровень нужен им для объединения сетей различных технологий, а протоколы нижних уровней — для взаимодействия с конкретными сетями, образующими составную сеть, например Ethernet или Frame Relay.

Коммутаторы глобальных сетей (например, ATM), работающие на основе технологии виртуальных каналов, могут поддерживать как два уровня протоколов, так и три. Протокол сетевого уровня нужен им в том случае, если они поддерживают процедуры автоматического установления виртуальных каналов. Так как топология глобальных сетей произвольная, без сетевого протокола в этом случае обойтись нельзя. Если же виртуальные соединения устанавливаются администраторами сети вручную, то коммутатору глобальной сети достаточно поддерживать только протоколы физического и канального уровней, чтобы передавать данные по уже проложенным виртуальным каналам.

Компьютеры, на которых работают сетевые приложения, должны поддерживать протоколы всех уровней. Протоколы прикладного уровня, пользуясь сервисами протоколов уровня представления и сеансового уровня, предоставляют приложениям набор сетевых услуг в виде сетевого интерфейса API. Протокол транспортного уровня также работает на всех конечных узлах. При передаче данных через сеть два модуля транспортного протокола, работающие на узле-отправителе и узле-получателе, взаимодействуют друг с другом для поддержания транспортного сервиса нужного качества. Коммуникационные устройства сети переносят сообщения транспортного протокола прозрачным образом, не вникая в их содержание.

В компьютерах коммуникационные протоколы всех уровней (кроме физического и части функций канального уровня) реализуются программно операционной системой или системными приложениями.

Конечные узлы сети (компьютеры и компьютеризованные устройства, например, мобильные телефоны) всегда предоставляют как информационные, так и транспортные услуги, а промежуточные узлы сети — только транспортные. Когда же мы говорим, что некоторая сеть предоставляет *только транспортные услуги*, то мы подразумеваем, что конечные узлы находятся за границей сети. Это обычно имеет место в обслуживающих клиентов коммерческих сетях.

Если же говорят, что сеть предоставляет *также информационные услуги*, то это значит, что компьютеры, предоставляющие эти услуги, включаются в состав сети. Примером является типичная ситуация, когда поставщик услуг Интернета поддерживает еще и собственные веб-серверы.

Вспомогательные протоколы транспортной системы

Настало время сказать, что на рис. 4.17 показан упрощенный вариант распределения протоколов между элементами сети. В реальных сетях некоторые из

коммуникационных устройств поддерживают не только протоколы трех нижних уровней, но и протоколы верхних уровней. Так, маршрутизаторы реализуют протоколы маршрутизации, позволяющие автоматически строить таблицы маршрутизации, а концентраторы и коммутаторы часто поддерживают протоколы SNMP и telnet, которые не нужны для выполнения основных функций этих устройств, но позволяют конфигурировать и управлять ими удаленно. Все эти протоколы являются протоколами прикладного уровня и выполняют некоторые вспомогательные (служебные) функции транспортной системы. Очевидно, что для работы прикладных протоколов сетевые устройства должны также поддерживать протоколы промежуточных уровней, таких как IP и TCP/UDP.

Вспомогательные протоколы можно разделить на группы, в соответствии с их функциями.

- Первую группу вспомогательных протоколов представляют протоколы *маршрутизации*, такие как RIP, OSPF, BGP. Без этих протоколов маршрутизаторы не смогут продвигать пакеты, так как таблица маршрутизации будет пустой (если только администратор не заполнит ее вручную, но это не очень хорошее решение для крупной сети). Если рассматривать не только стек TCP/IP, но и стеки протоколов сетей с виртуальными каналами, то в эту группу попадают служебные протоколы, которые используются для установления виртуальных каналов.
- Другая группа вспомогательных протоколов выполняет *преобразование адресов*. Здесь работает протокол DNS, который преобразует символьные имена узлов в IP-адреса. Протокол DHCP позволяет назначать IP-адреса узлам динамически, а не статически, что облегчает работу администратора сети.
- Третью группу образуют протоколы, которые используются для *управления сетью*. В стеке TCP/IP здесь находится протокол SNMP (Simple Network Management Protocol — простой протокол управления сетью), который позволяет автоматически собирать информацию об ошибках и отказах устройств, а также протокол telnet, с помощью которого администратор может удаленно конфигурировать коммутатор или маршрутизатор.

При рассмотрении вспомогательных протоколов мы столкнулись с ситуацией, когда деления протоколов на уровни иерархии (то есть деление «по вертикали»), которое существует в модели OSI, оказывается недостаточно. Полезным оказывается деление протоколов на группы «по горизонтали».

И хотя такое деление отсутствует в модели OSI, оно существует в других стеках протоколов. Например, при стандартизации сетей ISDN, которые, как мы уже упоминали, используют как принцип коммутации пакетов, так и принцип коммутации каналов, все протоколы разделены на три группы (рис. 4.18):

- **пользовательский слой** (user plane) образует группа протоколов, предназначенных для того, чтобы переносить пользовательский голосовой трафик;
- **слой управления** (control plane) составляют протоколы, необходимые для установления соединений в сети;
- в **слой менеджмента** (management plane) входят протоколы, поддерживающие операции менеджмента, такие как анализ ошибок и конфигурирование устройств.

Рис. 4.18. Три группы протоколов

И хотя такое «горизонтальное» деление протоколов пока не является общепринятым для компьютерных сетей, оно полезно, так как позволяет глубже понять назначение протоколов. Кроме того, оно объясняет сложности при соотнесении некоторых протоколов уровням модели OSI. Например, в книгах одних авторов протоколы маршрутизации могут находиться на сетевом уровне, а в книгах других — на прикладном. Это происходит не из-за небрежности авторов, а из-за объективных трудностей классификации. Модель OSI хорошо подходит для стандартизации протоколов, которые переносят пользовательский трафик, то есть протоколов пользовательского слоя. В то же время она в гораздо меньшей степени подходит для стандартизации вспомогательных протоколов. Поэтому многие авторы и помещают протоколы маршрутизации на сетевой уровень, чтобы каким-то образом отразить функциональную близость этих протоколов к транспортным услугам сети, которые реализуются протоколом IP.

Выводы

Эффективной моделью средств взаимодействия компьютеров в сети является многоуровневая структура, в которой модули вышележащего уровня при решении своих задач рассматривают средства нижележащего уровня как некий инструмент. Каждый уровень данной структуры поддерживает интерфейсы двух типов. Во-первых, это интерфейсы услуг с выше- и нижележащими уровнями «своей» иерархии средств. Во-вторых, это одноранговый интерфейс со средствами другой взаимодействующей стороны, расположенными на том же уровне иерархии. Этот интерфейс называют протоколом.

Иерархически организованный набор протоколов, достаточный для взаимодействия узлов в сети, называется стеком протоколов. Протоколы нижних уровней часто реализуются комбинацией программных и аппаратных средств, а протоколы верхних уровней — как правило, программными средствами. Программный модуль, реализующий некоторый протокол, называют протокольной сущностью, или тоже протоколом.

В начале 80-х годов ISO, ITU-T при участии некоторых других международных организаций по стандартизации разработали стандартную модель взаимодействия открытых систем (OSI). Модель OSI содержит описание обобщенного представления средств сетевого взаимодействия

и используется в качестве своего рода универсального языка сетевых специалистов, именно поэтому ее называют справочной моделью. Модель OSI определяет 7 уровней взаимодействия, дает им стандартные имена, указывает, какие функции должен выполнять каждый уровень.

Открытой системой может быть названа любая система (компьютер, компьютерная сеть, операционная система, программный пакет, другие аппаратные и программные продукты), которая построена в соответствии с общедоступными спецификациями и стандартами, принятыми в результате публичного обсуждения всеми заинтересованными сторонами.

В зависимости от области действия различают стандарты отдельных компаний, стандарты специальных комитетов и объединений, национальные стандарты, международные стандарты.

Важнейшим направлением стандартизации в области вычислительных сетей является стандартизация коммуникационных протоколов. Примерами стандартизованных стеков протоколов являются TCP/IP, IPX/SPX, NetBIOS/SMB, OSI, DECnet, SNA. Лидирующее положение занимает стек TCP/IP, он используется для связи десятков миллионов компьютеров всемирной информационной сети Интернет. Стек TCP/IP имеет 4 уровня: прикладной, транспортный, межсетевого взаимодействия и сетевых интерфейсов. Соответствие уровней стека TCP/IP уровням модели OSI достаточно условно.

Вопросы и задания

1. Что стандартизирует модель OSI?
2. Можно ли представить еще один вариант модели взаимодействия открытых систем с другим количеством уровней, например 8 или 5?
3. Ниже перечислены оригинальные (англоязычные) названия семи уровней модели OSI. Отметьте, какие из названий уровней не соответствуют стандарту?
   - ❍ physical layer
   - ❍ data-link layer
   - ❍ network layer
   - ❍ transport layer
   - ❍ seances layer
   - ❍ presentation layer
   - ❍ application layer
4. Какие из приведенных утверждений вы считаете ошибочными:
   - ❍ протокол — это программный модуль, решающий задачу взаимодействия систем;
   - ❍ протокол — это формализованное описание правил взаимодействия, включающих последовательность обмена сообщениями и их форматы;
   - ❍ термины «интерфейс» и «протокол», в сущности, являются синонимами.
5. На каком уровне модели OSI работает прикладная программа?
6. Как вы считаете, протоколы транспортного уровня устанавливаются только на конечных узлах, только на промежуточном коммуникационном оборудовании (маршрутизаторах) или и там, и там?

7. На каком уровне модели OSI работают сетевые службы?
8. Ниже перечислены некоторые сетевые устройства:
   - маршрутизатор;
   - коммутатор;
   - мост;
   - повторитель;
   - сетевой адаптер;
   - концентратор.

   В каком из этих устройств реализуются функции физического уровня модели OSI? Канального уровня? Сетевого уровня?
9. Какое название традиционно используется для единицы передаваемых данных на каждом из уровней? Заполните таблицу.

	Пакет	Сообщение	Кадр	Поток	Сегмент
Канальный уровень					
Сетевой уровень					
Транспортный уровень					
Сеансовый уровень					
Уровень представления					
Прикладной уровень					

10. Дайте определение открытой системы.
11. Пусть малоизвестная небольшая компания предлагает нужный вам продукт с характеристиками, превосходящими характеристики аналогичных продуктов известных фирм. В каком из перечисленных вариантов ваши действия можно считать согласующимися с принципом открытых систем:
    - приму предложение, проверив прилагаемую документацию и убедившись, что в ней указаны характеристики, превосходящие известные аналоги;
    - приму предложение только после того, как проведу тестирование и удостоверюсь, что характеристики действительно лучше;
    - в любом случае откажусь в пользу продукта известной фирмы, так как последняя наверняка следует стандартам, а значит, будет меньше проблем с совместимостью;
    - откажусь от продукта неизвестной компании, так как есть риск ее исчезновения, а значит, могут быть проблемы с поддержкой.
12. Какая организация разработала стандарты сетей Ethernet?
13. Какое из административных подразделений Интернета непосредственно занимается стандартизацией?

14. Какие из перечисленных терминов являются синонимами:
    - стандарт;
    - спецификация;
    - RFC;
    - никакие.
15. К какому типу стандартов могут относиться современные документы RFC:
    - к стандартам отдельных фирм;
    - к государственным стандартам;
    - к национальным стандартам;
    - к международным стандартам.
16. Какая организация стояла у истоков создания и стандартизации стека TCP/IP?
17. Определите основные особенности стека TCP/IP.
18. Сравните функции самых нижних уровней моделей TCP/IP и OSI.
19. Дайте определение транспортных и информационных услуг.
20. Какие протоколы относятся к слою управления (control plane)? А к слою менеджмента (management plane)?
21. Должны ли маршрутизаторами поддерживаться протоколы транспортного уровня?
22. Пусть на двух компьютерах установлено идентичное программное и аппаратное обеспечение за исключением того, что драйверы сетевых адаптеров Ethernet поддерживают отличающиеся интерфейсы с протоколом сетевого уровня IP. Будут ли эти компьютеры нормально взаимодействовать, если их соединить в сеть?
23. Как организовать взаимодействие двух компьютеров, если у них отличаются протоколы:
    - физического и канального уровней;
    - сетевого уровня;
    - прикладного уровня.
24. Опишите ваши действия в случае, если вам необходимо проверить, на каком этапе находится процесс стандартизации технологии MPLS?
25. Выясните, в каком направлении IETF работает в настоящее время наиболее интенсивно (в качестве критерия можно использовать, например, количество рабочих групп)?

ГЛАВА 5 Примеры сетей

В этой главе рассматриваются примеры наиболее популярных типов сетей — сетей операторов связи, корпоративных сетей и Интернет. Несмотря на различия между этими типами сетей, они имеют много общего и, прежде всего, схожую архитектуру. Так, любая телекоммуникационная сеть состоит из магистрали, сетей доступа, информационных центров и оборудования клиентов. Естественно, что эта обобщенная схема наполняется собственным содержанием для каждого конкретного типа сети.

Сети телекоммуникационных операторов отличаются тем, что они предоставляют публичные услуги. Традиционными услугами операторов связи являются услуги телефонии, а также предоставления каналов связи в аренду тем организациям, которые собираются строить на их основе собственные сети. С распространением компьютерных сетей операторы связи существенно расширили спектр своих услуг, добавив доступ к Интернету, виртуальные частные сети, веб-хостинг, электронную почту и IP-телефонию, а также широковещательную рассылку аудио- и видеосигналов.

С середины 80-х годов 20-го века в мире начался процесс лишения традиционных операторов связи монополии на предоставление публичных услуг. Этот процесс привел к появлению альтернативных операторов, пытающихся привлечь клиентов расширенным набором услуг и более привлекательным соотношением цены и качества услуг. Знание административного устройства телекоммуникационного мира полезно для понимания особенностей сетевых технологий, которые в некоторых случаях специально разработаны для операторов определенного типа.

Корпоративные сети имеют иерархическую структуру, аналогичную структуре сетей операторов связи, и отличаются от них тем, что, как правило, предоставляют услуги только сотрудникам предприятия, которое владеет этой сетью.

Заканчивается глава описанием Интернета. Эта сеть, уникальная во многих отношениях, оказала критическое влияние на развитие сетевых технологий в современном мире.

Обобщенная структура телекоммуникационной сети

Список ключевых слов: канал связи, сеть доступа, терминальное оборудование, офисный телефонный коммутатор, абонентское окончание, магистральная сеть, информационный центр, или центр управления сервисами.

Несмотря на сохраняющиеся различия между компьютерными, телефонными, телевизионными, радио и первичными сетями, в их структуре можно найти много общего. В общем случае телекоммуникационная сеть состоит из следующих компонентов (рис. 5.1):

- терминального оборудования пользователей (возможно, объединенного в сеть);
- сетей доступа;
- магистральной сети;
- информационных центров, или центров управления сервисами (Services Control Point (SCP)).

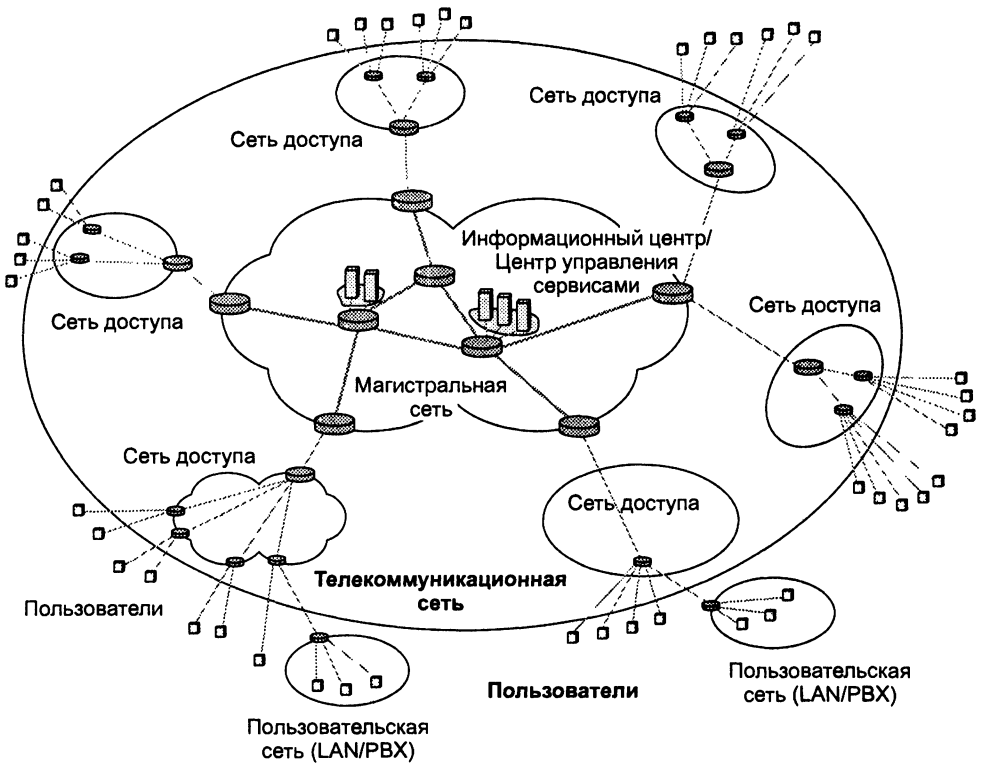


Рис. 5.1. Обобщенная структура телекоммуникационной сети

Как сеть доступа, так и магистральная сеть строятся на основе коммутаторов. Каждый коммутатор оснащен некоторым количеством портов, которые соединяются с портами других коммутаторов **каналами связи**.

Сеть доступа

Сеть доступа составляет нижний уровень иерархии телекоммуникационной сети. Основное назначение сети доступа — *концентрация информационных потоков*, поступающих по многочисленным каналам связи от оборудования клиентов, в сравнительно небольшом количестве узлов магистральной сети.

В случае компьютерной сети **терминальным оборудованием** являются компьютеры, телефонной — телефонные аппараты, телевизионной или радиосети — соответствующие теле- или радиоприемники. Терминальное оборудование пользователей может быть объединено в сети, которые не включаются в состав телекоммуникационной сети, так как принадлежат пользователям и размещаются на их территории. Компьютеры пользователей объединяются в LAN, а телефоны могут быть подключены к **офисному телефонному коммутатору** PBX (Private Branch Exchange).

Сеть доступа — это региональная сеть, отличающаяся большой разветвленностью. Как и телекоммуникационная сеть в целом, сеть доступа может состоять из нескольких уровней (на рис. 5.1 их показано два). Коммутаторы, установленные в узлах нижнего уровня, мультиплексируют информацию, поступающую по многочисленным абонентским каналам, часто называемым **абонентскими окончаниями**, и передают ее коммутаторам верхнего уровня, чтобы те в свою очередь передали ее коммутаторам магистрали.

Количество уровней сети доступа зависит от ее размера; небольшая сеть доступа может состоять из одного уровня, крупная — из двух-трех.

Магистральная сеть

Магистральная сеть объединяет отдельные сети доступа, обеспечивая транзит трафика между ними по высокоскоростным каналам.

Коммутаторы магистрали могут оперировать не только информационными соединениями между отдельными пользователями, но и агрегированными информационными потоками, переносящими данные большого количества пользовательских соединений. В результате информация с помощью магистрали попадает в сеть доступа получателей, где она демультиплексируется и коммутируется таким образом, чтобы на входной порт оборудования пользователя поступала только адресованная ему информация.

Пример

Вы можете легко заметить, что любая национальная сеть автомобильных дорог имеет ту же иерархическую структуру, что и крупная телекоммуникационная сеть. Обычно существует разветвленная инфраструктура небольших дорог, связывающих деревни и поселки. Эти дороги довольно узкие, так как интенсивность трафика между этими на-

селенными пунктами невысокая, и нет смысла делать подобные дороги многорядными. Такие дороги вливаются в более скоростные и соответственно более широкие дороги, которые, в свою очередь, имеют соединения с национальными супермагистралями. Иерархия автомобильных дорог, как и иерархия телекоммуникационных сетей, отражает интенсивность трафика между отдельными населенными пунктами и регионами страны и делает автомобильное движение более эффективным.

Информационные центры

Информационные центры, или **центры управления сервисами**, реализуют информационные услуги сети. В таких центрах может храниться информация двух типов:

- пользовательская информация, то есть информация, которая непосредственно интересует конечных пользователей сети;
- вспомогательная служебная информация, помогающая поставщику услуг предоставлять услуги пользователям.

Примером информационных ресурсов первого типа могут служить веб-порталы, на которых расположена разнообразная справочная и новостная информация, информация электронных магазинов и т. п. В телефонных сетях подобные центры оказывают услуги экстренного вызова (например, милиции, скорой помощи) и справочные услуги различных организаций и предприятий — вокзалов, аэропортов, магазинов и т. п.

К информационным центрам, хранящим ресурсы второго типа, можно отнести, например, различные системы аутентификации и авторизации пользователей, с помощью которых организация, владеющая сетью, проверяет права пользователей на получение тех или иных услуг; системы биллинга, которые в коммерческих сетях подсчитывают плату за полученные услуги; базы данных учетной информации пользователей, хранящие имена и пароли, а также перечни услуг, на которые подписан каждый пользователь. В телефонных сетях существуют централизованные центры управления сервисами (SCP), в которых установлены компьютеры, хранящие программы нестандартной обработки телефонных вызовов пользователей, например, вызовов к бесплатным справочным услугам коммерческих предприятий (так называемые услуги 800) или вызовов при проведении телеголосования.

Естественно, у сетей каждого конкретного типа имеется много особенностей, тем не менее их структура в целом соответствует описанной. В то же время, в зависимости от назначения и размера сети, в ней могут отсутствовать или же иметь несущественное значение некоторые составляющие обобщенной структуры. Например, в небольшой локальной компьютерной сети нет ярко выраженных сетей доступа и магистрали — они сливаются в общую и достаточно простую структуру. В корпоративной сети, как правило, отсутствует система биллинга, так как услуги сотрудником предприятия оказываются не на коммерческой основе. В некоторых телефонных сетях могут отсутствовать информационные центры, а в телевизионных сетях сеть доступа приобретает вид распределительной сети, так как информация в ней распространяется только в одном направлении — из сети к абонентам.

Сети операторов связи

Список ключевых слов: сеть оператора связи, корпоративная сеть, оператор связи, телефонные услуги, услуги компьютерных сетей, транспортные услуги, информационные услуги, интерактивные услуги, широковещательные услуги, массовый клиент, корпоративный клиент, виртуальная частная сеть, оператор операторов, лицензия, альтернативный местный оператор связи, уполномоченный местный оператор связи, транснациональный оператор, точка присутствия, центральный офис.

Как уже отмечалось, важным признаком классификации сетей является то, кому предназначаются услуги, предоставляемые сетью. **Сети операторов связи** (поставщиков услуг) оказывают общедоступные услуги, а **корпоративные сети** — услуги сотрудникам только того предприятия, которое владеет сетью.

Специализированное предприятие, которое создает телекоммуникационную сеть для оказания общедоступных услуг, владеет этой сетью и поддерживает ее работу, называется **оператором связи** (telecommunication carrier).

Операторы связи осуществляют свою деятельность на коммерческой основе, заключая договоры с потребителями услуг.

Операторы связи отличаются друг от друга:

- набором предоставляемых услуг;
- территорией, в пределах которой предоставляются услуги;
- типом клиентов, на которых ориентируются их услуги;
- имеющейся во владении оператора инфраструктурой — линиями связи, коммутационным оборудованием, информационными серверами и т. п.;
- отношением к монополии на предоставление услуг.

Услуги

Особенностью современных операторов связи является то, что они, как правило, оказывают услуги нескольких типов, например услуги телефонии и доступа в Интернет. Услуги можно разделить на несколько уровней и групп. На рис. 5.2 показаны только некоторые основные уровни и группы, но и эта неполная картина хорошо иллюстрирует широту спектра современных телекоммуникационных услуг и сложность их взаимосвязей. Услуги более высокого уровня опираются на услуги нижележащих уровней. Группы услуг выделены по типу сетей, которые их оказывают, — телефонные или компьютерные (для полноты картины нужно было бы дополнить рисунок услугами телевизионных и радио сетей).

Услуги предоставления каналов связи в аренду являются услугами самого нижнего уровня, так как пользователь при этом должен еще построить с помощью предоставленных каналов собственную сетевую инфраструктуру (установить телефонные коммутаторы или коммутаторы пакетных сетей), прежде чем начать извлекать из них какую-либо выгоду. Обычно такими услугами пользуются

либо другие операторы связи, не имеющие собственных каналов связи, либо крупные корпорации, которые на базе каналов строят свои частные корпоративные сети (которые рассматриваются далее).

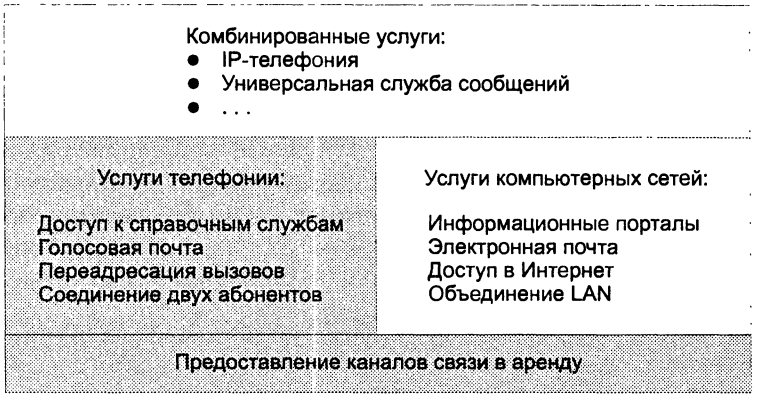


Рис. 5.2. Классификация услуг телекоммуникационной сети (закрашенные области соответствуют традиционным услугам операторов связи)

Следующий уровень составляют две большие группы услуг: телефонные услуги и услуги компьютерных сетей.

Телефонные услуги и предоставление каналов связи в аренду на протяжении очень долгого времени были традиционным набором услуг оператора связи.

Услуги компьютерных сетей стали предлагаться намного позже, чем телефонные услуги, и по абсолютному уровню доходов, приносимых операторам связи, они пока значительно отстают от традиционных телефонных услуг. Тем не менее подавляющее большинство операторов связи предоставляет услуги компьютерных сетей, и по темпам роста они намного опережают традиционные телефонные услуги, имея отличные перспективы. В объемном исчислении всемирный трафик данных уже превзошел телефонный трафик, но низкие тарифы на услуги передачи данных пока не позволяют им догнать традиционные услуги в стоимостном выражении.

Каждый из описанных уровней услуг, в свою очередь, можно разделить на подуровни. Например, оператор может предоставлять предприятию-клиенту на основе услуги доступа в Интернет, которая заключается в простом транспортном подключении компьютера или локальной сети к всемирной общедоступной сети, такие дополнительные услуги, как организацию виртуальной частной сети, надежно защищенной от остальных пользователей Интернета, или же создание информационного веб-портала предприятия и размещение его в своей сети.

Верхний уровень сегодня занимают комбинированные услуги, реализация которых требует совместного оперативного взаимодействия компьютерных и телефонных сетей. Ярким примером таких услуг является международная IP-телефония, которая отобрала у традиционной международной телефонии значительную часть клиентов.

Комбинированные услуги — это прямое следствие конвергенции сетей и главная движущая сила этого процесса.

Услуги можно разделить и по другому принципу — на **транспортные** и **информационные**. Телефонный разговор — это пример услуги первого типа, так как оператор доставляет голосовой трафик от одного абонента к другому. Примерами информационных услуг являются справочные услуги телефонной сети или веб-сайтов.

Именно этот тип различий услуг отражается на названиях телекоммуникационных компаний. Мы говорим «оператор» применительно к традиционным компаниям, основным бизнесом которых всегда были телефонные услуги и услуги предоставления каналов связи в аренду, то есть транспортные услуги. Название «поставщик услуг» стало популярно с массовым распространением Интернета и его информационной услуги WWW.

Услуги можно различать не только по виду предоставляемой информации, но и по степени их интерактивности. Так, телефонные сети оказывают **интерактивные услуги**, поскольку два абонента, участвующие в разговоре (или несколько абонентов, если это конференция), попеременно проявляют активность. Аналогичные услуги предоставляют компьютерные сети, пользователи которых могут активно участвовать в просмотре содержания веб-сайта, отвечая на вопросы анкеты или играя в игры.

С другой стороны, радиосети и телевизионные сети оказывают **широковещательные услуги**, при этом информация распространяется только в одну сторону — из сети к абонентам, по схеме «один ко многим».

Клиенты

Все множество клиентов — потребителей инфотелекоммуникационных услуг — можно разделить на два больших лагеря: массовые индивидуальные клиенты и корпоративные клиенты.

В первом случае местом потребления услуг выступает квартира или частный дом, а клиентами — жильцы, которым нужны, прежде всего, базовые услуги — телефонная связь, телевидение, радио, доступ в Интернет. Для **массовых клиентов** очень важна экономичность услуги — низкая месячная оплата, возможность использования стандартных терминальных устройств, таких как телефонные аппараты, телевизионные приемники, персональные компьютеры, а также возможность использования существующей в квартире проводки в виде телефонной пары и телевизионного коаксиального кабеля. Сложные в обращении и дорогие терминальные устройства, такие как, например, компьютеризованные телевизоры или IP-телефоны, вряд ли станут массовыми до тех пор, пока не приблизятся по стоимости к обычным телевизорам или телефонам и не будут поддерживать простой интерфейс с пользователем, не требующий для его освоения прослушивания специальных курсов. Существующая в наших домах проводка — это серьезное ограничение для предоставления услуг доступа в Интернет и новых услуг компьютерных сетей, так как она не была рассчитана на передачу данных, а подведение к каждому дому нового качественного кабеля, например волоконно-оп-

тического, — дело дорогое. Поэтому доступ в Интернет чаще всего осуществляется с помощью низкоскоростного модемного соединения по телефонной сети, но постепенно приобретают популярность новые технологии — так называемые технологии цифровых абонентских линий, позволяющие передавать по существующей телефонной проводке данные с гораздо более высокими скоростями, чем обычные модемы. Существуют также технологии доступа, которые используют для передачи данных имеющуюся в городе сеть кабельного телевидения.

Корпоративные клиенты — это предприятия и организации различного профиля. Мелкие предприятия по набору требуемых услуг не слишком отличаются от массовых клиентов — это те же базовые телефония и телевидение, стандартный модемный доступ к информационным ресурсам Интернета. Разве что телефонных номеров такому предприятию может потребоваться не один, а два или три.

Крупные же предприятия, состоящие из нескольких территориально рассредоточенных отделений и филиалов, а также имеющие сотрудников, часто работающих дома, нуждаются в расширенном наборе услуг. Прежде всего, такой услугой является **виртуальная частная сеть** (Virtual Private Network, VPN), когда оператор связи создает для предприятия иллюзию того, что все его отделения и филиалы соединены частной сетью, то есть сетью, полностью принадлежащей предприятию-клиенту и полностью управляемой предприятием-клиентом. На самом же деле для этих целей задействуется сеть оператора, то есть общедоступная сеть, которая одновременно передает данные многих клиентов.

В последнее время корпоративные пользователи все чаще получают не только транспортные, но и информационные услуги операторов, например, переносят собственные веб-сайты и базы данных на территорию оператора, поручая последнему поддерживать их работу и обеспечивать быстрый доступ к ним для сотрудников предприятия и, возможно, других пользователей сети оператора.

Инфраструктура

Помимо субъективных причин на формирование набора предлагаемых оператором услуг оказывает серьезное влияние материально-технический фактор. Так, для оказания услуг по аренде каналов оператор должен иметь в своем распоряжении транспортную сеть, например, первичную сеть PDH/SDH или же сеть с коммутацией каналов, такую как ISDN. Для оказания информационных веб-услуг требуется создать собственный сайт, который должен быть соединен с Интернетом, чтобы пользователи Интернета смогли получить к нему доступ.

В тех случаях, когда у оператора отсутствует вся необходимая инфраструктура для оказания некоторой услуги, он может воспользоваться услугами другого оператора, на базе которых, а также собственных элементов инфраструктуры, требуемая услуга может быть сконструирована. Например, для создания общедоступного веб-сайта электронной коммерции оператор связи может не иметь собственной IP-сети, соединенной с Интернетом. Для этого ему достаточно создать информационное наполнение сайта и поместить его на компьютере другого оператора, сеть которого имеет подключение к Интернету. Аренда физических каналов связи для создания собственной телефонной или компьютерной сети

является другим типичным примером предоставления услуг при отсутствии одного из элементов аппаратно-программной инфраструктуры. Оператора, который предоставляет услуги другим операторам связи, часто называют **оператором операторов** (carrier of carriers).

В большинстве стран мира операторы связи должны получать **лицензии** от государственных органов на оказание тех или иных услуг связи. Такое положение существовало не всегда — практически во всех странах были операторы, которые являлись фактическими монополистами на рынке телекоммуникационных услуг в масштабах страны. Сегодня во многих странах мира ситуация кардинально изменилась, и процесс демонополизации телекоммуникационных услуг (прежде всего, традиционных услуг, на которых, собственно, и была установлена монополия) протекает достаточно бурно. В результате монополисты теряют свои привилегии, а иногда и принудительно разукрупняются.

Демонополизация рынка телекоммуникационных услуг в США

В США компания AT&T до 1984 года была монополистом на предоставление как локальных услуг телефонии, так и услуг дальней связи. В 1984 году по решению суда AT&T была разделена на части, из которых наиболее важными были AT&T Long Lines, которой было разрешено предоставлять только услуги дальней связи, и 23 компании BOS (Bell Operating Systems), получившие право оказывать телефонные услуги только в локальных масштабах. Для предоставления услуг в масштабах регионов компании BOS были объединены в семь региональных компаний — Regional BOS (RBOS). Лишенным привилегий и разукрупненным национальным монополистам приходится бороться за клиентов с новыми операторами, которые приходят как на рынок локальных услуг, так и на региональные рынки и рынок дальней связи. Таких операторов обычно называют *альтернативными*. В США процесс конкурентного развития рынка телекоммуникационных услуг был ускорен в 1996 году, когда Конгрессом был принят документ Telecommunication Act, снимающий ограничения для оператора связи на предоставление услуг только в одном секторе рынка (либо дальней или региональной связи, либо локальных услуг). Сегодня в США работают большое количество **альтернативных местных операторов связи** (Competitive Local Exchange Carriers, CLEC) наряду с бывшими монополистами — **уполномоченными местными операторами связи** (Incumbent Local Exchange Carriers, ILEC). Не менее острая конкуренция идет и на рынке региональной и дальней связи США, где работает достаточно много крупных операторов, называемых **транснациональными операторами** (InterXchange Carriers, IXC). Эта терминология актуальна не только для читателей, живущих в США, так как она иногда используется при описании проектных решений и даже технологий, так что по типу оператора связи (IXC, CLEC или ILEC) легко понять его место в системе отношений операторов и специфику его услуг.

Территория покрытия

По степени покрытия территории, на которой предоставляются услуги, операторы делятся на локальных, региональных, национальных и транснациональных.

Локальный оператор работает на территории города или сельского района. Традиционный локальный оператор (ILEC в терминологии США) — это оператор городской телефонной сети, который владеет всей соответствующей транспортной инфраструктурой: физическими каналами между помещениями абонентов

(квартирами, домами и офисами) и узлом связи, автоматическими телефонными станциями (АТС) и каналами связи между телефонными станциями. Сегодня к традиционным локальным операторам добавились альтернативные (CLEC), которые часто являются поставщиками услуг нового типа, прежде всего, услуг Интернета, но иногда конкурируют с традиционными операторами и в секторе телефонии.

Несмотря на демонополизацию телекоммуникационной отрасли, физическими каналами доступа к абонентам по-прежнему в большинстве случаев владеют традиционные локальные операторы, такие как ILEC в США.

В таких неравноправных условиях альтернативным местным операторам достаточно трудно вести свой бизнес. У них есть несколько возможностей. Во-первых, они могут предоставлять только дополнительные услуги по передаче и обработке данных, например доступ в Интернет, размещение в своих узлах информационных ресурсов клиентов и т. п. А для организации доступа абонентов к этим ресурсам можно заключить договор с традиционным оператором, который будет направлять трафик непосредственно подключенных к нему абонентов в сеть альтернативного оператора. Здесь мы видим естественную специализацию операторов — каждый занимается тем делом, для которого в большей степени подходит его инфраструктура, при этом сотрудничество приносит дополнительный эффект, порождая новые услуги. Во-вторых, они могут брать в аренду у традиционных локальных операторов абонентские окончания. Обычно традиционные операторы идут на это очень неохотно, хотя законодательство в ряде стран принуждает или поощряет их к этому. Остается еще и третий вариант — создать собственную сеть абонентских окончаний, причем абонентские окончания могут быть проводными и беспроводными. Учитывая количество домов и квартир, сложности прокладки к ним кабеля и необходимость приобретения разрешения на эту прокладку у местных властей, проводной вариант часто оказывается нереальным. Это обстоятельство породило большой интерес к беспроводным решениям, которые сегодня бурно развиваются.

Региональные и национальные операторы оказывают услуги на большой территории, располагая соответствующей транспортной инфраструктурой. Традиционные операторы этого масштаба выполняют транзитную передачу телефонного трафика между телефонными станциями локальных операторов, имея в своем распоряжении крупные транзитные АТС, связанные высокоскоростными физическими каналами связи. Это операторы операторов, их клиентами являются, как правило, локальные операторы или крупные предприятия, имеющие отделения и филиалы в различных городах региона или страны. Располагая развитой транспортной инфраструктурой, такие операторы обычно оказывают услуги дальней связи, передавая транзитом большие объемы информации без какой-либо обработки.

Транснациональные операторы оказывают услуги в нескольких странах. Примерами таких операторов являются Cable & Wireless, Global One, Infonet. Они имеют собственные магистральные сети, покрывающие иногда несколько континентов. Часто такие операторы тесно сотрудничают с национальными операторами, используя их сети доступа для доставки информации клиентам.

Взаимоотношения между операторами связи различного типа

Взаимосвязи между операторами различного типа (а также их сетями) иллюстрирует рис. 5.3. На рисунке показаны клиенты двух типов — индивидуальные и корпоративные. Нужно иметь в виду, что каждый клиент обычно нуждается в услугах двух видов — телефонных и передачи данных. Индивидуальные клиенты имеют в своих домах или квартирах, как правило, телефон и компьютер, а у корпоративных клиентов имеются соответствующие сети — телефонная, поддерживаемая офисным телефонным коммутатором (PBX), и локальная сеть передачи данных, построенная на собственных коммутаторах.

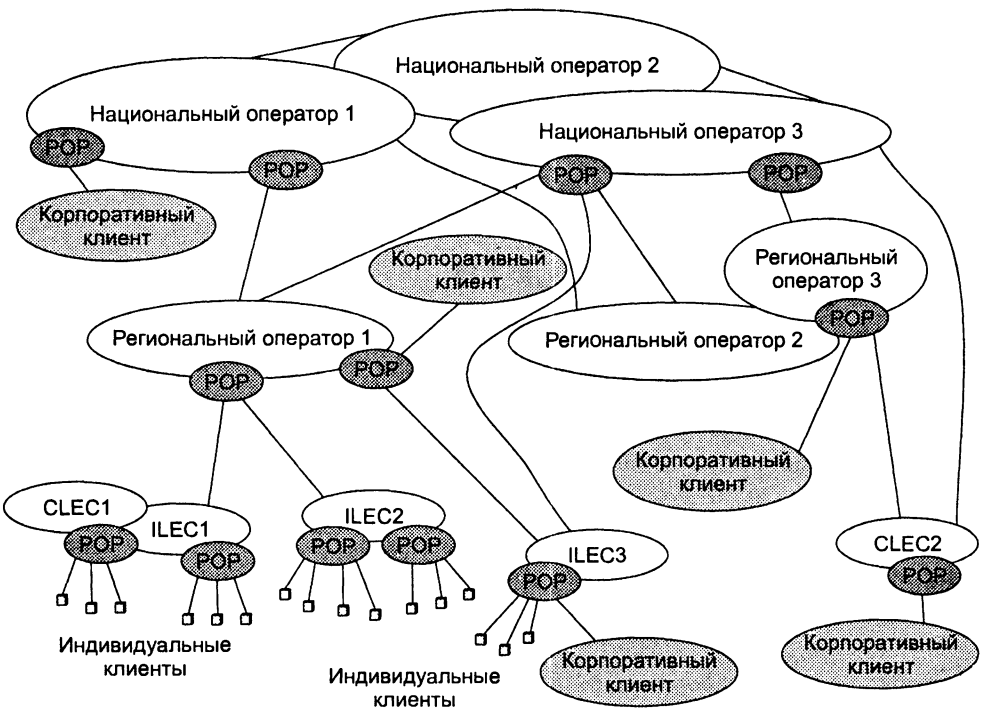


Рис. 5.3. Взаимоотношения между операторами связи различного типа

Для подключения оборудования клиентов операторы связи организуют, так называемые, **точки присутствия** (Point Of Presents, POP) — здания или помещения, в которых размещается оборудование доступа, способное подключить большое количество каналов связи, идущих от клиентов. Иногда такую точку называют **центральным офисом** (Central Office, CO) — это традиционное название для операторов телефонных сетей. К POP локальных операторов подключаются абоненты, а к POP операторов верхних уровней — операторы нижних уровней или крупные корпоративные клиенты, которым необходимы высокие скорости доступа

и большая территория покрытия, способная объединить их офисы и отделения в разных городах и странах.

Так как процесс конвергенции пока еще не привел нас к появлению единой сети для всех видов трафика, то за каждым овалом, представляющим на этом рисунке сети операторов, стоят две сети — телефонная и передачи данных.

Как видно из рисунка, в современном конкурентном телекоммуникационном мире нет строгой иерархии операторов, взаимосвязи между ними и их сетями могут быть достаточно сложными и запутанными. Например, сеть CLEC2 имеет непосредственную связь не только с сетью регионального оператора 3, как того требует иерархия, но и непосредственную связь с национальным оператором 3 (возможно, этот оператор предлагает более дешевые услуги по передаче международного трафика, чем это делает региональный оператор 3). Не все операторы на рисунке имеют собственную транспортную инфраструктуру (например, CLEC1). Как это часто бывает в таких случаях, оператор CLEC1 предоставляет только дополнительные информационные услуги, например предлагает клиентам оператора ILEC1 видео по требованию или разработку и поддержание их домашних страниц в Интернете. Свое оборудование (например, видеосервер) такой оператор часто размещает в POP другого оператора, как это и показано в данном случае.

Корпоративные сети

Список ключевых слов: корпоративная сеть, или сеть масштаба предприятия, сеть отдела, сеть рабочей группы, сеть здания, сеть кампуса, администратор сети, доступ к корпоративным ресурсам, интеграция программного и аппаратного обеспечения, глобальные связи, неоднородность, гетерогенность, масштабность.

Корпоративная сеть — это сеть, главным назначением которой является поддержание работы конкретного предприятия, владеющего данной сетью. Пользователями корпоративной сети являются только сотрудники данного предприятия. В отличие от сетей операторов связи, корпоративные сети, в общем случае, не оказывают услуг сторонним организациям или пользователям.

Хотя формально корпоративной сетью является сеть предприятия любого масштаба, обычно это название используют для сети крупного предприятия, имеющего отделения в различных городах и, возможно, разных странах. Поэтому корпоративная сеть является составной сетью, включающей как локальные, так и глобальные сети.

Структура корпоративной сети в целом соответствует обобщенной структуре телекоммуникационной сети, рассмотренной выше. Однако имеются и отличия. Например, локальные сети, объединяющие конечных пользователей, здесь включаются в состав корпоративной сети. Кроме того, названия структурных единиц корпоративной сети отражают не только территорию покрытия, но и организационную структуру предприятия. Так, принято делить корпоративную сеть на сети отделов и рабочих групп, сети зданий и кампусов, магистраль.

Сети отделов

Сети отделов — это сети, которые используются сравнительно небольшой группой сотрудников, работающих в одном отделе предприятия. Эти сотрудники решают некоторые общие задачи, например ведут бухгалтерский учет или занимаются маркетингом. Считается, что отдел может насчитывать до 100–150 сотрудников. Сеть отдела — это локальная сеть, которая охватывает все помещения, принадлежащие отделу. Это могут быть несколько комнат или этаж здания.

Главной целью сети отдела является разделение локальных ресурсов, таких как приложения, данные, лазерные принтеры и модемы. Обычно сети отделов не разделяются на подсети, имея в своем составе один или два файловых сервера и не более тридцати пользователей (рис. 5.4). В этих сетях локализуется большая часть трафика предприятия. Сети отделов обычно создаются на основе какой-либо одной сетевой технологии — Ethernet (или несколько технологий из семейства Ethernet — Ethernet, Fast Ethernet, реже Gigabit Ethernet), Token Ring или FDDI. Для такой сети характерен один или, максимум, два типа операционных систем.

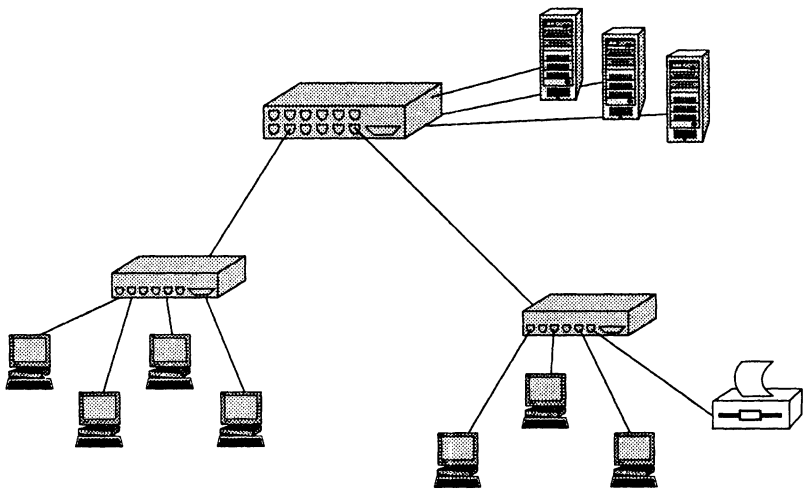


Рис. 5.4. Пример сети масштаба отдела

Задачи сетевого администрирования на уровне отдела относительно просты: добавление новых пользователей, устранение простых отказов, установка новых узлов и новых версий программного обеспечения. Такой сетью может управлять сотрудник, посвящающий обязанностям администратора только часть своего времени. Чаще всего администратор сети отдела не имеет специальной подготовки, но является тем человеком в отделе, который лучше всех разбирается в компьютерах, и само собой получается так, что он занимается администрированием сети.

Существует и другой тип сетей, близкий к сетям отделов, — **сети рабочих групп**. К таким сетям относят совсем небольшие сети, включающие до 10–20 компьютеров. Характеристики сетей рабочих групп практически не отличаются от описанных выше характеристик сетей отделов. Такие свойства, как простота сети и однородность, здесь проявляются в наибольшей степени, в то время как сети отделов могут приближаться в некоторых случаях к следующему по масштабу типу сетей — сетям зданий и кампусов.

В сетях рабочих групп еще часто используются технологии локальных сетей на разделяемых средах. По мере продвижения по иерархии вверх — к сетям отделов, зданий и кампусов, разделяемые среды встречаются все реже и реже, уступая место коммутируемым сетям.

Сеть отдела может входить в состав сети здания (кампуса) или же представлять собой сеть удаленного офиса предприятия. В первом случае сеть отдела подключается к сети здания или кампуса с помощью технологии локальной сети, которой сегодня, скорее всего, будет одна из представительниц семейства Ethernet. Во втором случае сеть удаленного офиса подключается непосредственно к магистрали сети с помощью WAN-технологии, например Frame Relay.

Сети зданий и кампусов

Сеть здания и кампуса объединяет сети различных отделов одного предприятия в пределах отдельного здания или в пределах одной территории (кампуса), покрывающей площадь в несколько квадратных километров. Для построения сетей зданий (кампусов) используются технологии локальных сетей, возможностей которых достаточно, чтобы обеспечить такую зону покрытия.

Обычно сеть здания (кампуса) строится по иерархическому принципу с собственной магистралью, построенной на базе технологии Gigabit Ethernet, к которой присоединяются сети отделов, использующие технологию Fast Ethernet или Ethernet (рис. 5.5). Магистраль Gigabit Ethernet практически всегда коммутируемая, хотя эта технология и имеет вариант на разделяемой среде.

Услуги такой сети включают взаимодействие между сетями отделов, доступ к общим базам данных предприятия, доступ к общим факс-серверам, высокоскоростным модемам и высокоскоростным принтерам. В результате сотрудники каждого отдела предприятия получают доступ к некоторым файлам и ресурсам сетей других отделов. Важной услугой, предоставляемой сетями кампусов, является доступ к корпоративным базам данных независимо от того, на каких типах компьютеров эти базы располагаются.

Именно на уровне сети кампуса возникают проблемы интеграции неоднородного аппаратного и программного обеспечения. Типы компьютеров, сетевых операционных систем, сетевого аппаратного обеспечения могут отличаться в каждом отделе. Отсюда вытекают сложности управления сетями кампусов. А поскольку сети отделов, входящие в сети кампуса, достаточно независимы и часто построены на базе различных технологий, объединяющей технологией обычно является IP.

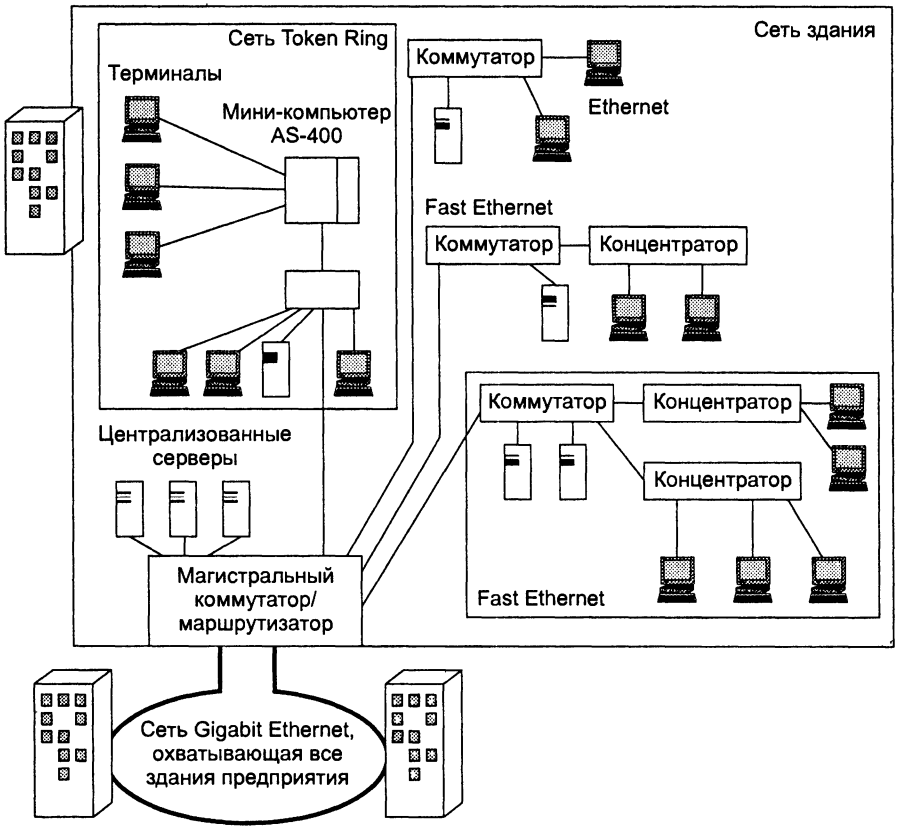


Рис. 5.5. Пример сети кампуса

Сети масштаба предприятия

Сети масштаба предприятия, или **корпоративные сети**, отличаются тем, что в них на первый план выходят *информационные услуги*. Эти сети не могут ограничиться только транспортными услугами. Если сети операторов связи могут и не предоставлять информационных услуг, так как компьютеры пользователей находятся за пределами зоны их ответственности, то корпоративные сети не могут себе этого позволить. Настольные компьютеры пользователей и серверы являются неотъемлемой частью любой корпоративной сети, и разработчики и специалисты по обслуживанию корпоративных сетей должны это учитывать. Можно сказать, что корпоративная сеть представляет собой пример инфокоммуникационной сети, где соблюдается паритет между двумя типа услуг. Корпоративную сеть можно представить в виде «островков» локальных сетей, «плавающих» в телекоммуникационной среде.

Другой особенностью корпоративной сети является ее *масштабность*. Сеть уровня отдела или здания редко называют корпоративной, хотя формально это так. Обычно название «корпоративная» применяют только для сети, включаю-

щей большое количество сетей масштаба отдела и здания, расположенных в разных городах и объединенных *глобальными связями*.

Число пользователей и компьютеров в корпоративной сети может измеряться тысячами, а число серверов — сотнями; расстояния между сетями отдельных территорий могут оказаться такими, что использование глобальных связей становится необходимым (рис. 5.6). Для соединения удаленных локальных сетей и отдельных компьютеров в корпоративной сети применяются разнообразные телекоммуникационные средства, в том числе каналы первичных сетей, радиоканалы, спутниковая связь.

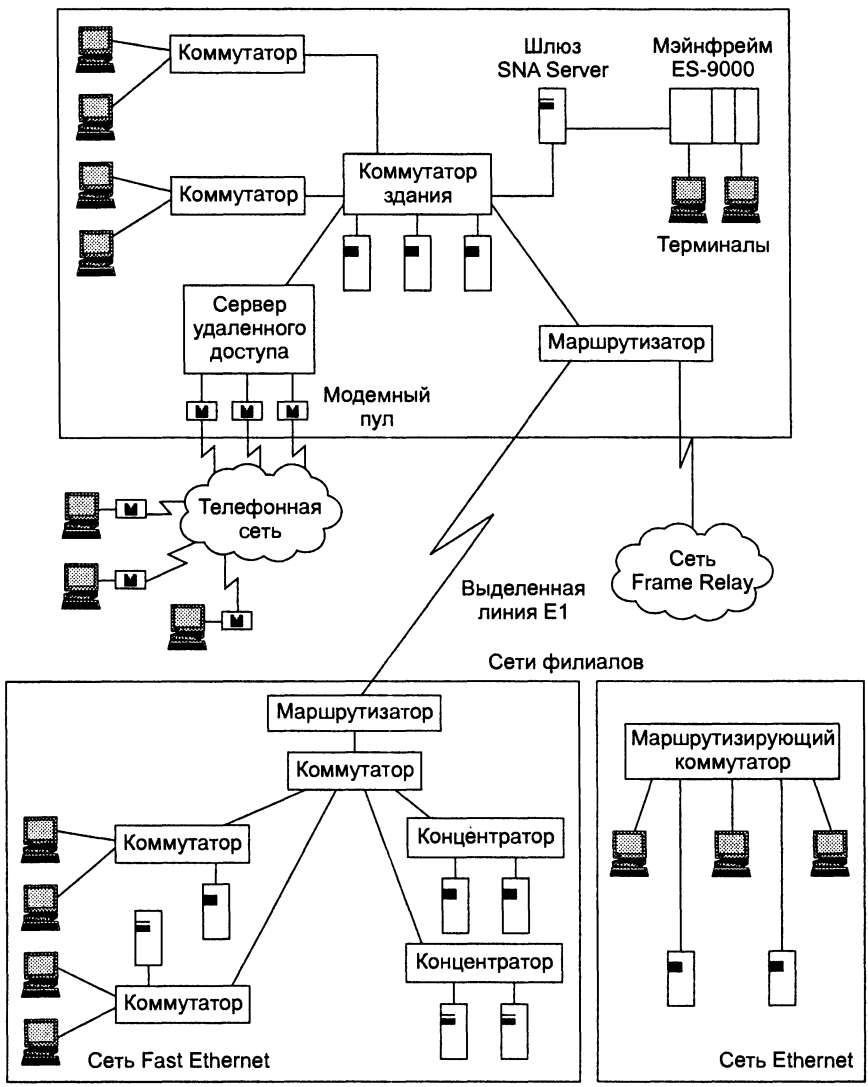


Рис. 5.6. Пример корпоративной сети

Непременным атрибутом такой сложной и крупномасштабной сети является *высокая степень неоднородности* (гетерогенности) — нельзя удовлетворить потребности тысяч пользователей с помощью однотипных программных и аппаратных средств. В корпоративной сети обязательно используются различные типы компьютеров — от мэйнфреймов до персональных компьютеров, несколько типов операционных систем и множество различных приложений. Неоднородные части корпоративной сети должны работать как единое целое, предоставляя пользователям по возможности удобный и простой доступ ко всем необходимым ресурсам.

Появление корпоративных сетей — это хорошая иллюстрация известного философского постулата *о переходе количества в качество*. При объединении в единую сеть отдельных сетей крупного предприятия, имеющего филиалы в разных городах и даже странах, многие количественные характеристики объединенной сети превосходят некоторый критический порог, за которым начинается новое качество. В этих условиях существующие методы и подходы к решению традиционных задач сетей меньших масштабов для корпоративных сетей оказались непригодными. На первый план вышли такие задачи и проблемы, которые в сетях рабочих групп, отделов и даже кампусов либо имели второстепенное значение, либо вообще не проявлялись. Примером может служить простейшая (для небольших сетей) задача — ведение учетных данных о пользователях сети.

Наиболее простой способ ее решения — помещение учетных данных всех пользователей в локальную базу учетных данных каждого компьютера, к ресурсам которого эти пользователи должны иметь доступ. При попытке доступа данные извлекаются из локальной учетной базы, и на их основе доступ предоставляется или не предоставляется. Для небольшой сети, состоящей из 5–10 компьютеров, этот подход работает очень хорошо. Но если в сети насчитывается несколько тысяч пользователей, каждому из которых нужен доступ к нескольким десяткам серверов, очевидно, что это решение становится крайне неэффективным. Администратор должен повторить несколько десятков раз (по числу серверов) операцию занесения учетных данных каждого пользователя. Сам пользователь также вынужден повторять процедуру логического входа каждый раз, когда ему нужен доступ к ресурсам нового сервера. Хорошее решение этой проблемы для крупной сети — использование централизованной справочной системы, в базе данной которой хранятся учетные записи всех пользователей сети. Администратор один раз выполняет операцию занесения данных пользователя в эту базу, а пользователь один раз выполняет процедуру логического входа, причем не в отдельный сервер, а в сеть целиком.

При переходе от более простого типа сетей к более сложному — от сетей отдела к корпоративной сети — географические расстояния увеличиваются, поддержание связи компьютеров становится все более сложным и дорогостоящим. По мере увеличения масштабов сети повышаются требования к ее надежности, производительности и функциональным возможностям. По сети циркулируют все возрастающие объемы данных, и сеть должна обеспечивать их безопасность и защищенность наряду с доступностью. Все это приводит к тому, что корпоративные сети строятся на основе наиболее мощного и разнообразного оборудования и программного обеспечения.

Интернет

Список ключевых слов: поставщик услуг Интернета, магистральный поставщик услуг, региональный поставщик услуг, центр обмена, контент, поставщик интернет-контента, поставщик информации, поставщик приложений, поставщик биллинговых услуг, брандмауэр, интрасеть.

Интернет представляет собой не только уникальную сеть, но и уникальное явление современной цивилизации. Изменения, причиной которых стал Интернет, многогранны. Гипертекстовая служба WWW изменила способ представления информации человеку, собрав на своих страницах все популярные ее виды — текст, графику и звук. Транспорт Интернета — недорогой и доступный практически всем предприятиям (а через телефонные сети и одиночным пользователям) — существенно облегчил задачу построения территориальной корпоративной сети, одновременно выдвинув на первый план задачу защиты корпоративных данных при передаче их через в высшей степени общедоступную сеть с многомиллионным «населением». Стек TCP/IP, на котором строится Интернет, стал самым популярным.

Интернет неуклонно движется к тому, чтобы стать общемировой сетью интерактивного взаимодействия людей. Он начинает все больше и больше использоваться не только для распространения информации, в том числе и рекламной, но и для осуществления самих деловых операций — покупки товаров и услуг, перемещения финансовых активов и т. п. Это в корне меняет для многих предприятий саму канву ведения бизнеса, поскольку изменяет поведение клиентов, значительная часть которых предпочитает совершать электронные сделки.

Уникальность Интернета

Уникальность Интернета проявляется во многих отношениях.

Прежде всего, это *самая большая в мире сеть*: по числу пользователей, по территории покрытия, по суммарному объему передаваемого трафика, по количеству входящих в ее состав сетей. Темпы роста Интернета, хотя и снизились по сравнению с периодом интернет-революции середины 90-х годов, остаются очень высокими и намного превышают темпы роста телефонных сетей.

Интернет — это *сеть, не имеющая единого центра управления* и в то же время работающая по единым правилам и предоставляющая всем своим пользователям единый набор услуг. Интернет — это «сеть сетей», но каждая входящая в Интернет сеть управляется независимым оператором — **поставщиком услуг Интернета** (Internet Service Provider, ISP). Некоторые центральные органы существуют, но они отвечают только за единую техническую политику, за согласованный набор технических стандартов, за централизованное назначение таких жизненно важных для гигантской составной сети параметров, как имена и адреса компьютеров и входящих в Интернет сетей, но не за ежедневное поддержание сети в работоспособном состоянии. Такая высокая степень децентрализации имеет свои достоинства и недостатки. Достоинства проявляются, например, в легкости наращивания Интернета. Так, новому ISP достаточно заключить соглашение, по

крайней мере, с одним из существующих ISP, после чего пользователи нового ISP получают доступ ко всем ресурсам Интернета. Негативные последствия децентрализации заключаются в сложности модернизации технологий и услуг Интернета. Такие коренные изменения требуют согласованных усилий всех поставщиков услуг, в случае «одного собственника» они проходили бы намного легче. Недаром многие новые технологии пока применяются только в пределах сети одного поставщика, примером может быть технология групповой рассылки, которая очень нужна для эффективной организации аудио- и видеовещания через Интернет, но все еще пока не может преодолеть границы, разделяющие сети различных ISP. Другой пример — не очень высокая надежность услуг Интернета, так как никто из поставщиков не отвечает за конечный результат, например, за доступ клиента А к сайту В, если они находятся в сетях разных поставщиков.

Интернет — *недорогая сеть*. Например, популярность новой услуги Интернета — интернет-телефонии — во многом объясняется существенно более низкими тарифами доступа в Интернет по сравнению с тарифами традиционных телефонных сетей. За низкой стоимостью стоит не временное снижение цен в надежде завоевать новый рынок, а вполне объективная причина — более низкая стоимость транспортной инфраструктуры Интернета как сети с коммутацией пакетов по сравнению с инфраструктурой телефонных сетей. Существуют, конечно, опасения, что по мере усовершенствования технологий и услуг доступ в Интернет будет обходиться все дороже и дороже. Эту опасность осознают разработчики технологий Интернета и поставщики услуг, проверяя каждое нововведение и с этой позиции.

Интернет не стал бы тем, чем он стал, если бы не еще одна его уникальная черта — *необъятное информационное наполнение и простота доступа к этой информации* для всех пользователей Интернета. Мы имеем в виду те терабайты информации, которые хранятся на серверах Интернета и доступны пользователям Интернета в форме веб-страниц. До 1991 года, когда этого сервиса не существовало, Интернет представлял собой уже всемирную и очень популярную сеть, но для сравнительно узкого круга пользователей — сотрудников и студентов университетов и исследовательских центров. Все другие клиенты — крупные корпорации, банки, правительственные организации, — которым нужно были услуги сетей передачи данных, пользовались услугами других пакетных сетей, а именно — сетей X.25.

Ничего подобного услуге WWW в сетях X.25 (и в пришедших им на смену в начале 90-х сетях Frame Relay) не было. С появлением WWW пользователи сразу поняли, что появилось нечто новое, очень удобное и нужное всем. Нельзя не отметить, что и до этого Интернет использовался в основном как информационная система, а не как транспортная. Электронная почта и ftp-архивы файлов существовали с первых лет существования Интернета. Но средства доступа к текстовой информации ftp-архивов, которые представляли собой в основном результаты исследований ученых, населявших Интернет, были очень примитивными, и поиск нужной информации по названиям файлов требовал часов и дней.

Удобная форма представления взаимосвязей между отдельными информационными фрагментами в виде гиперссылок и стандартный графический браузер, ко-

торый одинаково просто и эффективно работает во всех популярных операционных системах, совершили революцию. Интернет стал быстро заполняться самой разнообразной информацией в форме веб-страниц, превращаясь одновременно в энциклопедию, ежедневную газету, рекламное агентство и огромный магазин. Многие люди сегодня не представляют своей жизни без регулярного использования Интернета и для переписки со знакомыми, и для поиска информации (которая, как правило, нужна срочно), и для поиска работы, и для оплаты счетов.

ПРИМЕЧАНИЕ

Однако не нужно считать, что технология Интернета вытеснила или вытесняет все другие сетевые технологии. Это не так и вряд ли так случится в будущем. TCP/IP — это технология составных сетей, и она оставляет место и для других сетевых технологий, а именно тех технологий, которые работают внутри каждой сети, составляющей Интернет. Поэтому успех Интернета — не причина изучать только технологию TCP/IP. В современной сети технология TCP/IP тесно взаимодействует с множеством других технологий, таких как Ethernet, ATM, Framer Relay, MPLS, ADSL и пр.

Структура Интернета

Стремительный рост числа пользователей Интернета, привлекаемых информацией, содержащейся на его сайтах, изменил отношение корпоративных пользователей и операторов связи к этой сети. Сегодня Интернет поддерживается практически всеми традиционными операторами связи. Кроме того, к ним присоединилось большое количество новых операторов, построивших свой бизнес исключительно на услугах Интернета. Поэтому общая структура Интернета, показанная на рис. 5.7, во многом является отражением общей структуры всемирной телекоммуникационной сети, фрагмент которой мы уже рассматривали на рис. 5.3.

Магистральные поставщики услуг являются аналогами транснациональных операторов связи. Они обладают собственными транспортными магистралями, покрывающими крупные регионы (страна, континент, весь земной шар). Примерами магистральных ISP являются такие компании, как Cable & Wireless, WorldCom, Global One.

Соответственно, **региональные поставщики услуг** оказывают услуги Интернета в рамках определенного региона (штат, графство, округ — в зависимости от принятого в той или иной стране административного деления), а локальные поставщики услуг работают, как правило, в пределах одного города.

Связи между поставщиками услуг строятся на основе двухсторонних коммерческих соглашений о взаимной передаче трафика. Магистральный оператор обычно имеет подобные двусторонние (пиринговые) соглашения со всеми остальными магистральными операторами (так как их немного), а региональные операторы обычно заключают такие соглашения с одним из магистральных операторов и с несколькими другими региональными операторами. При этом операторы конфигурируют свое коммутационное оборудование так, чтобы трафик из одной сети проходил в другую, и наоборот.

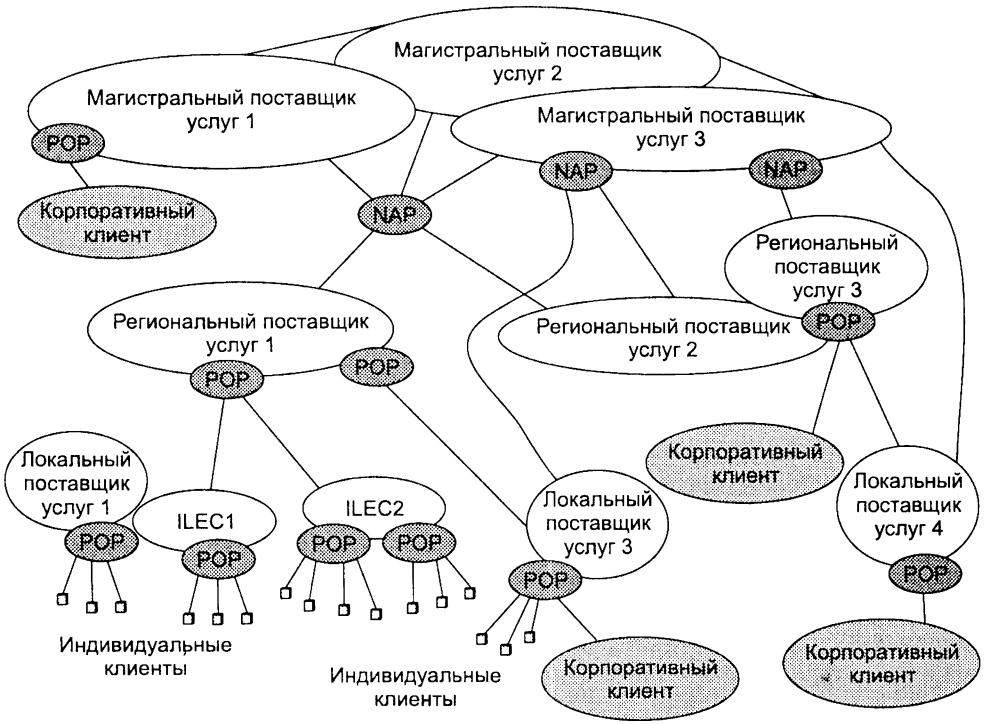


Рис. 5.7. Структура Интернета

Центры обмена NAP/IX

Для того чтобы региональным операторам было проще организовывать свои связи с другими операторами, в Интернете существуют специальные **центры обмена**, в которых соединяются сети большого количества операторов. Такой центр обмена может поддерживаться каким-либо оператором достаточно высокого уровня (национальным, международным) для операторов более низких уровней, присоединенных к его сети. Центр обмена может поддерживаться и специальной компанией, решающей только одну эту задачу. Такие центры обмена носят специальное название — чаще всего Internet eXchange (IX) или Network Access Point (NAP). В минимальном варианте такой центр просто предоставляет поставщикам услуг помещение для установки коммутационного оборудования. Все физические и логические соединения поставщики услуг выполняют самостоятельно. Более распространенным является участие коммутационного оборудования NAP/IX в обмене трафиком между операторами. При этом NAP/IX обеспечивает только физическое соединение оборудования всех поставщиков услуг, логические же связи между ISP-сетями таким подключением не создаются и трафик между ними не передается. Поэтому поставщики услуг, подключенные к NAP/IX, по-прежнему должны заключать пиринговые соглашения друг с другом. И наконец, существуют центры обмена данными, которые объединяют функции собственно обмена трафиком с коммерческими функциями. Эти центры, называемые также центрами взаиморасчетов, играют роль своеобразной биржи, на которой происходит оптовая торговля пропускной способностью. Все операторы, подключающиеся к такому центру, объявляют свою стоимость передачи данных, и центр исполняет роль посредника при заключении соглашений.

В соответствии с еще одним подходом к классификации, основанном на размере покрываемой территории, ISP делятся на четыре категории: Tier 1, Tier 2, Tier 3 и Tier 4 (www.nwfusion.com). Определения Tier 1, Tier 3 и Tier 4 для ISP совпадают с приведенными ранее определениями для магистральных, региональных и локальных операторов соответственно. Особое место занимает поставщик услуг Интернета категории Tier 2, который подобно локальному поставщику услуг работает непосредственно с конечными пользователями Интернета, но имеет существенно больший масштаб охвата клиентов. Примером ISP категории Tier 2 может служить компания America On Line (AOL). Появление ISP категории Tier 2 стало результатом договоров многочисленных локальных операторов связи, которые самостоятельно не предоставляют услуг Интернета.

На рис. 5.7 таким оператором является ILEC2. Оператор ILEC2 владеет абонентскими окончаниями, которые изначально были предназначены только для передачи телефонного трафика. Сегодня пользователи с помощью таких устройств, как модемы, могут задействовать эти физические каналы еще и для передачи трафика данных. Для абонентских окончаний в настоящее время применяются модемы двух типов: коммутируемые модемы и модемы на основе технологии асимметричной абонентской цифровой линии (Asymmetric Digital Subscriber Line, ADSL). Коммутируемый модем соединяет компьютер пользователя с ISP-сетью только временно, подобно тому, как телефон присоединяет пользователя к телефонной сети только на время разговора. Модем ADSL обеспечивает постоянное соединение компьютера с ISP-сетью.

Так как оператор ILEC2 предоставляет только телефонные услуги, он в своей точке присутствия (POP) отделяет телефонный трафик от трафика данных. Телефонный трафик оператор ILEC2 обрабатывает самостоятельно с помощью своих телефонных коммутаторов, а трафик данных направляет тому поставщику услуг, с которым у него заключено соглашение (на рисунке это региональный поставщик услуг 1). Если у поставщика услуг имеются соглашения с бóльшим количеством локальных операторов, то он становится поставщиком услуг категории Tier 2, не имея собственной инфраструктуры доступа к клиентам.

Поставщик услуг категории Tier 2 взаимодействует с другими поставщиками услуг, как правило, через поставщика услуг категории Tier 1, который передает их трафик на большие расстояния и оказывает другие полезные услуги, например обеспечивает взаиморасчеты[1].

Рассмотрим классификацию ISP по видам оказываемых услуг.

Общий термин **поставщик услуг Интернета** (Internet Service Provider, ISP) обычно относят к компаниям, которые только выполняют *транспортную* функцию для конечных пользователей — обеспечивают передачу их трафика в сети других ISP.

---

[1] Приведенная интерпретация терминов Tier 1–Tier 4 не является единственной. Например, в [4] при отнесении поставщика услуг к той или иной категории учитывается только территория предоставления услуг, таким образом, определения Tier 1–Tier 4 совпадают с определениями транснациональных, магистральных, региональных и локальных поставщиков услуг.

Поставщиком интернет-контента (Internet Content provider, ICP) называют ISP, который имеет собственные информационно-справочные ресурсы, предоставляя их содержание — **контент** (content) — в виде веб-сайтов. Многие ISP являются одновременно и ICP.

Поставщик услуг хостинга (Hosting Service Provider, HSP) — это компания, которая предоставляет свое помещение, свои каналы связи и серверы для *размещения контента*, созданного другими предприятиями.

Поставщики услуг по доставке контента (Content Delivery Provider, CDP) — это предприятия, которые не создают информационного наполнения, а занимаются *доставкой контента* в многочисленные точки доступа, максимально приближенные к пользователям, так чтобы повысить скорость доступа пользователей к информации.

Поставщики услуг по поддержке приложений (Application Service Provider, ASP) — предоставляют клиентам *доступ к крупным универсальным программным продуктам*, которые самим пользователям сложно поддерживать. Обычно это корпоративные пользователи, которых интересуют приложения класса управления предприятием, такие как SAP R3.

Так как Интернет стал уже явлением социальной жизни, то растет количество поставщиков, предоставляющих сугубо специализированные услуги, например, **поставщики биллинговых услуг** (Billing Service Provider, BSP) *обеспечивают оплату счетов по Интернету*, сотрудничая с муниципальными службами и поставщиками тепла и электроэнергии.

Границы Интернета

Возможно, чтение этого раздела вызвало у вас вопрос: А можно ли точно очертить границы Интернета, если он так тесно встроен в общую инфраструктуру операторов связи?

Чтобы ответить на этот вопрос, давайте более подробно рассмотрим типичные сети поставщика услуг и его клиентов, представленные на рис. 5.8.

Поставщик услуг имеет двух корпоративных и большое число индивидуальных клиентов. Он имеет также несколько соединений с другими поставщиками, через которых получает доступ ко всем остальным поставщикам услуг, входящим в Интернет. В ISP-сети находятся несколько серверов, на которых размещаются веб-сайты, доступные всем пользователям Интернета.

Однако у клиента CC1 есть и такие информационные ресурсы, которые содержат внутреннюю информацию корпорации и должны быть доступны только сотрудникам корпорации. Поэтому клиент CC1 установил в своей сети особое коммуникационное устройство — **брандмауэр**. Брандмауэр защищает ту часть сети, в которой находятся внутренние серверы, поэтому доступ к ним могут получить только клиенты, находящиеся внутри этой сети, а запросы всех внешних клиентов к этим серверам блокируются. В то же время брандмауэр не препятствует доступу внутренних клиентов к внешним информационным ресурсам, то есть к ресурсам Интернета, и пропускает ответы внешних серверов с необходимой сотрудникам предприятия информацией (если бы сотрудникам доступ к ресур-

сам Интернета был не нужен, то проще было бы не использовать брандмауэр, а физически отсоединить внутреннюю сеть от ISP-сети).

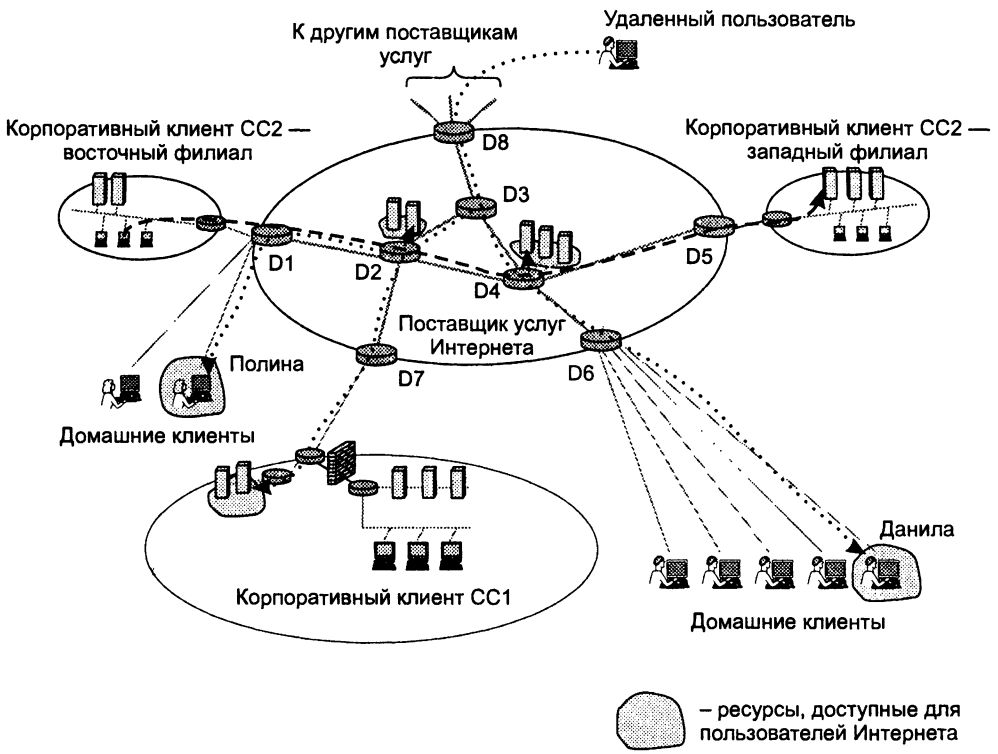


Рис. 5.8. Границы Интернета

Корпоративный клиент CC1 обладает собственной сетью, которая включает несколько серверов с размещенных на них информацией, а также большое количество клиентских компьютеров для сотрудников предприятия. Корпоративный клиент CC1 имеет постоянное высокоскоростное соединение с поставщиком услуг и пользуется базовой услугой доступа в Интернет, которая обеспечивает обмен данными его корпоративной сети с ISP-сетью и через нее транзитом — с сетями других поставщиков услуг. В интересах своего бизнеса клиент CC1 поддерживает несколько собственных веб-сайтов, доступных всем пользователям Интернета.

Основная часть домашних клиентов поставщика услуг подключаются к нему с помощью коммутируемого модема, а два их них — Данила и Полина — имеют постоянные соединения с ISP-сетью через ADSL-модемы. Данила и Полина развернули на своих домашних компьютерах собственные веб-сайты, так что каждый пользователь Интернета может подключаться к ним и пользоваться их информацией.

Корпоративный клиент CC2 имеет два филиала, расположенных в разных городах, — Западный и Восточный. В каждом из филиалов есть собственная локальная сеть. Эти сети работают на технологии TCP/IP — той же, что и Интернет.

Клиент CC2 пользуется услугами ISP не для доступа в Интернет, а для того, чтобы объединить сети своих подразделений в единую корпоративную сеть. Доступ в Интернет ему не нужен, более того, ему необходимо обеспечить высокую степень конфиденциальности своих передаваемых данных. Он не защищает свои ресурсы самостоятельно, как клиент CC1, а пользуется услугой виртуальной частной сети, которая гарантирует ему, что сети двух его филиалов будут полностью изолированы от трафика Интернета. Поэтому все информационные ресурсы его Западного и Восточного филиалов — веб-сайты, базы данных и т. п. — доступны только пользователям клиента CC1. Сотрудники клиента CC2 лишены возможности пользоваться услугами Интернета, так как их запросы в Интернет не передаются.

Давайте теперь попробуем ответить на вопрос, является ISP-сеть частью Интернета? С одной стороны, да, так как она передает транзитом трафик клиентов Интернета и, кроме того, содержит информационные ресурсы общего пользования. Действительно, показанный на рисунке удаленный пользователь Интернета может сделать запрос к любому веб-сайту нашего поставщика услуг.

В то же время часть транспортной инфраструктуры ISP к Интернету не относится. Это, например, коммуникационное устройство D5, которое используется исключительно для обслуживания клиента CC2, и канал связи между D5 и D4. В ISP-сети есть также устройства и каналы, которые обсуживают клиентов Интернета только частично — это устройства D1, D2, D4 и соединяющие их каналы.

Поэтому однозначное «Да» не точно отражает ситуацию — границы ISP-сети не являются границами Интернета, хотя мы и говорим, что Интернет состоит из ISP-сетей.

Аналогичная ситуация складывается и с сетью клиента CC1. С одной стороны, можно просто сказать, что его сеть не является частью Интернета, потому что он не поставщик услуг, а клиент. Кроме того, в сети CC1 есть ресурсы, защищенные брандмауэром и, очевидно, частью Интернета не являющиеся. С другой стороны, в сети CC1 есть веб-сайты, которые доступны всем пользователям Интернета, в том числе нашему удаленному пользователю. Для удаленного клиента эти веб-сайты ничем не отличаются от веб-сайтов, находящихся в ISP-сети.

Домашние клиенты, имеющие коммутируемое подключение к Интернету, как правило, не размещают свои веб-сайты на домашних компьютерах, так как большую часть времени они все равно останутся недоступными остальным пользователям Интернета. Однако домашние пользователи, имеющие постоянное подключение к Интернету через ADSL-модем, могут это делать. Данила и Полина поддерживают свои веб-сайты, так что удаленный пользователь при желании может воспользоваться размещенной на них информацией.

Таким образом, с точки зрения пользователей Интернет представляет собой набор информационных ресурсов, рассредоточенных по различным сетям — ISP-сетям, корпоративным сетям, а также сетям и компьютерам домашних пользователей.

Транспортные средства Интернета также достаточно «виртуальны» — можно считать, что они представляют собой часть ресурсов коммуникационных устройств и каналов операторов связи, которые обеспечивают передачу трафика Интернета,

то есть трафика между клиентами Интернета и его информационными ресурсами (или между двумя клиентами Интернета, если это трафик электронной почты или IP-телефонии).

В общем случае сеть отдельного поставщика услуг обычно называют частной IP-сетью, так как с помощью этой сети оператор оказывает как услуги Интернета, так и другие услуги, не относящиеся непосредственно к Интернету, например, услуги VPN. Если эти предоставляемые сервисы реализуются с помощью тех же технологий, на которых построен Интернет (то есть транспорта TCP/IP и информационного сервиса WWW), то такие услуги часто называют услугами **интрасети** (intranet).

Выводы

Компьютерные сети предоставляют услуги двух типов: информационные и транспортные. Часто под термином «сетевые услуги» понимают транспортные услуги, считая, что основной функцией сети является передача информации. Информационные услуги предоставляются конечными узлами сети — серверами, а транспортные — промежуточными узлами, которыми являются коммутаторы и маршрутизаторы сети.

Компьютерную сеть можно описать с помощью обобщенной структуры, которая справедлива для любой телекоммуникационной сети. Такая обобщенная структура состоит из сетей доступа, магистрали и информационных центров.

Специализированное предприятие, которое создает телекоммуникационную сеть для оказания общедоступных услуг, владеет этой сетью и поддерживает ее работу, называется оператором связи.

Операторы связи отличаются друг от друга набором предоставляемых услуг, территорией, в пределах которой предоставляются услуги, типом клиентов, на которых ориентируются их услуги, а также имеющейся во владении оператора инфраструктурой — линиями связи, коммутационным оборудованием, информационными серверами и т. п. Операторов связи, специализирующихся на предоставлении услуг компьютерных сетей, обычно называют поставщиками услуг.

Корпоративная сеть — это сеть, главным назначением которой является поддержание работы конкретного предприятия, владеющего сетью. Пользователями корпоративной сети являются только сотрудники данного предприятия.

Интернет является уникальной компьютерной сетью, предоставляющей разнообразные услуги во всемирном масштабе. Интернет использует стек TCP/IP для объединения сетей различных технологий в единую составную сеть. Популярность информационных услуг Интернета (электронная почта, WWW, чат) привела к тому, что транспортные протоколы стека TCP/IP стали основными протоколами любых составных сетей.

Вопросы и задания

1. Какой термин соответствует приведенному ниже определению сети?

   Сеть, предназначенная для концентрации информационных потоков, поступающих по многочисленным каналам связи от оборудования пользователей.

2. Приведите примеры информационных центров различных типов телекоммуникационных сетей.
3. Перечислите основные требования, которым должны удовлетворять сети доступа и магистральные сети.
4. Перечислите типы клиентов операторов связи.
5. Можно ли назвать сеть оператора связи корпоративной сетью?
6. Назовите основные характеристики сетей операторов связи.
7. Является ли аренда каналов связи традиционной или новой услугой для телекоммуникационных операторов?
8. Какие дополнительные услуги может предложить начинающий альтернативный оператор для привлечения клиентов?
9. Любые ли коммуникационные устройства, работающие в корпоративной сети, называют корпоративными?
10. В чем отличие ISP Tier 1 и Tier 2?
11. В чем заключается услуга по предоставлению доступа в Интернет?
12. Можно ли предоставлять услуги Интернета, не обладая собственными каналами связи?
13. Заполните представленную ниже таблицу, установив соответствие между описаниями сетей и их типами (один тип сети не описан).

Описание сети	**Корпоративная сеть**	**Сеть кампуса**	**Сеть отдела**	**Сеть оператора**
Сеть используется группой сотрудников до 100–150 человек				
Все сотрудники сети связаны с решением частной бизнес-задачи				
Сеть создана на основе какой-либо одной сетевой технологии				
Сеть включает тысячи пользовательских компьютеров, сотни серверов				
Сеть обладает высокой степенью гетерогенности компьютеров, коммуникационного оборудования, операционных систем и приложений				

Описание сети	Корпоративная сеть	Сеть кампуса	Сеть отдела	Сеть оператора
Сеть использует глобальные связи				
Сеть объединяет более мелкие сети в пределах отдельного здания или одной территории				
Глобальные соединения в сети не используются				
Службы сети предоставляют всем сотрудникам доступ к общим базам данных предприятия				

14. В сетях какого типа, корпоративных или ISP-сетях, доля локальных сетей больше?
15. На какие уровни иерархии можно разделить корпоративную сеть?
16. Назовите варианты специализации поставщиков услуг.
17. Входит ли корпоративная сеть в состав Интернета, если она имеет постоянное подключение к ISP-сети?
18. Каким способом альтернативный локальный оператор связи может обеспечить индивидуальный доступ клиентов к ресурсам своей сети?
19. Какие проблемы потребовалось решить при демонополизации телекоммуникационной отрасли?
20. Как вы могли бы интерпретировать понятие «виртуальная частная телефонная сеть»?
21. Опишите последовательность необходимых с вашей точки зрения действий руководства предприятия для того, чтобы это предприятие могло стать поставщиком услуг Интернета и начать предоставлять услуги клиентам.

ГЛАВА 6 Сетевые характеристики

Компьютерная сеть представляет собой сложную и дорогую систему, решающую ответственные задачи и обслуживающую большое количество пользователей. Поэтому очень важно, чтобы сеть не просто работала, но работала качественно.

Понятие «качество обслуживания» можно трактовать очень широко, включая в него все возможные и желательные для пользователя свойства сети и поставщика услуг, поддерживающего работу этой сети. Для того чтобы пользователь и поставщик услуг могли более конкретно обсуждать проблемы обслуживания и строить свои отношения на формальной основе, существует ряд общепринятых характеристик качества предоставляемых сетью услуг. Мы будем рассматривать в этой главе только характеристики качества транспортных услуг сети, которые намного проще поддаются формализации, чем характеристики качества информационных услуг. Характеристики качества транспортных услуг отражают такие важнейшие свойства сети, как производительность, надежность и безопасность.

Часть этих характеристик может быть оценена количественно и измерена при обслуживании пользователя. Пользователь и поставщик услуг могут заключить соглашение об уровне обслуживания, в котором оговорить требования к количественным значениям некоторых характеристик, например к доступности предоставляемых услуг.

Термин «качество обслуживания» часто употребляется в узком смысле, как одно из современных направлений в сетевых технологиях, цель которого состоит в разработке методов качественной передачи трафика через сеть. Характеристики качества обслуживания объединяет то, что все они отражают отрицательное влияние механизма очередей на передачу трафика, такие как временное снижение скорости передачи трафика, доставку пакетов с переменными задержками и потери пакетов из-за перегрузки буферов коммутаторов.

Типы характеристик

Список ключевых слов: надежность, производительность и безопасность транспортных услуг, планирование использования ресурсов и контроль над их использованием, масштабируемость сети, долговременные характеристики сети, среднесрочные характеристики сети, краткосрочные характеристики сети, методы контроля и предотвращения перегрузок, соглашение об уровне обслуживания.

Субъективные оценки качества

Если опросить пользователей, чтобы выяснить, что они вкладывают в понятие качественных сетевых услуг, то можно получить очень широкий спектр ответов. Среди них, скорее всего, встретятся следующие мнения:

- сеть работает быстро, без задержек;
- трафик передается надежно;
- услуги предоставляются бесперебойно по схеме 24×7 (то есть 24 часа в сутки семь дней в неделю);
- служба поддержки работает хорошо, давая полезные советы и помогая разрешить проблемы;
- услуги предоставляются по гибкой схеме, мне нравится, что можно в любой момент и в широких пределах повысить скорость доступа к сети и увеличить число точек доступа;
- поставщик не только передает мой трафик, но и защищает мою сеть от вирусов и атак злоумышленников;
- я всегда могу проконтролировать, насколько быстро и без потерь сеть передает мой трафик;
- поставщик предоставляет широкий спектр услуг, в частности помимо стандартного доступа в Интернет он предлагает хостинг для моего персонального веб-сайта и услуги IP-телефонии.

Эти *субъективные* оценки отражают пожелания пользователей к качеству сетевых сервисов. Пользователи, клиенты — это важнейшая сторона любого бизнеса, в том числе бизнеса сетей передачи данных, но существует и еще одна сторона — поставщик услуг (коммерческий, если это публичная сеть, и некоммерческий, если это корпоративная сеть). Для того чтобы пользователи и поставщики услуг могли обоснованно судить о качестве сервисов, существуют *формализованные характеристики качества сетевых услуг*, которые позволяют количественно оценить тот или иной аспект качества.

Характеристики и требования к сети

Работая в сети, пользователь формулирует определенные *требования* к ее характеристикам. Например, пользователь может потребовать, чтобы средняя скорость передачи его информации через сеть была не ниже 2 Мбит/с. Очевидно,

что он при этом использует характеристику «средняя скорость передачи информации через сеть» и определяет тот диапазон значений этой характеристики, который для него означает хорошее качество сервиса.

Все множество характеристик качества транспортных услуг сети можно отнести к одной из следующих групп:

- производительность;
- надежность;
- безопасность;
- характеристики, интересные только поставщику услуг.

Первые три группы соответствуют трем наиболее важным для пользователя характеристикам транспортных услуг — возможности без потерь и перерывов в обслуживании (**надежность**) передавать с заданной скоростью (**производительность**) защищенную от несанкционированного доступа и подмены информацию (**безопасность**). Понятно, что поставщик сетевых услуг, стремясь удовлетворить требования пользователей, также уделяет внимание этим характеристикам. В то же время существует ряд важных для поставщика характеристик сети, которые не интересуют пользователей.

Дело в том, что сеть обслуживает большое количество пользователей, и поставщику услуг нужно организовать работу своей сети таким образом, чтобы одновременно удовлетворить требования *всех* пользователей. Как правило, это сложная проблема, так как основные ресурсы сети — линии связи и коммутаторы (маршрутизаторы) — разделяются между информационными потоками пользователей. Поставщику необходимо найти такой баланс в распределении ресурсов между конкурирующими потоками, чтобы требования всех пользователей были соблюдены. Решение этой задачи включает **планирование** использования ресурсов и **контроль** над их использованием в процессе передачи пользовательского трафика. Поставщика интересуют те характеристики ресурсов, с помощью которых он обслуживает пользователей. Например, его интересует производительность коммутатора, так как поставщик должен оценить, какое количество потоков пользователей он может обслужить с помощью данного коммутатора. Для пользователя производительность коммутатора не представляет интерес, ему важен конечный результат — будет его поток обслужен качественно или нет.

Итак, в четвертой группе собраны характеристики качества, которые интересуют только поставщика услуг. Примером такой характеристики может служить **масштабируемость**, то есть способность сети работать качественно при значительном увеличении числа пользователей без изменения применяемой в этой сети технологии.

Временная шкала

Рассмотрим еще один способ классификации характеристик — в соответствии с временной шкалой, на которой эти характеристики определяются.

Долговременные характеристики определяются на промежутках времени от нескольких месяцев до нескольких лет. Их можно назвать характеристиками

проектных решений. Примерами таких характеристик являются набор моделей и количество коммутаторов в сети, топология и пропускная способность линий связи. Эти параметры сети прямо влияют на характеристики качества услуг сети. Одно проектное решение может оказаться удачным и сбалансированным, так что потоки трафика не будут испытывать перегрузок; другое может создавать узкие места для потоков, так что задержки и потери пакетов превысят допустимые пределы. Понятно, что полная замена или глубокая модернизация сети связана с большими затратами финансовых средств и времени, поэтому они происходят не слишком часто и продолжают оказывать влияние на качество сети в течение продолжительного времени.

Среднесрочные характеристики определяются на интервалах времени от нескольких секунд до нескольких дней. Примерами характеристик этого диапазона являются средние скорости потоков трафика или средние значения задержек пакетов, определяемые на достаточно продолжительном промежутке времени, который включает обслуживание большого количества пакетов. Примером методов, влияющих на среднесрочные характеристики, являются методы определения маршрутов трафика. Маршруты трафика могут быть неизменными в течение часов или дней, если топология сети и параметры трафика остаются постоянными, а каналы и коммутаторы сети не отказывают.

Краткосрочные характеристики относятся к темпу обработки отдельных пакетов и измеряются в микросекундном и миллисекундном диапазонах. Например, время буферизации или время пребывания пакета в очереди коммутатора или маршрутизатора являются характеристиками этой группы. Для анализа и обеспечения требуемого уровня краткосрочных характеристик разработано большое количество методов, получивших название **методов контроля и предотвращения перегрузок** (congestions control and congestion avoidance).

Соглашение об уровне обслуживания

Естественной основой нормального сотрудничества поставщика услуг и пользователей является договор. Договор всегда заключается между клиентами и поставщиками услуг публичных сетей передачи данных, однако не всегда в нем указываются количественные требования к эффективности предоставляемых услуг. Очень часто в договоре услуга специфицируется очень общо, например, «предоставление доступа в Интернет».

Однако существует и другой тип договора, называемый **соглашением об уровне обслуживания** (Service Level Agreement, SLA). В таком соглашении поставщик услуг и клиент описывают качество предоставляемой услуги в количественных терминах, пользуясь характеристиками эффективности сети. Например, в SLA может быть записано, что поставщик обязан передавать трафик клиента без потерь и с той средней скоростью, с которой пользователь направляет его в сеть. При этом оговорено, что это соглашение действует только в том случае, если средняя скорость трафика пользователя не превышает, например, 3 Мбит/с, в противном случае поставщик получает право просто не передавать избыточный трафик. Для того чтобы каждая сторона могла контролировать соблюдение этого соглашения, необходимо еще указать период времени, на котором будет

измеряться средняя скорость, например день, час или секунда. Еще более определенным соглашение SLA становится в том случае, когда в нем указываются средства и методы измерения характеристик сети, чтобы у поставщика и пользователя не было расхождений при контроле соглашения.

Соглашения SLA могут заключаться не только между поставщиками коммерческих услуг и их клиентами, но и между подразделениями одного и того же предприятия. В этом случае поставщиком сетевых услуг может являться, например, отдел информационных технологий, а потребителем — производственный отдел.

Производительность

Список ключевых слов: предложенная нагрузка, гистограмма распределения, задержка доставки пакета, плотность распределения задержки доставки пакета, статистические, или вероятностные, характеристики, среднее значение задержки, джиттер, коэффициент вариации, максимальная задержка, максимальная вариация задержки, время реакции сети, время оборота, скорость передачи данных, средняя скорость передачи данных, пиковая скорость передачи данных, период пульсации, величина пульсации, коэффициент пульсации трафика, узкое место составного пути.

Мы уже знакомы с такими важными долговременными характеристиками производительности сетевых устройств, как пропускная способность каналов или производительность коммутаторов и маршрутизаторов. Наибольший интерес данные характеристики представляют для поставщиков услуг — на их основе поставщик услуг может планировать свой бизнес, рассчитывая максимальное количество клиентов, которое он может обслужить, определяя рациональные маршруты прохождения трафика и т. п.

Однако клиента интересуют другие характеристики производительности, которые позволят ему количественно оценить, насколько быстро и качественно сеть передает его трафик. Для того чтобы определить эти характеристики, воспользуемся моделью идеальной сети.

Идеальная сеть

Будем считать, что сеть работает идеально, если она передает каждый бит информации с постоянной задержкой, равной скорости распространения света в физической среде. Пусть каналы идеальной сети обладают некоторой конечной (а не бесконечной, как хотелось бы) пропускной способностью, поэтому источник информации передает пакет в сеть не мгновенно, а за некоторое конечное время (которое равно, как мы уже знаем, частному от деления объема пакета в битах на пропускную способность канала доступа в сеть).

Результат передачи пакетов такой идеальной сетью иллюстрирует рис. 6.1. На верхней оси показаны значения времени поступления пакетов в сеть от узла отправителя, а на нижнем — значения времени поступления пакетов в узел назначения. Другими словами, можно сказать, верхняя ось показывает **предложенную**

нагрузку сети, а нижняя — результат передачи этой нагрузки через сеть. Мы отсчитываем значения времени отправления и поступления от момента попадания первого бита пакета в сеть или в узел назначения соответственно.

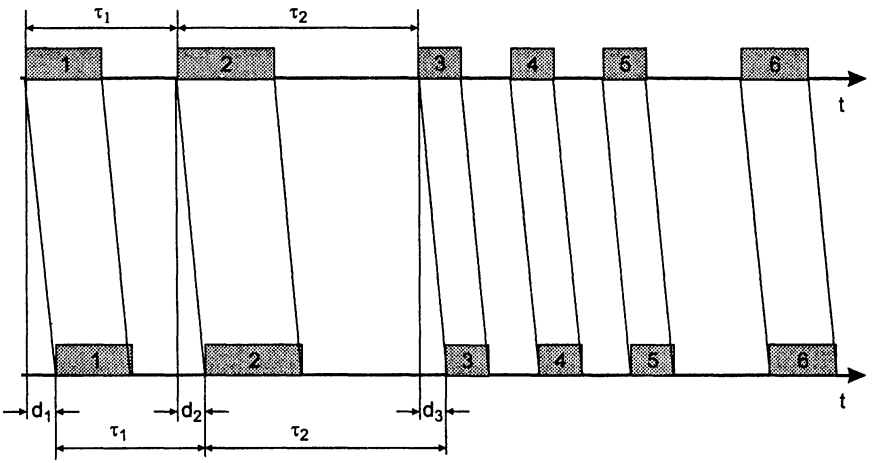


Рис. 6.1. Передача пакетов идеальной сетью

Как видно из рисунка, идеальная сеть доставляет все пакеты узлу назначения:

- не потеряв ни один из них (и не исказив информацию ни в одном из них);
- в том порядке, в котором они были отправлены;
- с одной и той же и минимально возможной задержкой ($d_1 = d_2$ и т. д.).

Важно, что все интервалы между соседними пакетами сеть сохраняет в неизменном виде. Например, если интервал между первым и вторым пакетами составляет при отправлении τ_1 секунд, а между вторым и третьим — τ_2, то такими же интервалы останутся в узле назначения.

Надежная доставка всех пакетов с минимально возможной задержкой и сохранением временны́х интервалов между ними удовлетворит любого пользователя сети независимо от того, трафик какого приложения он передает по сети — вебсервиса или IP-телефонии.

Теперь посмотрим, какие отклонения от идеала могут встречаться в реальной сети и какими характеристиками можно эти отклонения описывать (рис. 6.2).

Пакеты доставляются сетью узлу назначения *с различными задержками*. Как мы уже знаем, это неотъемлемое свойство сетей с коммутацией пакетов. Случайный характер процесса образования очереди приводит к случайным задержкам, при этом задержки отдельных пакетов могут быть значительными, в десятки раз превосходя среднюю величину задержек ($d_1 \neq d_2 \neq d_3$ и т. д.). Неравномерность задержек приводит к неравномерным интервалам между соседними пакетами. То есть изменяется характер временных соотношений между соседними пакетами, а это может катастрофически сказаться на качестве работы некоторых приложений. Например, при цифровой передаче речи (или более обобщенно — звука),

неравномерность интервалов между пакетами, несущими замеры голоса, приводит к существенным искажениям речи.

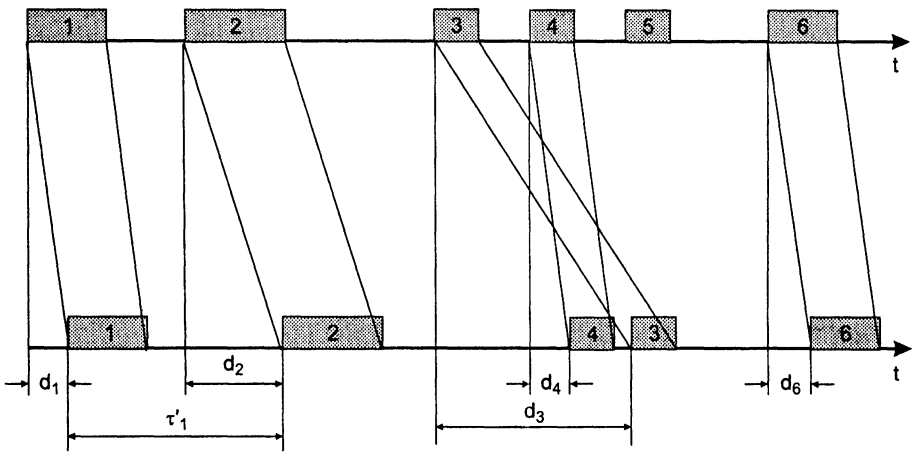


Рис. 6.2. Передача пакетов реальной сетью

Пакеты могут доставляться узлу назначения *не в том порядке*, в котором они были отправлены, например, на рис. 6.2 пакет 4 поступил в узел назначения раньше, чем пакет 3. Такие ситуации встречаются в дейтаграммных сетях, когда различные пакеты одного потока передаются через сеть различными маршрутами, а следовательно, ожидают обслуживания в разных очередях с разным уровнем задержек. Очевидно, что пакет 3 проходил через перегруженный узел или узлы, так что его суммарная задержка оказалась настолько большой, что пакет 4 прибыл раньше него.

Пакеты *могут теряться* в сети или же приходить в узел назначения *с искаженными данными*, что равносильно потере пакета, так как большинство протоколов не может восстановить искаженные данные, а только определяет этот факт по значению контрольной последовательности кадра (Frame Check Sequence, FCS).

Средняя скорость информационного потока на входе узла назначения может отличаться от средней скорости потока, направленного в сеть узлом отправителем. Виной этому являются не задержки пакетов, а их потери. Так, в примере, показанном на рис. 6.2, *средняя скорость исходящего потока уменьшается* из-за потери пакета 5. Чем больше потерь и искажений пакетов происходит в сети, тем ниже скорость информационного потока.

Очевидно, что множество отдельных значений времени передачи каждого отдельного пакета в узел назначения дают исчерпывающую характеристику качества передачи трафика сетью. Однако это слишком громоздкая и, более того, избыточная характеристика производительности сети. Для того чтобы представить характеристики качества передачи последовательности пакетов через сеть в компактной форме, применяются *статистические методы*. Статистические характеристики выявляют закономерности в поведении сети, которые устойчиво проявляются только на длительных периодах времени. Когда мы говорим о длительном пе-

риоде времени, то мы понимаем под этим период времени, в миллионы раз больший, чем время передачи одного пакета, которое в современной сети измеряется микросекундами. Так, время передачи пакета Fast Ethernet составляет около 100 мкс, Gigabit Ethernet — около 10 мкс, ячейки ATM — от долей микросекунды до 3 мкс (в зависимости от скорости передачи). Поэтому для получения устойчивых результатов нужно наблюдать поведение сети по крайней мере в течение минут, а лучше — нескольких часов.

Существует две группы статистических характеристик, которые относятся к производительности сети:

- характеристики задержек пакетов;
- характеристики скорости передачи данных.

Характеристики задержек пакетов

Основным инструментом статистики является так называемая **гистограмма** распределения оцениваемой величины. В данном случае оцениваемой величиной является **задержка доставки пакета**.

Будем считать, что нам удалось измерить задержку доставки каждого пакета и сохранить полученные результаты. Для того чтобы получить гистограмму распределения, мы должны разбить весь диапазон возможных задержек на несколько интервалов и подсчитать, сколько пакетов из нашей последовательности измерений попало в каждый интервал. В результате мы получим гистограмму, показанную на рис. 6.3. Здесь диапазон 25–75 мс, в который попали все значения задержек, разбит на шесть интервалов (сеть вносит фиксированную задержку в 25 мс, связанную с распространением сигнала и буферизацией пакета). Значит, мы можем использовать в качестве характеристики производительности нашей сети шесть чисел: n1, n2, n3, n4, n5 и n6. Это уже более компактная форма представления последовательности задержек. Но нужно соблюдать баланс между желанием сократить до минимума количество интервалов и информативностью полученных характеристик.

Гистограмма задержек дает хорошее представление о производительности сети. По ней можно судить, какие уровни задержек более вероятны, а какие менее. Чем больше период времени, в течение которого собираются данные для построения гистограммы, тем с более высокой степенью вероятности можно предсказать поведение сети в будущем. Например, пользуясь гистограммой на рис. 6.3, можно сказать, что с вероятностью 0,6 задержка пакета не превысит 50 мс. Для нахождения такой оценки мы сложили общее количество пакетов, задержки которых попали во все интервалы, меньшие 50 мс, и разделили эту величину на общее количество пакетов. Другими словами, мы нашли долю пакетов, задержки которых не превышают 50 мс.

При увеличении количества интервалов и времени наблюдения мы в пределе получаем непрерывную функцию, которая называется **плотностью распределения задержки доставки пакета** (показана пунктиром). В соответствии с теорией, вероятность того, что значение случайной величины окажется в определенном диапазоне, равна интегралу плотности распределения случайной величины от

нижней до верхней границ данного диапазона. Таким образом может быть вычислена вероятностное значение задержки пакета.

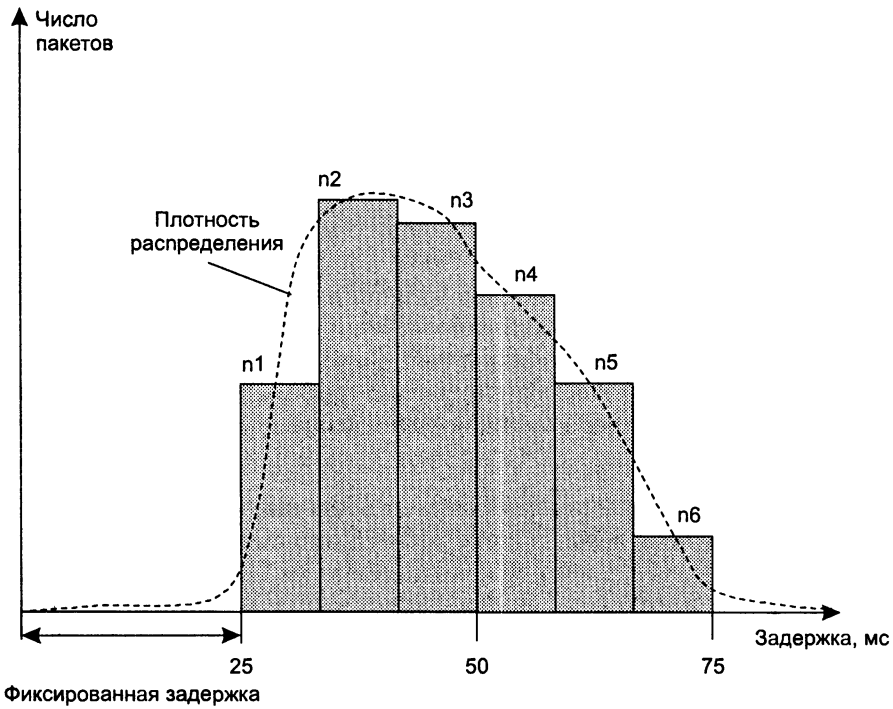


Рис. 6.3. Гистограмма распределения задержек

Только что мы впервые столкнулись с тем, что большое количество сетевых характеристик являются **статистическими (вероятностными)**. Мы не можем со стопроцентной уверенностью сказать, что характеристика имеет некоторое конкретное значение. Мы можем говорить об этом только с какой-то вероятностью, потому что процессы перемещения данных в сети с коммутацией пакетов являются *случайными процессами* по своей сути.

Определим еще несколько часто используемых статистических характеристик задержки пакета.

Среднее значение задержки (D) вычисляется как сумма всех задержек d_i, деленная на количество всех измерений N:

$$D = \sum \frac{d_i}{N}.$$

Джиттер[1] (J) представляет собой среднее отклонение каждой отдельной задержки от среднего значения задержки:

[1] Термин «джиттер» является примером сетевого «жаргона», математики называют эту величину *стандартным отклонением*.

$$J = \sqrt{\frac{\sum(d_i - D)^2}{N-1}}.$$

Как среднее значение задержки, так и джиттер измеряются в секундах. Очевидно, что если все задержки d_i равны между собой, то вариация отсутствует, что подтверждают приведенные формулы — в этом случае $D = d_i$ и $J = 0$.

Коэффициент вариации — это безразмерная величина, которая равна отношению джиттера к среднему значению задержки:

$$CV = J/D.$$

Коэффициент вариации характеризует трафик без привязки к абсолютным значениям временной оси. Идеальный равномерный поток данных всегда будет обладать нулевым значением коэффициента вариации. Коэффициент вариации, равный 1, означает достаточно пульсирующий трафик, так как средние отклонения интервалов от некоторого среднего периода следования пакетов равны этому периоду.

Максимальная задержка — это величина, которую задержки пакетов не должны превосходить с заданной вероятностью. Мы недавно вычисляли такую величину по гистограмме задержек. Чтобы получить оценку, достаточно определенно говорящую о качестве работы сети, обычно задают высокую вероятность, например 0,95 или 0,99. Действительно, если пользователю скажут, что сеть будет обеспечивать уровень задержек в 100 мс с вероятностью 0,5, то это его не очень обрадует, так как он ничего не будет знать об уровне задержек половины своих пакетов.

Максимальная вариация задержки — максимальное значение, на которое отклонение задержки от среднего значения задержки не превосходит с некоторой вероятностью.

Время реакции сети является интегральной характеристикой производительности сети с точки зрения пользователя. Именно эту характеристику имеет в виду пользователь, когда говорит: «Сегодня сеть работает медленно». Время реакции определяется как интервал времени между возникновением запроса пользователя к какой-либо сетевой службе и получением ответа на этот запрос. Время реакции сети можно представить в виде нескольких слагаемых, например (рис. 6.4): время подготовки запросов на клиентском компьютере ($t_{клиент1}$), время передачи запросов между клиентом и сервером через сеть ($t_{сеть}$), время обработки запросов на сервере ($t_{сервер}$), время передачи ответов от сервера клиенту через сеть (снова $t_{сеть}$) и время обработки получаемых от сервера ответов на клиентском компьютере ($t_{клиент2}$).

Время реакции сети характеризует сеть в целом, в том числе качество работы аппаратного и программного обеспечения серверов. Для того чтобы отдельно оценить транспортные возможности сети, чаще используется другая характеристика — время оборота данных по сети.

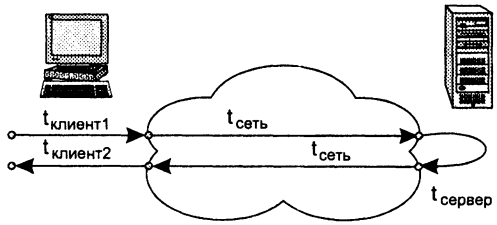


Рис. 6.4. Время реакции и время оборота

Время оборота (Round Trip Time, RTT) — это «чистое» время транспортировки данных от узла отправителя до узла назначения и обратно без учета времени, затраченного узлом назначения на подготовку ответа:

$$RTT = 2 \times t_{сеть}.$$

RTT является полезной характеристикой в том случае, когда значения времени передачи данных по сети в прямом и обратном направлениях отличаются друг от друга. Как и для односторонних задержек, значение RTT можно оценивать по его среднему и максимальному (с заданной вероятностью) значениям.

В зависимости от типа приложения клиент может использовать тот или иной набор характеристик задержек. Рассмотрим, например, работу приложения, воспроизводящего музыку через Интернет. Поскольку эта услуга не является интерактивной, она допускает значительные задержки передачи отдельных пакетов, например несколько минут. Однако пакеты должны приходить равномерно, то есть вариация задержки должна не превышать 100–150 мс, иначе качество воспроизведения музыки резко упадет. Поэтому в данном случае требования к сети должны включать ограничения на среднюю вариацию задержки или максимальное значение вариации задержки.

Характеристики скорости передачи

Скорость передачи данных (information rate) измеряется на каком-либо промежутке времени как частное от деления объема переданных данных за этот период на продолжительность периода. Таким образом, данная характеристика всегда является средней скоростью передачи данных.

Однако в зависимости от величины интервала, на котором измеряется скорость, для этой характеристики традиционно используется одно из двух наименований: средняя или пиковая скорость.

Средняя скорость передачи данных (Sustained Information Rate, SIR)[1] определяется на достаточно большом периоде времени. Это среднесрочная характеристика, период времени должен быть достаточным, чтобы можно было говорить об устойчивом поведении такой случайной величины, которой является скорость.

[1] Традиционно, для одной и той же характеристики может существовать несколько названий. Мы приводим только те из них, которые, по нашему мнению, наилучшим образом отражают их смысл.

Должен быть оговорен период контроля этой величины, например 10 секунд. Это означает, что каждые 10 секунд вычисляется скорость информационного потока и сравнивается с требованием к этой величине. Если такие контрольные измерения не проводить, это лишило бы пользователя возможности предъявлять претензии поставщику в некоторых конфликтных ситуациях. Например, если поставщик в один из дней месяца вообще не будет передавать пользовательский трафик, а в остальные дни разрешит пользователю превышать оговоренный предел, то средняя скорость за месяц окажется в норме. В этой ситуации только регулярный контроль скорости поможет пользователю отстоять свои права.

Пиковая скорость передачи данных (Peak Information Rate, PIR) — это наибольшая скорость, которую разрешается достигать пользовательскому потоку в течение оговоренного небольшого периода времени T.

Этот период обычно называют **периодом пульсации**. Очевидно, что при передаче трафика можно говорить об этой величине только с некоторой степенью вероятности. Например, требование к этой характеристике может быть сформулировано так: «Скорость информации не должна превышать 2 Мбит/с на периоде времени 10 мс с вероятностью 0,95». Часто значение вероятности опускают, подразумевая близость ее к единице. Пиковая скорость является краткосрочной характеристикой. PIR позволяет оценить способность сети справляться с пиковыми нагрузками, характерными для пульсирующего трафика и приводящими к перегрузке. Если в SLA оговорены обе скорости (SIR и PIR), очевидно, что периоды пульсации должны сопровождаться периодами относительного «затишья», когда скорость падает ниже средней. В противном случае показатель средней скорости соблюдаться не будет.

Величина пульсации (обычно обозначаемая B) используется для оценки емкости буфера коммутатора, необходимого для хранения данных во время перегрузки. Величина пульсации равна общему объему данных, поступающих на коммутатор в течение разрешенного интервала T (периода пульсации) передачи данных с пиковой скоростью (PIR):

$$B = PIR \times T.$$

В главе 3, обсуждая коммутацию каналов, мы упоминали еще об одном показателе — **коэффициенте пульсации трафика**. Мы определили его как отношение максимальной скорости на каком-либо небольшом периоде времени к средней скорости трафика, измеренной на длительном периоде времени. Неопределенность временны́х периодов делает коэффициент пульсации *качественной* характеристикой трафика.

Скорость передачи данных можно измерять между любыми двумя узлами, или точками, сети, например между клиентским компьютером и сервером, между входным и выходным портами маршрутизатора. Для анализа и настройки сети очень полезно знать данные о пропускной способности отдельных элементов сети. Важно отметить, что из-за последовательного характера передачи различными элементами сети общая пропускная способность сети любого составного пути в сети будет равна минимальной из пропускных способностей составляющих элементов маршрута. Поэтому максимальная скорость передачи данных всегда

ограничена пропускной способностью таких элементов. Для повышения пропускной способности составного пути необходимо в первую очередь обратить внимание на самые медленные элементы, называемые **узкими местами** (bottleneck).

Надежность

Список ключевых слов: доля потерянных пакетов, среднее время наработки на отказ, вероятность отказа, интенсивность отказов, доступность, отказоустойчивость, альтернативный маршрут следования трафика, методы повторной передачи пакетов, положительная квитанция, отрицательная квитанция, метод простоя источника, метод скользящего окна, размер окна.

Характеристики потерь пакетов

В качестве характеристики потери пакетов используется **доля потерянных пакетов** (обозначим ее L), равная отношению количества потерянных пакетов (NL) к общему количеству переданных пакетов (N):

$$L = NL/N.$$

Может также использоваться аналогичная характеристика, оперирующая не количествами потерянных и переданных пакетов, а объемами данных, содержавшихся в этих пакетах.

Доступность и отказоустойчивость

Для описания надежности отдельных устройств используются такие показатели надежности, как **среднее время наработки на отказ**, **вероятность отказа**, **интенсивность отказов**. Однако эти показатели пригодны только для оценки надежности простых элементов и устройств, которые при отказе любого своего компонента переходят в неработоспособное состояние. Сложные системы, состоящие из многих компонентов, могут при отказе одного из компонентов сохранять свою работоспособность. В связи с этим для оценки надежности сложных систем применяется другой набор характеристик.

Доступность (availability) означает долю времени, в течение которого система или служба находится в работоспособном состоянии.

Доступность является долговременной статистической характеристикой, поэтому измеряется на большом промежутке времени, которым может быть день, месяц или год. Примером высокого уровня доступности является коммуникационное оборудование телефонных сетей, лучшие представители которого обладают так называемой доступностью «пять девяток». Это означает, что доступность равна 0,99999, что соответствует чуть более 5 минутам простоя в год. Оборудование и услуги передачи данных только стремятся к такому рубежу, но рубеж трех девяток уже достигнут. Доступность услуги является универсальной характеристикой, которая используется как пользователями, так и поставщиками услуг.

Еще одной характеристикой надежности сложных систем является **отказоустойчивость** (fault tolerance). Под отказоустойчивостью понимается способность системы скрывать от пользователя отказ отдельных ее элементов.

Например, если коммутатор оснащен двумя коммутационными центрами, работающими параллельно, то отказ одного их них не приведет к полному останову коммутатора. Однако производительность коммутатора снизится, он будет обрабатывать пакеты вдвое медленней. В отказоустойчивой системе отказ одного из элементов приводит к некоторому снижению качества ее работы (**деградации**), а не к полному останову. В качестве еще одного примера можно назвать использование двух физических каналов для соединения коммутаторов. В нормальном режиме работы трафик передается по двум каналам со скоростью С Мбит/с, а при отказе одного из них трафик будут продолжать передаваться, но уже со скоростью С/2 Мбит/с. Однако из-за того, что во многих случаях количественно определить степени деградации системы или услуги достаточно сложно, отказоустойчивость чаще всего используется как качественная характеристика.

Далее мы рассмотрим наиболее часто применяемые методы обеспечения высокой надежности транспортных услуг.

Альтернативные маршруты

Доступность услуги можно улучшить двумя способами.

Первый состоит в использовании в сети *надежных элементов*, которые редко отказывают, но этот путь всегда ограничен возможностями технологии производства электронных компонентов (интегральных схем, печатных плат и т. п.).

Другой путь основан на введении *избыточности* в структуру системы: ключевые элементы системы должны существовать в нескольких экземплярах, чтобы при отказе одного из них функционирование системы обеспечивали другие. Поэтому коммутаторы и маршрутизаторы, работающие на магистрали сети, всегда строятся с использованием избыточных блоков — источников питания, процессоров и интерфейсов.

Для того чтобы обеспечить требуемый уровень доступности транспортного сервиса, поставщик должен иметь отказоустойчивую сеть. Другого пути здесь нет, так как в крупной системе, каковой является сеть, часто будут возникать отказы многочисленных составляющих ее элементов — каких-либо каналов или коммутаторов. Несмотря на то что эти элементы сами по себе являются отказоустойчивыми, предел их устойчивости может быть достигнут, и тогда сеть должна компенсировать их отказ другими средствами.

В хорошо спроектированной сети такими средствами являются **альтернативные маршруты следования трафика**. В примере сети на рис. 6.5, *а* альтернативные пути для передачи трафика между точками А и В отсутствуют. Поэтому такой дизайн сети не обладает свойством отказоустойчивости, поставщику сети приходится полагаться только на отказоустойчивость каналов связи и коммутаторов на пути между А и В.

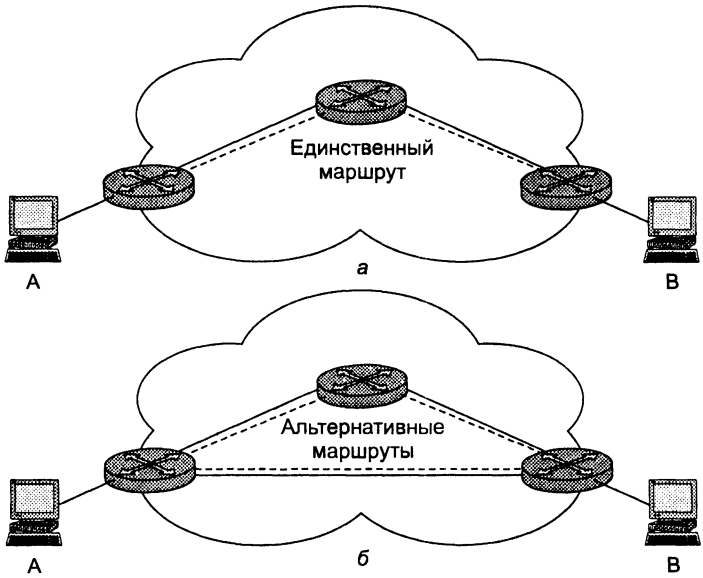


Рис. 6.5. Альтернативные маршруты

В примере на рис. 6.5, *б* сеть может передавать трафик между точками A и B по двум альтернативным маршрутам. Очевидно, что при отказе оборудования вдоль одного из маршрутов второй остается работоспособным, и сеть продолжает предоставлять услугу пользователю. При этом сеть может затратить некоторое время на переход от одного маршрута к другому. В течение переходного периода возможны потери пользовательского трафика, поэтому сокращение этого периода — одна из главных целей техники обеспечения отказоустойчивости сетей.

Существует несколько способов использования альтернативных маршрутов в сети.

- *Сеть определяет альтернативный маршрут только после отказа основного.* Это означает, что в таблицах продвижения коммутаторов сети для каждого информационного потока определен только *один* маршрут. После отказа линии связи или коммутатора на пути следования этого потока коммутаторы сети с помощью некоторого протокола маршрутизации начинают поиск альтернативного пути. Обычно нахождение такого пути занимает десятки секунд или несколько минут — в зависимости от масштаба сети и сложности ее топологии. Это самый медленный способ использования альтернативных маршрутов, и очевидно, он связан с потерями пользовательских данных в переходный период.
- *Сеть заранее находит и использует два маршрута для определенного потока, образуя избыточный поток, незаметный для пользователя.* Хотя данные передаются по обоим маршрутам, к пользователю попадают данные только одного. То есть один из маршрутов считается *основным*, другой — *резервным*. При отказе основного маршрута пользователь получает данные по резервному маршруту. Этот способ самый быстрый, поэтому он обеспечивает высшее качество обслуживания пользовательского потока. Однако он связан с большими

потерями производительности сети, так как вместо одного потока сеть организует два. Обычно такой способ применяют для поддержания небольшого количества критически важных для пользователя потоков, которым требуется высокий уровень качества обслуживания и доступности сервиса.

- *Сеть заранее находит два маршрута, однако использует только один.* При отказе основного маршрута переход на альтернативный маршрут происходит быстрее, чем при использовании первого способа, так как не тратится время на его поиск. Этот способ также гораздо более экономичен, чем второй. Но потери данных здесь выше, чем у второго способа, так как теряются те данные, которые уже были отправлены вдоль отказавшего маршрута.

В компьютерных сетях применяют в основном первый и третий способы использования альтернативных маршрутов. Технологии второго способа (два активных маршрута) задействуются в компьютерных сетях только тогда, когда к сетям предъявляются требования повышенной надежности. Широкое применение нашел второй способ в первичных высокоскоростных сетях, которые, как мы уже упоминали, были созданы для передачи телефонного трафика. Эти сети вносят свой существенный вклад в обеспечение высокого уровня доступности телефонных сетей, отсюда и особые требования к надежности.

Повторная передача и скользящее окно

В тех случаях, когда другие методы обеспечения надежности не срабатывают и пакеты теряются, применяют **методы повторной передачи пакетов**. Эти методы требуют использования протоколов, ориентированных на соединение.

Чтобы убедиться в необходимости повторной передачи данных, отправитель нумерует отправляемые кадры и для каждого кадра ожидает от приемника так называемой **положительной квитанции** (Positive Acknowledgment, ACK) — служебного кадра, извещающего о том, что исходный кадр получен и данные в нем корректны. Для того чтобы организовать такую нумерацию, и нужна процедура логического соединения — она дает точку отсчета, с которой начинается нумерация. Время ожидания квитанции ограничено — при отправке каждого кадра передатчик запускает таймер, и, если по истечении заданного времени положительная квитанция на получена, кадр считается утерянным. Приемник в случае получения кадра с искаженными данными может отправить **отрицательную квитанцию** (Negative Acknowledgment, NACK) — явное указание на то, что данный кадр нужно передать повторно.

Существует два метода организации процесса обмена квитанциями: метод простоя источника и метод скользящего окна.

Метод **простоя источника** требует, чтобы источник, пославший кадр, ожидал получения квитанции (положительной или отрицательной) от приемника и только после этого посылал следующий кадр (или повторял искаженный). Если же квитанция не приходит в течение тайм-аута, то кадр (или квитанция) считается утерянным и его передача повторяется. На рис. 6.6, *а* видно, что в этом случае производительность обмена данными ниже потенциально возможной, — хотя передатчик и мог бы послать следующий кадр сразу же после отправки предыду-

щего, он обязан ждать прихода положительной квитанции. (Далее, где это не искажает существо рассматриваемого вопроса, положительные квитанции для краткости будут называться просто «квитанциями».)

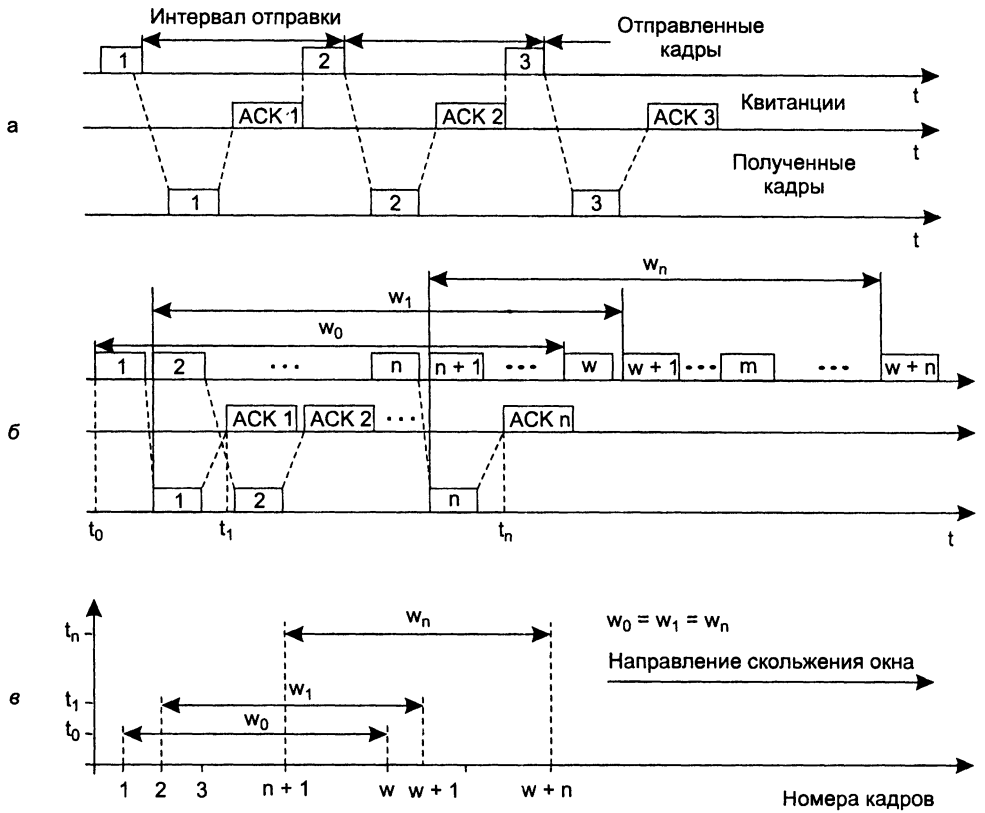


Рис. 6.6. Методы восстановления искаженных и потерянных кадров

Недостатки этого метода коррекции особенно заметны на низкоскоростных каналах связи, то есть в территориальных сетях.

Второй метод называется методом **скользящего окна** (sliding window). В этом методе для повышения скорости передачи данных источнику разрешается передать некоторое количество кадров в непрерывном режиме, то есть в максимально возможном для источника темпе без получения на эти пакеты квитанций. Количество пакетов, которые разрешается передавать таким образом, называется **размером окна**. Рисунок 6.6, *б* иллюстрирует применение данного метода для окна размером W пакетов.

В начальный момент, когда еще не послано ни одного пакета, окно определяет диапазон пакетов с номерами от 1 до W включительно. Источник начинает передавать пакеты и получать в ответ квитанции. Для простоты предположим, что квитанции поступают в той же последовательности, что и пакеты, которым они

соответствуют. В момент t_1 при получении первой квитанции K_1 окно сдвигается на одну позицию, определяя новый диапазон от 2 до (W + 1).

Процессы отправки пакетов и получения квитанций идут достаточно независимо друг от друга. Рассмотрим произвольный момент времени t_n, когда источник получает квитанцию на пакет с номером n. Окно сдвигается вправо и определяет диапазон разрешенных к передаче пакетов от (n + 1) до (W + n). Все множество пакетов, выходящих из источника, можно разделить на перечисленные ниже группы (см. рис. 6.6, *б*).

- Пакеты с номерами от 1 до n уже были отправлены и квитанции на них получены, то есть они находятся за пределами окна слева.
- Пакеты, начиная с номера (n + 1) и заканчивая номером (W + n), находятся в пределах окна и потому могут быть отправлены, не дожидаясь прихода какой-либо квитанции. Этот диапазон может быть разделен еще на два поддиапазона:
  - пакеты с номерами от (n + 1) до m уже отправлены, но квитанции на них еще не получены;
  - пакеты с номерами от m до (W + n) пока не отправлены, хотя запрета на это нет.
- Все пакеты с номерами, большими или равными (W + n + 1), находятся за пределами окна справа и поэтому пока не могут быть отправлены.

Перемещение окна вдоль последовательности номеров пакетов иллюстрирует рис. 6.6, *в*. Здесь t_0 — исходный момент, t_1 и t_n — моменты прихода квитанций на первый и n-й пакет соответственно. Каждый раз, когда приходит квитанция, окно сдвигается влево, но его размер при этом не меняется и остается равным W.

При отправке пакета в источнике устанавливается тайм-аут. Если за это время квитанция на отправленный пакет не придет, пакет (или квитанция на него) считается утерянным, и пакет передается снова.

Если же поток квитанций поступает регулярно в пределах допуска в W пакетов, то скорость обмена достигает максимально возможной величины для данного канала и принятого протокола.

В некоторых реализациях скользящего окна от приемника не требуется посылать квитанции на каждый принятый корректный пакет. Если среди пришедших пакетов нет «прогалин», то приемнику достаточно послать квитанцию только на последний принятый пакет, и эта квитанция будет свидетельствовать отправителю о том, что все предыдущие пакеты также дошли благополучно.

В других методах используются отрицательные квитанции. Отрицательные квитанции бывают двух типов — групповые и избирательные. Групповая квитанция содержит номер пакета, начиная с которого нужно повторить передачу всех пакетов, отправленных передатчиком в сеть. Избирательная отрицательная квитанция требует повторной передачи только одного пакета.

Метод скользящего окна имеет два параметра, которые могут существенно влиять на эффективность передачи данных между передатчиком и приемником, —

это размер окна и величина тайм-аута ожидания квитанции. Выбор тайм-аута зависит не от надежности сети, а от задержек передачи пакетов сетью.

В надежных сетях, в которых пакеты искажаются и теряются редко, для повышения скорости обмена данными размер окна нужно увеличивать, так как при этом передатчик будет посылать пакеты с меньшими паузами. В ненадежных сетях размер окна следует уменьшать, так как при частых потерях и искажениях пакетов резко возрастает объем вторично передаваемых через сеть пакетов, а значит, пропускная способность сети во многом расходуется вхолостую, и полезная пропускная способность сети падает.

Размер окна может быть постоянным параметром алгоритма скользящего окна. В этом случае он выбирается при установлении соединения и не меняется в течение сеанса. Существуют также адаптивные версии алгоритма, когда размер окна меняется в течение сеанса в соответствии с текущим состоянием сети и узла назначения.

Надежность сети в подобных алгоритмах определяется по таким признакам потерь пакетов, как истечение тайм-аута для положительной квитанции или приходе дубликата квитанции на определенный пакет. Дубликат говорит о том, что в узле назначения истек тайм-аут ожидания очередного пакета, и узел запрашивает отправку этого пакета вторично. При наступлении таких событий узел отправитель уменьшает размер окна, стараясь найти оптимальный режим передачи данных.

Размер окна может быть также изменен узлом назначения. Причиной уменьшения размера окна является перегрузка узла назначения, который не успевает обработать поступающие пакеты. Мы вернемся к этой проблеме позже, в разделе «Обратная связь» главы 7, когда будем изучать методы борьбы с перегрузками.

Существуют также реализации метода скользящего окна, в которых в качестве размера окна используется не количество пакетов, а количество байтов. Наиболее известным примером такого подхода является протокол TCP.

В общем случае метод скользящего окна более сложен в реализации, чем метод простоя источника, так как передатчик должен хранить в буфере все пакеты, на которые пока не получены положительные квитанции. Кроме того, при использовании данного метода требуется отслеживать несколько параметров алгоритма: размер окна W, номер пакета, на который получена квитанция, номер пакета, который еще можно передать до получения новой квитанции.

Безопасность

Список ключевых слов: уровень защищенности, компьютерная безопасность, сетевая безопасность, брандмауэр, конфиденциальность, доступность, целостность, шифрование, криптосистема, аутентификация, электронная подпись, идентификация, авторизация, аудит, защищенный канал.

Компьютерные сети — замечательное средство доступа к самой разнообразной информации, а также средство общения. Однако, как и у каждой красивой меда-

ли, у компьютерных сетей есть и обратная сторона. Она проявляется в разнообразных потенциальных угрозах целостности и конфиденциальности информации, которую вы доверяете сети. Например, предприятия, имеющие постоянное соединение с Интернетом, регулярно страдают от атак злоумышленников на их информационные ресурсы. Подвергаются атакам и пользователи, соединяющиеся с Интернетом по модему на небольшое время. Информация, находящаяся на их компьютерах, также может пострадать от вирусов, «живущих» в почтовых сообщениях или проникающих через «дырки» в системах мгновенной доставки сообщений, таких как ICQ.

Немного статистики

В отчете о проблемах и тенденциях в сферах компьютерной преступности и безопасности за 2002 год (Issues and Trends: 2002 CSI/FBI Computer Crime and Security Survey), опубликованном в апреле 2002 года, отмечается резкий рост числа обращений в правоохранительные органы по поводу компьютерных преступлений. 90 % респондентов (преимущественно из крупных компаний и правительственных структур) сообщили, что за последние 12 месяцев в их организациях имели место нарушения информационной безопасности; 80 % констатировали финансовые потери от этих нарушений; 44 % (223 респондента) смогли и/или захотели оценить потери количественно — общая сумма составила более 455 млн долларов.

Злоумышленники, которые пытаются получить несанкционированный доступ к информации или разрушить ее, могут использовать для атак не только Интернет, но и корпоративную сеть. Действительно, никогда нельзя ручаться за то, что кто-нибудь из служащих предприятия не захочет превысить свои служебные полномочия и попытаться прочитать документ, который он не имеет права читать. Возможны также попытки разрушить информацию предприятия изнутри (стереть файлы, вывести из строя компьютер и т. п.).

Очевидно, что пользователи сети хотели бы, чтобы их информация была надежно защищена от такого рода инцидентов. **Уровень защищенности** информации пользователя сетевых услуг является еще одной важной характеристикой сети. Защищенность не является количественной характеристикой, ее можно оценить только качественно — высокий (средний, низкий) уровень защищенности. Обычно уровень защищенности информации в определенной сети оценивается экспертом.

Компьютерная и сетевая безопасность

При всем своем многообразии средства защиты информации делятся на два больших класса:

- средства **компьютерной безопасности** предназначены для защиты внутренних информационных ресурсов, находящихся в локальной сети или на отдельном компьютере пользователя;
- средства **сетевой безопасности** предназначены для защиты информации в процессе ее передачи через сеть.

Это деление достаточно естественное, так как функции по обеспечению безопасности в этих двух случаях существенно различаются. В первом случае нужно защитить от несанкционированного доступа все ресурсы, находящиеся внутри собственной локальной сети. Это — аппаратные ресурсы (серверы, дисковые массивы, маршрутизаторы), программные ресурсы (операционные системы, СУБД, почтовые службы и т. п.) и сами данные, хранящиеся в файлах и обрабатываемые в оперативной памяти. Очевидно, что для этого необходимо контролировать трафик, входящий в сеть из публичной сети (сегодня это преимущественно Интернет), и стараться перекрыть доступ извне для любой информации, с помощью которой злоумышленник может попытаться использовать внутренние ресурсы сети во вред их владельцу.

Наиболее часто используемым средством защиты этого типа является брандмауэр, устанавливаемый в местах всех соединений внутренней сети с Интернетом. **Брандмауэр** (firewall) представляет собой межсетевой экран, который контролирует обмен сообщениями, ведущийся по протоколам всех уровней, и не пропускает подозрительный трафик в сеть.

Брандмауэр может использоваться и внутри сети, защищая одну подсеть от другой, что может быть необходимо в крупных компаниях с достаточно независимыми подразделениями. Помимо брандмауэра аналогичные проблемы призваны решать встроенные средства безопасности операционных систем и приложений, таких как базы данных, а также встроенные аппаратные средства компьютера.

В плане обеспечения сетевой безопасности приходится защищать информацию, которая находится вне пределов нашей досягаемости, а в виде IP-пакетов «путешествует» через сети поставщиков услуг Интернета. Интернет сегодня используется предприятиями не только как сверхмощный источник информации, хранящейся на многочисленных веб-сайтах, но и как дешевая транспортная среда, позволяющая объединить сеть центрального отделения с сетями филиалов, а также подключить к ресурсам предприятия телекомьютеров — сотрудников, находящихся дома или в командировке работающих с корпоративной сетью удаленно. При этом во многих случаях предприятию жизненно важно, чтобы передаваемая через Интернет информация не была искажена, уничтожена или просмотрена посторонними людьми. Для решения этой задачи сегодня широко используется механизм виртуальных частных сетей (VPN).

Автономно работающий компьютер можно эффективно защитить от внешних покушений разнообразными способами, например просто заперев на замок клавиатуру или сняв жесткий накопитель и поместив его в сейф. Компьютер, работающий в сети, по определению не может полностью отгородиться от мира, он должен общаться с другими компьютерами, возможно, даже удаленными от него на большое расстояние, поэтому обеспечение безопасности в сети является задачей значительно более сложной. Логический вход чужого пользователя в ваш компьютер является штатной ситуацией, если вы работаете в сети. Обеспечение безопасности в такой ситуации сводится к тому, чтобы сделать подобное проникновение контролируемым — каждому пользователю сети должны быть четко определены его права на доступ к информации, внешним устройствам, на выполнение системных действий на каждом из компьютеров сети.

Помимо проблем, порождаемых возможностью удаленного входа в сетевые компьютеры, сети по своей природе подвержены еще одному виду опасности — перехвату и анализу передаваемых по сети сообщений, а также созданию «ложного» трафика. Большая часть средств обеспечения сетевой безопасности направлена на предотвращение именно этого типа нарушений.

Вопросы сетевой безопасности приобретают особое значение сейчас, когда проектировщики корпоративных сетей все чаще отказываются от выделенных линий в пользу транспортных возможностей публичных сетей (Интернет, frame relay).

Конфиденциальность, целостность и доступность данных

Безопасная информационная система — это система, которая, во-первых, защищает данные от несанкционированного доступа, во-вторых, всегда готова предоставить их своим пользователям, в-третьих, надежно хранит информацию и гарантирует неизменность данных. Таким образом, безопасная система по определению обладает свойствами конфиденциальности, доступности и целостности.

- **Конфиденциальность** (confidentiality) — гарантия того, что секретные данные будут доступны только тем пользователям, которым этот доступ разрешен (такие пользователи называются авторизованными).
- **Доступность** (availability) — гарантия того, что авторизованные пользователи всегда получат доступ к данным.
- **Целостность** (integrity) — гарантия сохранности данными правильных значений, которая обеспечивается запретом для неавторизованных пользователей каким-либо образом изменять, модифицировать, разрушать или создавать данные.

Целью злоумышленников может быть нарушение каждой их составляющих информационной безопасности — доступности, целостности или конфиденциальности. Требования безопасности могут меняться в зависимости от назначения системы, характера используемых данных и типа возможных угроз. Трудно представить систему, для которой были бы не важны свойства целостности и доступности, но свойство конфиденциальности не всегда является обязательным. Например, если вы публикуете информацию в Интернете на веб-сервере и вашей целью является сделать ее доступной для самого широкого круга людей, то конфиденциальность в данном случае не требуется. Однако требования целостности и доступности остаются актуальными.

Действительно, если вы не предпримете специальных мер по обеспечению целостности данных, злоумышленник может, например, внести такие изменения в помещенный на веб-сайте прайс-лист, которые снизят конкурентоспособность вашего предприятия, или испортить коды свободно распространяемого вашей фирмой программного продукта, что, безусловно, скажется на ее деловом имидже.

Не менее важной в данном примере является и доступность данных. Затратив немалые средства на создание и поддержание сервера в Интернете, предприятие

вправе рассчитывать на отдачу: увеличение числа клиентов, количества продаж и т. д. Однако существует вероятность того, что злоумышленник предпримет атаку, в результате которой помещенные на сервер данные станут недоступными для тех, кому они предназначались. Примером таких злонамеренных действий может служить «бомбардировка» сервера IP-пакетами с неправильным обратным адресом, которые в соответствии с логикой работы протокола IP могут вызывать тайм-ауты и, в конечном счете, сделать сервер недоступным для всех остальных запросов.

ПРИМЕЧАНИЕ

Понятия конфиденциальности, доступности и целостности могут быть определены не только по отношению к информации, но и к другим ресурсам вычислительной сети, например внешним устройствам или приложениям. Возможность «незаконного» использования такого рода ресурсов способна привести к нарушению безопасности системы. Свойство конфиденциальности, примененное к устройству печати, означает, что доступ к нему имеют те и только те пользователи, которым этот доступ разрешен, причем они могут выполнять только те операции с устройством, которые для них определены. Свойство доступности устройства интерпретируется как его готовность к использованию всякий раз, когда в этом возникает необходимость. Благодаря свойству целостности злоумышленник не может изменить параметры настройки устройства, что могло бы привести к изменению очередности работ и даже к выводу устройства из строя. Легальность использования сетевых устройств важна не только постольку, поскольку она влияет на безопасность данных. Устройства могут предоставлять различные услуги (распечатка текстов, отправка факсов, доступ в Интернет, электронная почта и т. п.), незаконное потребление которых наносит материальный ущерб, что также является нарушением безопасности предприятия.

Сервисы сетевой безопасности

В разных программных и аппаратных продуктах, предназначенных для защиты данных, часто используются одинаковые подходы, приемы и технические решения. Такие средства, называемые также сервисами сетевой безопасности, решают самые разнообразные задачи по защите системы, обеспечивая, например, контроль доступа, включающий процедуры аутентификации и авторизации, аудит, шифрование информации, антивирусную защиту, контроль сетевого трафика и пр. Технические средства безопасности могут быть либо встроены в программное (операционные системы и приложения) и аппаратное (компьютеры и коммуникационное оборудование) обеспечение сети, либо реализованы в виде отдельных продуктов, созданных специально для решения проблем безопасности. Определим основные из них.

Шифрование — это краеугольный камень всех систем информационной безопасности, будь то система аутентификации или авторизации, защищенный канал или система безопасного хранения данных. Любая процедура шифрования, превращающая информацию из обычного «понятного» вида в «нечитабельный» зашифрованный вид, естественно должна быть дополнена процедурой дешифрирования, которая после применения к зашифрованному тексту снова делает его «понят-

ным». Пара процедур — шифрование и дешифрирование — называется **криптосистемой**.

Аутентификация предотвращает доступ к сети нежелательных лиц и разрешает вход для легальных пользователей. Термин «аутентификация» в переводе с латинского означает «установление подлинности». В качестве объектов, требующих аутентификации, могут выступать не только пользователи, но и различные устройства, приложения, текстовая и другая информация. Так, пользователь, обращающийся с запросом к корпоративному серверу, должен доказать ему свою легальность, но он также должен убедиться сам, что ведет диалог действительно с сервером своего предприятия. Другими словами, сервер и клиент должны пройти процедуру взаимной аутентификации. Здесь мы имеем дело с аутентификацией на уровне приложений. При установлении сеанса связи между двумя устройствами также часто предусматриваются процедуры взаимной аутентификации на более низком, канальном уровне. Аутентификация данных означает доказательство целостности этих данных, а также факт их поступления именно от того человека, который объявил об этом. Для этого используется механизм **электронной подписи**. Аутентификацию следует отличать от идентификации.

Идентификация заключается в сообщении пользователем системе своего идентификатора, в то время как аутентификация — это процедура доказательства пользователем того, что он является тем, за кого себя выдает, в частности доказательство того, что именно ему принадлежит введенный им идентификатор. Идентификаторы пользователей применяются в системе с теми же целями, что и идентификаторы любых других объектов (файлов, процессов, структур данных), и они не всегда связаны непосредственно с обеспечением безопасности.

Авторизация — процедура контроля доступа легальных пользователей к ресурсам системы с предоставлением каждому из них именно тех прав, которые определены ему администратором. Помимо предоставления пользователям прав доступа к каталогам, файлам и принтерам, система авторизации может контролировать возможность выполнения пользователями различных системных функций, таких как локальный доступ к серверу, установка системного времени, создание резервных копий данных, выключение сервера и т. п.

Аудит — фиксация в системном журнале событий, связанных с доступом к защищаемым системным ресурсам. Подсистема аудита современных ОС позволяет дифференцированно задавать перечень интересующих администратора событий с помощью удобного графического интерфейса. Средства учета и наблюдения обеспечивают возможность обнаружить и зафиксировать важные события, связанные с безопасностью; любые попытки (в том числе и неудачные) создать, получить доступ или удалить системные ресурсы.

Технология защищенного канала обеспечивает безопасность передачи данных по открытой транспортной сети, например по Интернету, за счет:

- взаимной аутентификации абонентов при установлении соединения;
- защиты передаваемых по каналу сообщений от несанкционированного доступа;
- обеспечения целостности поступающих по каналу сообщений.

Характеристики сети поставщика услуг

Список ключевых слов: расширяемость сети, масштабируемость сети, управляемость сети, совместимость, или интегрируемость, сети.

Рассмотрим основные характеристики, которыми оперирует поставщик услуг, оценивая эффективность своей сети. Эти характеристики часто являются качественными, то есть не могут быть выражены числами и соотношениями.

Расширяемость и масштабируемость

Термины «расширяемость» и «масштабируемость» иногда неверно используют как синонимы.

Расширяемость означает возможность сравнительно легкого добавления отдельных компонентов сети (пользователей, компьютеров, приложений, служб), наращивания длины сегментов кабелей и замены существующей аппаратуры более мощной.

При этом принципиально важно, что легкость расширения системы иногда может обеспечиваться в *определенных пределах*. Например, локальная сеть Ethernet, построенная на основе одного разделяемого сегмента толстого коаксиального кабеля, обладает хорошей расширяемостью в том смысле, что позволяет легко подключать новые станции. Однако такая сеть имеет ограничение на число станций — оно не должно превышать 30–40. Хотя сеть допускает физическое подключение к сегменту и большего числа станций (до 100), при этом резко снижается производительность сети. Наличие такого ограничения и является признаком плохой масштабируемости системы при ее хорошей расширяемости.

Масштабируемость означает, что сеть позволяет наращивать количество узлов и протяженность связей в очень широких пределах, при этом производительность сети не ухудшается.

Для обеспечения масштабируемости сети приходится применять дополнительное коммуникационное оборудование и специальным образом структурировать сеть. Обычно масштабируемое решение обладает многоуровневой иерархической структурой, которая позволяет добавлять элементы на каждом уровне иерархии без изменения главной идеи проекта.

Примером хорошо масштабируемой сети является Интернет, технология которого (TCP/IP) оказалась способной поддерживать сеть в масштабах земного шара. Организационная структура Интернета, которую мы рассмотрели в главе 5, образует несколько иерархических уровней: сети пользователей, сетей локальных поставщиков услуг и так далее вверх по иерархии вплоть до сетей межнациональных поставщиков услуг. Технология TCP/IP, на которой построен Интернет, также позволяет строить иерархические сети. Основной протокол Интернета (IP) основан на двухуровневой модели: нижний уровень составляют отдельные сети (чаще всего сети корпоративных пользователей), а верхний уровень — это

составная сеть, объединяющая эти сети. Стек TCP/IP поддерживает также концепцию автономной системы. В автономную систему входят все составные сети одного поставщика услуг, так что автономная система представляет собой более высокий уровень иерархии. Наличие автономных систем в Интернете позволяет упростить решение задачи нахождение оптимального маршрута — сначала ищется оптимальный маршрут между автономными системами, а затем каждая автономная система находит оптимальный маршрут внутри себя.

Не только сама сеть должна быть масштабируемой, но и устройства, работающие на магистрали сети, также должны обладать этим свойством, так как рост сети не должен приводить к необходимости постоянной смены оборудования. Поэтому магистральные коммутаторы и маршрутизаторы строятся обычно по модульному принципу, позволяя наращивать количество интерфейсов и производительность обработки пакетов в широких пределах.

Управляемость

Управляемость сети подразумевает возможность централизованно контролировать состояние основных элементов сети, выявлять и разрешать проблемы, возникающие при работе сети, выполнять анализ производительности и планировать развитие сети. Управляемость подразумевает наличие в сети некоторых автоматизированных средств администрирования, которые взаимодействуют с программным и аппаратным обеспечением сети с помощью коммуникационных протоколов.

В идеале средства администрирования сети представляют собой систему, осуществляющую *наблюдение и контроль* за каждым элементом сети — от простейших до самых сложных устройств, при этом сеть рассматривается как единое целое, а не как разрозненный набор отдельных устройств.

Хорошая система администрирования наблюдает за сетью и, обнаружив проблему, активизирует определенное действие, исправляет ситуацию и уведомляет администратора о том, что произошло и какие шаги предприняты. Одновременно с этим система администрирования должна *накапливать данные*, на основании которых можно планировать развитие сети. Наконец, система администрирования должна быть независима от производителя и обладать удобным интерфейсом, позволяющим выполнять все действия с одной консоли.

Решая тактические задачи, администраторы и технический персонал сталкиваются с ежедневными проблемами обеспечения работоспособности сети. Эти задачи требуют быстрого решения, обслуживающий сеть персонал должен оперативно реагировать на сообщения о неисправностях, поступающих от пользователей или автоматических средств администрирования сети. Постепенно становятся заметными более общие проблемы производительности, конфигурирования сети, обработки сбоев и безопасности данных, требующие стратегического подхода, то есть *планирования* сети. Планирование, кроме того, подразумевает умение прогнозировать изменения в требованиях пользователей к сети, решение вопросов применения новых приложений, новых сетевых технологий и т. п.

Полезность системы администрирования особенно ярко проявляется в больших сетях: корпоративных или публичных глобальных. Без системы администрирования в таких сетях требуется присутствие квалифицированных специалистов по эксплуатации в каждом здании каждого города, где установлено оборудование сети, что в итоге приводит к необходимости содержания огромного штата обслуживающего персонала.

В настоящее время в области систем администрирования сетей накопилось много нерешенных проблем. Явно недостаточно действительно удобных, компактных и многопротокольных средств администрирования. Большинство существующих средств вовсе не управляют сетью, а всего лишь обеспечивают *наблюдение* за ее работой и *фиксацию* важных событий, например отказов устройств. Реже системы администрирования выполняют активные действия, ликвидирующие последствия нежелательного события или предотвращающие его.

Совместимость

Совместимость, или **интегрируемость**, сети означает, что сеть способна включать в себя самое разнообразное программное и аппаратное обеспечение, то есть в ней могут сосуществовать различные операционные системы, поддерживающие разные стеки коммуникационных протоколов, а также аппаратные средства и приложения от разных производителей. Сеть, состоящая из разнотипных элементов, называется неоднородной, или гетерогенной, а если гетерогенная сеть работает без проблем, то она является интегрированной. Основной путь построения интегрированных сетей — использование модулей, выполненных в соответствии с открытыми стандартами и спецификациями.

Выводы

Главным требованием, предъявляемым к компьютерной сети, является обеспечение высокого качества предоставляемых сетью услуг. При широком понимании в понятие «качество обслуживания» включают все возможные характеристики услуг и сети, желательные для пользователя.

Формализованные характеристики позволяют обоснованно судить о качестве сетевых услуг. Требования к качеству услуг выражаются с помощью значений формализованных характеристик.

Качество транспортных услуг оценивается характеристиками четырех групп:

- производительность;
- надежность;
- безопасность;
- характеристики сети поставщика услуг (сюда входят расширяемость, масштабируемость, управляемость, совместимость).

Производительность сети оценивается с помощью статистических характеристик двух типов: характеристик скорости передачи информации и характеристик задержек передачи пакетов. В первую группу входит средняя скорость и максимальная скорость на периоде пульсации, а также длительность этого периода. Во вторую группу входят: средняя величина задержки,

средняя вариация задержки (джиттер), коэффициент вариации, а также максимальные значения задержки и вариации задержки.

Для оценки надежности сетей используются различные характеристики, в том числе: доля потерь пакетов, коэффициент доступности, означающий долю времени, в течение которого система может быть использована, отказоустойчивость — способность системы работать в условиях отказа некоторых ее элементов.

Надежность транспортных услуг сети обеспечивается надежностью ее компонентов (каналов и коммуникационного оборудования), наличием альтернативных маршрутов, а также повторной передачей потерянных или искаженных пакетов.

Средства безопасности компьютерных сетей включают:

- средства компьютерной безопасности, предназначенные для защиты внутренних информационных ресурсов, которые находятся в локальной сети или отдельном компьютере пользователя;
- средства сетевой безопасности, предназначенные для защиты информации в процессе ее передачи через сеть.

Основными характеристиками безопасной информационной системы являются:

- конфиденциальность — гарантия того, что секретные данные будут доступны только тем пользователям, которым этот доступ разрешен (такие пользователи называются авторизованными);
- доступность — гарантия того, что авторизованные пользователи всегда получат доступ к данным;
- целостность — гарантия сохранности данными правильных значений, которая обеспечивается запретом для неавторизованных пользователей каким-либо образом изменять, модифицировать, разрушать или создавать данные.

Для защиты информации в сетях используются механизмы шифрования, аутентификации, авторизации и аудита. Передача данных через сеть осуществляется с помощью комбинированной техники защищенного канала.

Вопросы и задания

1. Чем характеристика отличается от требования?
2. Какие характеристики входят в понятие «качество обслуживания» при его широкой трактовке?
3. Что объединяет характеристики QoS в узком значении этого термина?
4. Назовите характеристики качества обслуживания, которые интересуют только пользователя? Только поставщика услуг? И пользователя, и поставщика?
5. Какие характеристики производительности интересуют только поставщика услуг?
6. Между какими сторонами заключается соглашение об уровне обслуживания?
7. Предложите набор характеристик, которые вы хотели бы видеть в SLA, если вы хотите передавать через сеть трафик IP-телефонии.
8. Какой вид представления информации используется для результатов измерений задержек пакетов?

9. В чем преимущество использования такой характеристики, как коэффициент вариации по сравнению с джиттером?
10. Какая составляющая не учитывается при определении времени оборота?
11. Может ли трафик передаваться с большими задержками, но без джиттера?
12. Назовите параметры пульсации. Все ли эти параметры являются независимыми?
13. Зависит ли средняя скорость потока от величины задержек пакетов?
14. Какая характеристика надежности транспортной услуги используется в краткосрочном периоде, а какая в среднесрочном?
15. Опишите два основных подхода к обеспечению надежности сети.
16. Сколько существует основных способов использования альтернативных маршрутов для повышения надежности передачи трафика? Каковы их достоинства и недостатки?
17. Назовите две составляющие информационной безопасности.
18. Объясните разницу между масштабируемостью и расширяемостью.
19. Два коммутатора для повышения надежности связаны двумя физическими каналами (рис. 6.7). Оцените объем потерянных данных при отказе канала для двух вариантов использования этих каналов в качестве альтернативных маршрутов:
    - по схеме, в которой сеть заранее находит и использует оба маршрута;
    - по схеме, в которой сеть заранее находит оба маршрута, однако использует только один.

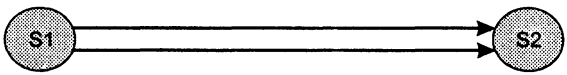


Рис. 6.7. Альтернативные маршруты

Протяженность каждого канала составляет 5000 км, скорость передачи данных — 155 Мбит/с, скорость распространения сигнала в канале равна 200 000 км/с. В обоих случаях коммутатор S2 обнаруживает факт отказа канала и переключается на резервный канал за 10 мс.

20. Оцените коэффициент использования канала, если данные по нему передаются с помощью протокола, использующего алгоритм простоя источника. Скорость передачи данных равна 100 Мбит/с, время оборота составляет 10 мс, пакеты не теряются и не искажаются. Размер пакета фиксирован и равен 1500 байт. Размером квитанции можно пренебречь.
21. Определите минимальный размер окна, который позволяет передавать пакеты по каналу без простоя источника. Скорость передачи данных равна 100 Мбит/с, время оборота составляет 10 мс, пакеты не теряются и не искажаются. Размер пакета фиксирован и равен 1500 байт. Размером квитанции можно пренебречь.

ГЛАВА 7 Методы обеспечения качества обслуживания

Методы обеспечения качества обслуживания (QoS) занимают сегодня одно из важнейших мест в арсенале технологий сетей с коммутацией пакетов, так как без их применения невозможна работа современных мультимедийных приложений, таких как IP-телефония, видео- и радиовещание, интерактивное дистанционное обучение и т. п. Эти методы оперируют параметрами, характеризующими скорость передачи данных, задержку пакетов и потерю пакетов.

Методы обеспечения качества обслуживания фокусируют внимание на влиянии очередей в коммуникационных устройствах на передачу трафика. В них используются различные алгоритмы управления очередями, резервирования и обратной связи, позволяющие снизить негативное влияние очередей до приемлемого для пользователей уровня.

Очереди являются неотъемлемым атрибутом сетей с коммутацией пакетов. Сам принцип работы таких сетей подразумевает наличие буфера у каждых входного и выходного интерфейсов коммутатора пакетов. Буферизация пакетов во время перегрузок представляет собой основной механизм поддержания пульсирующего трафика, обеспечивающий высокую производительность сетей этого типа. С другой стороны, очереди означают неопределенную задержку при передаче пакетов через сеть, а это главный источник проблем для чувствительного к задержкам трафика. Так как сегодня операторы пакетных сетей очень заинтересованы в передаче пульсирующего трафика, им необходимы средства обеспечения компромисса между стремлением предельно загрузить свою сеть и выполнением требований QoS одновременно для всех типов трафика.

В методах обеспечения качества обслуживания используются различные механизмы, направленные на уменьшение негативных последствий пребывания пакетов в очередях с сохранением в то же время положительной роли очередей. Набор механизмов достаточно широк, и в этой главе они рассматриваются достаточно подробно. Большинство из них учитывает и использует в своей работе факт существования в сети трафика различного типа.

К методам обеспечения качества обслуживания примыкают методы инжиниринга трафика. Эти методы служат для управления маршрутами передачи данных, что позволяет обеспечить сбалансированную загрузку всех ресурсов сети и исключить за счет этого переполнение очередей.

Приложения и качество обслуживания

Список ключевых слов: приложение с потоковым трафиком, постоянная битовая скорость, приложение с пульсирующим трафиком, переменная битовая скорость, асинхронное приложение, интерактивное приложение, изохронное приложение, приложение, сверхчувствительное к задержкам, приложение, чувствительное к потере данных, приложение, устойчивое к потере данных.

Требования к качеству обслуживания приложений разных типов

Современная тенденция конвергенции сетей различных типов, о чем рассказывалось в главе 1, привела к необходимости переноса сетью всех видов трафика, а не только традиционного для компьютерных сетей трафика приложений доступа к файлам и электронной почты.

В предыдущей главе мы перечислили различные характеристики QoS, с помощью которых оценивается качество передачи трафика через сеть. Характеристики QoS особенно важны в том случае, когда сеть передает одновременно трафик разного типа, например трафик веб-приложений и голосовой трафик. Это связано с тем, что различные типы трафика предъявляют разные требования к характеристикам QoS. Добиться одновременного соблюдения *всех* характеристик QoS для *всех* видов трафика очень сложно. Поэтому обычно используют следующий подход: классифицируют все виды трафика, существующие в сети, относя каждый из них к одному из распространенных типовых видов трафика, а затем добиваются одновременного выполнения определенного подмножества из набора требований для этих типов трафика.

К настоящему времени проделана большая работа по классификации трафика приложений. В качестве основных критериев классификации были приняты три характеристики трафика:

- относительная предсказуемость скорости передачи данных;
- чувствительность трафика к задержкам пакетов;
- чувствительность трафика к потерям и искажениям пакетов.

Предсказуемость скорости передачи данных

В отношении предсказуемости скорости передачи данных трафик приложений делится на два больших класса:

- потоковый трафик (stream);
- пульсирующий трафик (burst).

Приложения с потоковым трафиком порождают равномерный поток данных, который поступает в сеть с **постоянной битовой скоростью** (Constant Bit Rate, CBR). При использовании метода коммутации пакетов трафик таких приложений представляет собой последовательность пакетов одинакового размера (равного B бит), следующих друг за другом через один и тот же интервал времени T (рис. 7.1).

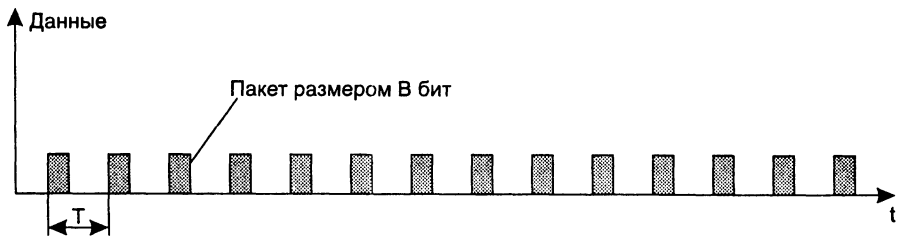


Рис. 7.1. Потоковый трафик

Постоянная скорость потокового трафика (CBR) может быть вычислена путем усреднения на одном периоде:

$$CBR = B/T \text{ бит/с}.$$

В общем случае постоянная скорость потокового трафика меньше номинальной максимальной битовой скорости протокола, с помощью которого передаются данные, так как между пакетами существуют паузы. Как будет показано в главе 12, максимальная скорость передачи данных с помощью протокола Ethernet составляет 9,76 Мбит/с (случай кадров максимальной длины), что меньше номинальной скорости этого протокола, равной 10 Мбит/с.

Приложения с пульсирующим трафиком отличаются высокой степенью непредсказуемости, когда периоды молчания сменяются пульсацией, в течение которой пакеты «плотно» следуют друг за другом. В результате трафик характеризуется **переменной битовой скоростью** (Variable Bit Rate, VBR), что иллюстрирует рис. 7.2. Так, при работе приложений файлового сервиса интенсивность трафика, генерируемого приложением, может падать до нуля, когда файлы не передаются, и повышаться до максимально доступной, ограниченной только возможностями сети, когда файловый сервер передает файл.

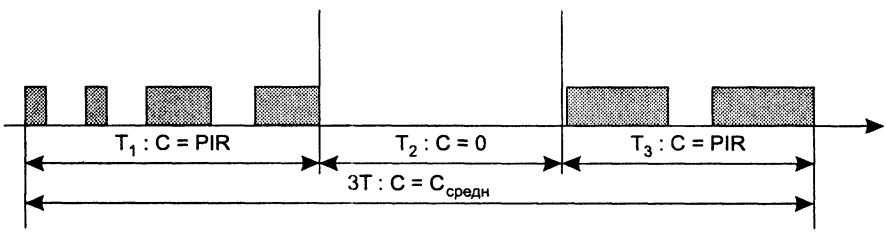


Рис. 7.2. Пульсирующий трафик

На рисунке показано три периода измерений T_1, T_2 и T_3. Для упрощения расчетов принято, что пиковые скорости на первом и третьем периодах равны между собой и равны PIR, а все три периода имеют одинаковую длительность T. Учитывая это, можно вычислить величину пульсации B, которая равна количеству битов, переданных на периоде пульсации:

$$B = PIR \times T.$$

Таким образом, величина пульсации для периодов T_1 и T_3 равна B, а на периоде T_2 — нулю.

Для приведенного примера можно подсчитать коэффициент пульсации. (Напомним, что он равен отношению пиковой скорости на каком-либо небольшом периоде времени к средней скорости трафика, измеренной на длительном периоде времени.) Так как пиковая скорость на периоде T_1 (или T_3) равна B/T, а средняя скорость на суммарном периоде $T_1 + T_2 + T_3$ равна 2B/3T, то коэффициент пульсации равен 3/2.

Практически любой трафик, даже трафик потоковых приложений, имеет ненулевой коэффициент пульсации. Просто значения коэффициентов пульсации у потокового и пульсирующего трафиков существенно различаются. У приложений с пульсирующим трафиком он обычно находится в пределах от 2:1 до 100:1, а у потоковых приложений близок к 1:1. В локальных сетях коэффициент пульсации обычно выше, чем в глобальных, поскольку на магистралях глобальных сетей трафик представляет собой сумму трафиков многих источников, что по закону больших чисел приводит к сглаживанию результирующего трафика.

Чувствительность трафика к задержкам пакетов

Еще один критерий классификации приложений по типу трафика — чувствительность к задержкам пакетов и их вариациям. Далее перечислены основные типы приложений в порядке повышения чувствительности к задержкам пакетов.

- **Асинхронные приложения.** Практически нет ограничений на время задержки (эластичный трафик). Пример такого приложения — электронная почта.
- **Интерактивные приложения.** Задержки могут быть замечены пользователями, но они не сказываются негативно на функциональности приложений. Пример — текстовый редактор, работающий с удаленным файлом.
- **Изохронные приложения.** Имеется порог чувствительности к вариациям задержек, при превышении которого резко снижается функциональность приложения. Пример — передача голоса, когда при превышении порога вариации задержек в 100–150 мс резко снижается качество воспроизводимого голоса.
- **Сверхчувствительные к задержкам приложения.** Задержка доставки данных сводит функциональность приложения к нулю. Пример — приложения, управляющие техническим объектом в реальном времени. При запаздывании управляющего сигнала на объекте может произойти авария.

Вообще говоря, интерактивность приложения всегда повышает его чувствительность к задержкам. Например, широковещательная рассылка аудиоинформации может выдерживать значительные задержки передачи пакетов (оставаясь чувствительным к вариациям задержек), а интерактивный телефонный или телевизионный разговор их не терпит, что хорошо заметно при трансляции разговора через спутник. Длительные паузы в разговоре вводят собеседников в заблуждение, часто они теряют терпение и начинают очередную фразу одновременно.

Наряду с приведенной выше классификацией, тонко дифференцирующей чувствительность приложений к задержкам и их вариациям, существует и более грубое деление приложений по этому же признаку на два класса — асинхронные и синхронные. К *асинхронным* относят те приложения, которые нечувствительны к задержкам передачи данных в очень широком диапазоне, вплоть до нескольких секунд, а все остальные приложения, на функциональность которых задержки влияют существенно, относят к *синхронным* приложениям.

Интерактивные приложения могут относиться как к асинхронным (например, текстовый редактор), так и к синхронным (например, видеоконференция).

Чувствительность трафика к потерям и искажениям пакетов

И, наконец, последним критерием классификации приложений является их чувствительность к потерям пакетов. Здесь обычно делят приложения на две группы.

- **Приложения, чувствительные к потере данных.** Практически все приложения, передающие алфавитно-цифровые данные (к которым относятся текстовые документы, коды программ, числовые массивы и т. п.), обладают высокой чувствительностью к потере отдельных, даже небольших, фрагментов данных. Такие потери часто ведут к полному обесцениванию остальной, успешно принятой информации. Например, отсутствие хотя бы одного байта в коде программы делает ее совершенно неработоспособной. Все традиционные сетевые приложения (файловый сервис, сервис баз данных, электронная почта и т. д.) относятся к этому типу приложений.
- **Приложения, устойчивые к потере данных.** К этому типу относятся многие приложения, передающие трафик с информацией об инерционных физических процессах. Устойчивость к потерям объясняется тем, что небольшое количество отсутствующих данных можно определить на основе принятых. Так, при потере одного пакета, несущего несколько последовательных замеров голоса, отсутствующие замеры при воспроизведении голоса могут быть заменены аппроксимацией на основе соседних значений. К такому типу относится большая часть приложений, работающих с мультимедийным трафиком (аудио- и видеоприложения). Однако устойчивость к потерям имеет свои пределы, поэтому процент потерянных пакетов не может быть большим (например, не более 1 %). Можно отметить также, что не любой мультимедийный трафик так устойчив к потерям данных, например, компрессированный

голос и видеоизображение очень чувствительны к потерям, поэтому относятся к первому типу приложений.

Классы приложений

Вообще говоря, между значениями трех характеристик качества обслуживания (относительная предсказуемость скорости передачи данных; чувствительность трафика к задержкам пакетов; чувствительность трафика к потерям и искажениям пакетов) нет строгой взаимосвязи. То есть приложение с равномерным потоком может быть как асинхронным, так и синхронным, а, например, синхронное приложение может быть как чувствительным, так и нечувствительным к потерям пакетов. Однако практика показывает, что из всего многообразия возможных сочетаний значений этих трех характеристик есть несколько таких, которые охватывают большую часть используемых сегодня приложений.

Например, следующее сочетание характеристик приложения «порождаемый трафик — равномерный поток, приложение изохронное, устойчивое к потерям» соответствует таким популярным приложениям, как IP-телефония, поддержка видеоконференций, аудиовещание через Интернет. Существуют и такие сочетания характеристик, для которых трудно привести пример приложения, например: «порождаемый трафик — равномерный поток, приложение асинхронное, чувствительное к потерям».

Устойчивых сочетаний характеристик, описывающих определенный класс приложений, существует не так уж много. Так, при стандартизации технологии ATM, которая изначально разрабатывалась для поддержания различных типов трафика, были определены 4 класса приложений: A, B, C и D. Для каждого класса рекомендуется использовать собственный набор характеристик QoS. Кроме того, для всех приложений, не включенных ни в один из этих классов, был определен класс X, в котором сочетание характеристик приложения может быть произвольным.

Классификация ATM является на сегодня наиболее детальной и общей, она не требует от нас знания технологий, используемых для передачи этих типов трафика, поэтому приведем ее здесь (табл. 7.1).

Таблица 7.1. Классы трафика

Класс трафика	Характеристики
A	Постоянная битовая скорость, чувствительность к задержкам, передача с установлением соединения (например, голосовой трафик, трафик телевизионного изображения). Параметры QoS: пиковая скорость передачи данных, задержка, джиттер
B	Переменная битовая скорость, чувствительность к задержкам, передача с установлением соединения (например, компрессированный голос, компрессированное видеоизображение). Параметры QoS: пиковая скорость передачи данных, пульсация, средняя скорость передачи данных, задержка, джиттер

продолжение

Таблица 7.1 (*продолжение*)

Класс трафика	Характеристики
C	Переменная битовая скорость, эластичность, передача с установлением соединения (например, трафик компьютерных сетей, в которых конечные узлы работают по протоколам с установлением соединений — frame relay, X.25, TCP). Параметры QoS: пиковая скорость передачи данных, пульсация, средняя скорость передачи данных
D	Переменная битовая скорость, эластичность, передача без установления соединения (например, трафик компьютерных сетей, в которых конечные узлы работают по протоколам без установления соединений — IP/UDP, Ethernet). Параметры QoS не определены
X	Тип трафика и его параметры определяются пользователем

Приведенная классификация приложений лежит в основе типовых требований к параметрам и механизмам обеспечения качества обслуживания в современных сетях.

Анализ очередей

Список ключевых слов: теория очередей, дисциплина первым пришел — первым обслужен, марковское, или пуассоновское, распределение, коэффициент использования.

Определить основные характеристики QoS и сформулировать требования к ним — это значит наполовину решить задачу. Пользователь формулирует свои требования к качеству обслуживания в виде некоторых предельных значений характеристик QoS, которые не должны быть превышены, например, он может указать, что предельное значение вариации задержки пакетов не должно превышать 50 мс с вероятностью 0,99.

Но как заставить сеть справиться с поставленной задачей? Какие меры нужно предпринимать, чтобы вариации задержек действительно не превысили эту величину? И как гарантировать пользователю, что средняя скорость передачи его потока через сеть будет соответствовать средней скорости входящего в сеть потока?

Долгое время эти вопросы не считались вопросами первостепенной важности. Сети с коммутацией пакетов были первоначально разработаны для передачи асинхронного трафика, так что с задержками можно было мириться. Однако сегодня, когда сети передачи данных начали переносить различные типы трафика, в том числе и чувствительного к задержкам, вопросы обеспечения показателей QoS вышли на первое место.

Для понимания механизмов поддержки QoS необходимо исследовать процесс образования очередей в сетевых устройствах и понять наиболее существенные факторы, влияющие на длину очереди.

Знакомство с моделью M/M/1

Существует ветвь прикладной математики, предметом которой являются процессы образования очередей. Эта дисциплина так и называется — **теория очередей**. Мы не будем углубляться в математические основы этой теории, приведем только некоторые ее выводы, существенные для рассматриваемой нами проблемы QoS.

На рис. 7.3 показана наиболее простая модель теории очередей, известная под названием M/M/1[1].

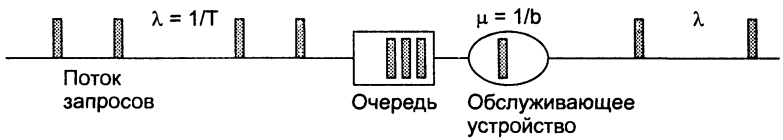


Рис. 7.3. Модель M/M/1

Основными элементами модели являются:

- входной поток абстрактных заявок на обслуживание;
- буфер;
- обслуживающее устройство;
- выходной поток обслуженных заявок.

Заявки поступают на вход буфера в случайные моменты времени. Если в момент поступления заявки буфер пуст и обслуживающее устройство свободно, то заявка сразу же передается в это устройство для обслуживания. Обслуживание также длиться случайное время.

Если в момент поступления заявки буфер пуст, но обслуживающее устройство занято обслуживанием ранее поступившей заявки, то заявка ожидает его завершения в буфере. Как только обслуживающее устройство завершает обслуживание очередной заявки, она передается на выход, а прибор выбирает из буфера следующую заявку (если буфер не пуст). Выходящие из обслуживающего устройства заявки образуют выходной поток. Буфер считается бесконечным, то есть заявки никогда не теряются из-за того, что исчерпана емкость буфера.

Если прибывшая заявка застает буфер не пустым, то она становится в очередь и ожидает обслуживания. Заявки выбираются из очереди в порядке поступления, то есть соблюдается дисциплина обслуживания **первым пришел — первым обслужен** (First-In, First-Out, FIFO).

Теория очередей позволяет оценить для этой модели среднюю длину очереди и среднея время ожидания заявки в очереди в зависимости от характеристик входного потока и времени обслуживания.

---

[1] Здесь 1 означает, что моделируется одно обслуживающее устройство, первая буква M обозначает тип распределения интервалов поступления заявок (марковское распределение), вторая — тип распределения значений времени обслуживания (тоже марковское).

Будем считать, что нам известно, что среднее время между поступлениями заявок равно T. Это значит, что интенсивность поступления заявок, которая традиционно обозначается в теории очередей символом λ, равна

$$\lambda = 1/T \text{ заявок в секунду.}$$

Случайный процесс поступления заявок описывается в этой модели функцией распределения интервалов между поступлениями заявок. Для упрощения получения компактных аналитических результатов обычно считают, что эти интервалы описываются так называемым **марковским** распределением (другое название — **пуассоновское**), плотность которого показана на рис. 7.4. Из рисунка видно, что входной поток является существенно пульсирующим, так как есть ненулевая вероятность того, что интервал между заявками будет очень небольшим, близким к нулю, а также того, что он будет очень большим. Среднее отклонение интервалов также равно T, так что стандартное отклонение равно $T/T = 1$.

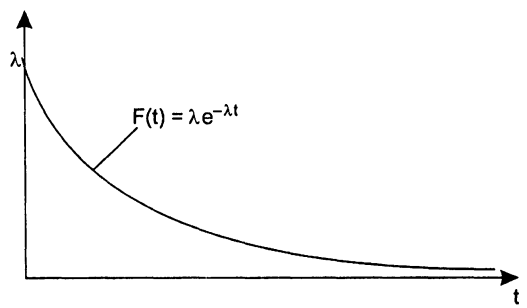


Рис. 7.4. Плотность распределения входного потока

Будем также считать, что среднее время обслуживания заявки равно b. Это означает, что обслуживающий прибор способен продвигать заявки на выход с интенсивностью $1/b = \mu$. Опять же для получения аналитического результата считают, что время обслуживания — это случайная величина с пуассоновской плотностью распределения.

Принятие таких предположений дает простой результат для среднего времени ожидания заявки в очереди, которое мы обозначим w:

$$w = \rho \frac{b}{1-\rho}. \qquad (1)$$

Здесь через ρ обозначено отношение λ/μ.

Параметр ρ называют **коэффициентом использования** (utilization) обслуживающего прибора. Для любого периода времени этот показатель равен отношению времени занятости обслуживающего прибора к величине этого периода.

Зависимость среднего времени ожидания заявки w от ρ иллюстрирует рис. 7.5. Как видно из поведения кривой, параметр ρ играет ключевую роль в образовании очереди. Если значение ρ близко к нулю, то и среднее время ожидания очень

близко к нулю. А это означает, что заявки почти никогда не ожидают обслуживания в буфере (в момент их прихода он оказывается пустым), а сразу попадают в обслуживающее устройство. И наоборот, если ρ приближается к 1, то время ожидания растет очень быстро и нелинейно (и в пределе равно бесконечности). Такое поведение очереди интуитивно понятно, ведь ρ — это отношение средней интенсивности входного потока к средней интенсивности его обслуживания. Чем ближе средние значения интервалов между пакетами к среднему времени обслуживания, тем сложнее обслуживающему устройству справляться с нагрузкой.

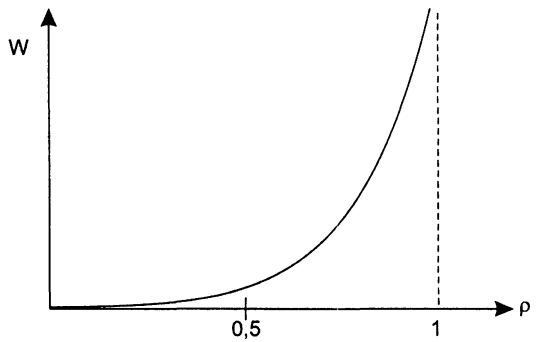


Рис. 7.5. Зависимость среднего времени ожидания заявки от коэффициента использования ресурса

M/M/1 как модель обработки пакетов

С помощью модели M/M/1 можно моделировать сеть с коммутацией пакетов (рис. 7.6).

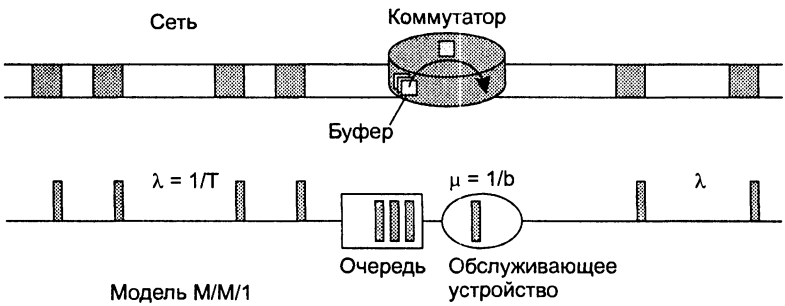


Рис. 7.6. Соответствие модели M/M/1 элементам сети

Пусть входной поток заявок играет роль потока пакетов, поступающих на вход интерфейса коммутатора, буфер модели соответствует буферу входного интерфейса коммутатора, а абстрактное обслуживающее устройство моделирует процессор, обрабатывающий пакеты и направляющий их на выходной интерфейс. Следовательно, среднее время обслуживания заявки соответствует среднему

времени продвижения пакета процессором коммутатора из входного буфера в выходной канал.

Понятно, что приведенная модель очень упрощенно описывает процессы, происходящие в сети. Она не учитывает многих особенностей обработки пакетов, например конечного размера буфера коммутатора, ненулевого времени поступления пакета в буфер и других. Тем не менее она очень полезна для понимания основных факторов, влияющих на величину очереди.

Сетевые инженеры хорошо знакомы с графиком, представленным на рис. 7.5. Они интерпретируют этот график как зависимость задержек в сети от ее загрузки. Параметр ρ модели соответствует коэффициенту использования сетевого ресурса, который участвует в передаче трафика, то есть интерфейса коммутатора, процессора коммутатора, канала или разделяемой среды.

В приведенном графике есть и нечто неожиданное. Трудно представить, что обслуживающее устройство (сетевой ресурс) практически перестает справляться со своими обязанностями, когда его коэффициент использования приближается к 1. Ведь в этом случае нагрузка не превышает его возможностей, а только приближается к этому пределу. Интуитивно не очень понятна также причина существования очередей при значениях ρ в окрестностях 0,5. Интенсивность обработки трафика вдвое превышает интенсивность нагрузки, а очереди существуют!

Такие парадоксальные на первый взгляд результаты характерны для систем, в которых протекают случайные процессы. Так как λ и μ — это средние значения интенсивностей потоков на больших промежутках времени, то на небольших промежутках времени они могут *существенно отклоняться* от этих значений. Очередь создается на тех промежутках, на которых интенсивность поступления пакетов намного превосходит интенсивность обслуживания.

Перегрузка ресурсов может привести к полной деградации сети, когда, несмотря на то что сеть передает пакеты, полезная скорость передачи данных оказывается равной нулю. Это происходит в том случае, если задержки доставки всех пакетов превосходят некоторый порог, и пакеты по тайм-ауту отбрасываются узлом назначения, как устаревшие. Если же протоколы, работающие в сети, используют надежные процедуры передачи данных на основе квитирования и повторной передачи утерянных пакетов, то процесс перегрузки будет нарастать лавинообразно.

Существует еще один важный параметр, оказывающий непосредственное влияние на образование очередей в сетях с коммутацией пакетов. Этим параметром является вариация интервалов входного потока пакетов, то есть пульсация входного трафика. Мы анализировали поведение модели теории очередей в предположении, что входной поток описывается пуассоновским распределением, которое имеет довольно большое стандартное отклонение вариации (напомним, что средняя вариация его равна T при среднем значении интервала T, а коэффициент вариации равен 1). А что будет, если вариация интервалов входного потока будет меньше? Или входной поток будет сверхпульсирующим?

К сожалению, модели теории очередей не дают для этих случаев простых аналитических зависимостей, подобных формуле (1). Поэтому для получения результатов приходится применять методы имитационного моделирования сетей или проводить измерения в реальной сети.

На рис. 7.7 показано семейство зависимостей w от ρ, полученных для разных значений коэффициента вариации CV входного потока. Имитационная модель учитывает фиксированную задержку в сети. Одна из кривых, у которой CV = 1, соответствует пуассоновскому входному потоку. Из рисунка видно, что чем меньше пульсирует входной поток (CV приближается к нулю), тем меньше проявляется эффект лавинообразного образования очереди при приближении коэффициента загрузки ресурса к 1. И наоборот, чем больше CV, тем раньше (при меньших значениях ρ) начинает этот эффект проявляться. Из поведения графиков на рисунке можно сделать два вывода.

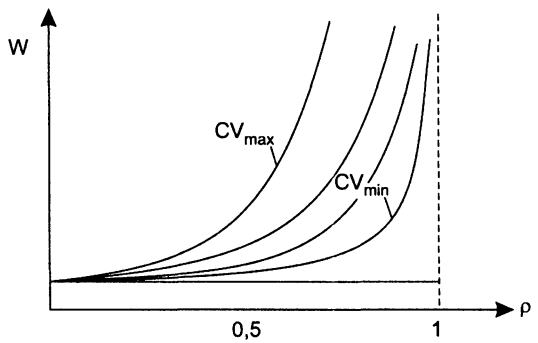


Рис. 7.7. Влияние степени пульсации потока на задержки

Для оценки значений задержек в очередях на коммутаторах сети недостаточно информации о коэффициенте загрузки ρ, необходимо также знать параметры пульсации трафика.

Для снижения уровня задержек нужно снижать значение ρ и сглаживать трафик.

Механизмы обеспечения качества обслуживания

Список ключевых слов: контроль перегрузки, коэффициент использования, работа в недогруженном режиме, нагруженная сеть, трафик реального времени, эластичный трафик, период пульсаций.

Работа в недогруженном режиме

Итак, через сеть одновременно протекает большое количество потоков. Каждый из них требует обслуживания в соответствии с определенными требованиями QoS. Каждый поток проходит на пути следования от одного конечного узла до другого через несколько коммутаторов, и в каждом из коммутаторов он проходит через две очереди — к процессору коммутатора и к выходному каналу коммутатора. Мы уже выяснили, что главным фактором, влияющим на величину задержек, а значит, и на качество обслуживания, является коэффициент использования

ресурса. Поэтому для обеспечения определенного качества обслуживания важно, чтобы коэффициент использования каждого ресурса (то есть процессоров и коммутаторов), который обслуживает поток на пути его следования, не превышал определенной величины.

Самым простым способом обеспечения требований QoS для всех потоков является *работа сети в недогруженном режиме*, когда все коммутаторы и каналы работают на 20–30 % от своей максимальной производительности.

Однако это сводит «на нет» основное достоинство сети с коммутацией пакетов, а именно ее высокую производительность при передаче пульсирующего трафика.

Введение разных классов обслуживания

Единственно приемлемым для практики является обеспечение качества обслуживания в *нагруженной сети*. Для упрощения понимания будем пока делить все потоки на два класса — чувствительный к задержкам (трафик реального времени, например голосовой) и эластичный, допускающий большие задержки, но чувствительный к потерям данных.

Мы точно не знаем зависимость задержек от коэффициента использования ресурса, но знаем общий вид этой зависимости. Если мы обеспечим для чувствительного к задержкам трафика коэффициент загрузки каждого ресурса не более 0,2, то, очевидно, что задержки в каждой очереди будут небольшими и скорее всего приемлемыми для многих типов приложений этого класса. Для эластичного трафика, слабо чувствительного к задержкам, можно допустить более высокий коэффициент загрузки, но не более 0,9. Для того чтобы пакеты этого класса не терялись, нужно предусмотреть для них буферную память, достаточную для хранения всех пакетов периода пульсации. Эффект от такого распределения загрузки ресурса иллюстрирует рис. 7.8.

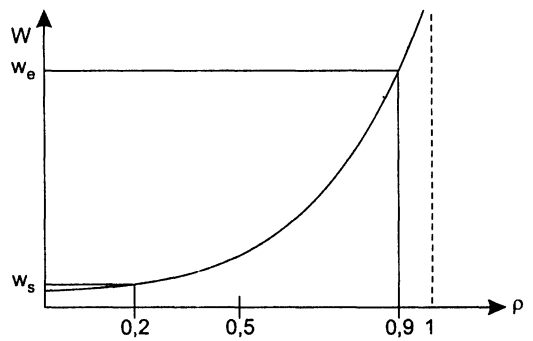


Рис. 7.8. Обслуживание эластичного и чувствительного к задержкам трафика

Задержки чувствительного к задержкам трафика равны w_s, а задержки эластичного трафика — w_e.

Чтобы добиться различных коэффициентов использования ресурсов для разных классов трафика, нужно в каждом коммутаторе для каждого ресурса поддержи-

вать две разные очереди. Алгоритм выборки пакетов из очередей должен отдавать предпочтение очереди чувствительных к задержкам пакетов. Если бы все пакеты первой очереди обслуживались приоритетно, а пакеты второй очереди — только тогда, когда первая очередь пуста, то для трафика первой очереди трафик второй очереди фактически перестал бы существовать. Поэтому если отношение средней интенсивности приоритетного трафика λ_1 к производительности ресурса μ равно 0,2, то и коэффициент загрузки для него равен 0,2. А вот для эластичного трафика, пакеты которого всегда ждут обслуживания приоритетных пакетов, коэффициент загрузки подсчитывается по-другому. Если средняя интенсивность эластичного трафика равна λ_2, то для него ресурс будет загружен на $(\lambda_1 + \lambda_2)/\mu$. Так что если мы хотим, чтобы для эластичного трафика коэффициент загрузки составлял 0,9, то его интенсивность должна находиться из соотношения $\lambda_2/\mu = 0{,}7$.

Основная идея, лежащая в основе всех методов поддержания характеристик QoS, заключается в следующем: *общая производительность каждого ресурса должна быть разделена между разными классами трафика неравномерно.*

Можно ввести более чем два класса обслуживания и стараться, чтобы каждый класс работал на своей части кривой задержек. Если такая задача решена, то можно обеспечить улучшение характеристик QoS за счет других методов, например снижая пульсацию трафика. Осталось выяснить, каким образом можно обеспечить такие условия для разных классов трафика в каждом узле сети.

Эта задача решается на протяжении всего времени существования пакетных сетей, то есть уже более тридцати лет. Долгое время пакетные сети передавали только эластичный трафик, поэтому основными требованиями QoS были минимизация потерь пакетов и удержание коэффициента нагрузки каждого ресурса сети не выше 0,9. Методы, решающие эту задачу, носят название методов **контроля перегрузки**.

С появлением в начале 90-х чувствительного к задержкам трафика ситуация усложнилась и начались поиски новых методов. Собственно, термин «качество обслуживания» появился именно в это время, отражая более детальные и дифференцированные требования разных типов трафика к сети.

Алгоритмы управления очередями

Список ключевых слов: алгоритм FIFO, алгоритм приоритетного обслуживания, приоритет, классификация трафика, точка классификации трафика, размер буфера, класс трафика, гранулярность, агрегат трафика, алгоритм взвешенных очередей, взвешенное справедливое обслуживание, комбинированные алгоритмы обслуживания очередей, средства профилирования трафика.

Алгоритмы управления очередями нужны для работы в периоды временных перегрузок, когда сетевое устройство не справляется с передачей пакетов на выходной интерфейс в том темпе, в котором они поступают. Если причиной перегрузки является недостаточная производительность процессорного блока сетевого устройства, то необработанные пакеты временно накапливаются во входной оче-

реди соответствующего входного интерфейса. Очередей к входному интерфейсу может быть несколько, если мы дифференцируем запросы на обслуживание по нескольким классам. В том же случае, когда причина перегрузки заключается в ограниченной пропускной способности выходного интерфейса, пакеты временно сохраняются в выходной очереди (или очередях) этого интерфейса.

Алгоритм FIFO

В традиционном **алгоритме FIFO** в случае перегрузки все пакеты помещаются в *одну* общую очередь и выбираются из нее в том порядке, в котором поступили. Во всех устройствах с коммутацией пакетов алгоритм FIFO используется по умолчанию. Достоинствами его является простота реализации и отсутствие потребности в конфигурировании. Однако ему присущ и коренной недостаток — *невозможность дифференцированной обработки пакетов различных потоков*. Все пакеты стоят в общей очереди на равных основаниях. Вместе оказываются и пакеты чувствительного к задержкам голосового трафика, и пакеты нечувствительного к задержкам, но очень интенсивного трафика резервного копирования, длительные пульсации которого могут надолго задержать голосовой пакет.

Приоритетное обслуживание

Алгоритмы приоритетного обслуживания очень популярны во многих областях вычислительной техники, в частности в операционных системах, когда одним приложениям нужно отдать предпочтение перед другими при обработке их в мультипрограммной смеси. Применяются эти алгоритмы и для преимущественной по сравнению с другими обработки одного класса трафика.

Механизм приоритетного обслуживания основан на разделении всего сетевого трафика на небольшое количество классов и последующего назначения каждому классу некоторого числового признака — **приоритета**.

Классификация трафика представляет собой отдельную задачу. Пакеты могут разбиваться на приоритетные классы на основании различных признаков: адреса назначения, адреса источника, идентификатора приложения, генерирующего этот трафик, любых других комбинаций признаков, которые содержатся в заголовках пакетов. Правила классификации пакетов представляют собой часть политики администрирования сети.

Точка классификации трафика может размещаться в каждом коммуникационном устройстве. Более масштабируемое решение — размещение функций классификации трафика в одном или нескольких устройствах, расположенных на границе сети (например, в коммутаторах корпоративной сети, к которым подключаются компьютеры пользователей, или во входных маршрутизаторах сети поставщика услуг). В этом случае необходимо специальное поле в пакете, в котором можно запомнить назначенное значение приоритета, чтобы им могли воспользоваться остальные сетевые устройства, обрабатывающие трафик после классифицирующего устройства. Такое поле имеется в заголовке многих протоколов. В тех же случаях, когда специального поля приоритета в заголовке нет, разрабатывается

дополнительный протокол, который вводит новый заголовок с таким полем (так произошло, например, с протоколом Ethernet).

Приоритеты могут назначаться не только коммутатором или маршрутизатором, но и приложением на узле-отправителе. Необходимо также учитывать, что если в сети отсутствует централизованная политика назначения приоритетов, каждое сетевое устройство может не согласиться с приоритетом, назначенным данному пакету в другой точке сети. В этом случае оно перепишет значение приоритета в соответствии с локальной политикой, принятой непосредственно на данном устройстве.

В сетевом устройстве, поддерживающем приоритетное обслуживание, имеется *несколько* очередей (буферов), по одной для каждого приоритетного класса. Пакет, поступивший в период перегрузок, помещается в очередь, соответствующую его приоритетному классу[1]. На рис. 7.9 приведен пример использования четырех приоритетных очередей с высоким, средним, нормальным и низким приоритетом. До тех пор пока из более приоритетной очереди не будут выбраны все имеющиеся в ней пакеты, устройство не переходит к обработке следующей, менее приоритетной очереди. Поэтому пакеты с низким приоритетом обрабатываются только тогда, когда пусты все вышестоящие очереди: с высоким, средним и нормальным приоритетами.

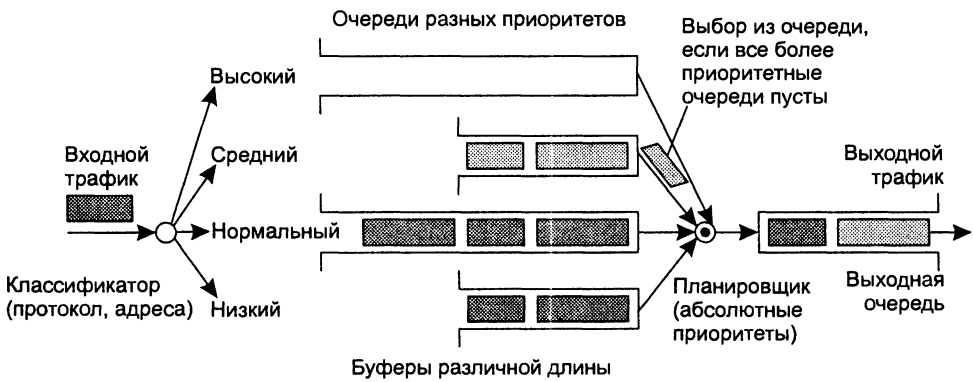


Рис. 7.9. Приоритетные очереди

Размер буфера сетевого устройства определяет максимальную длину очереди ожидающих обслуживания пакетов, если пакет поступает при заполненном буфере, то он просто отбрасывается. Обычно по умолчанию всем приоритетным очередям отводятся одинаковые буферы, но многие устройства разрешают администратору назначать каждой очереди буфер индивидуального размера. Размер буфера определяется в идеальном случае таким образом, чтобы его хватало с некоторым запасом для хранения очереди среднестатистической длины. Однако установить это значение достаточно сложно, так как оно изменяется в зависимости

[1] Иногда несколько очередей изображают в виде одной очереди, в которой находятся заявки различных классов. Если заявки выбираются из очереди в соответствии с их приоритетами, то это просто другое представление одного и того же механизма.

от нагрузки сети, поэтому требуется постоянное и длительное наблюдение за работой сети. В общем случае, чем выше значимость трафика для предприятия, чем больше его интенсивность и пульсации, тем больший размер буфера требуется этому трафику. В примере, приведенном на рис. 7.9, для трафика высшего и нормального приоритета выбраны большие размеры буферов, а для остальных двух классов — меньшие. Мотивы принятого решения для высшего приоритета очевидны, а трафик нормального приоритета имеет, очевидно, высокую интенсивность и значительный коэффициент пульсаций.

Приоритетное обслуживание очередей обеспечивает высокое качество обслуживания для пакетов из самой приоритетной очереди. Если средняя интенсивность их поступления в устройство не превосходит пропускной способности выходного интерфейса (и производительности внутренних продвигающих блоков самого устройства), то пакеты высшего приоритета всегда получают ту пропускную способность, которая им нужна. Уровень задержек высокоприоритетных пакетов также минимален. Однако он не нулевой и зависит в основном от характеристик потока этих пакетов — чем выше пульсации потока и его интенсивность, тем вероятнее возникновение очереди, образованной пакетами данного высокоприоритетного потока. Трафик всех остальных приоритетных классов почти прозрачен для пакетов высшего приоритета. Слово «почти» относится к ситуации, когда высокоприоритетный пакет вынужден ждать завершения обслуживания низкоприоритетного пакета, если его приход совпадает по времени с началом продвижения низкоприоритетного пакета на выходной интерфейс.

Что же касается остальных приоритетных классов, то качество их обслуживания будет ниже, чем у пакетов самого высокого приоритета, причем уровень снижения может быть очень существенным. Если коэффициент нагрузки выходного интерфейса, определяемый только трафиком высшего приоритетного класса, приближается в какой-то период времени к единице, то трафик остальных классов просто на это время замораживается. Поэтому приоритетное обслуживание обычно применяется для класса трафика, чувствительного к задержкам, имеющего небольшую интенсивность. При таких условиях обслуживание этого класса не слишком ущемляет обслуживание остального трафика. Например, голосовой трафик чувствителен к задержкам, но его интенсивность обычно не превышает 8–16 Кбит/с, так что при назначении ему высшего приоритета ущерб остальным классам трафика будет не очень значительным.

Однако в сети могут наблюдаться и другие ситуации. Например, видеотрафик также требует приоритетного обслуживания, но имеет гораздо более высокую интенсивность. Для таких случаев разработаны алгоритмы обслуживания очередей, дающие низкоприоритетному трафику некоторые гарантии даже в периоды повышения интенсивности высокоприоритетного трафика.

Внимательный читатель, очевидно, уже обратил внимание на то, что при описании работы приоритетных очередей фигурировали не отдельные потоки, а **классы трафика**. Это важная особенность, которая относится не только к приоритетным алгоритмам, но и ко многим другим механизмам поддержания качества обслуживания.

Сеть может обслуживать трафик с различной степенью **гранулярности**. Отдельный поток представляет собой минимальную единицу обслуживания, которую принимают во внимание механизмы обеспечения заданных параметров QoS. Если мы обеспечиваем каждому потоку собственные параметры QoS, то это поддержание качества обслуживания на уровне потоков. Если мы объединяем несколько потоков в единый поток и перестаем различать отдельные потоки при обеспечении параметров QoS, то это поддержание качества обслуживания на уровне классов трафика. Такие классы также называют **агрегатами** трафика.

ВНИМАНИЕ

Для того чтобы потоки можно было объединить в агрегат, нужно, чтобы они предъявляли одинаковые требования к качеству обслуживания и имели общие точки входа в сеть и выхода из сети.

Взвешенные очереди

Алгоритм взвешенных очередей разработан для того, чтобы можно было предоставить всем классам трафика определенный минимум пропускной способности или гарантировать некоторые требования к задержкам. Под *весом* данного класса понимается процент предоставляемой классу трафика пропускной способности от полной пропускной способности выходного интерфейса.

При взвешенном обслуживании так же, как при приоритетном, трафик делится на несколько классов, и для каждого класса ведется отдельная очередь пакетов. Но с каждой очередью связывается *не приоритет, а процент пропускной способности* ресурса, гарантируемый данному классу трафика при перегрузках этого ресурса. Для входного потока таким ресурсом является процессор, а для выходного потока (после выполнения коммутации) — выходной интерфейс.

Пример

Показанное на рис. 7.10 устройство для 5 классов трафика поддерживает 5 очередей к выходному интерфейсу коммутатора. Этим очередям при перегрузках выделяется соответственно 10 %, 10 %, 30 %, 20 % и 30 % пропускной способности выходного интерфейса.

Достигается поставленная цель тем, что очереди обслуживаются последовательно и циклически, и в каждом цикле обслуживания из каждой очереди выбирается такое число байтов, которое соответствует весу данной очереди. Например, если цикл просмотра очередей в рассматриваемом примере равен одной секунде, а скорость выходного интерфейса составляет 100 Мбит/с, то при перегрузках в каждом цикле первой очереди уделяется 10 % времени, то есть 100 мс и выбирается 10 Мбит данных, из второй — тоже 10 Мбит, из третьей — 30 Мбит, из четвертой — 20 Мбит, из пятой — 30 Мбит.

В результате каждому классу трафика достается гарантированный минимум пропускной способности, что во многих случаях является более желательным результатом, чем подавление низкоприоритетных классов высокоприоритетным.

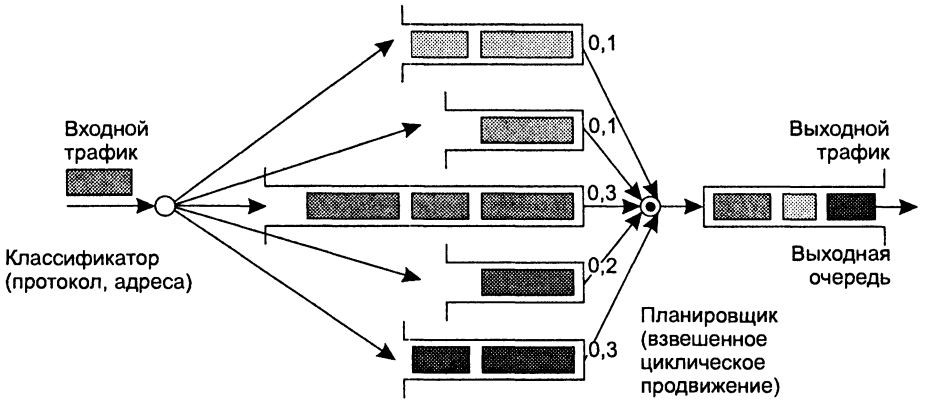


Рис. 7.10. Взвешенные очереди

Так как данные выбираются из очереди пакетами, а не битами, то реальное распределение пропускной способности между классами трафика всегда немного отличается от планируемого. Например, вместо 10 % первый класс трафика может получить при перегрузках 9 или 12 %. Чем больше время цикла, тем точнее соблюдаются требуемые пропорции между классами трафика, так как из каждой очереди выбирается большое число пакетов и влияние размера каждого пакета усредняется.

С другой стороны, длительный цикл приводит к большим задержкам передачи пакетов. Например, при выбранном нами для примера цикле в одну секунду задержка может составить одну секунду и больше — ведь арбитр возвращается к каждой очереди не чаще, чем раз в секунду, кроме того, в очереди может находиться более одного пакета. Поэтому при выборе времени цикла нужно обеспечить баланс между точностью соблюдения пропорций пропускной способности и стремлением к уменьшению задержки.

Для нашего примера время цикла в 1000 мкс является примером такого баланса. С одной стороны, оно обеспечивает обслуживание очереди каждого класса каждые 1000 мкс, а значит — более низкий уровень задержек. С другой стороны, этого времени достаточно, чтобы выбрать из каждой очереди в среднем по несколько пакетов (первой очереди в нашем примере будет отводиться 100 мкс, что достаточно для передачи в выходной канал одного пакета Fast Ethernet или десяти пакетов Gigabit Ethernet).

На уровень задержек и вариации задержек пакетов для некоторого класса трафика при взвешенном обслуживании в значительной степени влияет коэффициент использования. В этом случае коэффициент подсчитывается как отношение интенсивности входного трафика класса к пропускной способности, выделенной этому классу в соответствии с его весом. Например, если мы выделили первой очереди 10 % от общей пропускной способности выходного интерфейса, то есть 10 Мбит/с, а средняя интенсивность потока, который попадает в эту очередь, равна 3 Мбит/с, то коэффициент использования для этого потока составит 3/10 = 0,3. Зависимость на рис. 7.5 показывает, что задержки при таком значе-

нии коэффициента использования будут незначительными. Если бы интенсивность входного потока этой очереди была 9 Мбит/с, то очереди были бы значительными, а при превышении предела 10 Мбит/с часть пакетов потока постоянно бы отбрасывалась из-за переполнения очереди.

Качественное поведение очереди и, соответственно, задержек здесь выглядит примерно так же, как и в случае очереди FIFO — чем меньше коэффициент загрузки, тем меньше средняя длина очереди и тем меньше задержки.

Как и для приоритетного обслуживания, при взвешенном обслуживании администратор может назначать разным классам очередей буферы разных размеров. Уменьшение размеров буферов для очередей ведет к росту числа потерь пакетов при перегрузках, но зато снижает время ожидания для тех пакетов, которые не были отброшены и попали в очередь.

Существует также такой вид взвешенного обслуживания, как **взвешенное справедливое обслуживание** (Weighted Fair Queuing, WFQ). В случае подобного обслуживания пропускная способность ресурса делится между всеми потоками поровну, то есть «справедливо».

ВНИМАНИЕ

Взвешенное обслуживание обеспечивает требуемые соотношения между интенсивностями трафика различных очередей только в периоды перегрузок, когда каждая очередь постоянно заполнена. Если же какая-нибудь из очередей пуста (то есть для трафика данного класса текущий период не является периодом перегрузки), то при просмотре очередей она пропускается, и ее время обслуживания распределяется между остальными очередями в соответствии с их весом. Поэтому в отдельные периоды трафик определенного класса может обладать большей интенсивностью, чем соответствующий процент от пропускной способности выходного интерфейса.

Комбинированные алгоритмы обслуживания очередей

Каждый из описанных подходов имеет свои достоинства и недостатки. Приоритетное обслуживание, обеспечивая минимальный уровень задержек для очереди наивысшего приоритета, не дает никаких гарантий в отношении средней пропускной способности для трафика очередей более низких приоритетов. Взвешенное обслуживание обеспечивает заданное распределение средней пропускной способности, но не учитывает требований к задержкам.

Существуют **комбинированные алгоритмы обслуживания очередей.** В наиболее популярном алгоритме подобного рода поддерживается одна приоритетная очередь при обслуживании остальных очередей в соответствии с взвешенным алгоритмом. Обычно приоритетная очередь используется для чувствительного к задержкам трафика, а остальные — для эластичного трафика нескольких классов. Каждый класс эластичного трафика получает некоторый минимум пропускной способности при перегрузках. Этот минимум вычисляется как процент от пропускной способности, оставшейся от приоритетного трафика. Очевидно, что нужно как-то ограничить приоритетный трафик, чтобы он не поглощал всю пропускную

способность ресурса. Обычно это делается **средствами профилирования трафика**, которые рассматриваются далее.

Обратная связь

Список ключевых слов: механизм контроля перегрузки, механизм предотвращения перегрузки, контроль потока, признак перегрузки, максимальная скорость передачи, кредит.

Назначение

Алгоритмы управления очередями не предотвращают перегрузок, а лишь некоторым «справедливым» образом в условиях дефицита перераспределяют ресурсы между различными потоками или классами трафика. Алгоритмы управления очередями относятся к **механизмам контроля перегрузок**, которые начинают работать, когда сеть уже перегружена.

Существует другой класс средств, которые носят название **механизмов предотвращения перегрузок**. Очевидно, что предотвратить перегрузку сети можно в том случае, когда суммарная интенсивность всех потоков, передаваемых каждым интерфейсом каждого коммутатора сети, меньше пропускной способности этого интерфейса. Добиться этого можно двумя способами — увеличивая пропускную способность интерфейса или уменьшая интенсивности потоков. Первый вариант относится к средствам проектирования и планирования сети и поэтому здесь не рассматривается.

Второй вариант — уменьшение интенсивности потоков — можно реализовать также двумя принципиально различными способами. Первый способ основан на использовании механизма *обратной связи*, с помощью которого перегруженный узел сети, реагируя на перегрузку, просит предыдущие узлы, расположенные вдоль маршрута следования потока (или потоков, принадлежащих к одному классу), временно снизить скорость трафика. После того как перегрузка в данном узле исчезнет, он посылает другое сообщение, разрешающее повысить скорость передачи данных. Другой способ основан на предварительном *резервировании* пропускной способности для потоков, протекающих через сеть. Для этого ему необходима предварительная информация об интенсивностях потоков. Принципы резервирования ресурсов мы рассмотрим позже, а сейчас остановимся на механизмах обратной связи.

Участники обратной связи

Существует несколько механизмов обратной связи. Они отличаются информацией, которая передается по обратной связи, а также тем, какой тип узла генерирует эту информацию и кто реагирует на эту информацию — конечный узел (компьютер) или промежуточный (коммутатор или маршрутизатор).

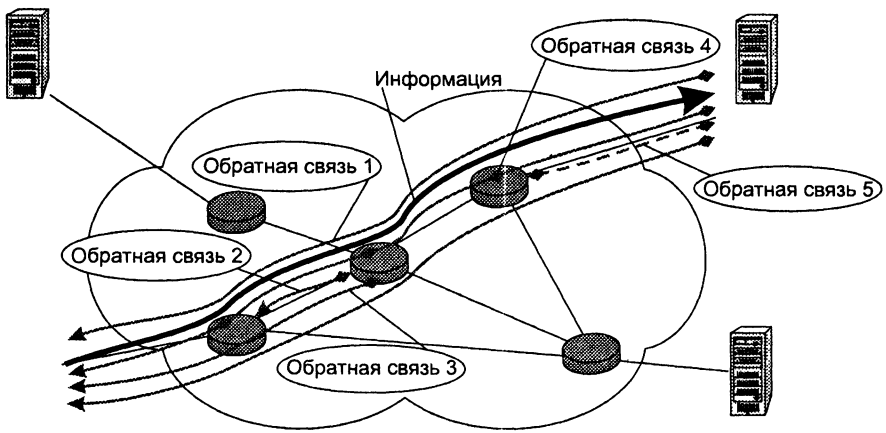


Рис. 7.11. Участники обратной связи

На рис. 7.11 показаны различные варианты организации обратной связи.

Обратная связь 1 организована между двумя конечными узлами сети. Это наиболее радикальный способ снижения нагрузки на сеть, так как только конечный узел может снизить скорость поступления информации в сеть. Однако этот вид обратной связи не относят к методам контроля перегрузок, так как его назначение — борьба с перегрузками узла назначения, а не с перегрузками сетевых устройств. Принципиально эта та же самая проблема, так как она является следствием временного превышения скорости поступления пакетов в ресурс над скоростью обработки этих пакетов. Только ресурсом в данном случае выступает не коммутатор сети, а конечный узел. Но традиционно за этим видом обратной связи закрепилось собственное название — **контроль потока**. Устройства сети не принимают участие в работе этого вида механизма обратной связи, они только передают соответствующие сообщения между конечными узлами. Несмотря на разные названия, в методах контроля перегрузки и контроля потока используются общие механизмы.

При организации обратной связи важно учитывать время передачи информации по сети. В высокоскоростных глобальных сетях за время передачи сообщения о перегрузке узла назначения узел-источник может успеть передать через сеть тысячи пакетов, так что перегрузка не будет ликвидирована вовремя. Из теории автоматического управления известно, что задержки в контуре обратной связи могут приводить ко многим нежелательным эффектам, прямо противоположным первоначальным целям. Например, в системе могут начаться колебательные процессы, и она никогда не сможет прийти в равновесное состояние. Подобные явления наблюдались на ранней стадии развития Интернета, когда из-за несовершенства алгоритмов обратной связи и алгоритмов маршрутизации в нем возникали участки перегрузок, которые периодически перемещались по сети. Причина такой проблемы интуитивно понятна — задержка в контуре обратной связи приводит к тому, что регулирующий элемент получает устаревшую информацию о состоянии регулируемого элемента. В данном случае узел-источник получает информацию о состоянии очереди узла назначения с задержкой. Поэтому воз-

можны ситуации, когда узел-источник начинает снижать скорость передачи информации, когда в действительности очередь в узле назначения уже не существует, и, наоборот, повышать скорость передачи информации в тот момент, когда узел назначения начал испытывать перегрузку. Для борьбы с такими явлениями в контур обратной связи обычно вводится интегрирующий элемент, который на каждом шаге обрабатывает не только текущее сообщение обратной связи, но и несколько предыдущих сообщений, что позволяет учесть динамику изменения ситуации и реагировать адекватно.

Обратная связь 2 организована между двумя соседними коммутаторами. Коммутатор сообщает соседу, находящемуся выше по течению потока, что он испытывает перегрузку и его буфер заполнился до критической величины. Получив такое сообщение, сосед, расположенный выше по течению, должен снизить на некоторое время скорость передачи данных в направлении перегруженного коммутатора и тем самым решить проблему перегрузки. Это менее эффективное для сети в целом решение, так как поток будет продолжать течь от узла-источника с той же скоростью, что и раньше. Но для коммутатора, который испытывает перегрузку, это является хорошим выходом, так как он получает время для того, чтобы разгрузить переполнившуюся очередь. Правда, проблема переносится в коммутатор, расположенный выше по течению, в котором теперь может возникнуть перегрузка, так как он начинает передавать данные из своего буфера с меньшей скоростью. Достоинством такого метода является низкая задержка обратной связи, так как узлы являются соседями (если они, конечно, не соединены спутниковым каналом).

Обратная связь 3 организована между промежуточным коммутатором и узлом-источником. Сообщения обратной связи хотя и передаются несколькими коммутаторами сети в направлении узла-источника, но они на него не реагируют.

Обратная связь 4. Здесь, как и в случае обратной связи 1, сообщение о перегрузке порождается узлом назначения и передается узлу-источнику. Однако имеется и важное отличие: в данном случае каждый промежуточный коммутатор реагирует на это сообщение. Во-первых, он снижает скорость передачи данных в направлении узла назначения, во-вторых, он может изменить содержание сообщения. Например, если узел назначения просит снизить скорость до 30 Мбит/с, то промежуточный коммутатор может снизить эту величину до 20 Мбит/с, оценив состояние своего буфера. Кроме того, породить сообщение обратной связи может любой коммутатор сети, а не только узел назначения.

При описании различных вариантов организации обратной связи мы подразумевали, что сообщение о перегрузке идет в направлении, обратном направлению передачи пользовательской информации (собственно, поэтому этот механизм так и называется). Однако некоторые коммуникационные протоколы не предусматривают возможности генерации подобных сообщений промежуточными узлами. В таких условиях часто используется искусственный прием — передача сообщения о перегрузке узлу назначения, который преобразует его в сообщение обратной связи и отправляет в нужном направлении, то есть в направлении источника. Этот вариант показан на рисунке как *обратная связь 5.*

Информация обратной связи

В применяемых сегодня методах обратной связи используются следующие основные типы сообщений:

- признак перегрузки;
- максимальная скорость передачи;
- максимальный объем данных (кредит);
- косвенные признаки.

Признак перегрузки не говорит о степени перегруженности сети или узла, он только фиксирует факт наличия перегрузки. Реакция узла, получившего такое сообщение, может быть разной. В некоторых протоколах узел обязан прекратить передачу информации в определенном направлении до тех пор, пока не будет получено другое сообщение обратной связи, разрешающее продолжение передачи. В других протоколах узел ведет себя адаптивно, он снижает скорость на некоторую величину и ожидает реакции сети. Если сообщения с признаком перегрузки продолжают поступать, то он продолжает снижение скорости.

Во втором типе сообщений указывается **максимальная скорость передачи**, то есть порог скорости, который должен соблюдать источник или промежуточный узел, расположенный выше по течению потока. В этом случае обязательно нужно учитывать время передачи сообщения по сети, чтобы исключить колебательные процессы в сети и обеспечить нужную скорость реакции на перегрузку. Поэтому в территориальных сетях такой способ обычно реализуется силами всех коммутаторов сети (обратная связь 4 в нашем примере).

Сообщение «максимальный объем данных» используется в широко применяемом в пакетных сетях алгоритме скользящего окна (см. главу 6). Этот алгоритм позволяет не только обеспечивать надежную передачу данных, но и реализовать обратную связь для контроля потока между конечными узлами. Параметром, несущим информацию обратной связи, является «окно» — число, тесно связанное с текущим размером свободного пространства в буфере принимающего узла. Окно также называют **кредитом**, который принимающий дает передающему узлу. Передающий узел может с любой скоростью передать объем информации (или определенное количество пакетов, если окно измеряется в пакетах), соответствующий кредиту. Но если кредит исчерпан, то передающий узел не имеет права передавать информацию, пока не получит следующий кредит. При перегрузках принимающий узел уменьшает размер окна, тем самым снижая нагрузку. Если эффект перегрузки исчезает, то принимающий узел увеличивает размер окна. Недостатком этого алгоритма является то, что он работает только в протоколах с установлением соединения.

И, наконец, в некоторых случаях передающий узел определяет, что принимающий узел (или узлы) испытывает перегрузку, по некоторым *косвенным признакам* без получения сообщения обратной связи. Такими косвенными признаками могут быть факты потери пакетов. Для того чтобы протокол мог обнаруживать факты потерь пакетов, это должен быть протокол с установлением соединения. Тогда истечение тайм-аута или приход дубликата положительной квитанции косвенно свидетельствуют о том, что пакет потерян. Однако потеря пакета не

всегда свидетельствует о перегрузке сети. Перегрузка сети — это только одна из возможных причин потери пакета, другой причиной может быть ненадежная работа коммуникационных устройств (отказы оборудования, искажения данных из-за помех). Но так как реакция на перегрузки и ненадежную работу сети должна быть одинаковой и состоять в снижении скорости передачи, то неоднозначность причины потери пакета не является проблемой.

Примером протокола, использующего неявную информацию о перегрузках, является протокол TCP. Этот протокол с помощью явной информации обратной связи (о размере окна) осуществляет контроль потока, а с помощью неявной (потери пакетов, дубликаты квитанций) — контроль перегрузки.

Резервирование ресурсов

Список ключевых слов: механизм обслуживания очередей, протокол резервирования ресурсов, сигнальный протокол, профиль трафика, протокол установления виртуального канала, кондиционирование трафика, классификация трафика, профилирование трафика, формирование трафика.

Резервирование ресурсов и коммутация пакетов

Как уже было сказано выше, еще одним механизмом предотвращения перегрузки в сети, наряду с обратной связью, является резервирование ресурсов. Главная идея резервирования состоит в том, чтобы ограничить уровень перегрузок некоторой приемлемой величиной. Эта величина должна быть такой, чтобы алгоритмы контроля перегрузки, работающие в коммутаторах сети, справлялись с кратковременными перегрузками и без обратной связи обеспечивали требуемые значения характеристик QoS.

Резервирование ресурсов в сетях с коммутацией пакетов принципиально отличается от подобной процедуры в сетях с коммутацией каналов. В сетях с коммутацией каналов для каждого канала резервируется (выделяется) фиксированная доля пропускной способности линии связи (физического канала). Поток передается через сеть с постоянной скоростью, равной зарезервированной для него пропускной способности. При этом пропускная способность соединения всегда закреплена за этим потоком, она *не может* динамически перераспределяться среди других потоков. Предварительное резервирование является *неотъемлемым свойством* сети с коммутацией каналов.

В сетях с коммутацией пакетов резервирование *не является обязательным*. Иногда в определении метода коммутации пакетов отсутствие резервирования фигурирует как основное свойство этого типа сетей. Но и в тех случаях, когда резервирование в сетях с коммутацией пакетов выполняется, оно отличается от резервирования ресурсов в сетях с коммутацией каналов тем, что здесь учитывается пульсирующий характер трафика и возможность динамического перераспределения пропускной способности сети между потоками (агрегатами).

Резервирование заключается в том, что все сетевые устройства вдоль следования потока должны выделить этому потоку (агрегату) некоторую часть пропускной способности своих интерфейсов и производительности процессоров, равную средней требуемой скорости передачи данных потока. Поясним это на примере.

Пример

Предположим, что в исходном состоянии ресурсы сети, показанной на рис. 7.12, не были зарезервированы. Затем было решено выделить некоторые ресурсы сети потоку 1. Для этого необходимо знать, по крайней мере, такой параметр потока, как среднюю требуемую скорость передачи данных. Предположим, что эта скорость для потока 1 равна 15 Мбит/с, а пропускные способности всех каналов связи (а значит, и интерфейсов коммутаторов) равны 100 Мбит/с. Будем для упрощения считать, что каждый входной интерфейс оснащен собственным процессором, производительность которого превышает производительность данного интерфейса, так что процессор не может быть узким местом, и мы не будем принимать его в расчет при принятии решения о выделении ресурсов.

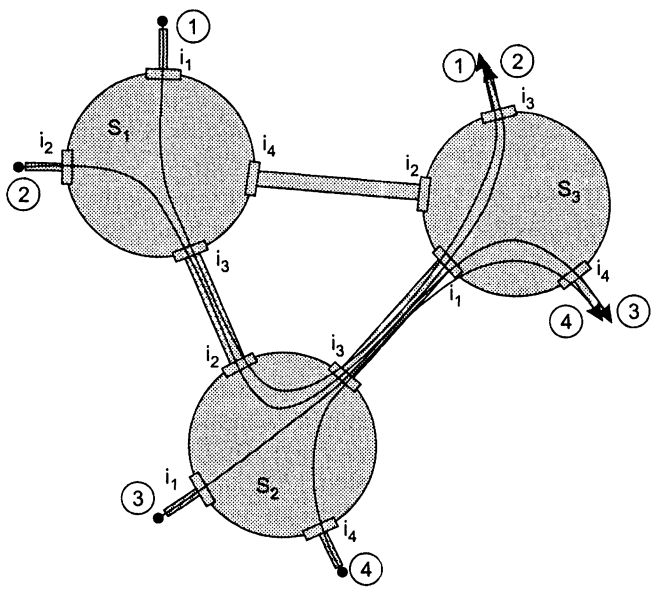


Рис. 7.12. Резервирование ресурсов в сетях с коммутацией пакетов

Поток 1 может быть принят на обслуживание, потому что все интерфейсы на его пути обладают достаточной для его обслуживания производительностью (15 < 100). Поэтому резервирование выполняется, и каждый интерфейс вдоль пути потока запоминает, что он уже выделил 15 Мбит/с своей производительности потоку 1.

Допустим, что после этого возникла потребность в резервировании ресурсов для потока 2, который обладает средней скоростью передачи данных 70 Мбит/с. Такое резервирование также может быть сделано, так как у всех интерфейсов вдоль маршрута потока 2 свободная (не зарезервированная) пропускная способность интерфейсов превышает 70 Мбит/с. У тех интерфейсов, через которые проходят как поток 1, так

и поток 2 (интерфейсы i_3 и i_1 коммутаторов S_2 и S_3 соответственно), остается 85 Мбит/с свободной пропускной способности, а у остальных интерфейсов — 100 Мбит/с. После резервирования у интерфейсов i_3 и i_1 остается по 15 Мбит/с свободной пропускной способности.

Также оказывается успешной попытка резервирования пропускной способности для потока 3, средняя скорость которого равна 10 Мбит/с. Однако резервирование для потока 4, средняя скорость которого 20 Мбит/с, оказывается невозможным, так как у интерфейсов i_3 и i_1 осталось только по 5 Мбит/с свободной пропускной способности.

Этот пример показывает, что сеть отказывается принять на обслуживание поток, если она не может гарантировать ему требуемый уровень качества обслуживания. Мы, конечно, упростили схему резервирования ресурсов. В действительности, сеть может гарантировать потоку не только соблюдение его средней скорости, о которой мы говорили в примере, но и обеспечить другие характеристики QoS, такие как максимальная задержка, максимальная вариация задержки и допустимый уровень потерь данных. Однако для этого сеть должна знать некоторые дополнительные параметры потока, например его максимальный уровень пульсации, чтобы зарезервировать необходимое пространство в буфере.

Свободная пропускная способность для чувствительного к задержкам трафика и для эластичного трафика должна при резервировании учитываться отдельно. Чтобы обеспечить для приоритетного трафика приемлемый уровень задержек и их вариаций, максимальная суммарная резервируемая пропускная способность не должна превышать 30–50 % от общей пропускной способности каждого ресурса. Для иллюстрации этого воспользуемся предыдущим примером. Пусть мы решили отвести чувствительному к задержкам трафику 30 % пропускной способности ресурсов. Тогда, если чувствительными к задержкам являются потоки 1 и 3, то резервирование для них возможно. Если же такими потоками являются потоки 1 и 2, то нет, так как суммарная средняя скорость этих потоков равна 85 Мбит/с, что больше чем 30 Мбит/с (30 % от 100 Мбит/с).

Если мы подразумеваем, что чувствительный к задержкам трафик будет обслуживаться в приоритетной очереди, то при резервировании пропускной способности для эластичного трафика нужно учитывать, что ему может быть выделена только та часть пропускной способности, которая осталась от чувствительного к задержкам трафика. Например, если потоки 1 и 3 являются чувствительными к задержкам и мы выделили им требуемую среднюю пропускную способность 30 Мбит/с, то для эластичных потоков остается только 70 Мбит/с свободой пропускной способности.

Что же меняется в сети после того, как в ней выполнено резервирование? Ничего принципиально нового. Просто сеть оказывается загруженной рациональным образом. В ней нет ресурсов, которые работают с перегрузкой. Механизмы организации очередей по-прежнему обеспечивают временную буферизацию пакетов в периоды пульсаций. Так как мы планировали загрузку ресурсов из расчета средних скоростей передачи данных, то на периодах пульсаций в течение некоторого ограниченного времени скорости потоков могут превышать средние скорости, так что механизмы борьбы с перегрузками по-прежнему нужны. Для обеспечения требуемых средних скоростей потоков на периодах перегрузок соответствующие потоки могут обслуживаться с помощью взвешенных очередей.

Главное преимущество метода коммутации пакетов также сохраняется: если некоторый поток не расходует отведенной ему пропускной способности, то она может быть использована для обслуживания другого потока. Нормальной практикой является резервирование пропускной способности только для части потоков, в то время как другие потоки обслуживаются без резервирования, получая обслуживание по возможности (с максимальными усилиями). Временно свободная пропускная способность может для таких потоков выделяться динамически, без нарушения взятых обязательств по обслуживанию потоков, для которых ресурсы зарезервированы.

Сеть с коммутацией каналов подобного перераспределения ресурсов выполнить не может, так как у нее в распоряжении нет независимо адресуемых единиц информации — пакетов!

Пример

Проиллюстрируем принципиальное отличие резервирования ресурсов в сетях с коммутацией пакетов и в сетях с коммутацией каналов на примере автомобильного трафика. Пусть в некотором городе решили обеспечить некоторые привилегии для движения машин скорой помощи. В ходе обсуждения этого проекта возникли две конкурирующие идеи его реализации. Первый вариант предусматривал выделение для автомобилей скорой помощи отдельной полосы на всех дорогах города, недоступной для другого транспорта ни при каких условиях, даже если в какой-то период времени машин скорой помощи на дороге нет. Во втором случае для машин скорой помощи также выделялась отдельная полоса, но в отсутствии привилегированных машин по ней разрешалось двигаться и другому транспорту. В случае же появления машины скорой помощи автомобили, занимающие выделенную полосу, обязаны были ее освободить. Нетрудно заметить, что первый вариант соответствует принципу резервирования в сетях с коммутацией каналов — пропускная способность выделенной полосы монопольно используется автомобилями скорой помощи и не может быть перераспределена даже тогда, когда она им не нужна. Второй вариант является аналогией резервирования в сетях с коммутацией пакетов. Пропускная способность дороги здесь используется более эффективно, но для потока автомобилей скорой помощи такой вариант менее благоприятен, так как при необходимости освобождения полосы возникают помехи, создаваемые непривилегированными машинами.

Возвращаясь от автомобильного трафика к сетям с коммутацией пакетов, следует отметить: для того чтобы соблюсти гарантии обслуживания каждого потока, описанной схемы резервирования недостаточно. Мы предположили, что точно знаем среднюю пропускную способность и параметры пульсаций потоков. На практике такие сведения не всегда бывают достоверными. А что случится, если поток будет поступать в сеть со скоростью, превышающей ту, которую мы учитывали при резервировании? И еще один немаловажный вопрос остается открытым — как обеспечить автоматическое резервирование пропускной способности вдоль маршрута следования потока?

Для решения поставленных задач в сети необходима система обеспечения качества обслуживания, в которую помимо механизмов управления очередями входят некоторые дополнительные механизмы.

Системы обеспечения качества обслуживания, основанные на резервировании

Система обеспечения качества обслуживания имеет распределенный характер, так как ее элементы должны присутствовать на всех сетевых устройствах, продвигающих пакеты: коммутаторах, маршрутизаторах, серверах. С другой стороны, работу отдельных сетевых устройств по поддержанию характеристик QoS нужно координировать, чтобы качество обслуживания было однородным вдоль всего пути, по которому следуют пакеты потока. Поэтому служба QoS должна включать также элементы централизованного управления, с помощью которых администратор сети мог бы согласованно конфигурировать механизмы поддержания характеристик QoS в отдельных устройствах сети.

Система обеспечения качества обслуживания, базирующаяся на резервировании ресурсов, состоит из подсистем нескольких типов (рис. 7.13):

- механизмов обслуживания очередей;
- протокола резервирования ресурсов;
- механизмов кондиционирования трафика.

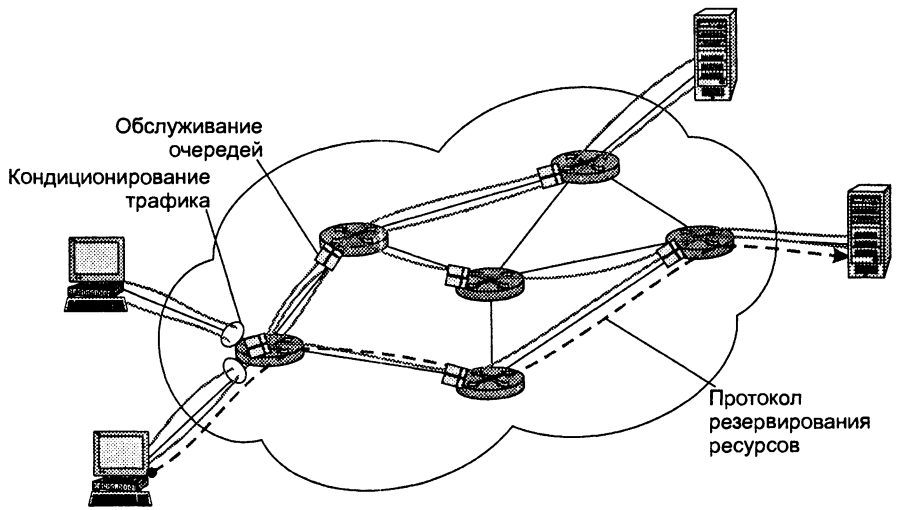


Рис. 7.13. Архитектура системы обеспечения качества обслуживания, основанная на резервировании ресурсов

Механизмы обслуживания очередей используются в периоды временных перегрузок. При этом обычно применяются комбинации приоритетной очереди с очередями с взвешенным обслуживанием.

Протокол резервирования ресурсов нужен для автоматизации процедуры резервирования на всем пути следования некоторого потока, то есть «из конца в конец». Протокол резервирования является аналогом протоколов установления соединения в сетях с коммутацией каналов, поэтому он иногда называется **сиг-**

нальным протоколом, в соответствии с терминологией, принятой для этого типа сетей.

Сообщения протокола резервирования ресурсов делают два «прохода» по сети. Сначала источник генерирует сообщение, которое проходит сеть в прямом направлении до приемника информации. В этом сообщении протокола резервирования содержится так называемый **профиль трафика**, то есть такие его характеристики, как средняя скорость, параметры пульсации, а также требования к качеству обслуживания, например к уровню задержек. На основании этого профиля и требований QoS каждый коммутатор на пути следования потока принимает решение о возможности или невозможности выполнить резервирование для этого потока. Если он «соглашается» выполнить резервирование, то сообщение передается дальше, а коммутатор *делает отметки* о параметрах проведенного резервирования. Если все коммутаторы вдоль маршрута согласны с запрошенными параметрами резервирования, то последний коммутатор передает новое сообщение протокола резервирования в обратном направлении. При прохождении этого сообщения каждый коммутатор *фиксирует* параметры резервирования для данного потока.

Инициировать работу сигнального протокола может не только конечный узел, но и промежуточное устройство. В этом случае гарантированное обслуживание потока будет выполняться не на всем пути следования трафика, а только в пределах определенного участка сети, что, конечно, снижает качество обслуживания трафика.

Протокол резервирования ресурсов позволяет выполнять резервирование как для отдельных потоков, так и для классов трафика. Принципы его работы в обоих случаях остаются одинаковыми. Однако инициатором резервирования ресурсов для класса трафика является не конченый узел, которого интересует собственный поток, а один из коммутаторов сети. Таким коммутатором чаще всего становится пограничный коммутатор сети поставщика услуг, который принимает потоки различных пользователей.

В сетях с виртуальными каналами функции протокола резервирования ресурсов обычно выполняет **протокол установления виртуального канала** — это является его дополнительной задачей. В дейтаграммных сетях протокол резервирования является самостоятельным протоколом. Примером такого протокола является **протокол резервирования ресурсов** (ReSource reserVation Protocol, RSVP), который работает в IP-сетях. Резервирование может выполняться и вручную администратором сети, который должен сконфигурировать параметры резервирования для каждого потока в каждом коммутаторе сети.

Механизмы кондиционирования трафика следят за тем, чтобы текущие параметры потоков соответствовали заявленным при резервировании. Это своего рода контрольно-пропускные пункты, которые проверяют трафик на входе в коммутатор. Без таких механизмов невозможно выполнение гарантий обслуживания трафика. Если средние скорости потоков или пульсации превысят тот уровень, который был указан при резервировании, то задержки и потери пакетов выше допустимых. Такое превышение может произойти по разным причинам. Скажем, потому, что достаточно трудно точно оценить параметры трафика. Предварительно-

ные измерения средней скорости и пульсации могут не дать правильного результата, потому что эти характеристики могут меняться с течением времени, и через неделю они уже не будут соответствовать действительности. Кроме того, нельзя исключать умышленного искажения характеристик трафика, что возможно при использовании коммерческих услуг.

Механизм кондиционирования трафика обычно выполняет несколько функций.

- **Классификация трафика.** Эта функция выделяет из общей последовательности пакетов, поступающих в устройство, пакеты одного потока, имеющего общие требования к качеству обслуживания. В сетях с виртуальными каналами признаком потока является метка виртуального пути, поэтому дополнительной классификации не требуется. В дейтаграммных сетях такого признака, как правило, нет, поэтому классификация выполняется на основе нескольких формальных признаков пакета — адресов источника и назначения, идентификаторов приложений и т. п. Без классификации пакетов в дейтаграммных сетях невозможно обеспечить требуемое качество обслуживания.
- **Профилирование трафика.** Для каждого входного потока в каждом коммутаторе имеется соответствующий ему набор параметров QoS, то есть профиль трафика. Профилирование трафика подразумевает проверку соответствия каждого входного потока параметрам его профиля. Существуют алгоритмы, которые позволяют выполнить такую проверку автоматически в темпе поступления пакетов на входной интерфейс коммутатора. Примерами алгоритмов профилирования являются алгоритмы «дырявого ведра» и «ведра маркеров». Эти алгоритмы будут рассмотрены при изучении отдельных технологий, таких как IP, Frame Relay и ATM.

  В случае нарушения параметров профиля (например, превышения длительности пульсации или средней скорости) происходит отбрасывание или маркировка пакетов этого потока. Отбрасывание некоторых пакетов снижает интенсивность потока и приводит его параметры в соответствие с указанными в профиле. Маркировка пакетов без отбрасывания нужна для того, чтобы пакеты все же были обслужены данным узлом (или последующими по потоку), но с качеством обслуживания, отличным от указанного в профиле (например, с увеличенным значением задержки).
- **Формирование трафика** (shaping). Эта функция предназначена для придания прошедшему профилирование трафику нужной временной «формы». В основном с помощью данной функции стремятся сгладить пульсации трафика, чтобы пакеты на выходе устройства появлялись более равномерно, чем на входе. Сглаживание пульсаций сократит очереди в сетевых устройствах, которые будут обрабатывать трафик далее по потоку. Его также целесообразно использовать для восстановления временных соотношений трафика приложений, работающих с равномерными потоками, например голосовых приложений.

Механизмы кондиционирования трафика могут поддерживаться каждым узлом сети или реализовываться только в пограничных устройствах. Последний вариант часто используют поставщики услуг, кондиционируя трафик своих клиентов.

Существует принципиальное отличие поведения описанной системы для обеспечения средней скорости потока, с одной стороны, и для обеспечения требуемых порогов задержек и вариаций задержек, с другой стороны.

Требуемое значение средней скорости обслуживания обеспечивается за счет конфигурирования выделяемого процента пропускной способности при взвешенном обслуживании. Поэтому сеть может выполнить запрос на любое значение средней скорости для потока, если оно не превышает свободной пропускной способности ресурсов сети вдоль этого потока.

Однако сеть не может сконфигурировать алгоритм приоритетного обслуживания так, чтобы он строго обеспечил какой-либо заранее заданный порог задержек и их вариации. Направление пакетов в приоритетную очередь только позволяет гарантировать, что задержки будут достаточно низкими — существенно меньше, чем у пакетов, которые обрабатываются по алгоритму взвешенного обслуживания. Но аналитически оценить количественные значения задержек очень сложно. Каким же образом поставщик услуг может выполнить свои обязательства перед клиентами?

Как правило, решение данной проблемы основывается на *измерении трафика в сети*. Поставщик услуг должен организовать приоритетное обслуживание трафика с одной или несколькими приоритетными очередями, измеряя задержки реального трафика и обрабатывая результаты статистическими методами. Это означает, что он должен строить гистограммы распределения задержек для различных путей следования потоков и определять средние задержки, средние вариации, максимальные задержки и максимальные вариации для каждого класса трафика, чувствительного к задержкам. На основании этих характеристик поставщик выбирает некоторые предельные значения характеристик QoS, которые он может гарантировать своим клиентам. Обычно такие предельные значения выбираются с некоторым запасом, чтобы при появлении некоторого количества новых клиентов сеть могла соблюдать заявленные гарантии.

Инжиниринг трафика

Список ключевых слов: методы инжиниринга трафика, качество обслуживания, альтернативные маршруты, распараллеливание потока, топология «рыба», предложенная нагрузка, агрегированные потоки, классы трафика, фоновый и оперативный режимы.

При рассмотрении системы обеспечения качества обслуживания, основанной на резервировании, мы не стали затрагивать вопрос маршрутов следования потоков через сеть. Точнее, мы считали, что они каким-то образом выбраны, причем этот выбор делается без учета требований QoS. И в условиях заданности маршрутов мы старались обеспечить прохождение по этим маршрутам такого набора потоков, для которого можно гарантировать соблюдение требований QoS.

Очевидно, что задачу поддержки требований QoS можно решить более эффективно, если считать, что маршруты следования трафика не фиксированы, а также подлежат выбору. Это позволило бы сети обслуживать больше потоков с га-

рантиями QoS при тех же характеристиках самой сети, то есть пропускной способности каналов и производительности коммутаторов и маршрутизаторов.

Задачу выбора маршрутов для потоков (или классов) трафика с учетом соблюдения требований QoS решают **методы инжиниринга трафика** (Traffic Engineering, TE). С помощью этих методов стремятся добиться еще одной цели — по возможности максимально и сбалансировано загрузить все ресурсы сети, чтобы сеть при заданном уровне качества обслуживания обладала как можно боле высокой суммарной производительностью.

Методы TE, как и другие рассмотренные ранее методы, основаны на резервировании ресурсов. То есть они не только позволяют найти рациональный маршрут для потока, но и резервируют для него пропускную способность ресурсов сети, находящихся вдоль этого маршрута.

Методы инжиниринга трафика являются сравнительно новыми для сетей с коммутацией пакетов. Это объясняется во многом тем, что передача эластичного трафика не предъявляла строгих требований к параметрам QoS. Кроме того, Интернет долгое время не являлся коммерческой сетью, поэтому максимальное использование ресурсов не считалось первоочередной задачей для IP-технологий, лежащих в основе Интернета.

Сегодня ситуация изменилась. Сети с коммутацией пакетов должны передавать различные виды трафика с заданным качеством обслуживания, максимально используя возможности своих ресурсов. Однако для этого им нужно изменить некоторые, ставшие уже традиционными, подходы к выбору маршрутов.

Недостатки традиционных методов маршрутизации

Основным принципом работы протоколов маршрутизации в сетях с коммутацией пакетов вот уже долгое время является выбор маршрута на основе топологии сети без учета информации о ее текущей загрузке.

Для каждой пары «адрес источника — адрес назначения» такие протоколы выбирают единственный маршрут, не принимая во внимание информационные потоки, протекающие через сеть. В результате все потоки между парами конечных узлов сети идут по *кратчайшему* (в соответствии с некоторой метрикой) маршруту. Выбранный маршрут может быть более рациональным, например, если в расчет принимается номинальная пропускная способность каналов связи или вносимые ими задержки, либо менее рациональным, если учитывается только количество промежуточных маршрутизаторов между исходным и конечным узлами.

ВНИМАНИЕ

Традиционные методы маршрутизации рассматривают наилучший выбранный маршрут в качестве единственно возможного, даже если существуют другие, хотя и несколько худшие маршруты.

Классическим примером неэффективности такого подхода является так называемая «рыба» — сеть с топологией, приведенной на рис. 7.14. Несмотря на то что между коммутаторами A и E существует два пути (верхний — через коммутатор B и нижний — через коммутаторы C и D), весь трафик от коммутатора A к коммутатору E в соответствии с традиционными принципами маршрутизации направляется по верхнему пути. Только потому, что нижний путь немного (на один ретрансляционный участок) длиннее, чем верхний, он игнорируется, хотя мог бы работать «параллельно» с верхним путем.

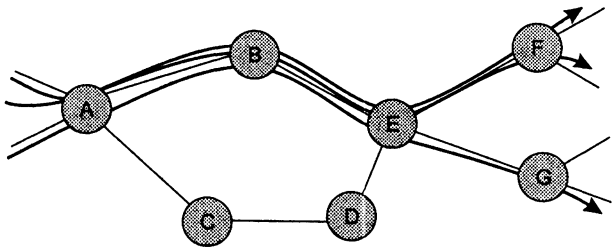


Рис. 7.14. Неэффективность кратчайших путей

Такой подход приводит к тому, что даже если кратчайший путь перегружен, пакеты все равно посылаются по этому пути. Так, в сети, представленной на рис. 7.14, верхний путь будет продолжать использоваться даже тогда, когда его ресурсов перестанет хватать для обслуживания трафика от коммутатора A к коммутатору E, а нижний путь будет простаивать, хотя, возможно, ресурсов коммутаторов B и C хватило бы для качественной передачи этого трафика.

Налицо явная ущербность методов распределения ресурсов сети — одни ресурсы работают с перегрузкой, а другие не используются вовсе. Традиционные методы борьбы с перегрузками эту проблему решить не могут, нужны качественно иные механизмы.

Методы инжиниринга трафика

Исходными данными для методов инжиниринга трафика являются:

- характеристики передающей сети — ее топология, а также производительность составляющих ее коммутаторов и линий связи (рис. 7.15);
- сведения о предложенной нагрузке сети, то есть о потоках трафика, которые сеть должна передать между своими пограничными коммутаторами (рис. 7.16).

Пусть производительность процессора каждого коммутатора достаточна для обслуживания трафика всех его входных интерфейсов, даже если трафик поступает на интерфейс с максимально возможной скоростью, равной пропускной способности интерфейса. Поэтому при резервировании ресурсов будем считать ресурсами пропускную способность линий связи между коммутаторами, которая определяет также пропускную способность двух интерфейсов, связанных этой линией.

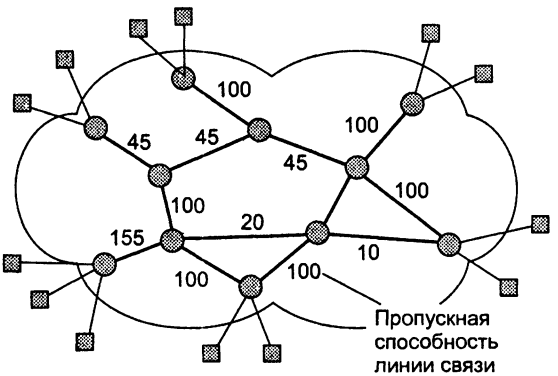


Рис. 7.15. Топология сети и производительность ее ресурсов

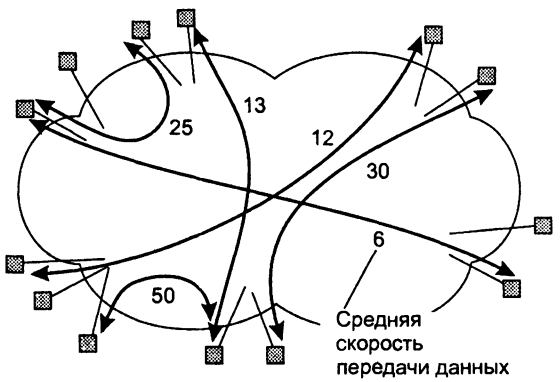


Рис. 7.16. Предложенная нагрузка

Каждый поток характеризуется точкой входа в сеть, точкой выхода из сети и профилем трафика. Для получения оптимальных решений можно использовать детальное описание каждого потока, например учитывать величину возможной пульсации трафика или требования QoS. Однако поскольку количественно оценить их влияние на работу сети достаточно сложно, а влияние этих параметров на характеристики QoS менее значимо, то для нахождения субоптимального распределения путей прохождения потоков через сеть, как правило, учитываются только их средние скорости передачи данных, что и показано на рис. 7.16.

Методы TE чаще работают не с отдельными потоками, а с *агрегированными* потоками, которые являются объединением нескольких потоков. Так как мы ищем общий маршрут для нескольких потоков, то агрегировать можно только потоки, имеющие общие точки входа в сеть и выхода из сети. Агрегированное задание потоков позволяет упростить задачу выбора путей, так как при индивидуальном рассмотрении каждого пользовательского потока промежуточные коммутаторы должны хранить слишком большие объемы информации, поскольку индивидуальных потоков может быть очень много. Необходимо, однако, подчеркнуть, что агрегирование отдельных потоков в один возможно только в том случае, когда

все составляющие потоки предъявляют одни и те же требования к качеству обслуживания. Далее в этом разделе мы будем для краткости пользоваться термином «поток» как для индивидуального потока, так и для агрегированного, поскольку принципы TE от этого не меняются.

Задача TE состоит в определении маршрутов прохождения потоков трафика через сеть, то есть для каждого потока требуется найти точную последовательность промежуточных коммутаторов и их интерфейсов. При этом маршруты должны быть такими, чтобы все ресурсы сети были нагружены до максимально возможного уровня, а каждый поток получал требуемое качество обслуживания.

Максимальный уровень использования ресурсов выбирается таким образом, чтобы механизмы контроля перегрузки могли обеспечить требуемое качество обслуживания. Это означает, что для эластичного трафика максимальное значение выбирается не больше, чем 0,9, а для чувствительного к задержкам трафика — не больше, чем 0,5. Так как обычно резервирование производится не для всех потоков, то нужно оставить часть пропускной способности для свободного использования. Поэтому приведенные максимальные значения обычно уменьшают до 0,75 и 0,25 соответственно. Для упрощения рассуждений мы будем считать далее, что в сети передается один вид трафика, а потом покажем, как обобщить методы TE для случая трафика нескольких типов.

Существуют различные формальные математические определения задачи TE. Мы здесь ограничимся наиболее простым определением, тем более что сегодня оно чаще всего используется на практике.

Будем считать, что решением задачи TE является такой набор маршрутов для заданного множества потоков трафика, для которого все значения коэффициентов использования ресурсов вдоль маршрута следования каждого потока не превышают некоторого заданного порога K_{max}.

На рис. 7.17 показано одно из возможных решений задачи, иллюстрируют которую рис. 7.15 и 7.16. Найденные маршруты гарантируют, что максимальный коэффициент использования любого ресурса для любого потока не превышает 0,6.

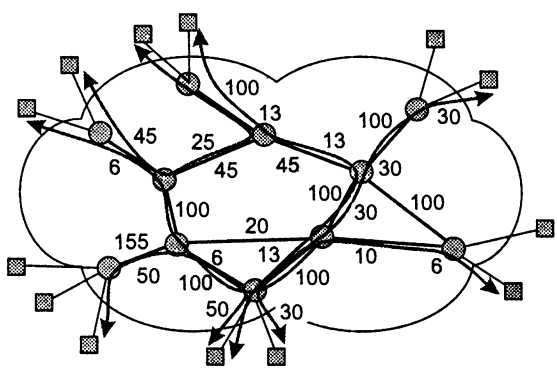


Рис. 7.17. Распределение нагрузки по сети — выбор путей передачи трафика

Решение задачи TE можно искать по-разному. Во-первых, можно искать его заблаговременно, *в фоновом режиме*. Для этого нужно знать исходные данные: топологию и производительность сети, а также входные и выходные точки потоков трафика и среднюю скорость передачи данных в них. После этого задачу рационального распределения путей следования трафика при фиксированных точках входа и выхода, а также заданном уровне максимального значения коэффициента использования ресурса можно передать некоторой программе, которая, например, с помощью направленного перебора вариантов найдет решение. Результатом работы программы будут точные маршруты для каждого потока с указанием всех промежуточных коммутаторов.

Во-вторых, можно решать задачу TE *в оперативном режиме*, поручив ее самим коммутаторам сети. Для этого используются модификации стандартных протоколов маршрутизации. Модификация протоколов маршрутизации состоит в том, что они сообщают друг другу не только топологическую информацию, но и текущее значение свободной пропускной способности у каждого ресурса.

После того как решение найдено, нужно его реализовать, то есть воплотить в таблицах маршрутизации. На этом этапе может возникнуть проблема — в том случае, если мы хотим проложить эти маршруты в дейтаграммной сети. Дело в том, что таблицы маршрутизации этих сетей учитывают только адреса назначения пакетов. Коммутаторы и маршрутизаторы таких сетей (например, IP-сетей) не работают с потоками, для них поток в явном виде не существует, каждый пакет при его продвижении является независимой единицей коммутации. Можно сказать, что таблицы продвижения этих сетей отражают только топологию сети (направления продвижения к определенным адресам назначения).

Поэтому привнесение методов резервирования в дейтаграммные сети происходит с большими трудностями. В протоколах резервирования, подобных упомянутому ранее протоколу RSVP, используется некоторый дополнительный набор признаков помимо адреса назначения, чтобы определить поток для дейтаграммного маршрутизатора. При этом понятие потока требуется только на этапе резервирования, а при продвижении пакетов по-прежнему работает традиционная для этого типа сетей схема, учитывающая только адрес назначения.

Теперь представим ситуацию, когда у нас имеется несколько потоков между двумя конечными узлами, и мы хотим направить их по разным маршрутам. Мы приняли такое решение, исходя из баланса загрузки сети, когда решали задачу инжиниринга трафика. Дейтаграммный коммутатор или маршрутизатор не имеет возможности реализовать наше решение, потому что для всех этих потоков у него в таблице продвижения есть только одна запись, соответствующая общему адресу назначения пакетов этих потоков. Изменять логику работы коммутаторов и маршрутизаторов дейтаграммных сетей достаточно нецелесообразно, поскольку это слишком принципиальная модернизация.

Поэтому методы инжиниринга трафика сегодня используются только в сетях с виртуальными каналами, для которых не составляет труда реализовать найденное решение для группы потоков. Каждому потоку (или группе потоков с одинаковыми маршрутами) выделяется виртуальный канал, который прокладывается в соответствии с выбранным маршрутом. Методы инжиниринга трафика успешно

применяются в сетях ATM и Frame Relay, работающих на основе техники виртуальных каналов. IP-сети также опираются на методы TE, когда те используются в сетях ATM или Frame Relay, работающих в составной сети, построенной на основе протокола IP. Существует также новая технология MPLS, которая разработана специально в качестве средства привнесения техники виртуальных каналов в IP-сети. На основе технологии MPLS в IP-сетях можно также решать задачи TE.

Мы рассмотрим особенности методов TE для каждой отдельной технологии при детальном изучении этих технологий в следующих частях книги.

Инжиниринг трафика различных классов

При решении задачи инжиниринга трафика мы считали, что все потоки трафика предъявляли одинаковые требования к качеству обслуживания. То есть пользователей сети удовлетворяло, что все потоки обслуживаются с заданной средней скоростью (она, естественно, у каждого потока своя, отличающаяся от других).

Более реальной является ситуация, когда у каждого пользователя сети имеется *несколько классов трафика*, и эти классы отличаются разными требованиями к качеству обслуживания. Мы уже обсуждали эту проблему при рассмотрении вопросов резервирования ресурсов.

Методы TE, учитывающие наличие в сети трафика с различными требованиями QoS, решают проблему точно так же, как и методы резервирования ресурсов отдельных узлов. Если у нас имеется, например, два класса трафика, то мы задаемся двумя уровнями максимального использования ресурсов.

Для достижения такого результата с каждым ресурсом должно быть связано два счетчика свободной пропускной способности — один для приоритетного трафика, второй для эластичного трафика. При определении возможности прохождения маршрута через конкретный ресурс для приоритетного трафика средняя интенсивность нового потока должна сравниваться со свободной пропускной способностью для приоритетного трафика.

Если свободной пропускной способности достаточно и новый поток будет проходить через данный интерфейс, то значение средней скорости передачи данных для нового потока необходимо вычесть как из счетчика загрузки приоритетного трафика, так и из счетчика загрузки эластичного трафика, так как приоритетный трафик всегда будет обслуживаться перед эластичным и создаст дополнительную нагрузку для эластичного трафика. Если же задача TE решается для эластичного трафика, то его средняя скорость передачи данных сравнивается со свободной пропускной способностью счетчика эластичного трафика и в случае положительного решения значение этой скорости вычитается только из счетчика эластичного трафика, так как для приоритетного трафика эластичный трафик прозрачен.

Модифицированные протоколы маршрутизации должны распространять по сети информацию о двух параметрах свободной пропускной способности — для каждого класса трафика отдельно. Если же задача обобщается для случая передачи через сеть трафика нескольких классов, то, соответственно, с каждым ресурсом должно быть связано *столько счетчиков, сколько классов трафика существует*

в сети, а протоколы маршрутизации должны распространять вектор свободных пропускных способностей соответствующей размерности.

Выводы

Качество обслуживания в его узком смысле фокусирует внимание на характеристиках и методах передачи трафика через очереди коммуникационных устройств. Методы обеспечения качества обслуживания занимают сегодня одно из важнейших мест в арсенале технологий сетей с коммутацией пакетов, так как без их применения невозможна работа современных мультимедийных приложений, таких как IP-телефония, видео- и радиовещание, интерактивное дистанционное обучение и т. п.

Характеристики QoS отражают отрицательные последствия пребывания пакетов в очередях, которые проявляются в снижении скорости передачи, задержках пакетов и их потерях.

Приоритетные и взвешенные очереди, а также резервирование и обратная связь позволяют гарантировать качество обслуживания для чувствительного к задержкам и эластичного трафика.

Алгоритм скользящего окна обеспечивает не только надежную передачу пакетов, но и является эффективным средством обратной связи.

Архитектура основанной на резервировании системы поддержания качества обслуживания включает:

- механизмы очередей;
- протоколы резервирования, позволяющие автоматически выделять необходимые ресурсы для «сквозного» потока;
- средства кондиционирования трафика, выполняющие классификацию, профилирование и формирование трафика.

Методы инжиниринга трафика состоят в выборе рациональных маршрутов прохождения потоков через сеть. Выбор маршрутов обеспечивает максимизацию загрузки ресурсов сети при одновременном соблюдении необходимых гарантий в отношении параметров качества обслуживания трафика.

Вопросы и задания

1. В чем состоят позитивные и негативные эффекты применения очередей в коммутаторах пакетов?
2. Какой параметр в наибольшей степени влияет на размер очереди? Какой параметр является вторым по значимости?
3. Какие типы трафика передает сеть с коммутацией пакетов? Какие требования эти типы трафика предъявляют к сети?
4. В чем состоят достоинства и недостатки приоритетного обслуживания?
5. Для трафика какого типа в наибольшей степени подходит взвешенное обслуживание?
6. Можно ли комбинировать приоритетное и взвешенное обслуживание?

7. Назовите отличия между резервированием пропускной способности в сетях с коммутацией каналов и пакетов?
8. Из каких компонентов состоит система обеспечения параметрв QoS, базирующаяся на резервировании?
9. Какую задачу решают методы инжиниринга трафика?
10. Какой параметр трафика меняется при инжиниринге трафика?
11. Пусть некоторый поток данных принадлежит классу CBR. Данные передаются пакетами размером в 125 байт по каналу 100 Мбит/с. Профиль трафика имеет следующие параметры: пиковая скорость передачи данных (PIR) на периоде пульсаций равна 25 Мбит/с, максимальное отклонение периода следования пакетов составляет 10 мкс, период пульсации равен 600 мкс. Если трафик соответствует своему профилю, то какова максимальная величина пульсации?
12. Какой из 5 потоков будет меньше в среднем задерживаться в очереди к выходному интерфейсу 100 Мбит/с, если потоки обслуживаются взвешенными очередями, при этом потокам отведено 40, 15, 10, 30 и 5 % пропускной способности интерфейса. Потоки имеют средние скорости: 35, 2, 8, 3 и 4 Мбит/с соответственно. Коэффициент вариации интервалов следования пакетов одинаков у всех потоков.
13. Что является причиной того, что поток, который обслуживается в очереди самого высокого приоритета, все равно сталкивается с необходимостью ожидания в очереди:
    - очереди более низких приоритетов;
    - собственная пульсация;
    - пульсации низкоприоритетного трафика.
14. К выходному интерфейсу 10 Мбит/с организовано три очереди, которые обслуживаются по алгоритму взвешенных очередей. В первой очереди имеется три пакета: пакет 1 размером 1500 байт, пакет 2 размером 625 байт и пакет 3 размером 750 байт. Во второй очереди тоже находятся три пакета: пакет 4 размером 500 байт, пакет 5 размером 1500 байт и пакет 6 размером 1500 байт. В третьей очереди находятся четыре пакета: пакет 7 размером 100 байт, пакет 8 размером 275 байт, пакет 9 размером 1500 байт и пакет 10 размером 1500 байт. В очередях пакеты размещены в порядке возрастания номеров, то есть в первой очереди первым стоит пакет 1, во второй — пакет 4, в третьей — пакет 7.

    В каком порядке появятся на выходе интерфейса 2 Мбит/с пакеты, если цикл работы алгоритма равен 10 мс и очередям отведено 50, 30 и 20 % пропускной способности ресурса соответственно? Алгоритм всегда берет из очереди пакет в каждом цикле (если очередь не пуста), даже если размер пакета таков, что его передача превысит время, отведенное данной очереди.
15. Какое время длится каждый из двух циклов обслуживания очередей в предыдущем примере? С какой скоростью обслуживается каждый поток на этом интервале, состоящем из двух циклов?

16. Как нужно изменить время цикла работа алгоритма из задания 14, чтобы скорости потоков были ближе к запланированным? Увеличить или уменьшить?
17. На входе сети некоторый поток профилируется в соответствии с профилем 3 Мбит/с. Для этого же потока отведено 30 % пропускной способности выходного интерфейса 10 Мбит/с в промежуточном коммутаторе сети. Какое из приведенных ниже объяснений является правильным?
    - Результат действия этих механизмов один и тот же, поэтому резервирование в коммутаторе можно не применять.
    - Результат действия этих механизмов один и тот же, но резервирование в коммутаторе необходимо, так как на входе в сеть и внутри коммутатора поток конкурирует за ресурсы с другими потоками.
    - Результат работы этих механизмов разный, на входе сеть ограничивает скорость потока пределом в 3 Мбит/с, а в коммутаторе для этого потока гарантируется скорость 3 Мбит/с даже в периоды перегрузок.
18. Может ли отсутствовать очередь в системе, коэффициент использования которой близок к единице?
19. Какие из перечисленных механизмов нужно использовать для качественной передачи голосового трафика через сеть с коммутацией пакетов, образующего поток 64 Кбит/с:
    - резервирование пропускной способности в 64 Кбит/с на всех коммутаторах вдоль маршрута следования потока;
    - обслуживание потока в приоритетной очереди всех коммутаторов вдоль маршрута следования потока;
    - применение входного буфера пакетов у приемного узла сети;
    - сглаживание трафика в выходных очередях всех коммутаторов вдоль маршрута следования потока.
20. Верно ли утверждение, что резервирование ресурсов в сети с коммутацией пакетов лишает ее возможности динамического перераспределения пропускной способности между потоками?
21. Какой механизм нужно применять для того, чтобы высокоприоритетный трафик не подавил низкоприоритетный?

Часть II

Технологии физического уровня

Физической основой любой компьютерной (и телекоммуникационной) сети являются линии связи. Без таких линий коммутаторы не смогут обмениваться пакетами, и компьютеры останутся изолированными устройствами.

После изучения принципов построения компьютерных сетей в воображении читателя могла возникнуть достаточно простая картина компьютерной сети — компьютеры и коммутаторы, соединенные друг с другом отрезками кабеля. Однако при более детальном рассмотрении компьютерной сети все оказывается сложнее, чем это казалось при изучении модели OSI.

Дело в том, что специально выделенные кабели используются для соединения сетевых устройств только на небольших расстояниях, то есть в локальных сетях. При построении сетей WAN и MAN такой подход крайне расточителен из-за высокой стоимости протяженных линий связи. К тому же, на их прокладку необходимо получать разрешение. Поэтому гораздо чаще для связи коммутаторов в сетях WAN и MAN используются уже существующие телефонные или первичные территориальные сети с коммутацией каналов. В этом случае в сети с коммутацией каналов создается составной канал, который выполняет те же функции, что и отрезок кабеля — обеспечивает физическое двухточечное соединение. Конечно, составной канал представляет собой гораздо более сложную техническую систему, чем кабель, но для компьютерной сети эти сложности прозрачны. Первичные сети специально строятся для создания канальной инфраструктуры, поэтому их каналы более эффективны по соотношению цена/пропускная способность. Сегодня в распоряжении проектировщика компьютерной сети имеются каналы первичных сетей для широкого диапазона скоростей — от 64 Кбит/с до 10 Гбит/с.

Несмотря на различия в физической и технической природе линий связи, их можно описать с помощью единого набора характеристик. Важнейшими характеристиками любой линии связи при передаче дискретной информации являются полоса пропускания, измеряемая в герцах (Гц), и емкость, или пропускная способность, измеряемая в битах в секунду (бит/с). Пропускная способность представляет собой скорость битового потока, передаваемого линией связи. Пропускная способность зависит от полосы пропускания линии и способа кодирования дискретной информации.

Все большую популярность приобретают беспроводные каналы. Они являются единственным типом каналов, обеспечивающих мобильность пользователей компьютерной сети. Кроме того, беспроводная связь применяется в тех случаях, когда кабели проложить невозможно или невыгодно — в малонаселенных районах, при доступе к жилым домам, уже охваченным кабельной инфраструктурой конкурентов и т. п. При беспроводной связи используются электромагнитные волны различной частоты — радиоволны, микроволны, инфракрасное излучение и видимый свет. Высокий уровень помех и сложные пути распространения волн требуют применения в беспроводных каналах особых способов кодирования и передачи сигналов.

- Глава 8. Линии связи
- Глава 9. Кодирование и мультиплексирование данных
- Глава 10. Беспроводная передача данных
- Глава 11. Первичные сети

ГЛАВА 8 Линии связи

При построении сетей применяются линии связи, в которых используются различные физические среды: подвешенные в воздухе телефонные и телеграфные провода, проложенные под землей и по дну океана медные коаксиальные и волоконно-оптические кабели, опутывающие все современные офисы медные витые пары, все проникающие радиоволны.

В этой главе рассматриваются общие характеристики линий связи, не зависящие от их физической природы, такие как полоса пропускания, пропускная способность, помехоустойчивость и достоверность передачи. Ширина полосы пропускания является фундаментальной характеристикой канала связи, так как определяет максимально возможную информационную скорость канала, которая называется пропускной способностью канала. Формула Найквиста выражает эту зависимость для идеального канала, а формула Шеннона учитывает наличие в реальном канале шума. Завершает главу рассмотрение конструкций и стандартов современных кабелей, которые составляют основу проводных линий связи.

Классификация линий связи

Список ключевых слов: звено, канал, составной канал, линия связи, первичная сеть, наложенная сеть, физическая среда передачи данных, проводная (воздушная) линия связи, кабельная линия связи, неэкранированная витая пара, экранированная витая пара, медный кабель, радиоканал, диапазон широковещательного радио, диапазон очень высоких частот, диапазон ультравысоких частот, диапазон микроволн, аппаратура передачи данных, модем, терминальный адаптер сетей ISDN, устройства для подключения к цифровым каналам, оконечное оборудование данных, промежуточная аппаратура, повторитель, концентратор, усилитель, регенератор, мультиплексор, демультиплексор, коммутатор, аналоговая линия связи, цифровая линия связи.

Первичные сети, линии и каналы связи

При описании технической системы, которая передает информацию между узлами сети, в литературе можно встретить несколько названий: *линия связи*, *составной канал*, *канал*, *звено*. Часто эти термины используются как синонимы, и во многих случаях это не вызывает проблем. В то же время есть и специфика в их употреблении.

- **Звено** (link) — это сегмент, обеспечивающий передачу данных между двумя соседними узлами сети. То есть звено не содержит промежуточных устройств коммутации и мультиплексирования.
- **Каналом** (channel) чаще всего обозначают часть пропускной способности звена, используемую независимо при коммутации. Например, звено первичной сети может состоять из 30 каналов, каждый из которых обладает пропускной способностью 64 Кбит/с.
- **Составной канал** (circuit) — это путь между двумя конечными узлами сети. Составной канал образуется отдельными каналами промежуточных звеньев и внутренними соединениями в коммутаторах. Часто эпитет «составной» опускается и термин «канал» используется для обозначения как составного канала, так и канала между соседними узлами, то есть в пределах звена.
- **Линия связи** может использоваться как синоним для любого из трех остальных терминов.

Не стоит относиться к путанице в терминологии очень строго. Особенно это относится к различиям в терминологии традиционной телефонии и более новой области — компьютерных сетей. Процесс конвергенции только усугубил проблему терминологии, так как многие механизмы этих сетей стали общими, но сохранили за собой по паре (иногда и больше) названий, пришедших из каждой области.

Кроме того, существуют объективные причины для неоднозначного понимания терминов. На рис. 8.1 показаны два варианта линии связи. В первом случае (рис. 8.1, *а*) линия состоит из сегмента кабеля длиной несколько десятков мет-

ров и представляет собой звено. Во втором случае (рис. 8.1, *б*) линия связи представляет собой составной канал, развернутый в сети с коммутацией каналов. Такой сетью может быть **первичная сеть** или телефонная сеть.

Однако для компьютерной сети эта линия представляет собой звено, так как соединяет два соседних узла, и вся коммутационная промежуточная аппаратура является прозрачной для этих узлов. Повод для взаимного непонимания на уровне терминов компьютерных специалистов и специалистов первичных сетей здесь очевиден.

Первичные сети специально создаются для того, чтобы предоставлять услуги каналов передачи данных для компьютерных и телефонных сетей, про которые в таких случаях говорят, что они работают «поверх» первичных сетей и являются **наложенными сетями**.

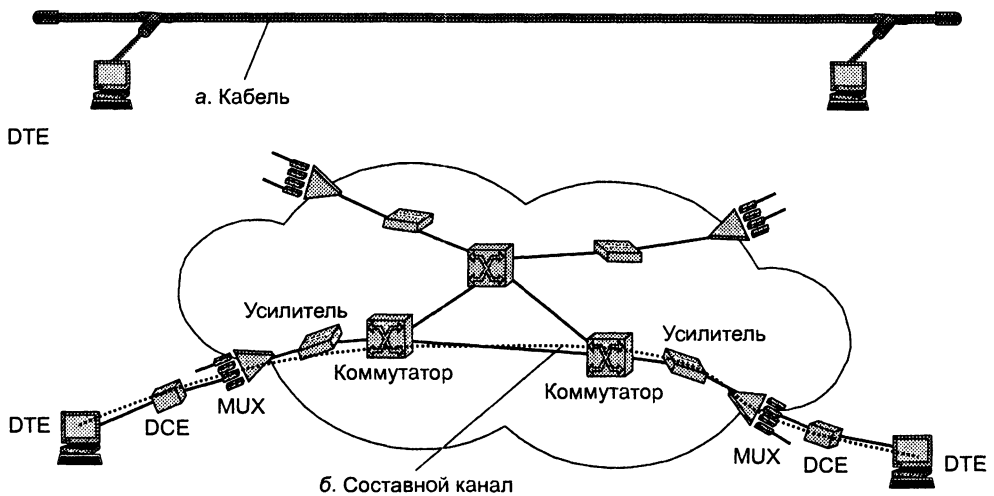


Рис. 8.1. Состав линии связи

Физическая среда передачи данных

Линии связи отличаются также физической средой, которую они используют для передачи информации.

Физическая среда передачи данных может представлять набор проводников, по которым передаются сигналы. На основе таких проводников строятся проводные (воздушные) или кабельные линии связи (рис. 8.2). В качестве среды также используется земная атмосфера или космическое пространство, через которое распространяются информационные сигналы. В первом случае говорят о *проводной среде*, а во втором — о *беспроводной*.

В современных телекоммуникационных системах информация передается с помощью электрического тока или напряжения, радиосигналов или световых сигналов — все эти физические процессы представляют собой колебания электромагнитного поля различной частоты.

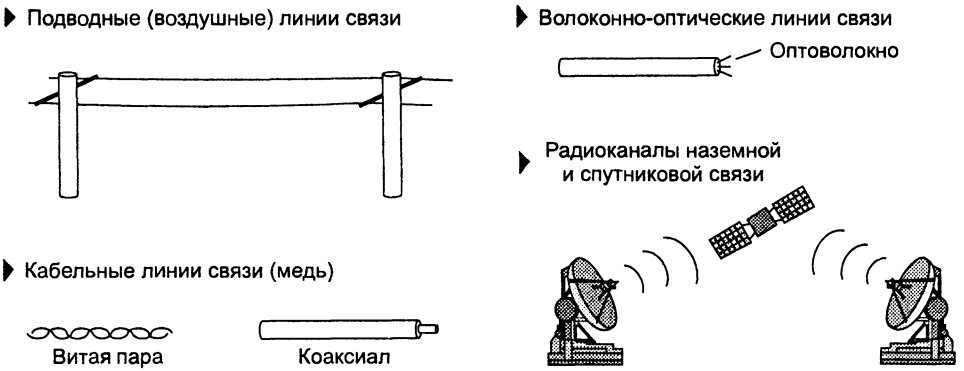


Рис. 8.2. Типы сред передачи данных

Проводные (воздушные) линии связи представляют собой провода без каких-либо изолирующих или экранирующих оплеток, проложенные между столбами и висящие в воздухе. Еще в недалеком прошлом такие линии связи были основными для передачи телефонных или телеграфных сигналов. Сегодня проводные линии связи быстро вытесняются кабельными. Но кое-где они все еще сохранились и при отсутствии других возможностей продолжают использоваться и для передачи компьютерных данных. Скоростные качества и помехозащищенность этих линий оставляют желать много лучшего.

Кабельные линии имеют достаточно сложную конструкцию. Кабель состоит из проводников, заключенных в несколько слоев изоляции: электрической, электромагнитной, механической и, возможно, климатической. Кроме того, кабель может быть оснащен разъемами, позволяющими быстро выполнять присоединение к нему различного оборудования. В компьютерных (и телекоммуникационных) сетях применяются три основных типа кабеля: кабели на основе скрученных пар медных проводов — **неэкранированная витая пара** (Unshielded Twisted Pair, UTP) и **экранированная витая пара** (Shielded Twisted Pair, STP), **коаксиальные кабели** с медной жилой, волоконно-оптические кабели. Первые два типа кабелей называют также **медными кабелями**.

Радиоканалы наземной и спутниковой связи образуются с помощью передатчика и приемника радиоволн. Существует большое разнообразие типов радиоканалов, отличающихся как используемым частотным диапазоном, так и дальностью канала. **Диапазоны широковещательного радио** (длинных, средних и коротких волн), называемые также **АМ-диапазонами**, или диапазонами амплитудной модуляции (Amplitude Modulation, AM), обеспечивают дальнюю связь, но при невысокой скорости передачи данных. Более скоростными являются каналы, использующие **диапазоны очень высоких частот** (Very High Frequency, VHF), для которых применяется частотная модуляция (Frequency Modulation, FM). Для передачи данных также используются **диапазоны ультравысоких частот** (Ultra High Frequency, UHF), называемые еще **диапазонами микроволн** (свыше 300 МГц). При частоте свыше 30 МГц сигналы уже не отражаются ионосферой Земли, и для устойчивой связи требуется наличие прямой видимости между передатчиком и приемником. Поэтому такие частоты используют либо спутниковые каналы, либо

радиорелейные каналы, либо локальные или мобильные сети, где это условие выполняется.

В компьютерных сетях сегодня применяются практически все описанные типы физических сред передачи данных. Хорошие возможности предоставляют волоконно-оптические кабели, обладающие широкой полосой пропускания и низкой чувствительностью к помехам. На них сегодня строятся как магистрали крупных территориальных и городских сетей, так и высокоскоростные локальные сети. Популярной средой является также витая пара, которая характеризуется отличным отношением качества к стоимости, а также простотой монтажа. Беспроводные каналы используются чаще всего в тех случаях, когда кабельные линии связи применить нельзя — например, при прохождении канала через малонаселенную местность или же для связи с мобильными пользователями сети. Обеспечение мобильности затронуло в первую очередь телефонные сети, компьютерные сети в этом отношении пока отстают. Тем не менее построение компьютерных сетей на основе беспроводных технологий, например, Radio Ethernet, считаются сегодня одним из самых перспективных направлений телекоммуникаций. Линии связи на основе беспроводной среды изучаются в главе 10.

Аппаратура передачи данных

Как показано на рис. 8.1, линии связи состоят не только из среды передачи, но и аппаратуры. Даже в том случае, когда линия связи не проходит через первичную сеть, а основана на кабеле, в ее состав входит аппаратура передачи данных.

Аппаратура передачи данных (Data Circuit Equipment, DCE) в компьютерных сетях непосредственно присоединяет компьютеры или коммутаторы к линиям связи и является, таким образом, пограничным оборудованием. Традиционно аппаратуру передачи данных включают в состав линии связи. Примерами DCE являются **модемы** (для телефонных линий), **терминальные адаптеры сетей ISDN**, **устройства для подключения к цифровым каналам** первичных сетей DSU/CSU (Data Service Unit/Circuit Service Unit).

DCE работает на физическом уровне модели OSI, отвечая за передачу информации в физическую среду (в линию) и прием из нее сигналов нужной формы, мощности и частоты. Аппаратура пользователя линии связи, вырабатывающая данные для передачи по линии связи и подключаемая непосредственно к аппаратуре передачи данных, носит обобщенное название **оконечное оборудование данных** (Data Terminal Equipment, DTE). Примером DTE могут служить компьютеры, коммутаторы и маршрутизаторы. Эту аппаратуру не включают в состав линии связи.

ПРИМЕЧАНИЕ

Разделение оборудования на DCE и DTE в локальных сетях является достаточно условным. Например, адаптер локальной сети можно считать как принадлежностью компьютера, то есть оборудованием DTE, так и составной частью канала связи, то есть аппаратурой DCE. Точнее, одна часть сетевого адаптера выполняет функции DTE, а его другая, оконечная его часть, непосредственно принимающая и передающая сигналы, относится к DCE.

Для подключения устройств DCE к устройствам DTE (то есть компьютерам или коммутаторам/маршрутизаторам) существуют несколько *стандартных интерфейсов*[1]. Работают эти устройства на коротких расстояниях друг от друга, как правило, несколько метров.

Промежуточная аппаратура обычно используется на линиях связи большой протяженности. Она решает две основные задачи:

- улучшение качества сигнала;
- создание постоянного составного канала связи между двумя абонентами сети.

В локальных сетях промежуточная аппаратура может совсем не использоваться, если протяженность физической среды — кабелей или радиоэфира — позволяет одному сетевому адаптеру принимать сигналы непосредственно от другого сетевого адаптера без дополнительного усиления. В противном случае применяется промежуточная аппаратура, роль которой здесь играют устройства типа **повторителей** и **концентраторов**.

В *глобальных сетях* необходимо обеспечить качественную передачу сигналов на расстояния в сотни и тысячи километров. Поэтому без **усилителей** (повышающих мощность сигналов) и **регенераторов** (наряду с повышением мощности восстанавливающих форму импульсных сигналов, исказившихся при передаче на большое расстояние), установленных через определенные расстояния, построить территориальную линию связи невозможно.

В первичных сетях помимо рассмотренного выше оборудования, обеспечивающего качественную передачу сигналов, необходима промежуточная коммутационная аппаратура — **мультиплексоры** (MUX), **демультиплексоры** и **коммутаторы**. Эта аппаратура создает между двумя абонентами сети постоянный составной канал из отрезков физической среды — кабелей с усилителями.

В зависимости от типа промежуточной аппаратуры все линии связи делятся на аналоговые и цифровые. В **аналоговых линиях** промежуточная аппаратура предназначена для усиления аналоговых сигналов, то есть сигналов, которые имеют непрерывный диапазон значений. Такие линии связи традиционно применялись в телефонных сетях для связи телефонных коммутаторов между собой. Для создания высокоскоростных каналов, которые мультиплексируют несколько низкоскоростных аналоговых абонентских каналов, при аналоговом подходе обычно используется *техника частотного мультиплексирования* (Frequency Division Multiplexing, FDM).

В **цифровых линиях** связи передаваемые сигналы имеют конечное число состояний. Как правило, элементарный сигнал, то есть сигнал, передаваемый за один такт работы передающей аппаратуры, имеет 2, 3 или 4 состояния, которые в линиях связи воспроизводятся импульсами или потенциалами прямоугольной формы. С помощью таких сигналов передаются как компьютерные данные, так и оцифрованные речь и изображение (именно благодаря одинаковому способу представления информации современными компьютерными, телефонными и телевизи-

[1] Интерфейсы DTE-DCE описываются стандартами серии V CCITT, а также стандартами EIA серии RS (Recommended Standards — рекомендуемые стандарты). Две линии стандартов во многом дублируют друг друга. Наиболее популярными стандартами являются RS-232, RS-530, V.35 и HSSI.

онными сетями стало возможным появление общих для всех первичных сетей). В цифровых линиях связи используется специальная промежуточная аппаратура — регенераторы, которые улучшают форму импульсов и восстанавливают период их следования. Промежуточная аппаратура мультиплексирования и коммутации первичных сетей работает по принципу *временного мультиплексирования каналов* (Time Division Multiplexing, TDM).

Характеристики линий связи

Список ключевых слов: гармоника, спектральное разложение (спектр) сигнала, ширина спектра сигнала, формулы Фурье, внешние помехи, внутренние помехи, или наводки, затухание сигнала, погонное затухание, окно прозрачности, абсолютный уровень мощности, относительный уровень мощности, порог чувствительности приемника, волновое сопротивление, помехоустойчивость линии, электрическая связь, магнитная связь, наведенный сигнал, перекрестные наводки на ближнем конце, перекрестные наводки на дальнем конце, защищенность кабеля, достоверность передачи данных, интенсивность битовых ошибок, полоса пропускания, пропускная способность, физическое, или линейное, кодирование, несущий сигнал, несущая частота, модуляция, такт, бод.

Спектральный анализ сигналов на линиях связи

Важная роль при определении параметров линий связи отводится спектральному разложению передаваемого по этой линии сигнала. Из теории гармонического анализа известно, что *любой периодический процесс можно представить в виде суммы синусоидальных колебаний различных частот и различных амплитуд* (рис. 8.3).

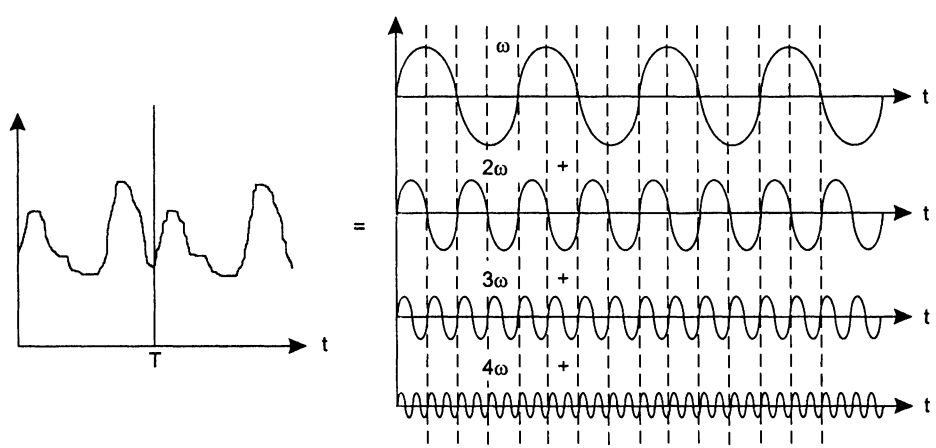


Рис. 8.3. Представление периодического сигнала суммой синусоид

Каждая составляющая синусоида называется также **гармоникой**, а набор всех гармоник называют **спектральным разложением**, или **спектром**, исходного сигнала.

Под **шириной спектра сигнала** понимается разность между максимальной и минимальной частотами того набора синусоид, которые в сумме дают исходный сигнал.

Непериодические сигналы можно представить в виде интеграла синусоидальных сигналов с непрерывным спектром частот. В частности, спектральное разложение идеального импульса (единичной мощности и нулевой длительности) имеет составляющие всего спектра частот, от $-\infty$ до $+\infty$ (рис. 8.4).

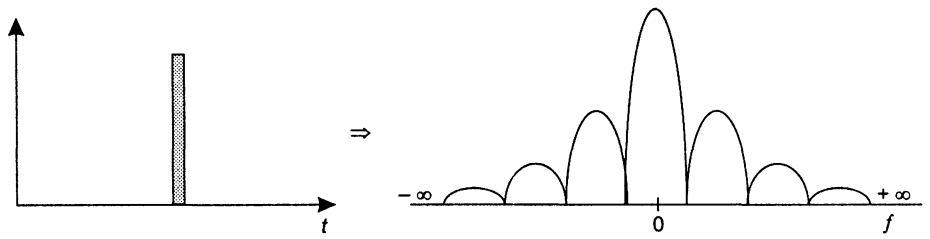


Рис. 8.4. Спектральное разложение идеального импульса

Техника нахождения спектра любого исходного сигнала хорошо известна. Для некоторых сигналов, которые описываются аналитически (например, для последовательности прямоугольных импульсов одинаковой длительности и амплитуды), спектр легко вычисляется на основании **формул Фурье**.

Для сигналов произвольной формы, встречающихся на практике, спектр можно найти с помощью специальных приборов — спектральных анализаторов, которые измеряют спектр реального сигнала и отображают амплитуды составляющих гармоник на экране, распечатывают их на принтере или передают для обработки и хранения в компьютер.

Искажение передающей линией связи синусоиды какой-либо частоты приводит, в конечном счете, к искажению амплитуды и формы передаваемого сигнала любого вида. Искажения формы проявляются в том случае, когда синусоиды различных частот искажаются неодинаково. Если это аналоговый сигнал, передающий речь, то изменяется тембр голоса за счет искажения обертонов — боковых частот. При передаче импульсных сигналов, характерных для компьютерных сетей, искажаются низкочастотные и высокочастотные гармоники, в результате фронты импульсов теряют свою прямоугольную форму (рис. 8.5) и сигналы могут плохо распознаваться на приемном конце линии.

Передаваемые сигналы искажаются из-за несовершенства линий связи. Идеальная передающая среда, не вносящая никаких помех в передаваемый сигнал, должна, по меньшей мере, иметь нулевые сопротивление, емкость и индуктивность. Однако на практике медные провода, например, всегда представляют собой некоторую распределенную по длине комбинацию активного сопротивления, емкостной и индуктивной нагрузок (рис. 8.6). В результате синусоиды различных частот передаются этими линиями по-разному.

Помимо искажений сигналов, возникающих из-за не идеальных физических параметров линии связи, существуют и **внешние помехи**, которые вносят свой вклад в искажение формы сигналов на выходе линии. Эти помехи создаются различ-

ными электрическими двигателями, электронными устройствами, атмосферными явлениями и т. д. Несмотря на защитные меры, предпринимаемые разработчиками кабелей, и наличие усилительной и коммутирующей аппаратуры, полностью компенсировать влияние внешних помех не удается. Помимо внешних помех в кабеле существуют и **внутренние помехи** — так называемые **наводки** одной пары проводников на другую. В результате сигналы на выходе линии связи могут иметь искаженную форму (как это и показано на рис. 8.5).

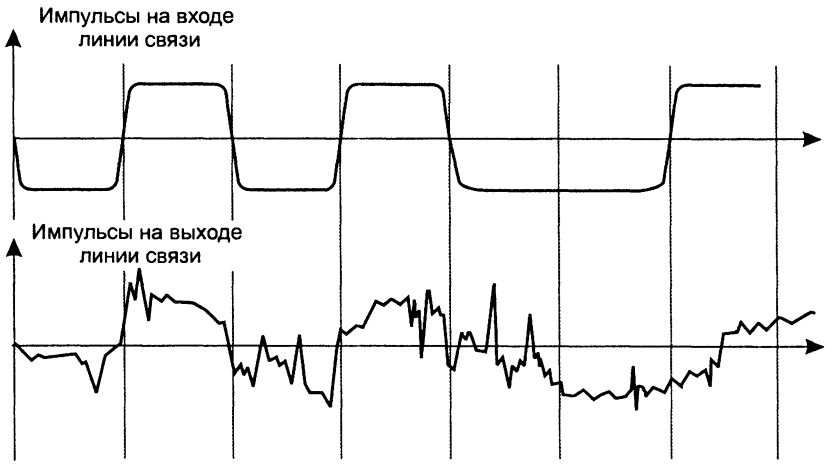


Рис. 8.5. Искажение импульсов в линии связи

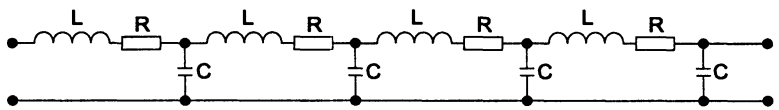


Рис. 8.6. Представление линии как распределенной индуктивно-емкостной нагрузки

Затухание и волновое сопротивление

Степень искажения синусоидальных сигналов линиями связи оценивается такими характеристиками, как затухание и полоса пропускания.

Затухание показывает, насколько уменьшается мощность эталонного синусоидального сигнала на выходе линии связи по отношению к мощности сигнала на входе этой линии. Затухание (A) обычно измеряется в децибелах (дБ) и вычисляется по следующей формуле:

$$A = 10 \lg P_{out}/P_{in}.$$

Здесь P_{out} — мощность сигнала на выходе линии, P_{in} — мощность сигнала на входе линии. Так как затухание зависит от длины линии связи, то в качестве характеристики линии связи используется так называемое **погонное затухание**, то есть затухание на линии связи определенной длины. Для кабелей локальных сетей в качестве такой длины обычно используют 100 м, так как это значение является максимальной длиной кабеля для многих LAN-технологий. Для территориальных линий связи погонное затухание измеряют для расстояния в 1 км.

Обычно затуханием характеризуют пассивные участки линии связи, состоящие из кабелей и кроссовых секций, без усилителей и регенераторов. Так как мощность выходного сигнала кабеля без промежуточных усилителей меньше, чем мощность входного сигнала, затухание кабеля всегда является *отрицательной величиной.*

Степень затухания мощности синусоидального сигнала зависит от частоты синусоиды, и эта зависимость также используется для характеристики линии связи (рис. 8.7).

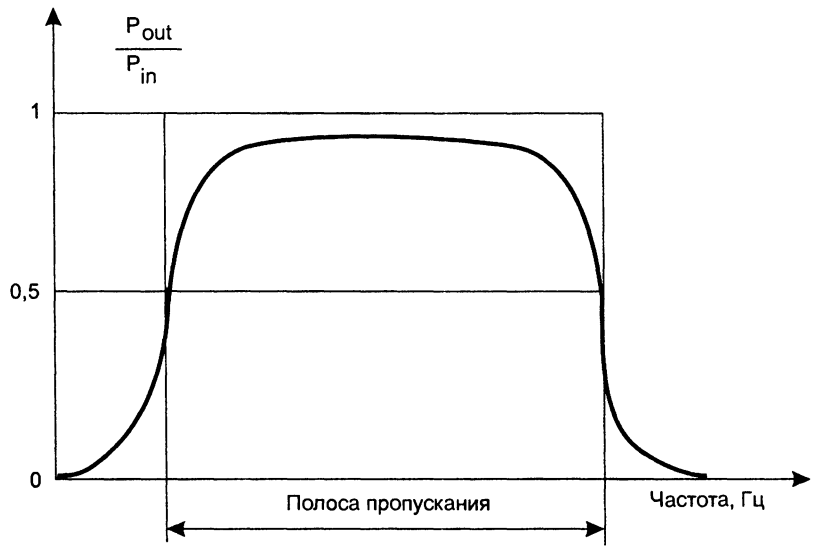


Рис. 8.7. Зависимость затухания от частоты

Чаще всего при описании параметров линии связи приводятся значения затухания всего *для нескольких значений частот.* Это объясняется, с одной стороны, стремлением упростить измерения при проверке качества линии. С другой стороны, на практике часто заранее известна основная частота передаваемого сигнала, то есть та частота, гармоника которой имеет наибольшую амплитуду и мощность. Поэтому достаточно знать затухание на этой частоте, чтобы приблизительно оценить искажения передаваемых по линии сигналов.

ВНИМАНИЕ

Как было сказано выше, затухание всегда имеет отрицательное значение, однако знак минус часто опускают, при этом иногда возникает путаница. Совершенно корректно утверждение, что качество линии связи тем выше, чем больше (с учетом знака) затухание. Если же игнорировать знак, то есть иметь в виду абсолютное значение затухания, то у более качественной линии затухание меньше. Приведем пример. Для внутренней проводки в зданиях используется кабель на витой паре категории 5. Этот кабель, на котором работают практически все технологии локальных сетей, характеризуется затуханием не меньше, чем –23,6 дБ для частоты 100 МГц при длине кабеля 100 м. Более качественный кабель категории 6 имеет на частоте 100 МГц затухание не меньше, чем –20,6 дБ. Получаем, что $-20{,}6 > -23{,}6$, но $20{,}6 < 23{,}6$.

На рис. 8.8 показаны типовые зависимости затухания от частоты для кабелей на неэкранированной витой паре категорий 5 и 6.

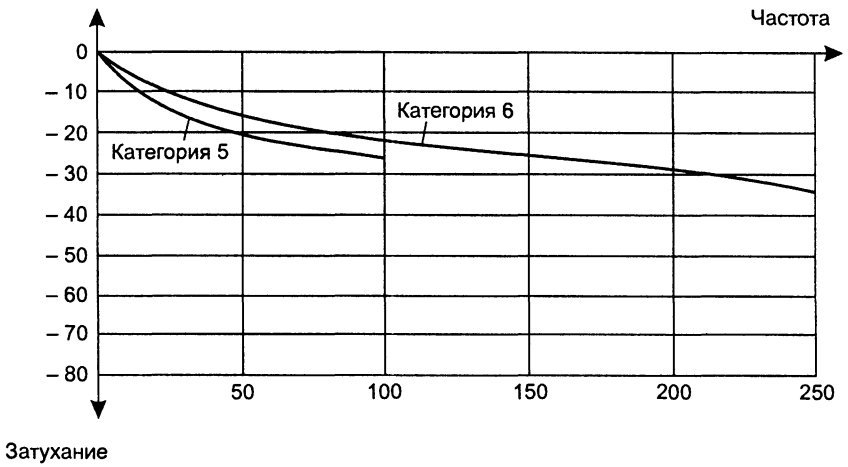


Рис. 8.8. Затухание неэкранированного кабеля на витой паре

Оптический кабель имеет существенно меньшие (по абсолютной величине) величины затухания, обычно в диапазоне от –0,2 до –3 дБ при длине кабеля в 1000 м, а значит, является более качественным, чем кабель на витой паре. Практически все оптические волокна имеют сложную зависимость затухания от длины волны, которая имеет три так называемых **окна прозрачности**. На рис. 8.9 показана характерная зависимость затухания для оптического волокна. Из рисунка видно, что область эффективного использования современных волокон ограничена волнами длин 850 нм, 1300 нм и 1550 нм (соответственно 35 ТГц, 23 ТГц и 19,4 ТГц). Окно 1550 нм обеспечивает наименьшие потери, а значит, максимальную дальность при фиксированной мощности передатчика и фиксированной чувствительности приемника.

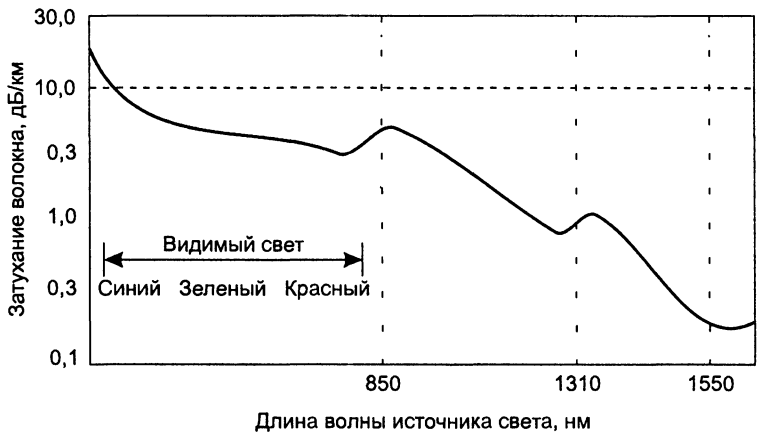


Рис. 8.9. Окна прозрачности оптического волокна

В качестве характеристики мощности сигнала используются абсолютный и относительный уровни мощности. **Абсолютный уровень мощности** измеряется в ваттах, **относительный уровень мощности**, как и затухание, измеряется в децибелах. При этом в качестве базового значения мощности, относительно которого измеряется мощность сигнала, принимается значение в 1 мВт. Таким образом, относительный уровень мощности p вычисляется по следующей формуле:

$$p = 10 \lg P/1\text{мВт} \ [\text{дБм}].$$

Здесь P — абсолютная мощность сигнала в милливаттах, а дБм — единица измерения относительного уровня мощности (децибел на 1 мВт). Относительные значения мощности удобно использовать при *расчетах энергетического бюджета линий связи.*

Пример

Пусть требуется определить относительную минимальную мощность x (дБм) передатчика, достаточную для того, чтобы на выходе линии относительная мощность сигнала была не ниже некоторого порогового значения y (дБм). Затухание линии известно и равно A. Пусть X и Y — абсолютные значения мощности сигнала на входе и выходе линии соответственно. И пусть они заданы в милливаттах. По определению $A = 10 \lg X/Y$. Используя свойства логарифмов имеем:

$$A = 10 \lg X - 10 \lg Y = 10 \lg X/1\ \text{мВт} - 10 \lg Y/1\ \text{мВт}.$$

Заметим, что два последних члена уравнения по определению являются относительными значениями мощности сигналов на выходе и на входе, поэтому приходим к простому соотношению $A = x - y$, из которого следует, что минимальная требуемая мощность передатчика может быть определена как сумма затухания и мощности сигнала на выходе: $x = A + y$.

Предельная простота расчета стала возможной благодаря тому, что в качестве исходных данных были использованы относительные значения мощности входного и выходного сигналов. Использованная в примере величина y называется **порогом чувствительности приемника** и представляет собой минимальную мощность сигнала на входе приемника, при котором он способен корректно распознавать дискретную информацию, содержащуюся в сигнале. Очевидно, что для нормальной работы линии связи необходимо, чтобы минимальная мощность сигнала передатчика, даже ослабленная затуханием линии связи, превосходила порог чувствительности приемника: $x - A > y$. Проверка этого условия и является сутью расчета энергетического бюджета линии.

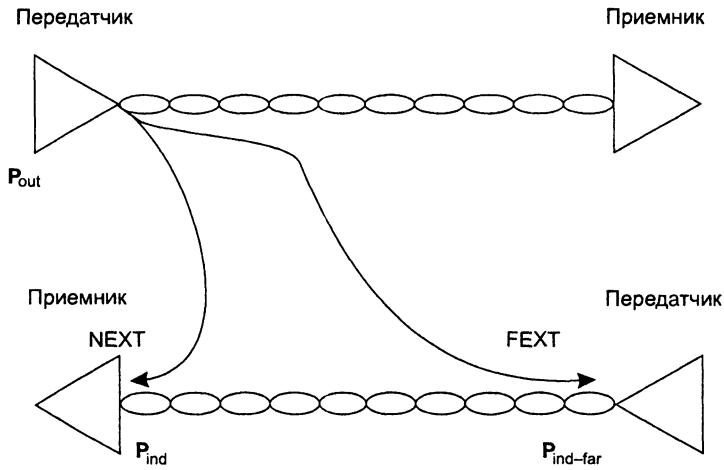
Важным параметром медной линии связи является ее **волновое сопротивление**, представляющее собой полное (комплексное) сопротивление, которое встречает электромагнитная волна определенной частоты при распространении вдоль однородной цепи. Волновое сопротивление измеряется в омах и зависит от таких параметров линии связи, как активное сопротивление, погонная индуктивность и погонная емкость, а также от частоты самого сигнала. Выходное сопротивление передатчика должно быть согласовано с волновым сопротивлением линии, иначе затухание сигнала будет чрезмерно большим.

Помехоустойчивость и достоверность

Помехоустойчивость линии, как и следует из названия, определяет способность линии противостоять влиянию помех, создаваемых во внешней среде или на внутренних проводниках самого кабеля. Помехоустойчивость линии зависит от типа используемой физической среды, а также от экранирующих и подавляющих помехи средств самой линии. Наименее помехоустойчивыми являются радиолинии, хорошей устойчивостью обладают кабельные линии и отличной — волоконно-оптические линии, малочувствительные к внешнему электромагнитному излучению. Обычно для уменьшения помех, создаваемых внешними электромагнитными полями, проводники экранируют и/или скручивают.

Электрическая и магнитная связь — это параметры медного кабеля, также являющиеся результатом помех. **Электрическая связь** определяется отношением наведенного тока в цепи, подверженной влиянию, к напряжению, действующему во влияющей цепи. **Магнитная связь** — это отношение электродвижущей силы, наведенной в цепи, подверженной влиянию, к току во влияющей цепи. Результатом электрической и магнитной связи являются **наведенные сигналы** (наводки) в цепи, подверженной влиянию. Существует несколько различных параметров, характеризующих устойчивость кабеля к наводкам.

Перекрестные наводки на ближнем конце (Near End Cross Talk, NEXT) определяют устойчивость кабеля в том случае, когда наводка образуется в результате действия сигнала, генерируемого передатчиком, подключенным к одной из соседних пар на том же конце кабеля, на котором работает подключенный к подверженной влиянию паре приемник (рис. 8.10). Показатель NEXT, выраженный в децибелах, равен 10 lg P_{out}/P_{ind}, где P_{out} — мощность выходного сигнала, P_{ind} — мощность наведенного сигнала.



$P_{ind-far}$ – мощность наведенного сигнала на дальнем конце кабеля

Рис. 8.10. Переходное затухание

Чем меньше значение NEXT, тем лучше кабель. Так, для витой пары категории 5 показатель NEXT должен быть меньше –27 дБ на частоте 100 МГц.

Перекрестные наводки на дальнем конце (Far End Cross Talk, FEXT) позволяют оценить устойчивость кабеля к наводкам для случая, когда передатчик и приемник подключены к разным концам кабеля. Очевидно, что этот показатель должен быть лучше, чем NEXT, так как до дальнего конца кабеля сигнал приходит ослабленный затуханием каждой пары.

Показатели NEXT и FEXT обычно применяются к кабелю, состоящему из нескольких витых пар, так как в этом случае взаимные наводки одной пары на другую могут достигать значительных величин. Для одинарного коаксиального кабеля (то есть состоящего из одной экранированной жилы) этот показатель не имеет смысла, а для двойного коаксиального кабеля он также не применяется вследствие высокой степени защищенности каждой жилы. Оптические волокна тоже не создают сколько-нибудь заметных взаимных помех.

В связи с тем, что в некоторых новых технологиях данные передаются одновременно по нескольким витым парам, в последнее время стали применяться также показатели перекрестных наводок с приставкой PS (PowerSUM — объединенная наводка), такие как PS NEXT и PS FEXT. Эти показатели отражают устойчивость кабеля к суммарной мощности перекрестных наводок на одну из пар кабеля от всех остальных передающих пар (рис. 8.11).

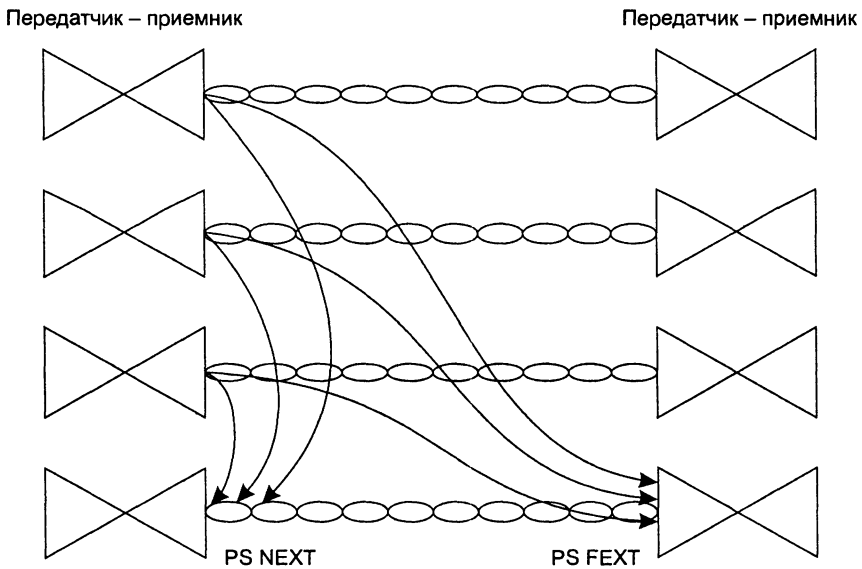


Рис. 8.11. Суммарное переходное затухание

Еще одним практически важным показателем является **защищенность кабеля** (Attenuation/Crosstalk Ratio, ACR). Защищенность определяется как разность между уровнями полезного сигнала и помех. Чем больше значение защищенности кабеля, тем в соответствии с формулой Шеннона с потенциально более высокой

скоростью можно передавать данные по этому кабелю. На рис. 8.12 показана типичная характеристика зависимости защищенности кабеля на неэкранированной витой паре от частоты сигнала.

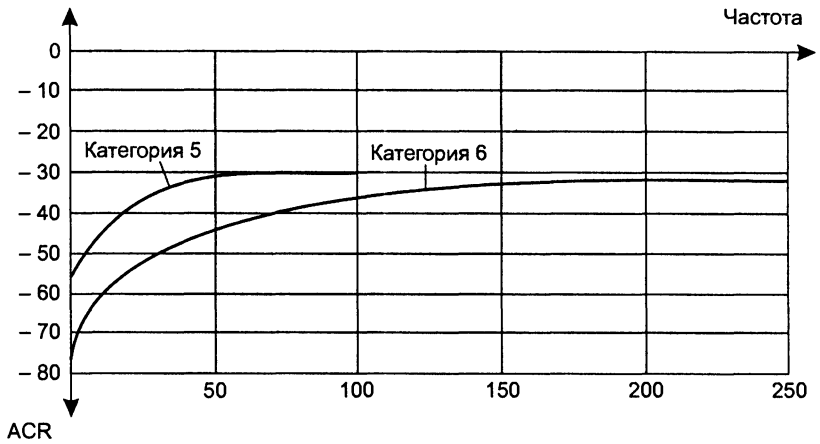


Рис. 8.12. Защищенность витой пары

Достоверность передачи данных характеризует вероятность искажения каждого передаваемого бита данных. Иногда этот же показатель называют **интенсивностью битовых ошибок** (Bit Error Rate, BER). Величина BER для линий связи без дополнительных средств защиты от ошибок (например, самокорректирующихся кодов или протоколов с повторной передачей искаженных кадров) составляет, как правило, 10^{-4}–10^{-6}, в оптоволоконных линиях связи — 10^{-9}. Значение достоверности передачи данных, например 10^{-4}, говорит о том, что в среднем из 10 000 бит искажается значение одного бита.

Полоса пропускания и пропускная способность

Полоса пропускания — это непрерывный диапазон частот, для которого затухание не превышает некоторый заранее заданный предел. То есть полоса пропускания определяет диапазон частот синусоидального сигнала, при которых этот сигнал передается по линии связи без значительных искажений.

Часто граничными частотами считаются частоты, на которых мощность выходного сигнала уменьшается в два раза по отношению к входному, что соответствует затуханию в –3 дБ. Как мы увидим далее, *ширина* полосы пропускания в наибольшей степени влияет на максимально возможную скорость передачи информации по линии связи. Полоса пропускания зависит от типа линии и ее протяженности. На рис. 8.13 показаны полосы пропускания линий связи различных типов, а также наиболее часто используемые в технике связи частотные диапазоны.

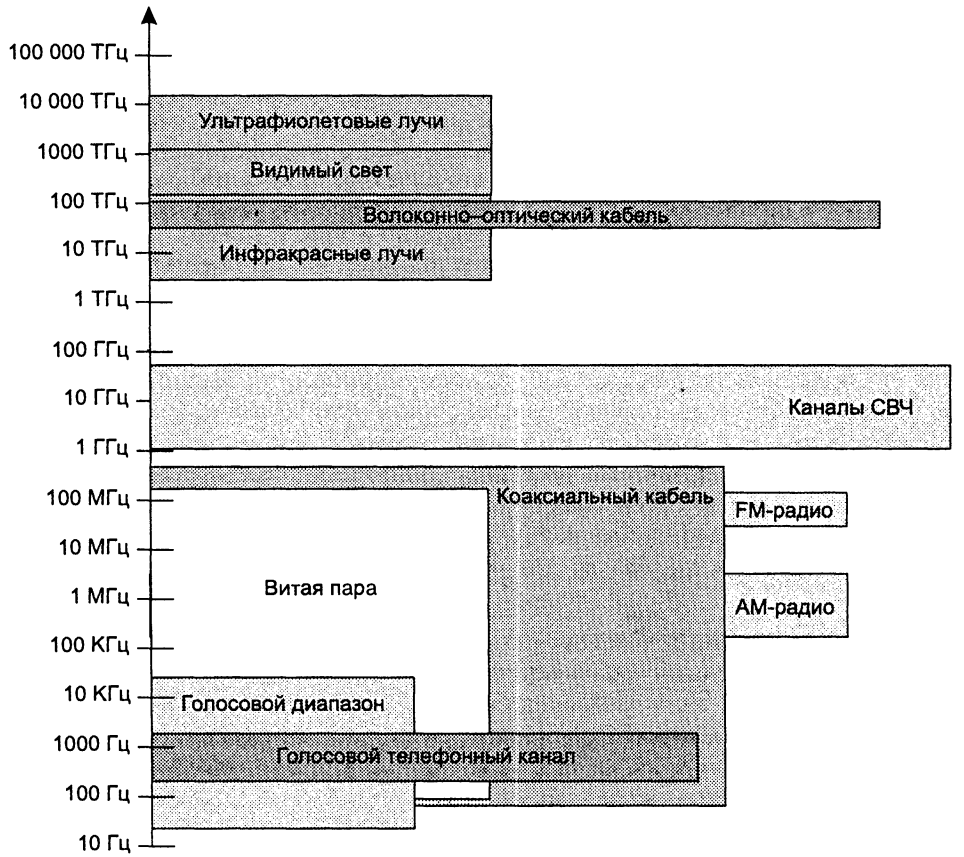


Рис. 8.13. Полосы пропускания линий связи и популярные частотные диапазоны

Пропускная способность линии характеризует максимально возможную скорость передачи данных, которая может быть достигнута на этой линии. Особенностью пропускной способности является то, что, с одной стороны, эта характеристика зависит от параметров *физической среды*, а с другой — определяется *способом передачи данных*. Следовательно, нельзя говорить о пропускной способности линии связи до того, как для нее определен протокол физического уровня.

Например, поскольку для цифровых линий всегда определен протокол физического уровня, задающий битовую скорость передачи данных, то для них всегда известна и пропускная способность — 64 Кбит/с, 2 Мбит/с и т. п.

В тех же случаях, когда только предстоит выбрать, какой из множества существующих протоколов использовать на данной линии, очень важными являются остальные характеристики линии, такие как полоса пропускания, перекрестные наводки, помехоустойчивость и др.

Пропускная способность, как и скорость передачи данных, измеряется в битах в секунду (бит/с), а также в производных единицах, таких как килобит в секунду (Кбит/с) и т. д.

ВНИМАНИЕ

Пропускная способность линий связи и коммуникационного сетевого оборудования традиционно измеряется в битах в секунду, а не в байтах в секунду. Это связано с тем, что данные в сетях передаются последовательно, то есть побитно, а не параллельно, байтами, как это происходит между устройствами внутри компьютера. Такие единицы измерения, как килобит, мегабит или гигабит, в сетевых технологиях строго соответствуют степеням 10 (то есть килобит — это 1000 бит, а мегабит — это 1 000 000 бит), как это принято во всех отраслях науки и техники, а не близким к этим числам степеням двойки, как это принято в программировании, где приставка «кило» равна 2^{10} = 1024, а «мега» — 2^{20} = 1 048 576.

Пропускная способность линии связи зависит не только от ее характеристик, таких как затухание и полоса пропускания, но и от спектра передаваемых сигналов. Если значимые гармоники сигнала (то есть те гармоники, амплитуды которых вносят основной вклад в результирующий сигнал) попадают в полосу пропускания линии, то такой сигнал будет хорошо передаваться данной линией связи, и приемник сможет правильно распознать информацию, отправленную по линии передатчиком (рис. 8.14, *а*). Если же значимые гармоники выходят за границы полосы пропускания линии связи, то сигнал будет значительно искажаться, а приемник будет ошибаться при распознавании информации (рис. 8.14, *б*).

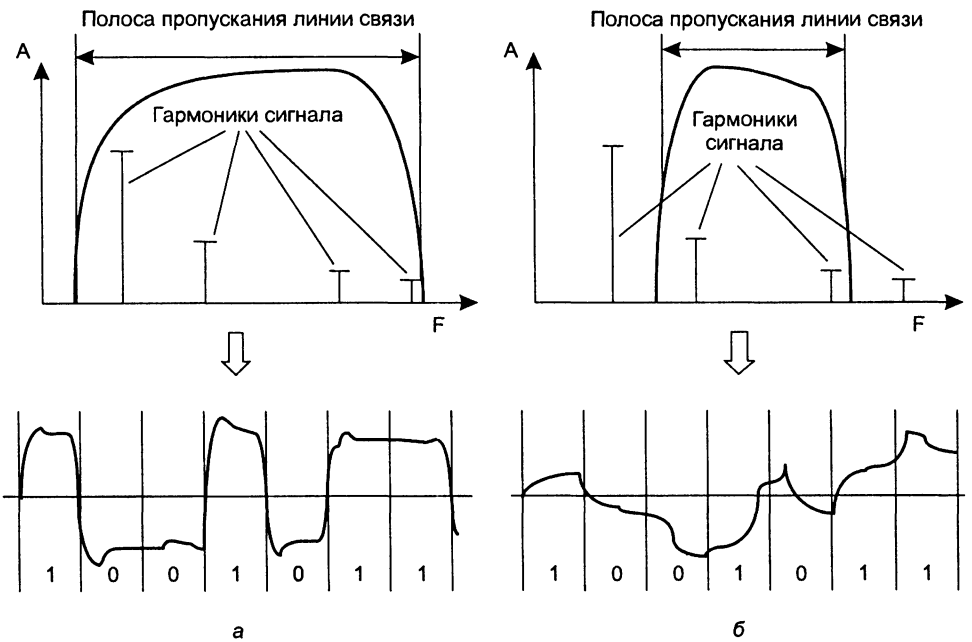


Рис. 8.14. Соответствие между полосой пропускания линии связи и спектром сигнала

Биты и боды

Выбор способа представления дискретной информации в виде сигналов, подаваемых на линию связи, называется **физическим**, или **линейным**, **кодированием**.

От выбранного способа кодирования зависит спектр сигналов и, соответственно, пропускная способность линии.

Таким образом, для одного способа кодирования линия может обладать одной пропускной способностью, а для другого — другой. Например, витая пара категории 3 может передавать данные с пропускной способностью 10 Мбит/с при способе кодирования стандарта физического уровня 10Base-T и 33 Мбит/с при способе кодирования стандарта 100Base-T4.

ВНИМАНИЕ

В соответствии с основным постулатом теории информации любое различимое непредсказуемое изменение принимаемого сигнала несет в себе информацию. Отсюда следует, что синусоида, у которой амплитуда, фаза и частота остаются неизменными, информации не несет, так как изменение сигнала хотя и происходит, но является абсолютно предсказуемым. Аналогично, не несут в себе информации импульсы на тактовой шине компьютера, так как их изменения тоже постоянны во времени. А вот импульсы на шине данных предсказать заранее нельзя, это и делает их информационными, они переносят информацию между отдельными блоками или устройствами компьютера.

В большинстве способов кодирования используется изменение какого-либо параметра периодического сигнала — частоты, амплитуды и фазы синусоиды или же знака потенциала последовательности импульсов. Периодический сигнал, параметры которого подвергаются изменениям, называют **несущим сигналом**, а его частоту, если сигнал синусоидальный, — **несущей частотой**. Процесс изменения параметров несущего сигнала в соответствии с передаваемой информацией называется **модуляцией**.

Если сигнал изменяется так, что можно различить только два его состояния, то любое его изменение будет соответствовать наименьшей единице информации — биту. Если же сигнал может иметь более двух различимых состояний, то любое его изменение будет нести *несколько битов информации.*

Передача дискретной информации в телекоммуникационных сетях осуществляется тактировано, то есть изменение сигнала происходит через фиксированный интервал времени, называемый **тактом**. Приемник информации считает, что вначале каждого такта на его вход поступает новая информация. При этом независимо от того, повторяет ли сигнал состояние предыдущего такта или же он имеет состояние, отличное от предыдущего, приемник получает новую информацию от передатчика. Например, если такт равен 0,3 с, а сигнал имеет два состояния и 1 кодируется потенциалом 5 вольт, то присутствие на входе приемника сигнала 5 вольт в течение 3 секунд означает получение информации, представленной двоичным числом 1111111111.

Количество изменений информационного параметра несущего периодического сигнала в секунду измеряется в **бодах**. Один бод равен одному изменению информационного параметра в секунду. Например, если такт передачи информации равен 0,1 секунды, то сигнал изменяется со скоростью 10 бод. Таким образом, скорость в бодах целиком определяется величиной такта.

Информационная скорость измеряется в битах в секунду и в общем случае *не совпадает* со скоростью в бодах. Она может быть как выше, так и ниже скорости

изменения информационного параметра, измеряемого в *бодах*. Это соотношение зависит от числа состояний сигнала. Например, если сигнал имеет более двух различимых состояний, то при равных тактах и соответствующем методе кодирования информационная скорость в битах в секунду может быть *выше*, чем скорость изменения информационного сигнала в бодах.

Пусть информационными параметрами являются фаза и амплитуда синусоиды, причем различаются 4 состояния фазы в 0, 90, 180 и 270° и два значения амплитуды сигнала, тогда информационный сигнал может иметь 8 различимых состояний. Это означает, что любое состояние этого сигнала несет информацию в 3 бит. В этом случае модем, работающий со скоростью 2400 бод (меняющий информационный сигнал 2400 раз в секунду), передает информацию со скоростью 7200 бит/с, так как при одном изменении сигнала передается 3 бит информации.

Если сигнал имеет два состояния (то есть несет информацию в 1 бит), то информационная скорость обычно совпадает с количеством бодов. Однако может наблюдаться и обратная картина, когда информационная скорость оказывается *ниже* скорости изменения информационного сигнала в бодах. Это происходит в тех случаях, когда для надежного распознавания приемником пользовательской информации каждый бит в последовательности кодируется несколькими изменениями информационного параметра несущего сигнала. Например, при кодировании единичного значения бита импульсом положительной полярности, а нулевого значения бита импульсом отрицательной полярности физический сигнал дважды изменяет свое состояние при передаче каждого бита. При таком кодировании скорость линии в битах в секунду в два раза ниже, чем в бодах.

Чем выше частота несущего периодического сигнала, тем выше может быть частота модуляции и тем выше может быть пропускная способность линии связи.

Однако, с другой стороны, с увеличением частоты периодического несущего сигнала увеличивается и ширина спектра этого сигнала.

Линия передает этот спектр синусоид с теми искажениями, которые определяются ее полосой пропускания. Чем больше несоответствие между полосой пропускания линии и шириной спектра передаваемых информационных сигналов, тем больше сигналы искажаются и тем вероятнее ошибки в распознавании информации принимающей стороной, а значит, возможная скорость передачи информации оказывается меньше.

Соотношение полосы пропускания и пропускной способности

Связь между полосой пропускания линии и ее пропускной способностью вне зависимости от принятого способа физического кодирования установил *Клод Шеннон*:

$$C = F \log_2(1 + P_c/P_ш).$$

Здесь C — пропускная способность линии в битах в секунду, F — ширина полосы пропускания линии в герцах, P_c — мощность сигнала, $P_ш$ — мощность шума.

Из этого соотношения следует, что теоретического предела пропускной способности линии с фиксированной полосой пропускания не существует. Однако на практике такой предел имеется. Действительно, повысить пропускную способность линии можно за счет увеличения мощности передатчика или же уменьшения мощности шума (помех) в линии связи. Обе эти составляющие поддаются изменению с большим трудом. Повышение мощности передатчика ведет к значительному увеличению его габаритов и стоимости. Снижение уровня шума требует применения специальных кабелей с хорошими защитными экранами, что весьма дорого, а также снижения шума в передатчике и промежуточной аппаратуре, чего достичь весьма не просто. К тому же влияние мощностей полезного сигнала и шума на пропускную способность ограничено логарифмической зависимостью, которая растет далеко не так быстро, как прямо-пропорциональная. Так, при достаточно типичном исходном отношении мощности сигнала к мощности шума в 100 раз повышение мощности передатчика в два раза даст только 15 % увеличения пропускной способности линии.

Близким по сути к формуле Шеннона является другое соотношение, полученное *Найквистом*, которое также определяет максимально возможную пропускную способность линии связи, но без учета шума в линии:

$$C = 2F \log_2 M.$$

Здесь M — количество различимых состояний информационного параметра.

Если сигнал имеет два различимых состояния, то пропускная способность равна удвоенному значению ширины полосы пропускания линии связи (рис. 8.15, *а*). Если же передатчик использует более двух устойчивых состояний сигнала для кодирования данных, то пропускная способность линии повышается, так как за один такт работы передатчик передает несколько битов исходных данных, например 2 бита при наличии четырех различимых состояний сигнала (рис. 8.15, *б*).

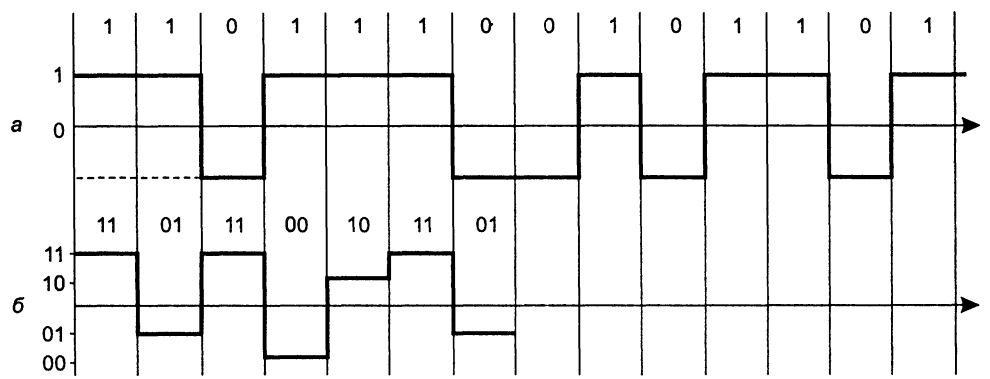


Рис. 8.15. Повышение скорости передачи за счет дополнительных состояний сигнала

Хотя в формуле Найквиста наличие шума в явном виде не учитывается, косвенно его влияние отражается в выборе количества состояний информационного сигнала. Для повышения пропускной способности линии связи следовало бы уве-

личивать количество состояний, но на практике этому препятствует шум на линии. Например, пропускную способность линии, сигнал которой показан на рис. 8.15, *б*, можно увеличить еще в два раза, применив для кодирования данных не 4, а 16 уровней. Однако если амплитуда шума время от времени превышает разницу между соседними уровнями, то приемник не сможет устойчиво распознавать передаваемые данные. Поэтому количество возможных состояний сигнала фактически ограничивается соотношением мощности сигнала и шума, а формула Найквиста определяет предельную скорость передачи данных в том случае, когда количество состояний уже выбрано с учетом возможностей устойчивого распознавания приемником.

Типы кабелей

Список ключевых слов: витая пара, экранированная витая пара, неэкранированная витая пара, коаксиальный кабель, «толстый» коаксиальный кабель, «тонкий» коаксиальный кабель, телевизионный кабель, волоконно-оптический кабель, мода луча, одномодовый кабель, многомодовый кабель, структурированная кабельная система.

Сегодня как для внутренней (кабели зданий), так и для внешней проводки чаще всего применяются три класса проводных линий связи:

- витая пара;
- коаксиальные кабели;
- волоконно-оптические кабели.

Экранированная и неэкранированная витая пара

Витой парой называется скрученная пара проводов. Этот вид среды передачи данных очень популярен и составляет основу большого количества как внутренних, так и внешних кабелей. Кабель может состоять из нескольких скрученных пар (внешние кабели иногда содержат до нескольких десятков таких пар).

Скручивание проводов снижает влияние внешних и взаимных помех на полезные сигналы, передаваемые по кабелю.

Основные особенности конструкции кабелей схематично показаны на рис. 8.16.

Кабели на основе витой пары являются *симметричными*, то есть они состоят из двух одинаковых в конструктивном отношении проводников. Симметричный кабель на основе витой пары может быть как *экранированным*, так и *неэкранированным*.

Нужно отличать *электрическую* изоляцию проводящих жил, которая имеется в любом кабеле, от *электромагнитной* изоляции. Первая состоит из непроводящего диэлектрического слоя — бумаги или полимера, например поливинилхлорида или полистирола. Во втором случае помимо электрической изоляции проводящие жилы помещаются также внутрь электромагнитного экрана, в качестве которого чаще всего применяется проводящая медная оплетка.

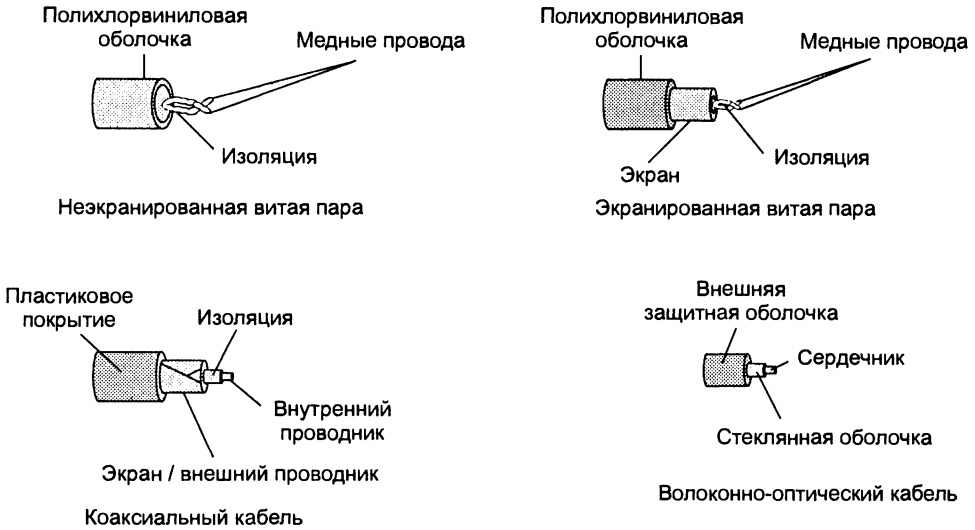


Рис. 8.16. Устройство кабелей

Кабель на основе **неэкранированной витой пары**, используемый для проводки внутри здания, разделяется в международных стандартах на *категории* (от 1 до 7).

- Кабели *категории 1* применяются там, где требования к скорости передачи минимальны. Обычно это кабель для цифровой и аналоговой передачи голоса и низкоскоростной (до 20 Кбит/с) передачи данных. До 1983 года это был основной тип кабеля для телефонной разводки.
- Кабели *категории 2* были впервые применены фирмой IBM при построении собственной кабельной системы. Главное требование к кабелям этой категории — способность передавать сигналы со спектром до 1 МГц.
- Кабели *категории 3* были стандартизованы в 1991 году. Стандарт EIA-568 определил электрические характеристики кабелей для частот в диапазоне до 16 МГц. Кабели категории 3, предназначенные как для передачи данных, так и для передачи голоса, составляют сейчас основу многих кабельных систем зданий.
- Кабели *категории 4* представляют собой несколько улучшенный вариант кабелей категории 3. Кабели категории 4 обязаны выдерживать тесты на частоте передачи сигнала 20 МГц и обеспечивать повышенную помехоустойчивость и низкие потери сигнала. На практике используются редко.
- Кабели *категории 5* были специально разработаны для поддержки высокоскоростных протоколов. Их характеристики определяются в диапазоне до 100 МГц. Большинство высокоскоростных технологий (FDDI, Fast Ethernet, ATM и Gigabit Ethernet) ориентируются на использование витой пары категории 5. Кабель категории 5 пришел на замену кабелю категории 3, и сегодня все новые кабельные системы крупных зданий строятся именно на этом типе кабеля (в сочетании с волоконно-оптическим).

- Особое место занимают кабели *категорий 6 и 7*, которые промышленность начала выпускать сравнительно недавно. Для кабеля категории 6 характеристики определяются до частоты 250 МГц, а для кабелей категории 7 — до 600 МГц. Кабели категории 7 обязательно экранируются, причем как каждая пара, так и весь кабель в целом. Кабель категории 6 может быть как экранированным, так и неэкранированным. Основное назначение этих кабелей — поддержка высокоскоростных протоколов на отрезках кабеля большей длины, чем кабель UTP категории 5.

Все кабели UTP независимо от их категории выпускаются в 4-парном исполнении. Каждая из четырех пар кабеля имеет определенный цвет и шаг скрутки. Обычно две пары предназначены для передачи данных, а две — для передачи голоса.

Экранированная витая пара хорошо защищает передаваемые сигналы от внешних помех, а также меньше излучает электромагнитные колебания вовне, что, в свою очередь, защищает пользователей сетей от вредного для здоровья излучения. Наличие заземляемого экрана удорожает кабель и усложняет его прокладку.

Основным стандартом, определяющим параметры экранированной витой пары для применения внутри зданий, является фирменный *стандарт IBM*. В этом стандарте кабели делятся не на категории, а на *типы* от 1 до 9 включительно.

Рассмотрим для примера кабель *типа 1* стандарта IBM. Он состоит из двух пар скрученных проводов, экранированных проводящей оплеткой, которая заземляется. Электрические параметры кабеля типа 1 примерно соответствуют параметрам кабеля UTP категории 5. Однако волновое сопротивление кабеля типа 1, равное 150 Ом, значительно выше волнового сопротивления UTP категории 5 (100 Ом), поэтому невозможно «улучшение» кабельной проводки сети путем простой замены неэкранированной пары экранированной парой типа 1. Передатчики, рассчитанные на работу с кабелем, имеющим волновое сопротивление 100 Ом, будут плохо работать на волновое сопротивление 150 Ом.

Коаксиальный кабель

Коаксиальный кабель состоит из несимметричных пар проводников. Каждая пара представляет собой внутреннюю медную жилу и соосную с ней внешнюю жилу, которая может быть полой медной трубой или оплеткой, отделенной от внутренней жилы диэлектрической изоляцией. Внешняя жила играет двоякую роль — по ней передаются информационные сигналы и она является экраном, защищающим внутреннюю жилу от внешних электромагнитных полей. Существует несколько типов коаксиального кабеля, отличающихся характеристиками и областями применения: для локальных компьютерных сетей, для глобальных телекоммуникационных сетей, для кабельного телевидения и т. п.

Современные стандарты не считают коаксиальный кабель хорошим вариантом для построения структурированной кабельной системы зданий. Ниже приводятся основные типы и характеристики этих кабелей.

«Толстый» коаксиальный кабель разработан для сетей Ethernet 10Base-5 с волновым сопротивлением 50 Ом и внешним диаметром около 12 мм. Этот кабель имеет достаточно толстый внутренний проводник диаметром 2,17 мм, который обеспечивает хорошие механические и электрические характеристики (затухание на частоте 10 МГц — не хуже 18 дБ/км). Зато этот кабель сложно монтировать — он плохо гнется.

«Тонкий» коаксиальный кабель предназначен для сетей Ethernet 10Base-2. Обладая внешним диаметром около 50 мм и тонким внутренним проводником 0,89 мм, этот кабель не так прочен, как «толстый» коаксиал, зато обладает гораздо большей гибкостью, что удобно при монтаже. «Тонкий» коаксиальный кабель также имеет волновое сопротивление 50 Ом, но его механические и электрические характеристики хуже, чем у «толстого» коаксиального кабеля. Затухание в этом типе кабеля выше, чем в «толстом» коаксиальном кабеле, что приводит к необходимости уменьшать длину кабеля для получения одинакового затухания в сегменте.

Телевизионный кабель с волновым сопротивлением 75 Ом широко применяется в кабельном телевидении. Существуют стандарты локальных сетей, позволяющие использовать такой кабель для передачи данных.

Волоконно-оптический кабель

Волоконно-оптический кабель состоит из тонких (5–60 микрон) гибких стеклянных волокон (волоконных световодов), по которым распространяются световые сигналы. Это наиболее качественный тип кабеля — он обеспечивает передачу данных с очень высокой скоростью (до 10 Гбит/с и выше) и к тому же лучше других типов передающей среды обеспечивает защиту данных от внешних помех (в силу особенностей распространения света такие сигналы легко экранировать).

Каждый световод состоит из центрального проводника света (сердцевины) — стеклянного волокна, и стеклянной оболочки, обладающей меньшим показателем преломления, чем сердцевина. Распространяясь по сердцевине, лучи света не выходят за ее пределы, отражаясь от покрывающего слоя оболочки. В зависимости от распределения показателя преломления и величины диаметра сердечника различают:

- многомодовое волокно со ступенчатым изменением показателя преломления (рис. 8.17, *а*);
- многомодовое волокно с плавным изменением показателя преломления (рис. 8.17, *б*);
- одномодовое волокно (рис. 8.17, *в*).

Понятие «мода» описывает режим распространения световых лучей в сердцевине кабеля.

В **одномодовом кабеле** (Single Mode Fiber, SMF) используется центральный проводник очень малого диаметра, соизмеримого с длиной волны света — от 5 до 10 мкм. При этом практически все лучи света распространяются вдоль оптической оси световода, не отражаясь от внешнего проводника. Изготовление сверх-

тонких качественных волокон для одномодового кабеля представляет собой сложный технологический процесс, что делает одномодовый кабель достаточно дорогим. Кроме того, в волокно такого маленького диаметра достаточно сложно направить пучок света, не потеряв при этом значительную часть его энергии.

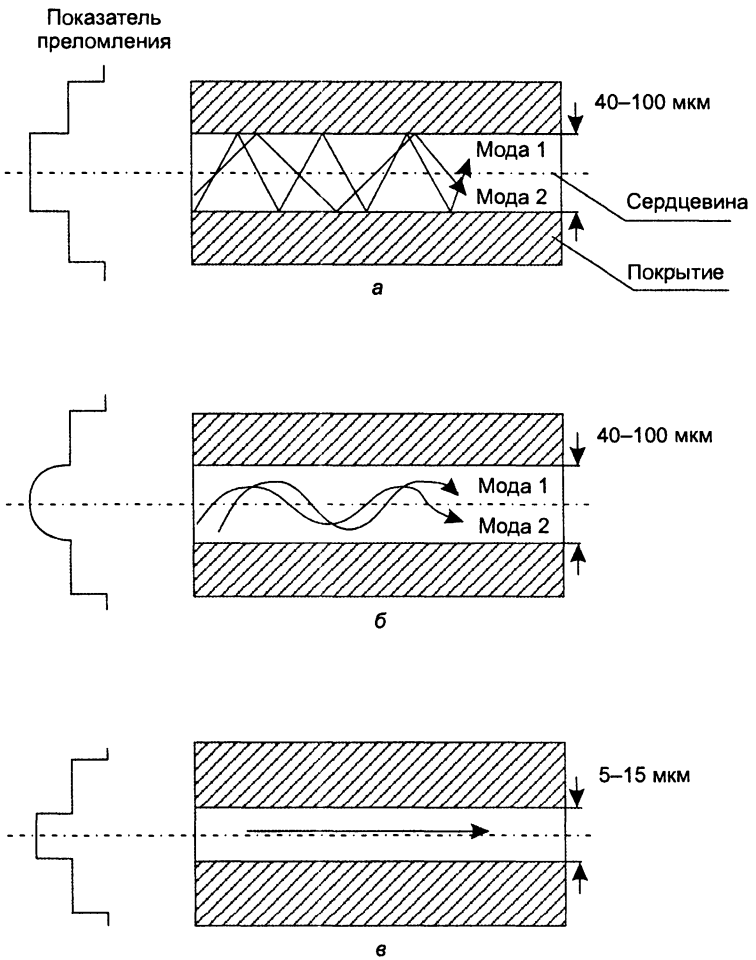


Рис. 8.17. Типы оптического кабеля

В **многомодовых кабелях** (Multi Mode Fiber, MMF) используются более широкие внутренние сердечники, которые легче изготовить технологически. В многомодовых кабелях во внутреннем проводнике одновременно существует несколько световых лучей, отражающихся от внешнего проводника под разными углами. Угол отражения луча называется **модой** луча. В многомодовых кабелях с плавным изменением коэффициента преломления режим отражения лучей имеет сложный характер. Возникающая при этом интерференция ухудшает качество передаваемого сигнала, что приводит к искажениям передаваемых импульсов

в многомодовом оптическом волокне. По этой причине технические характеристики многомодовых кабелей хуже, чем одномодовых.

В результате многомодовые кабели используются в основном для передачи данных на скоростях не более 1 Гбит/с на небольшие расстояния (до 300–2000 м), а одномодовые — для передачи данных со сверхвысокими скоростями в несколько десятков гигабит в секунду (а при использовании технологии DWDM — до нескольких терабит в секунду) на расстояния до нескольких десятков и даже сотен километров (дальняя связь).

В качестве источников света в волоконно-оптических кабелях применяются:

- светодиоды, или светоизлучающие диоды (Light Emitted Diode, LED);
- полупроводниковые лазеры, или лазерные диоды.

Для одномодовых кабелей применяются только лазерные диоды, так как при таком малом диаметре оптического волокна световой поток, создаваемый светодиодом, невозможно без больших потерь направить в волокно — он имеет чересчур широкую диаграмму направленности излучения, в то время как лазерный диод — узкую. Более дешевые светодиодные излучатели используются только для многомодовых кабелей.

Стоимость волоконно-оптических кабелей ненамного превышает стоимость кабелей на витой паре, но проведение монтажных работ с оптоволокном обходится намного дороже из-за трудоемкости операций и высокой стоимости применяемого монтажного оборудования.

Структурированная кабельная система зданий

Структурированная кабельная система (Structured Cabling System, SCS) здания — это набор коммутационных элементов (кабелей, разъемов, коннекторов, кроссовых панелей и шкафов), а также методика их совместного использования, которая позволяет создавать регулярные, легко расширяемые структуры связей в вычислительных сетях. Здание само по себе представляет собой достаточно регулярную структуру — оно состоит из этажей, а каждый этаж, в свою очередь, состоит из определенного количества комнат, соединенных коридорами. Структура здания предопределяет структуру кабельной системы здания.

Структурированная кабельная система здания представляет своего рода «конструктор», с помощью которого проектировщик сети строит нужную ему конфигурацию из стандартных кабелей, соединенных стандартными разъемами и коммутируемых на стандартных кроссовых панелях. При необходимости конфигурацию связей можно легко изменить — добавить компьютер, сегмент, коммутатор, изъять ненужное оборудование, поменять соединение между компьютером и концентратором.

Наиболее детально на сегодня разработаны стандарты кабельных систем зданий, при этом иерархический подход к процессу создания такой кабельной системы позволяет назвать ее структурированной. На основе SCS здания работает одна

или несколько локальных сетей организаций или подразделений одной организации, размещенной в этом здании. SCS планируется и строится иерархически, с главной магистралью и многочисленными ответвлениями от нее (рис. 8.18).

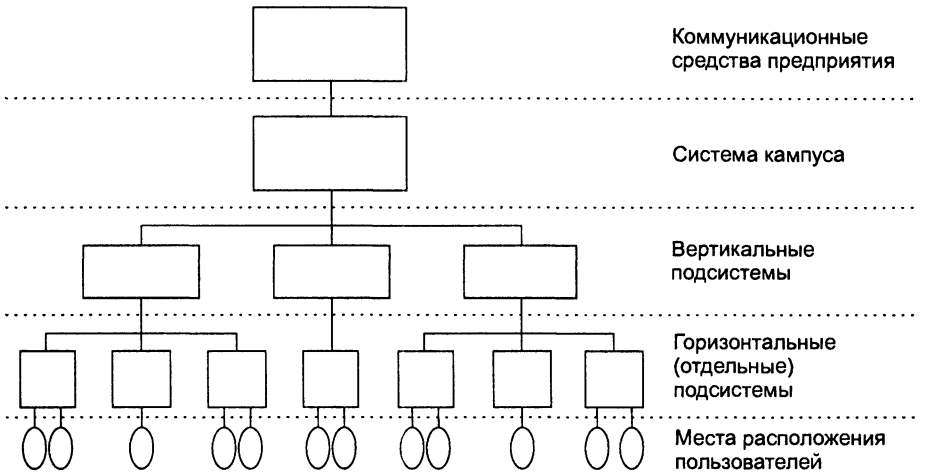


Рис. 8.18. Иерархия структурированной кабельной системы

Типичная иерархическая структура SCS включает (рис. 8.19):

- *горизонтальные подсистемы,* соответствующие этажам здания, — они соединяют кроссовые шкафы этажа с розетками пользователей;
- *вертикальные подсистемы,* соединяющие кроссовые шкафы каждого этажа с центральной аппаратной здания;
- *подсистема кампуса,* объединяющая несколько зданий с главной аппаратной всего кампуса. Эта часть кабельной системы обычно называется магистралью.

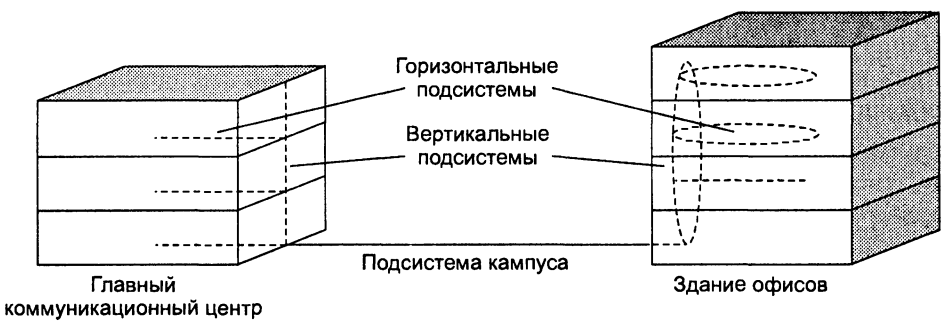


Рис. 8.19. Структура кабельных подсистем

Использование структурированной кабельной системы вместо хаотически проложенных кабелей дает предприятию много преимуществ. Система SCS при

продуманной организации может стать *универсальной средой* для передачи компьютерных данных в локальной вычислительной сети, организации локальной телефонной сети, передачи видеоинформации и даже для передачи сигналов от датчиков пожарной безопасности или охранных систем. Подобная универсализация позволяет автоматизировать многие процессы контроля, мониторинга и управления хозяйственными службами и системами жизнеобеспечения предприятия.

Кроме того, использование SCS делает *более экономичным* добавление новых пользователей и изменения их мест размещения. Известно, что стоимость кабельной системы определяется в основном не стоимостью кабеля, а стоимостью работ по его прокладке. Поэтому выгоднее провести однократную работу по прокладке кабеля, возможно, с большим запасом по длине, чем несколько раз выполнять прокладку, наращивая длину кабеля.

Выводы

В зависимости от типа промежуточной аппаратуры все линии связи делятся на аналоговые и цифровые. В аналоговых линиях промежуточная аппаратура предназначена для усиления аналоговых сигналов. В аналоговых линиях используется частотное мультиплексирование.

В цифровых линиях связи передаваемые сигналы имеют конечное число состояний. В таких линиях используется специальная промежуточная аппаратура — регенераторы, которые улучшают форму импульсов и обеспечивает их ресинхронизацию, то есть восстанавливают период их следования. Промежуточная аппаратура мультиплексирования и коммутации первичных сетей работает по принципу временного мультиплексирования каналов, когда каждому низкоскоростному каналу выделяется определенная доля времени (тайм-слот, или квант) высокоскоростного канала.

Полоса пропускания определяет диапазон частот, которые передаются линией связи с приемлемым затуханием.

Пропускная способность линии связи зависит от ее внутренних параметров, в частности — полосы пропускания, внешних параметров — уровня помех и степени ослабления помех, а также принятого способа кодирования дискретных данных.

Формула Шеннона определяет максимально возможную пропускную способность линии связи при фиксированных значениях полосы пропускания линии и отношении мощности сигнала к шуму.

Формула Найквиста выражает максимально возможную пропускную способность линии связи через полосу пропускания и количество состояний информационного сигнала.

Кабели на основе витой пары делятся на неэкранированные (UTP) и экранированные (STP). Кабели UTP проще в изготовлении и монтаже, зато кабели STP обеспечивают более высокий уровень защищенности.

Волоконно-оптические кабели обладают отличными электромагнитными и механическими характеристиками, недостаток их состоит в сложности и высокой стоимости монтажных работ.

Структурированная кабельная система представляет собой набор коммуникационных элементов — кабелей, разъемов, коннекторов, кроссовых панелей и шкафов, которые удовлетворяют стандартам и позволяют создавать регулярные, легко расширяемые структуры связей.

Вопросы и задания

1. Чем звено отличается от составного канала связи?
2. Может ли составной канал состоять из звеньев? А наоборот?
3. Может ли цифровой канал передавать аналоговые данные?
4. В чем заключаются функции устройств DTE и DCE? К какому из этих двух типов устройств относится сетевой адаптер?
5. К какому типу характеристик линии связи относятся: уровень шума, полоса пропускания, погонная емкость?
6. Какие меры можно предпринять для увеличения информационной скорости звена:
   - уменьшить длину кабеля;
   - выбрать кабель с меньшим сопротивлением;
   - выбрать кабель с более широкой полосой пропускания;
   - применить метод кодирования с более узким спектром.
7. Почему не всегда можно увеличить пропускную способность канала за счет увеличения числа состояний информационного сигнала?
8. За счет какого механизма подавляются помехи в кабелях UTP?
9. Какой кабель более качественно передает сигналы — с большим значением параметра NEXT или с меньшим?
10. Какова ширина спектра идеального импульса?
11. Назовите типы оптического кабеля.
12. Что произойдет, если в работающей сети заменить кабель UTP кабелем STP? Варианты ответов:
    - в сети снизится доля искаженных кадров, так как внешние помехи будут подавляться более эффективно;
    - ничего не изменится;
    - в сети увеличится доля искаженных кадров, так как выходное сопротивление передатчиков не совпадает с импедансом кабеля.
13. Назовите основные преимущества структурированной кабельной системы.
14. Какие типы кабелей используются для горизонтальной подсистемы SCS?
15. Почему проблематично использовать волоконно-оптический кабель в горизонтальной подсистеме?
16. Известными величинами являются:
    - минимальная мощность передатчика P_{out} (дБм);
    - погонное затухание кабеля A (дБ/км);
    - порог чувствительности приемника P_{in} (дБм).

    Требуется найти максимально возможную длину линии связи, при которой сигналы передаются нормально.

17. Каким будет теоретический предел скорости передачи данных в битах в секунду по линии связи с шириной полосы пропускания 20 кГц, если мощность передатчика составляет 0,01 мВт, а мощность шума в линии связи равна 0,0001 мВт?
18. Определите пропускную способность дуплексной линии связи для каждого из направлений, если известно, что ее полоса пропускания равна 600 кГц, а в методе кодирования используется 10 состояний сигнала.
19. Рассчитайте задержку распространения сигнала и задержку передачи данных для случая передачи пакета в 128 байт (считайте скорость распространения сигнала равной скорости света в вакууме 300 000 км/с):
    - по кабелю витой пары длиной в 100 м при скорости передачи 100 Мбит/с;
    - по коаксиальному кабелю длиной в 2 км при скорости передачи в 10 Мбит/с;
    - по спутниковому каналу протяженностью в 72 000 км при скорости передачи 128 Кбит/с.
20. Подсчитайте скорость линии связи, если известно, что тактовая частота передатчика равно 125 МГц, а сигнал имеет 5 состояний.
21. Приемник и передатчик сетевого адаптера подключены к соседним парам кабеля UTP. Какова мощность наведенной помехи на входе приемника, если передатчик имеет мощность 30 дБм, а показатель NEXT кабеля равен –20 дБ?
22. Пусть известно, что модем передает данные в дуплексном режиме со скоростью 33,6 Кбит/с. Сколько состояний имеет его сигнал, если полоса пропускания линии связи равна 3,43 кГц?

ГЛАВА 9 Кодирование и мультиплексирование данных

Проводные среды, которые мы рассмотрели в предыдущей главе, предоставляют только потенциальную возможность передачи дискретной информации. Для того чтобы передатчик и приемник, соединенные некоторой средой, могли обмениваться информацией, им необходимо договориться о том, какие сигналы будут соответствовать двоичным единицам и нулям дискретной информации. Для представления дискретной информации в среде передачи данных применяются сигналы двух типов: прямоугольные импульсы и синусоидальные волны. В первом случае используют термин «кодирование», во втором — «модуляция».

Существует большое количество способов кодирования, которые отличаются шириной спектра сигнала при одной и той же скорости передачи данных. Для передачи данных с минимальным числом ошибок полоса пропускания канала должна быть шире, чем спектр сигнала, иначе выбранные для представления единиц и нулей сигналы значительно исказятся, и приемник не сможет правильно распознать переданную информацию. Поэтому спектр сигнала является одним из главных критериев оценки эффективности способа кодирования.

Кроме того, способ кодирования должен способствовать синхронизации приемника с передатчиком, а также обеспечивать приемлемое соотношение мощности сигнала к шуму. Эти требования являются взаимно противоречивыми, поэтому каждый применяемый на практике способ кодирования представляет собой компромисс между основными требованиями.

Битовые ошибки в каналах связи нельзя исключить полностью, даже если выбранный код обеспечивает хорошую степень синхронизации и высокий уровень отношения сигнала к шуму. Поэтому при передаче дискретной информации применяются специальные коды, которые позволяют обнаруживать (а некоторые даже исправлять) битовые ошибки.

Завершает главу рассмотрение методов мультиплексирования, которые позволяют образовать в одной линии связи несколько каналов передачи.

Модуляция

Список ключевых слов: амплитудная модуляция, частотная модуляция, канал тональной частоты, модем, двоичная частотная манипуляция, четырехуровневая частотная манипуляция, многоуровневая частотная манипуляция, фазовая модуляция, двоичная фазовая манипуляция, квадратурная фазовая манипуляция, квадратурная амплитудная модуляция, решетчатый код, основная гармоника.

Модуляция при передаче аналоговых сигналов

Исторически модуляция начала применяться для *аналоговой информации* и только потом для дискретной.

Необходимость в модуляции аналоговой информации возникает, когда нужно передать низкочастотный аналоговый сигнал через канал, находящийся в высокочастотной области спектра. Примерами такой ситуация является передача голоса по радио или телевидению. Голос имеет спектр шириной примерно в 10 кГц, а радиодиапазоны включают гораздо более высокие частоты, от 30 кГц до 300 мГц. Еще более высокие частоты использует телевидение. Очевидно, что непосредственно голос через такую среду передать нельзя.

Для решения этой проблемы амплитуду высокочастотного несущего сигнала изменяют (модулируют) в соответствии с изменением низкочастотного голосового сигнала (рис. 9.1). При этом спектр результирующего сигнала попадает в нужный высокочастотный диапазон. Такой тип модуляции называется **амплитудной модуляцией** (Amplitude Modulation, AM).

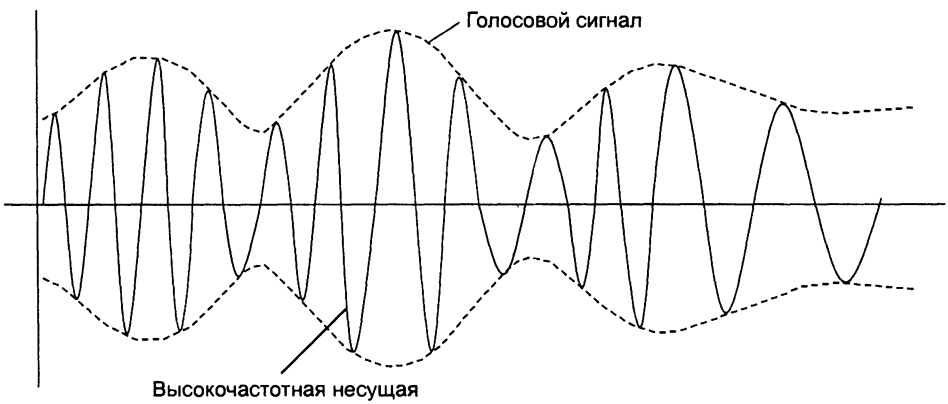


Рис. 9.1. Модуляция голосовым сигналом

В качестве информационного параметра используют не только амплитуду несущего синусоидального сигнала, но и частоту. В этих случаях мы имеем дело с **частотной модуляцией** (Frequency Modulation, FM)[1].

[1] Заметим, что при модуляции аналоговой информации фаза не применяется как информационный параметр.

Модуляция при передаче дискретных сигналов

При передаче *дискретной информации* посредством модуляции единицы и нули кодируются изменением амплитуды, частоты или фазы несущего синусоидального сигнала. В случае когда модулированные сигналы передают дискретную информацию, вместо термина «модуляция» иногда используется термин «манипуляция»: амплитудная манипуляция (Amplitude Shift Keying, ASK), частотная манипуляция (Frequency Shift Keying, FSK), фазовая манипуляция (Phase Shift Keying, PSK).

Пожалуй, самый известный пример применения модуляции при передаче дискретной информации — это передача компьютерных данных по телефонным каналам. Типичная амплитудно-частотная характеристика стандартного абонентского канала, называемого также **каналом тональной частоты**, представлена на рис. 9.2. Этот составной канал проходит через коммутаторы телефонной сети и соединяет телефоны абонентов. Канал тональной частоты передает частоты в диапазоне от 300 до 3400 Гц, таким образом, его полоса пропускания равна 3100 Гц. Такая узкая полоса пропускания вполне достаточна для качественной передачи голоса, однако она недостаточно широка для передачи компьютерных данных в виде прямоугольных импульсов. Решение проблемы было найдено благодаря аналоговой модуляции. Устройство, которое выполняет функцию *модуляции* несущей синусоиды на передающей стороне и обратную функцию *демодуляции* на приемной стороне, носит название **модем** (**мод**улятор-**дем**одулятор).

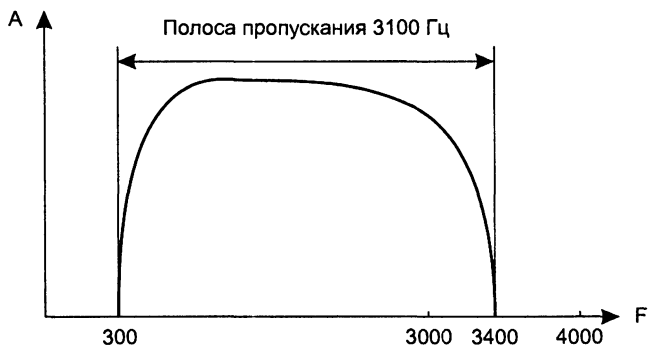


Рис. 9.2. Амплитудно-частотная характеристика канала тональной частоты

На рис. 9.3 показаны различные типы модуляции, применяемые при передаче дискретной информации. Исходная последовательность битов передаваемой информации приведена на диаграмме, представленной на рис. 9.3, *а*.

При *амплитудной модуляции* для логической единицы выбирается один уровень амплитуды синусоиды несущей частоты, а для логического нуля — другой (рис. 9.3, *б*). Этот способ редко используется в чистом виде на практике из-за низкой помехоустойчивости, но часто применяется в сочетании с другим видом модуляции — фазовой модуляцией.

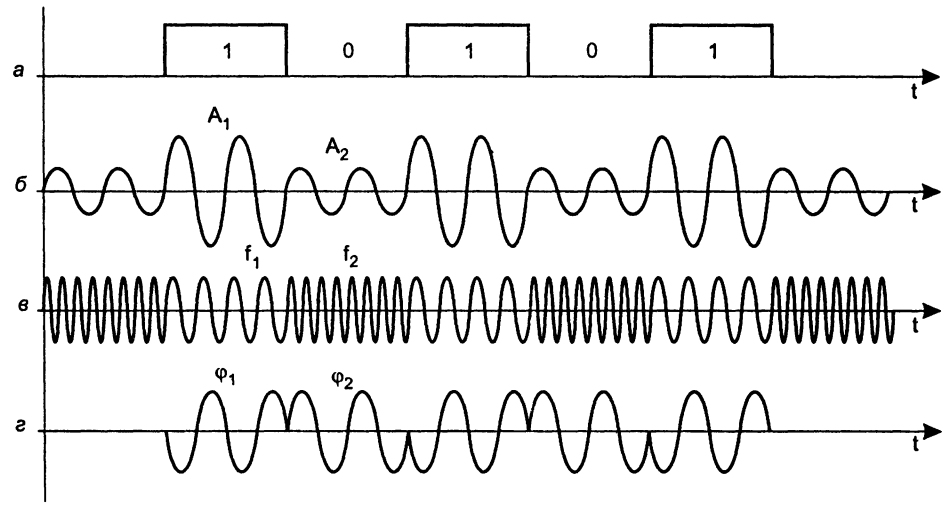


Рис. 9.3. Различные типы модуляции

При *частотной модуляции* значения нуля и единицы исходных данных передаются синусоидами с различной частотой — f_0 и f_1 (рис. 9.3, *в*). Этот способ модуляции не требует сложных схем и обычно применяется в низкоскоростных модемах, работающих на скоростях 300 и 1200 бит/с. При использовании только двух частот за один такт передается один бит информации, поэтому такой способ называется **двоичной частотной манипуляцией** (Binary FSK, BFSK). Может также использоваться четыре различных частоты для кодирования двух битов информации в одном такте, такой способ носит название **четырехуровневой частотной манипуляции** (four-level FSK). Применяется также название **многоуровневая частотная манипуляция** (Multilevel FSK, MFSK).

При **фазовой модуляции** значениям данных 0 и 1 соответствуют сигналы одинаковой частоты, но различной фазы, например 0 и 180° или 0, 90, 180 и 270° (рис. 9.3, *г*). В первом случае такая модуляция носит название **двоичной фазовой манипуляции** (Binary PSK, BPSK), а во втором — **квадратурной фазовой манипуляции** (Quadrature PSK, QPSK).

Комбинированные методы модуляции

Для повышения скорости передачи данных используют комбинированные методы модуляции. Наиболее распространенными являются методы **квадратурной амплитудной модуляции** (Quadrature Amplitude Modulation, QAM). Эти методы основаны на сочетании фазовой и амплитудной модуляций.

На рис. 9.4 показан вариант модуляции, в котором используется 8 различных значений фазы и 4 значения амплитуды. Однако из 32 возможных комбинаций сигнала задействовано только 16, так как разрешенные значения амплитуд у соседних фаз отличаются. Это повышает помехоустойчивость кода, но вдвое снижает скорость передачи данных. Другим решением, повышающим надежность

кода за счет введения избыточности, являются так называемые **решетчатые коды**. Эти коды добавляют к каждым четырем битам информации пятый бит, который даже при наличии ошибок позволяет с большой степенью вероятности определить правильный набор четырех информационных битов.

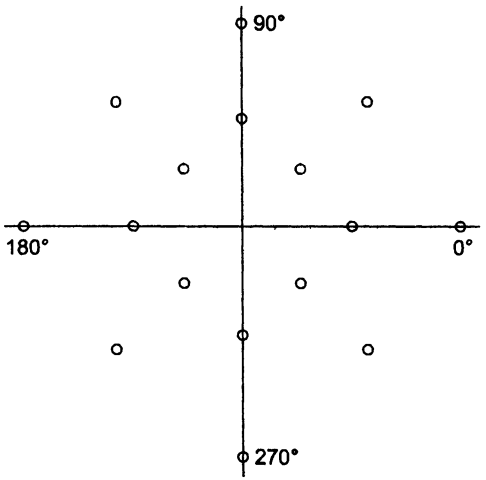


Рис. 9.4. Квадратурная амплитудная модуляция с 16-ю состояниями сигнала

Спектр результирующего модулированного сигнала зависит от *типа модуляции* и скорости модуляции, то есть желаемой *скорости передачи* битов исходной информации.

Рассмотрим сначала спектр сигнала при потенциальном кодировании. Пусть логическая единица кодируется положительным потенциалом, а логический ноль — отрицательным потенциалом такой же величины. Для упрощения вычислений предположим, что передается информация, состоящая из бесконечной последовательности чередующихся единиц и нулей, как показано на рис. 9.3, *а*.

Спектр непосредственно получается из формул Фурье для периодической функции. Если дискретные данные передаются с битовой скоростью N бит/с, то спектр состоит из постоянной составляющей нулевой частоты и бесконечного ряда гармоник с частотами f_0, $3f_0$, $5f_0$, $7f_0$, ..., где $f_0 = N/2$. Частота f_0 — первая частота спектра — называется **основной гармоникой**.

Амплитуды этих гармоник убывают достаточно медленно — с коэффициентами 1/3, 1/5, 1/7, ... от амплитуды гармоники f_0 (рис. 9.5, *а*). В результате спектр потенциального кода требует для качественной передачи широкую полосу пропускания. Кроме того, нужно учесть, что реально спектр сигнала постоянно меняется в зависимости от того, какие данные передаются по линии связи. Например, передача длинной последовательности нулей или единиц сдвигает спектр в сторону низких частот, а в крайнем случае, когда передаваемые данные состоят только из единиц (или только из нулей), спектр состоит из гармоники нулевой частоты. При передаче чередующихся единиц и нулей постоянная составляющая отсутствует. Поэтому спектр результирующего сигнала потенциального кода

при передаче произвольных данных занимает полосу от некоторой величины, близкой к 0 Гц, до примерно $7f_0$ (гармониками с частотами выше $7f_0$ можно пренебречь из-за их малого вклада в результирующий сигнал). Для канала тональной частоты верхняя граница при потенциальном кодировании достигается для скорости передачи данных в 971 бит/с, а нижняя неприемлема для любых скоростей, так как полоса пропускания канала начинается с 300 Гц. В результате потенциальные коды на каналах тональной частоты никогда не используются.

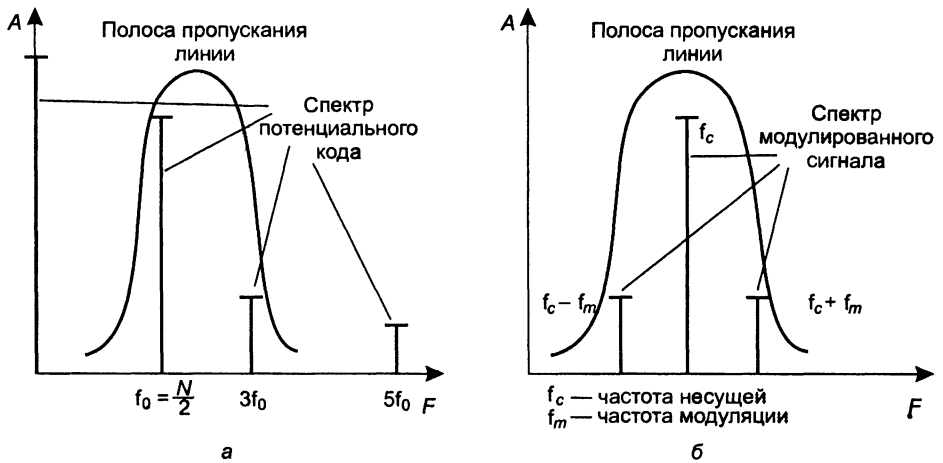


Рис. 9.5. Спектры сигналов при потенциальном кодировании и амплитудной модуляции

При амплитудной модуляции спектр состоит из синусоиды несущей частоты f_c, двух боковых гармоник $(f_c + f_m)$ и $(f_c - f_m)$, а также боковых гармоник $(f_c + 3f_m)$ и $(f_c - 3f_m)$, где f_m — частота изменения информационного параметра синусоиды, которая совпадает со скоростью передачи данных при использовании двух уровней амплитуды (рис. 9.5, *б*). Частота f_m определяет пропускную способность линии при данном способе кодирования. На небольшой частоте модуляции ширина спектра сигнала также оказывается небольшой (равной $2f_m$), если пренебречь гармониками $3f_m$, мощность которых незначительна.

При фазовой и частотной модуляциях спектр сигнала получается более сложным, чем при амплитудной модуляции, так как боковых гармоник здесь образуется более двух, но они тоже симметрично расположены относительно основной несущей частоты, а их амплитуды быстро убывают.

Дискретизация аналоговых сигналов

Список ключевых слов: дискретная модуляция, импульсно-кодовая модуляция, аналого-цифровой преобразователь, цифро-аналоговый преобразователь, элементарный канал цифровых телефонных сетей.

В предыдущем разделе мы познакомились с преобразованием дискретной формы представления информации в аналоговую. В этом разделе рассмотрим обрат-

ную задачу — преобразование и передача аналоговой информации в дискретной форме.

Такая задача была решена на практике в 60-е годы, когда голос начал передаваться по телефонным сетям в виде последовательности единиц и нулей. Основной причиной такого преобразования является невозможность улучшения качества данных, переданных в аналоговой форме, если они существенно исказились при передаче. Сам аналоговый сигнал не дает никаких указаний ни о том, что произошло искажение, ни о том, как его исправить, поскольку форма сигнала может быть любой, в том числе и такой, которую зафиксировал приемник. Улучшение же качества линий, особенно территориальных, требует огромных усилий и капиталовложений. Поэтому на смену аналоговой технике записи и передачи звука и изображения пришла цифровая техника. В этой технике используется так называемая **дискретная модуляция** исходных непрерывных во времени аналоговых процессов.

Импульсно-кодовая модуляция

Рассмотрим принципы дискретной модуляции на примере **импульсно-кодовой модуляции** (Pulse Code Modulation, PCM), которая широко применяется в цифровой телефонии.

Дискретные способы модуляции основаны на дискретизации непрерывных процессов, как по амплитуде, так и по времени (рис. 9.6).

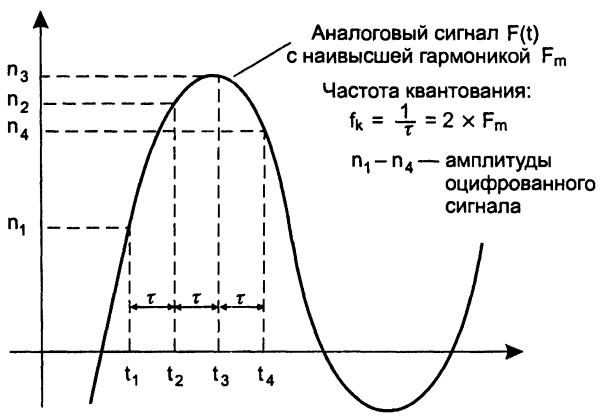


Рис. 9.6. Дискретная модуляция непрерывного процесса

Амплитуда исходной непрерывной функции измеряется с заданным периодом — за счет этого происходит *дискретизация по времени.*

Затем каждый замер представляется в виде двоичного числа определенной разрядности, что означает *дискретизацию по значениям* — непрерывное множество возможных значений амплитуды заменяется дискретным множеством ее значений.

Устройство, которое выполняет подобную функцию, называется **аналого-цифровым преобразователем** (АЦП). После этого замеры передаются по линиям связи в виде последовательности единиц и нулей. При этом применяются те же методы кодирования (с ними мы познакомимся позднее), что и при передаче изначально дискретной информации.

На приемной стороне линии коды преобразуются в исходную последовательность битов, а специальная аппаратура, называемая **цифро-аналоговым преобразователем** (ЦАП), производит демодуляцию оцифрованных амплитуд, восстанавливая исходную непрерывную функцию времени.

Дискретная модуляция основана на теории отображения Найквиста. В соответствии с этой теорией, аналоговая непрерывная функция, переданная в виде последовательности ее дискретных по времени значений, может быть точно восстановлена, если частота дискретизации была в два или более раз выше, чем частота самой высокой гармоники спектра исходной функции.

Если это условие не соблюдается, то восстановленная функция будет существенно отличаться от исходной.

Преимуществом цифровых методов записи, воспроизведения и передачи аналоговой информации является возможность контроля достоверности считанных с носителя или полученных по линии связи данных. Для этого можно применять те же методы, что и в случае компьютерных данных, — вычисление контрольной суммы, повторная передача искаженных кадров, применение самокорректирующихся кодов.

Оцифровывание голоса

Для качественной передачи голоса в методе PCM используется частота квантования амплитуды звуковых колебаний в 8000 Гц. Это связано с тем, что в аналоговой телефонии для передачи голоса был выбран диапазон от 300 до 3400 Гц, который достаточно качественно передает все основные гармоники собеседников. В соответствии с *теоремой Найквиста—Котельникова* для качественной передачи голоса достаточно выбрать частоту дискретизации, в два раза превышающую самую высокую гармонику непрерывного сигнала, то есть $2 \times 3400 = 6800$ Гц. Выбранная в действительности частота дискретизации 8000 Гц обеспечивает некоторый запас качества. В методе PCM обычно используется 7 или 8 бит кода для представления амплитуды одного замера. Соответственно, это дает 127 или 256 градаций звукового сигнала, что оказывается вполне достаточным для качественной передачи голоса.

При использовании метода PCM для передачи одного голосового канала необходима пропускная способность 56 или 64 Кбит/с в зависимости от того, каким количеством битов представляется каждый замер. Если для этих целей применяется 7 бит, то при частоте передачи замеров в 8000 Гц получаем:

$$8000 \times 7 = 56000 \text{ бит/с или } 56 \text{ Кбит/с.}$$

А для случая 8 бит:

$$8000 \times 8 = 64000 \text{ бит/с или } 64 \text{ Кбит/с.}$$

Стандартным является цифровой канал 64 Кбит/с, который также называется **элементарным каналом цифровых телефонных сетей**.

Передача непрерывного сигнала в дискретном виде требует от сетей жесткого соблюдения временного интервала в 125 мкс (соответствующего частоте дискретизации 8000 Гц) между соседними замерами, то есть требует синхронной передачи данных между узлами сети.

При отсутствии синхронности прибывающих замеров исходный сигнал восстанавливается неверно, что приводит к искажению голоса, изображения или другой мультимедийной информации, передаваемой по цифровым сетям. Так, искажение синхронизации в 10 мс может привести к эффекту «эха», а сдвиги между замерами в 200 мс приводят к невозможности распознавания произносимых слов.

В то же время потеря одного замера при соблюдении синхронности между остальными замерами практически не сказывается на воспроизводимом звуке. Это происходит за счет сглаживающих устройств в цифро-аналоговых преобразователях, работа которых основана на свойстве инерционности любого физического сигнала — амплитуда звуковых колебаний не может мгновенно измениться на большую величину.

На качество сигнала после ЦАП влияет не только синхронность поступления на его вход замеров, но и погрешность дискретизации амплитуд этих замеров. В теореме Найквиста—Котельникова предполагается, что амплитуды функции измеряются точно, в то же время использование для их хранения двоичных чисел с ограниченной разрядностью несколько искажает эти амплитуды. Соответственно искажается восстановленный непрерывный сигнал, что называется шумом дискретизации (по амплитуде).

Методы кодирования

Список ключевых слов: спектр сигнала, синхронизация передатчика и приемника, самосинхронизирующийся код, фронт, код без возвращения к нулю, биполярный код с альтернативной инверсией, потенциальный код с инверсией при единице, биполярный импульсный код, манчестерский код, потенциальный код 2D1Q, избыточный код, логический код 4B/5B, запрещенный код, скремблирование, скремблер, дескремблер, код B8ZS, код HDB3, компрессия, или сжатие, адаптивная компрессия, относительное кодирование, символьное подавление, код переменной длины, статистическое кодирование, алгоритм Хафмана.

Выбор способа кодирования

При выборе способа кодирования нужно одновременно стремиться к достижению нескольких целей:

- минимизировать ширину спектра сигнала, полученного в результате кодирования;
- обеспечивать синхронизацию между передатчиком и приемником;
- обеспечивать устойчивость к шумам;

- обнаруживать и по возможности исправлять битовые ошибки;
- минимизировать мощность передатчика.

Более узкий **спектр сигнала** позволяет на одной и той же линии (с одной и той же полосой пропускания) добиваться более высокой скорости передачи данных. Спектр сигнала в общем случае зависит как от способа кодирования, так и от тактовой частоты передатчика. Пусть мы разработали два способа кодирования, причем в каждом такте передается один бит информации. Пусть также в первом способе ширина спектра сигнала F равна тактовой частоте смены сигналов f, то есть F = f, а второй способ дает зависимость F = 0,8f. Тогда при одной и той же полосе пропускания B первый способ позволит передавать данные со скоростью B бит/с, а второй (1/0,8)B = 1,25 B бит/с.

Синхронизация передатчика и приемника нужна для того, чтобы приемник точно знал, в какой момент времени необходимо считывать новую порцию информации с линии связи. При передаче дискретной информации время всегда разбивается на такты одинаковой длительности, и приемник старается считать новый сигнал в середине каждого такта, то есть синхронизировать свои действия с передатчиком.

Проблема синхронизации в сетях решается сложнее, чем при обмене данными между близко расположенными устройствами, например между блоками внутри компьютера или же между компьютером и принтером. На небольших расстояниях хорошо работает схема, основанная на отдельной *тактирующей линии связи* (рис. 9.7), так что информация снимается с линии данных только в момент прихода тактового импульса. В сетях использование этой схемы вызывает трудности из-за неоднородности характеристик проводников в кабелях. На больших расстояниях неравномерность скорости распространения сигнала может привести к тому, что тактовый импульс придет настолько позже или раньше соответствующего сигнала данных, что бит данных будет пропущен или считан повторно. Другой причиной, по которой в сетях отказываются от использования тактирующих импульсов, является экономия проводников в дорогостоящих кабелях.

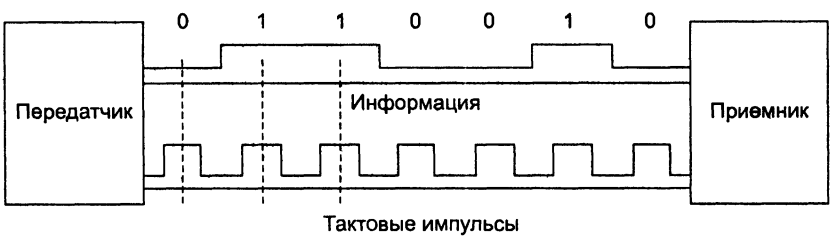


Рис. 9.7. Синхронизация приемника и передатчика на небольших расстояниях

В сетях для решения проблемы синхронизации применяются так называемые **самосинхронизирующиеся коды**, сигналы которых несут для приемника указания о том, в какой момент времени нужно осуществлять распознавание очередного бита (или нескольких битов, если код ориентирован более чем на два состояния сигнала). Любой резкий перепад сигнала — **фронт** — может служить указанием на необходимость синхронизации приемника с передатчиком.

При использовании синусоид в качестве несущего сигнала результирующий код обладает свойством самосинхронизации, так как изменение амплитуды несущей частоты дает возможность приемнику определить момент очередного такта.

Распознавание и коррекцию искаженных данных сложно осуществить средствами физического уровня, поэтому чаще всего эту работу берут на себя протоколы, лежащие выше: канальный, сетевой, транспортный или прикладной. С другой стороны, распознавание ошибок на физическом уровне экономит время, так как приемник не ждет полного помещения кадра в буфер, а отбраковывает его сразу при распознавании ошибочных битов внутри кадра.

Требования, предъявляемые к методам кодирования, являются взаимно противоречивыми, поэтому каждый из рассматриваемых ниже популярных методов кодирования обладает своими достоинствами и недостатками в сравнении с другими.

Потенциальный код NRZ

Рисунок 9.8, *а* иллюстрирует уже упомянутый ранее метод *потенциального кодирования*, называемый также кодированием **без возвращения к нулю** (Non Return to Zero, NRZ). Последнее название отражает то обстоятельство, что при передаче последовательности единиц сигнал, в отличие от других методов кодирования, не возвращается к нулю в течение такта.

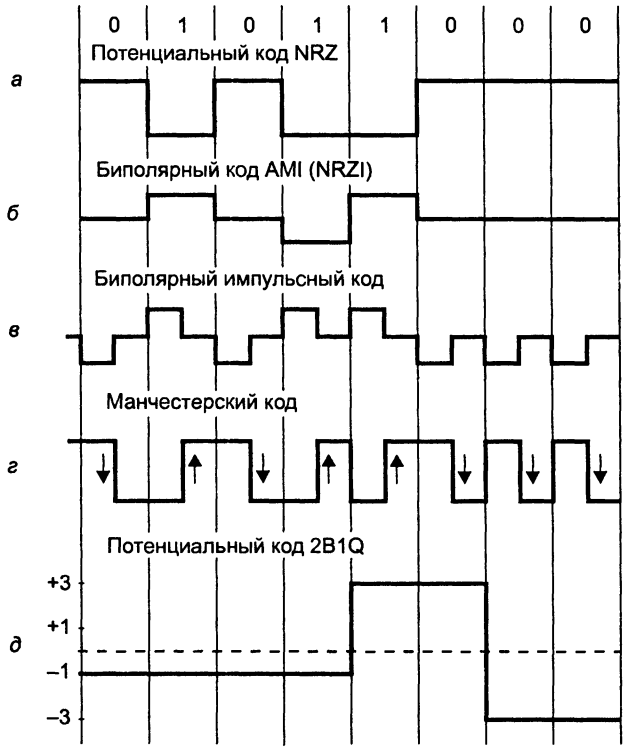


Рис. 9.8. Способы дискретного кодирования данных

Итак, достоинства метода NRZ.

- Простота реализации.
- Метод обладает хорошей распознаваемостью ошибок (благодаря наличию двух резко отличающихся потенциалов).
- Основная гармоника f_0 имеет достаточно низкую частоту (равную N/2 Гц, как было показано в предыдущем разделе), что приводит к узкому спектру.

Теперь недостатки метода NRZ.

- Метод не обладает свойством самосинхронизации. Даже при наличии высокоточного тактового генератора приемник может ошибиться с выбором момента съема данных, так как частоты двух генераторов никогда не бывают полностью идентичными. Поэтому при высоких скоростях обмена данными и длинных последовательностях единиц или нулей небольшое рассогласование тактовых частот может привести к ошибке в целый такт и, соответственно, считыванию некорректного значения бита.
- Вторым серьезным недостатком метода NRZ является наличие низкочастотной составляющей, которая приближается к постоянному сигналу при передаче длинных последовательностей единиц или нулей. Из-за этого многие линии связи, не обеспечивающие прямого гальванического соединения между приемником и источником, этот вид кодирования не поддерживают. Поэтому в сетях код NRZ в основном используется в виде различных его модификаций, в которых устранены как плохая самосинхронизация кода NRZ, так и проблемы постоянной составляющей.

Биполярное кодирование AMI

Одной из модификаций метода NRZ является метод биполярного кодирования **с альтернативной инверсией** (Alternate Mark Inversion, AMI). В этом методе применяются три уровня потенциала — отрицательный, нулевой и положительный (рис. 9.8, *б*). Для кодирования логического нуля используется нулевой потенциал, а логическая единица кодируется либо положительным потенциалом, либо отрицательным, при этом потенциал каждой новой единицы противоположен потенциалу предыдущей.

Код AMI частично решает проблемы наличия постоянной составляющей и отсутствия самосинхронизации, присущие коду NRZ. Это происходит при передаче *длинных последовательностей единиц*. В этих случаях сигнал на линии представляет собой последовательность разнополярных импульсов с тем же спектром, что и у кода NRZ, передающего чередующиеся нули и единицы, то есть без постоянной составляющей и с основной гармоникой N/2 Гц (где N — битовая скорость передачи данных). *Длинные же последовательности нулей* так же опасны для кода AMI, как и для кода NRZ — сигнал вырождается в постоянный потенциал нулевой амплитуды.

В целом, для различных комбинаций битов на линии использование кода AMI приводит к *более узкому спектру сигнала*, чем для кода NRZ, а значит, и к более

высокой пропускной способности линии. Например, при передаче чередующихся единиц и нулей основная гармоника f_0 имеет частоту N/4 Гц.

Код AMI предоставляет также некоторые возможности по распознаванию ошибочных сигналов. Так, нарушение строгой очередности в полярности сигналов говорит о ложном импульсе или исчезновении с линии корректного импульса.

В коде AMI используются не два, а три уровня сигнала на линии. Дополнительный уровень требует увеличение мощности передатчика примерно на 3 дБ для обеспечения той же достоверности приема битов на линии, что является общим недостатком кодов с несколькими состояниями сигнала по сравнению с кодами, которые различают только два состояния.

Потенциальный код NRZI

Существует код, похожий на AMI, но только с двумя уровнями сигнала. При передаче нуля он передает потенциал, который был установлен на предыдущем такте (то есть не меняет его), а при передаче единицы потенциал инвертируется на противоположный. Этот код называется **потенциальным кодом с инверсией при единице** (Non Return to Zero with ones Inverted, NRZI). Он удобен в тех случаях, когда наличие третьего уровня сигнала весьма нежелательно, например в оптических кабелях, где устойчиво распознаются только два состояния сигнала — свет и темнота.

Для улучшения потенциальных кодов, подобных AMI и NRZI, используются два метода. Первый метод основан на добавлении в исходный код избыточных битов, содержащих логические единицы. Очевидно, что в этом случае длинные последовательности нулей прерываются, и код становится самосинхронизирующимся для любых передаваемых данных. Исчезает также постоянная составляющая, а значит, еще более сужается спектр сигнала. Но этот метод снижает полезную пропускную способность линии, так как избыточные единицы пользовательской информации не несут.

Другой метод основан на предварительном «перемешивании» исходной информации таким образом, чтобы вероятность появления единиц и нулей на линии становилась близкой. Устройства, или блоки, выполняющие такую операцию, называются **скремблерами**. При скремблировании используется известный алгоритм, поэтому приемник, получив двоичные данные, передает их на **дескремблер**, который восстанавливает исходную последовательность битов.

Биполярный импульсный код

Помимо *потенциальных кодов* в сетях используются и *импульсные коды*, в которых данные представлены полным импульсом или же его частью — фронтом. Наиболее простым случаем такого подхода является **биполярный импульсный код**, в котором единица представляется импульсом одной полярности, а ноль — другой (рис. 9.8, *в*). Каждый импульс длится половину такта. Такой код обладает отличными самосинхронизирующими свойствами, но постоянная составляющая может присутствовать, например, при передаче длинной последовательности

единиц или нулей. Кроме того, спектр у него шире, чем у потенциальных кодов. Так, при передаче всех нулей или единиц частота основной гармоники кода будет равна N Гц, что в два раза выше основной гармоники кода NRZ и в четыре раза выше основной гармоники кода AMI при передаче чередующихся единиц и нулей. Из-за слишком широкого спектра биполярный импульсный код используется редко.

Манчестерский код

В локальных сетях до недавнего времени самым распространенным методом кодирования был так называемый **манчестерский код** (рис. 9.8, *г*). Он применяется в технологиях Ethernet и Token Ring.

В манчестерском коде для кодирования единиц и нулей используется перепад потенциала, то есть фронт импульса. При манчестерском кодировании каждый такт делится на две части. Информация кодируется перепадами потенциала, происходящими в середине каждого такта. Единица кодируется перепадом от низкого уровня сигнала к высокому, а ноль — обратным перепадом. В начале каждого такта может происходить служебный перепад сигнала, если нужно представить несколько единиц или нулей подряд. Так как сигнал изменяется по крайней мере один раз за такт передачи одного бита данных, то манчестерский код обладает хорошими самосинхронизирующими свойствами. Полоса пропускания манчестерского кода уже, чем у биполярного импульсного. У него также нет постоянной составляющей, а основная гармоника в худшем случае (при передаче последовательности единиц или нулей) имеет частоту N Гц, а в лучшем (при передаче чередующихся единиц и нулей) — N/2 Гц, как и у кодов AMI и NRZ. В среднем ширина полосы манчестерского кода в полтора раза уже, чем у биполярного импульсного кода, а основная гармоника колеблется вблизи значения 3N/4. Манчестерский код имеет еще одно преимущество перед биполярным импульсным кодом. В последнем для передачи данных используются три уровня сигнала, а в манчестерском — два.

Потенциальный код 2B1Q

На рис. 9.8, *д* показан потенциальный код с четырьмя уровнями сигнала для кодирования данных. Это код **2B1Q**, название которого отражает его суть — каждые два бита (2B) передаются за один такт (1) сигналом, имеющим четыре состояния (Q — Quadra). Паре битов 00 соответствует потенциал –2,5 В, паре 01 — потенциал –0,833 В, паре 11 — потенциал +0,833 В, а паре 10 — потенциал +2,5 В.

При этом способе кодирования требуются дополнительные меры по борьбе с длинными последовательностями одинаковых пар битов, так как при этом сигнал превращается в постоянную составляющую. При случайном чередовании битов спектр сигнала в два раза уже, чем у кода NRZ, так как при той же битовой скорости длительность такта увеличивается в два раза. Таким образом, с помощью кода 2B1Q можно по одной и той же линии передавать данные в два раза быстрее, чем с помощью кода AMI или NRZI. Однако для его реализации мощность

передатчика должна быть выше, чтобы четыре уровня четко различались приемником на фоне помех.

Для улучшения потенциальных кодов типа AMI, NRZI или 2Q1B используются избыточные коды и скремблирование.

Избыточный код 4B/5B

Избыточные коды основаны на разбиении исходной последовательности битов на порции, которые часто называют *символами*. Затем каждый исходный символ заменяется новым с бóльшим количество битов, чем исходный.

Например, в логическом коде **4B/5B**, используемом в технологиях FDDI и Fast Ethernet, исходные символы длиной 4 бит заменяются символами длиной 5 бит. Так как результирующие символы содержат избыточные биты, то общее количество битовых комбинаций в них больше, чем в исходных. Так, в коде 4B/5B результирующие символы могут содержать 32 битовых комбинации, в то время как исходные символы — только 16 (табл. 9.1). Поэтому в результирующем коде можно отобрать 16 таких комбинаций, которые не содержат большого количества нулей, а остальные считать **запрещенными кодами** (code violations). Помимо устранения постоянной составляющей и придания коду свойства самосинхронизации, избыточные коды позволяют приемнику распознавать искаженные биты. Если приемник принимает запрещенный код, значит, на линии произошло искажение сигнала.

Таблица 9.1. Соответствие исходных и результирующих кодов 4B/5B

Исходный код	Результирующий код	Исходный код	Результирующий код
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

После разбиения получившийся код 4B/5B передается по линии путем преобразования с помощью какого-либо из методов потенциального кодирования, чувствительного только к длинным последовательностям нулей. Символы кода 4B/5B длиной 5 бит гарантируют, что при любом их сочетании на линии не встретятся более трех нулей подряд.

ПРИМЕЧАНИЕ

Буква B в названии кода 4B/5B означает, что элементарный сигнал имеет два состояния (от английского binary — двоичный). Имеются также коды и с тремя состояниями сигнала, например, в коде 8B/6T для кодирования 8 бит исходной информации используется код из 6 сигналов, каждый из которых имеет три состояния. Избыточность кода 8B/6T выше, чем кода 4B/5B, так как на 256 исходных кодов приходится $3^6 = 729$ результирующих символов.

Использование таблицы перекодировки является очень простой операцией, поэтому этот подход не усложняет сетевые адаптеры и интерфейсные блоки коммутаторов и маршрутизаторов.

Для обеспечения заданной пропускной способности линии передатчик, использующий избыточный код, должен работать с повышенной тактовой частотой. Так, для передачи кодов 4B/5B со скоростью 100 Мбит/с передатчик должен работать с тактовой частотой 125 МГц. При этом спектр сигнала на линии расширяется по сравнению со случаем, когда по линии передается не избыточный код. Тем не менее спектр избыточного потенциального кода оказывается у́же спектра манчестерского кода, что оправдывает дополнительный этап логического кодирования, а также работу приемника и передатчика на повышенной тактовой частоте.

Скремблирование

Скремблирование заключается в побитном вычислении результирующего кода на основании битов исходного кода и полученных в предыдущих тактах битов результирующего кода. Например, скремблер может реализовывать следующее соотношение:

$$B_i = A_i B_{i-3} B_{i-5}.$$

Здесь B_i — двоичная цифра результирующего кода, полученная на i-м такте работы скремблера, A_i — двоичная цифра исходного кода, поступающая на i-м такте на вход скремблера, B_{i-3} и B_{i-5} — двоичные цифры результирующего кода, полученные на предыдущих тактах работы скремблера (соответственно на 3 и на 5 тактов ранее текущего такта) и объединенные операцией исключающего ИЛИ (сложение по модулю 2).

Например, для исходной последовательности 110110000001 скремблер даст следующий результирующий код (первые три цифры результирующего кода будут совпадать с исходным кодом, так как еще нет нужных предыдущих цифр):

$$B_1 = A_1 = 1$$
$$B_2 = A_2 = 1$$
$$B_3 = A_3 = 0$$
$$B_4 = A_4 B_1 = 1\ 1 = 0$$
$$B_5 = A_5 B_2 = 1\ 1 = 0$$
$$B_6 = A_6 B_3 B_1 = 0\ 0\ 1 = 1$$
$$B_7 = A_7 B_4 B_2 = 0\ 0\ 1 = 1$$

$$B_8 = A_8\ B_5\ B_3 = 0\ 0\ 0 = 0$$
$$B_9 = A_9\ B_6\ B_4 = 0\ 1\ 0 = 1$$
$$B_{10} = A_{10}\ B_7\ B_5 = 0\ 1\ 0 = 1$$
$$B_{11} = A_{11}\ B_8\ B_6 = 0\ 0\ 1 = 1$$
$$B_{12} = A_{12}\ B_9\ B_7 = 1\ 1\ 1 = 1$$

Таким образом, на выходе скремблера появится последовательность 110001101111, в которой нет последовательности из шести нулей, присутствовавшей в исходном коде.

После получения результирующей последовательности приемник передает ее дескремблеру, который восстанавливает исходную последовательность на основании обратного соотношения:

$$C_i = B_i\ B_{i-3}\ B_{i-5} = (A_i\ B_{i-3}\ B_{i-5})\ B_{i-3}\ B_{i-5} = A_i.$$

Различные алгоритмы скремблирования отличаются количеством слагаемых, дающих цифру результирующего кода, и сдвигом между слагаемыми. Так, в сетях ISDN при передаче данных от сети к абоненту используется преобразование со сдвигами на 5 и 23 позиции, а при передаче данных от абонента в сеть — со сдвигами на 18 и 23 позиции.

Существуют и более простые методы борьбы с последовательностями единиц, также относимые к классу скремблирования. Для улучшения биполярного кода AMI используются два метода, основанные на искусственном искажении последовательности нулей *запрещенными символами*.

Рисунок 9.9 иллюстрирует использование методов **B8ZS** (Bipolar with 8-Zeros Substitution) и **HDB3** (High-Density Bipolar 3-Zeros) для корректировки кода AMI. Исходный код состоит из двух длинных последовательностей нулей: в первом случае — из 8, а во втором — из 5.

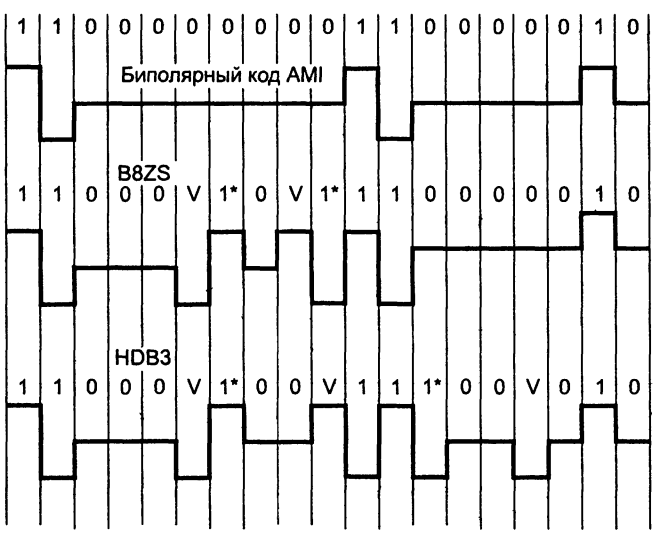
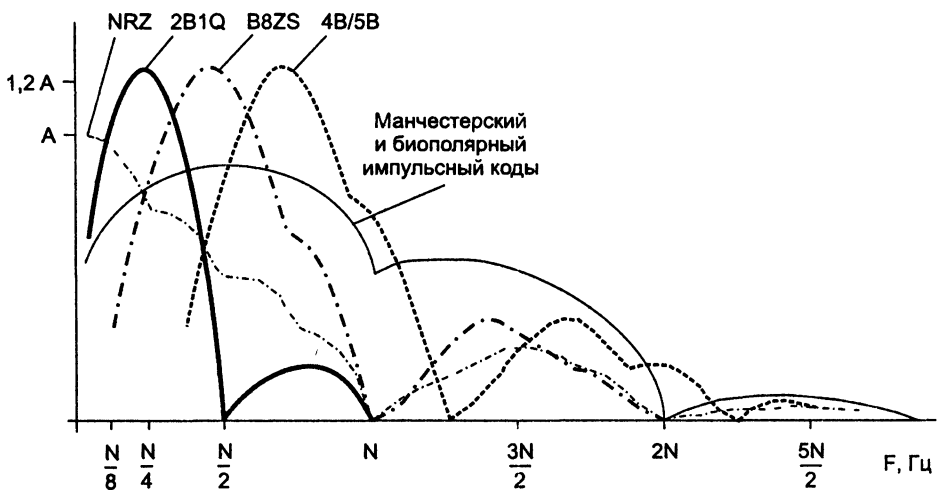


Рис. 9.9. Коды B8ZS и HDB3

Код B8ZS исправляет только последовательности, состоящие из 8 нулей. Для этого он после первых трех нулей вместо оставшихся пяти нулей вставляет пять цифр: V-1*-0-V-1*. Здесь V обозначает сигнал единицы, запрещенной (Violations) для данного такта полярности, то есть сигнал, не изменяющий полярность предыдущей единицы, 1* — сигнал единицы корректной полярности (знак звездочки отмечает тот факт, что в исходном коде в этом такте была не единица, а ноль). В результате на 8 тактах приемник наблюдает 2 искажения — очень маловероятно, что это случается из-за шума на линии или других сбоев передачи. Поэтому приемник считает такие нарушения кодировкой 8 последовательных нулей и после приема заменяет их исходными 8 нулями. Код B8ZS построен так, что его постоянная составляющая равна нулю при любых последовательностях двоичных цифр.

Код HDB3 исправляет любые четыре смежных нуля в исходной последовательности. Правила формирования кода HDB3 более сложные, чем кода B8ZS. Каждые четыре нуля заменяются четырьмя сигналами, в которых имеется один сигнал V. Для подавления постоянной составляющей полярность сигнала V чередуется при последовательных заменах. Кроме того, для замены используются два образца четырехтактовых кодов. Если перед заменой исходный код содержал нечетное число единиц, задействуется последовательность 000V, а если число единиц было четным — последовательность 1*00V.

Улучшенные потенциальные коды обладают достаточно узкой полосой пропускания для любых последовательностей единиц и нулей, которые встречаются в передаваемых данных. На рис. 9.10 приведены спектры сигналов разных кодов, полученные при передаче произвольных данных, в которых различные сочетания нулей и единиц в исходном коде равновероятны.



N — скорость передачи данных, бит/с;
A — амплитуда сигнала

Рис. 9.10. Спектры потенциальных и импульсных кодов

При построении графиков спектр усреднялся по всем возможным наборам исходных последовательностей. Естественно, что результирующие коды могут иметь и другое распределение нулей и единиц. Из рисунка видно, что потенциальный код NRZ обладает хорошим спектром с одним недостатком — у него имеется постоянная составляющая. Коды, полученные из потенциального кода путем логического кодирования, обладают более узким спектром, чем манчестерский код, даже при повышенной тактовой частоте (на рисунке спектр кода 4B/5B должен был бы примерно совпадать с кодом B8ZS, но он сдвинут в область более высоких частот, так как его тактовая частота повышена на 1/4 по сравнению с другими кодами). Этим объясняется преимущественное применение потенциальных избыточных и скремблированных кодов в современных технологиях, подобных FDDI, Fast Ethernet, Gigabit Ethernet, ISDN и т. п. вместо манчестерского и биполярного импульсного кодирования.

Компрессия данных

Компрессия, или **сжатие**, данных применяется для сокращения времени их передачи. Так как на компрессию данных передающая сторона тратит дополнительное время, к которому нужно еще прибавить аналогичные затраты времени на декомпрессию этих данных принимающей стороной, то выгоды от сокращения времени на передачу сжатых данных обычно бывают заметны только на низкоскоростных каналах. Соответствующий порог скорости для современной аппаратуры составляет около 64 Кбит/с. Многие программные и аппаратные средства сети способны выполнять *динамическую компрессию* данных в отличие от *статической*, когда данные сначала компрессируются (например, с помощью популярных архиваторов типа WinZip), а уже затем отсылаются в сеть.

На практике может использоваться ряд алгоритмов компрессии, каждый из которых применим к определенному типу данных. Некоторые модемы (называемые интеллектуальными) предлагают **адаптивную компрессию**, при которой в зависимости от передаваемых данных выбирается определенный алгоритм компрессии. Рассмотрим некоторые из общих алгоритмов компрессии данных.

Когда данные состоят только из чисел, значительную экономию можно получить путем уменьшения количества используемых на цифру битов с 7 до 4, используя простое двоичное кодирование десятичных цифр вместо кода ASCII. Просмотр таблицы ASCII показывает, что старшие три бита всех кодов десятичных цифр содержат комбинацию 011. Если все данные в кадре информации состоят из десятичных цифр, то, поместив в заголовок кадра соответствующий управляющий символ, можно существенно сократить длину кадра. Этот метод носит название **десятичной упаковки**.

Альтернативой десятичной упаковке при передаче числовых данных с небольшими отклонениями между последовательными цифрами является передача только этих отклонений вместе с известным опорным значением. Такой метод называется **относительным кодированием** и используется, в частности, при цифровом кодировании голоса с помощью кода ADPCM, когда в каждом такте передается только разница между соседними замерами голоса.

Часто передаваемые данные содержат большое количество повторяющихся байтов. Например, при передаче черно-белого изображения черные поверхности будут порождать большое количество нулевых значений, а максимально освещенные участки изображения — большое количество байтов, состоящих из всех единиц. Передатчик сканирует последовательность передаваемых байтов, и если обнаруживает последовательность из трех или более одинаковых байтов, заменяет ее специальной трехбайтовой последовательностью, в которой указывает значение байта, количество его повторений, а также отмечает начало этой последовательности специальным управляющим символом. Этот метод носит название **символьного подавления**.

В методе кодирования с помощью **кодов переменной длины** используется тот факт, что не все символы в передаваемом кадре встречаются с одинаковой частотой. Поэтому во многих схемах кодирования коды часто встречающихся символов заменяют кодами меньшей длины, а редко встречающихся — кодами большей длины. Такое кодирование называется также **статистическим кодированием**. Из-за того что символы имеют разную длину, для передачи кадра возможна только бит-ориентированная передача. При статистическом кодировании коды выбираются таким образом, чтобы при анализе последовательности битов можно было бы однозначно определить соответствие определенной порции битов тому или иному символу или же запрещенной комбинации битов. Если данная последовательность битов представляет собой запрещенную комбинацию, то необходимо к ней добавить еще один бит и повторить анализ. Например, если при неравномерном кодировании для наиболее часто встречающегося символа «Р» выбран код 1, состоящий из одного бита, то значение 0 однобитного кода будет запрещенным. Иначе мы сможем закодировать только два символа. Для другого часто встречающегося символа «О» можно использовать код 01, а код 00 оставить как запрещенный. Тогда для символа «А» можно выбрать код 001, для символа «П» — код 0001 и т. п.

Неравномерное кодирование наиболее эффективно, когда неравномерность распределения частот передаваемых символов велика, как при передаче длинных текстовых строк. Напротив, при передаче двоичных данных, например кодов программ, оно малоэффективно, так как 8-битные коды при этом распределены почти равномерно.

Одним из наиболее распространенных алгоритмов, на основе которых строятся неравномерные коды, является **алгоритм Хафмана**, позволяющий строить коды автоматически на основании известных частот появления символов. Существуют адаптивные модификации метода Хафмана, которые позволяют строить дерево кодов «на ходу», по мере поступления данных от источника.

Многие модели коммуникационного оборудования, такие как модемы, мосты, коммутаторы и маршрутизаторы, поддерживают протоколы динамической компрессии, позволяющие сократить объем передаваемой информации в 4, а иногда и в 8 раз. В таких случаях говорят, что протокол обеспечивает коэффициент сжатия 1:4 или 1:8. Существуют стандартные протоколы компрессии, например V.42bis, а также большое количество нестандартных, фирменных протоколов. Реальный

коэффициент компрессии зависит от типа передаваемых данных. Так, графические и текстовые данные обычно сжимаются хорошо, а коды программ — хуже.

Обнаружение и коррекция ошибок

Список ключевых слов: контрольная сумма, или контрольная последовательность кадра, вертикальный и горизонтальный контроль по паритету, циклический избыточный контроль, прямая коррекция ошибок, расстояние Хемминга, код Хемминга, сверточный, или решетчатый, код.

Надежную передачу информации обеспечивают различные методы. В главе 6 были рассмотрены принципы работы протоколов, которые обеспечивают надежность за счет повторной передачи искаженных или потерянных пакетов. Такие протоколы основаны на том, что приемник в состоянии распознать факт искажения информации в принятом кадре. Еще одним, более эффективным подходом, чем повторная передача пакетов, является использование самокорректирующихся кодов, которые позволяют не только обнаруживать, но и исправлять ошибки в принятом кадре.

Методы обнаружения ошибок

Методы обнаружения ошибок основаны на передаче в составе блока данных избыточной служебной информации, по которой можно судить с некоторой степенью вероятности о достоверности принятых данных. В сетях с коммутацией пакетов такой единицей информации может быть PDU любого уровня, для определенности будем считать, что мы контролируем кадры.

Избыточную служебную информацию принято называть **контрольной суммой**, или **контрольной последовательностью кадра** (Frame Check Sequence, FCS). Контрольная сумма вычисляется как функция от основной информации, причем *не обязательно путем суммирования*. Принимающая сторона повторно вычисляет контрольную сумму кадра по известному алгоритму и в случае ее совпадения с контрольной суммой, вычисленной передающей стороной, делает вывод о том, что данные были переданы через сеть корректно. Рассмотрим несколько распространенных алгоритмов вычисления контрольной суммы, отличающихся вычислительной сложностью и способностью обнаруживать ошибки в данных.

Контроль по паритету представляет собой наиболее простой метод контроля данных. В то же время это наименее мощный алгоритм контроля, так как с его помощью можно обнаружить только одиночные ошибки в проверяемых данных. Метод заключается в суммировании по модулю 2 всех битов контролируемой информации. Нетрудно заметить, что для информации, состоящей из нечетного числа единиц, контрольная сумма всегда равна 1, а при четном числе единиц — 0. Например, для данных 100101011 результатом контрольного суммирования будет значение 1. Результат суммирования также представляет собой один дополнительный бит данных, который пересылается вместе с контролируемой информацией. При искажении в процессе пересылки любого одного бита исходных

данных (или контрольного разряда) результат суммирования будет отличаться от принятого контрольного разряда, что говорит об ошибке. Однако двойная ошибка, например 110101010, будет неверно принята за корректные данные. Поэтому контроль по паритету применяется к небольшим порциям данных, как правило, к каждому байту, что дает коэффициент избыточности для этого метода 1/8. Метод редко применяется в компьютерных сетях из-за значительной избыточности и невысоких диагностических способностей.

Вертикальный и горизонтальный контроль по паритету представляет собой модификацию описанного выше метода. Его отличие состоит в том, что исходные данные рассматриваются в виде матрицы, строки которой составляют байты данных. Контрольный разряд подсчитывается отдельно для каждой строки и для каждого столбца матрицы. Этот метод обнаруживает большую часть двойных ошибок, однако обладает еще большей избыточностью. На практике этот метод сейчас также почти не применяется при передаче информации по сети.

Циклический избыточный контроль (Cyclic Redundancy Check, CRC) является в настоящее время наиболее популярным методом контроля в вычислительных сетях (и не только в сетях, например, этот метод широко применяется при записи данных на гибкие и жесткие диски). Метод основан на рассмотрении исходных данных в виде одного многоразрядного двоичного числа. Например, кадр стандарта Ethernet, состоящий из 1024 байт, будет рассматриваться как одно число, состоящее из 8192 бит. Контрольной информацией считается остаток от деления этого числа на известный делитель R. Обычно в качестве делителя выбирается семнадцати- или тридцатитрехразрядное число, чтобы остаток от деления имел длину 16 разрядов (2 байт) или 32 разряда (4 байт). При получении кадра данных снова вычисляется остаток от деления на тот же делитель R, но при этом к данным кадра добавляется и содержащаяся в нем контрольная сумма. Если остаток от деления на R равен нулю, то делается вывод об отсутствии ошибок в полученном кадре, в противном случае кадр считается искаженным.

Этот метод обладает более высокой вычислительной сложностью, но его диагностические возможности гораздо выше, чем у методов контроля по паритету. Метод CRC обнаруживает все одиночные ошибки, двойные ошибки и ошибки в нечетном числе битов. Метод обладает также невысокой степенью избыточности. Например, для кадра Ethernet размером 1024 байт контрольная информация длиной 4 байт составляет только 0,4 %.

Методы коррекции ошибок

Техника кодирования, которая позволяет приемнику не только понять, что присланные данные содержат ошибки, но и исправить их, называется **прямой коррекцией ошибок** (Forward Error Correction, FEC). Коды, которые обеспечивают прямую коррекцию ошибок, требуют введения бóльшей избыточности в передаваемые данные, чем коды, которые только обнаруживают ошибки.

При применении любого избыточного кода не все комбинации кодов являются разрешенными. Например, контроль по паритету делает разрешенными только половину кодов. Если мы контролируем три информационных бита, то раз-

решенными 4-битными кодами с дополнением до нечетного количества единиц будут:

000 1, 001 0, 010 0, 011 1, 100 0, 101 1, 110 1, 111 0,

то есть всего 8 кодов из 16 возможных.

Для того чтобы оценить количество дополнительных битов, требуемых для исправления ошибок, нужно знать так называемое расстояние Хемминга между разрешенными комбинациями кода. **Расстоянием Хемминга** называется минимальное число битовых разрядов, в которых отличается любая пара разрешенных кодов. Для схем контроля по паритету расстояние Хемминга равно 2.

Можно доказать, что если мы сконструировали избыточный код с расстоянием Хемминга, равным n, то такой код будет в состоянии распознавать (n–1)-кратные ошибки и исправлять (n–1)/2-кратные ошибки. Так как коды с контролем по паритету имеют расстояние Хемминга, равное 2, то они могут только обнаруживать однократные ошибки и не могут исправлять ошибки.

Коды Хемминга эффективно обнаруживают и исправляют изолированные ошибки, то есть отдельные искаженные биты, которые разделены большим количеством корректных битов. Однако при появлении длинной последовательности искаженных битов (пульсации ошибок) коды Хемминга не работают.

Пульсации ошибок характерны для *беспроводных каналов*, в которых применяют **сверточные коды**. Поскольку для распознавания наиболее вероятного корректного кода в этом методе используется решетчатая диаграмма, то такие коды еще называют **решетчатыми**. Эти коды используются не только в беспроводных каналах, но и в модемах.

Мультиплексирование и коммутация

Список ключевых слов: частотное мультиплексирование, уплотненный канал, волновое мультиплексирование, уплотненное волновое мультиплексирование, временное мультиплексирование, асинхронный режим временного мультиплексирования, синхронный режим временного мультиплексирования, тайм-слот, синхронный режим передачи, статистическое временное мультиплексирование, асинхронный режим передачи, дуплексная связь с частотным разделением, дуплексная связь с временны́м разделением.

Методы кодирования и коррекции ошибок позволяют создать в некоторой среде, например в медных проводах кабеля, линию связи. Однако для эффективного соединения пользователей сети этого недостаточно. Нужно образовать в этой линии отдельные каналы передачи данных, которые можно использовать для коммутации информационных потоков пользователей. Для создания пользовательского канала коммутаторы первичных сетей должны поддерживать какую-либо технику мультиплексирования и коммутации. Методы коммутации тесно связаны с используемым методом мультиплексирования, поэтому здесь они изучаются совместно.

В настоящее время для мультиплексирования абонентских каналов используются:

- частотное мультиплексирование (Frequency Division Multiplexing, FDM);
- волновое мультиплексирование (Wave Division Multiplexing, WDM).
- временное мультиплексирование (Time Division Multiplexing, TDM);
- множественный доступ с кодовым разделением (Code Division Multiple Access, CDMA).

Метод TDM используется при коммутации как каналов, так пакетов. Методы FDM, WDM и CDMA пригодны исключительно для коммутации каналов. Метод CDMA применяется только в технике расширенного спектра и рассматривается в следующей главе, посвященной беспроводной передаче.

Коммутация каналов на основе методов FDM и WDM

Техника **частотного мультиплексирования** (FDM) была разработана для телефонных сетей, но применяется она и для других видов сетей, например первичных сетей (микроволновые каналы) или сетей кабельного телевидения.

Основная идея этого метода состоит в выделении каждому соединению своего собственного диапазона частот в общей полосе пропускания линии связи.

На основе этого диапазона и создается **канал**. Данные, передаваемые в канале, модулируются с помощью одного из описанных выше методов с использованием несущей частоты, принадлежащей диапазону канала. Мультиплексирование выполняется с помощью смесителя частот, а демультплексирование — с помощью узкополосного фильтра, ширина которого равна ширине диапазона канала.

Рассмотрим особенности этого вида мультиплексирования на примере телефонной сети.

На входы FDM-коммутатора поступают исходные сигналы от абонентов телефонной сети. Коммутатор выполняет перенос частоты каждого канала в выделенный каналу диапазон за счет модуляции определенной несущей частоты. Чтобы низкочастотные составляющие сигналов разных каналов не смешивались между собой, полосы делают шириной в 4 кГц, а не в 3,1 кГц, оставляя между ними страховой промежуток в 900 Гц (рис. 9.11). В линии связи между двумя FDM-коммутаторами одновременно передаются сигналы всех абонентских каналов, но каждый из них занимает *свою* полосу частот. Такой канал называют **уплотненным**.

Выходной FDM-коммутатор выделяет модулированные сигналы каждой несущей частоты и передает их на соответствующий выходной канал, к которому непосредственно подключен абонентский телефон.

FDM-коммутаторы могут выполнять как динамическую, так и постоянную коммутацию. При *динамической коммутации* один абонент инициирует соединение с другим абонентом, посылая в сеть номер вызываемого абонента. Коммутатор динамически выделяет данному абоненту одну из свободных полос своего уп-

лотненного канала. При постоянной коммутации за абонентом полоса в 4 кГц закрепляется на длительный срок путем настройки коммутатора по отдельному входу, недоступному пользователям.

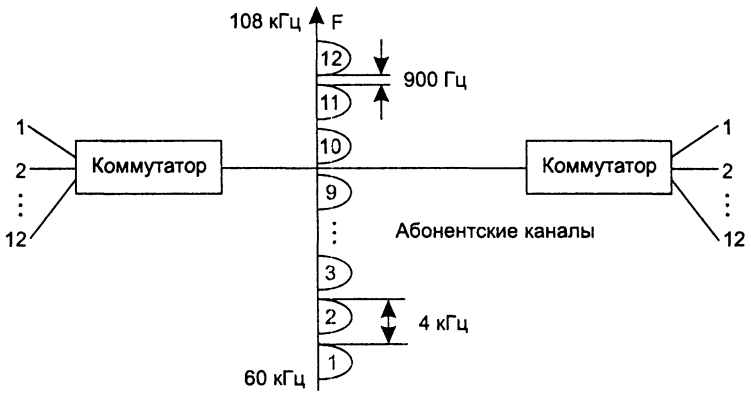


Рис. 9.11. FDM-коммутация

Принцип коммутации на основе разделения частот остается неизменным и в сетях другого вида, меняются только границы полос, выделяемых отдельному абонентскому каналу, а также количество низкоскоростных каналов в высокоскоростном канале.

В методе **волнового мультиплексирования** (WDM) используется тот же принцип частотного разделения каналов, но только в другой области электромагнитного спектра. Информационным сигналом является не электрический ток и не радиоволны, а свет. Для организации WDM-каналов в волоконно-оптическом кабеле используются волны инфракрасного диапазона длиной от 850 до 1565 нм, что соответствует частотам от 196 до 350 ТГц.

В магистральном канале обычно мультиплексируется несколько спектральных каналов — до 16, 32, 40, 80 или 160, причем, начиная с 16 каналов, такая техника мультиплексирования называется **уплотненным волновым мультиплексированием** (Dense Wave Division Multiplexing, DWDM). Внутри такого спектрального канала данные могут кодироваться как дискретным способом, так и аналоговым. По сути WDM и DWDM — это реализации идеи частотного аналогового мультиплексирования, но в другой форме. Отличие сетей WDM/DWDM от сетей FDM — в предельных скоростях передачи информации. Если сети FDM обычно обеспечивают на магистральных каналах одновременную передачу до 600 разговоров, что соответствует суммарной скорости в 36 Мбит/с (для сравнения с цифровыми каналами скорость пересчитана из расчета 64 Кбит/с на один разговор), то сети DWDM обеспечивают общую пропускную способность до сотен гигабитов и даже нескольких терабитов в секунду.

Более подробно технология DWDM рассматривается в главе 11.

Коммутация каналов на основе метода TDM

FDM-коммутация разрабатывалась в расчете на передачу голосовых аналоговых сигналов. Переход к цифровой форме представления голоса стимулировал разработку новой техники мультиплексирования, ориентированной на дискретный характер передаваемых данных и носящей название **временно́го мультиплексирования** (TDM). Принцип временно́го мультиплексирования заключается в выделении канала каждому соединению на определенный период времени. Применяются два типа временно́го мультиплексирования — асинхронный и синхронный. С **асинхронным режимом TDM** мы уже знакомы — он используется в сетях с коммутацией пакетов. Каждый пакет занимает канал определенное время, необходимое для его передачи между конечными точками канала. Между различными информационными потоками нет синхронизации, каждый пользователь пытается занять канал тогда, когда у него возникает потребность в передаче информации.

Рассмотрим теперь **синхронный режим TDM**[1]. В этом случае доступ всех информационных потоков к каналу синхронизируется таким образом, чтобы каждый информационный поток периодически получал канал в свое распоряжение на фиксированный промежуток времени.

Рисунок 9.12 поясняет принцип коммутации каналов на основе техники TDM при передаче голоса.

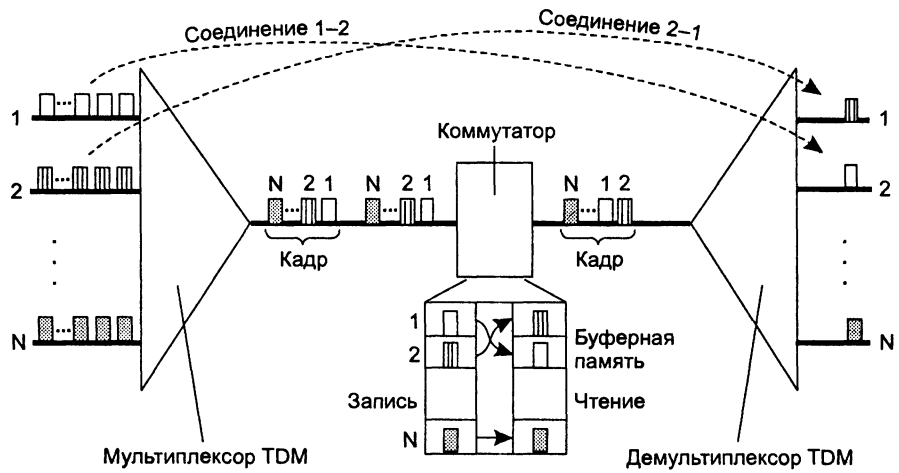


Рис. 9.12. Коммутация на основе разделения канала во времени

Аппаратура TDM-сетей — мультиплексоры, коммутаторы, демультиплексоры — работает в режиме разделения времени, поочередно обслуживая в течение цикла своей работы все абонентские каналы. Цикл равен 125 мкс, что соответствует периоду следования замеров голоса в цифровом абонентском канале. Это значит,

[1] Когда аббревиатура TDM используется без уточнения режима работы, то она всегда обозначает синхронный режим TDM.

что мультиплексор или коммутатор успевает вовремя обслужить любой абонентский канал и передать его очередной замер далее по сети. Каждому соединению выделяется один квант времени цикла работы аппаратуры, называемый также **тайм-слотом**. Длительность тайм-слота зависит от числа абонентских каналов, обслуживаемых мультиплексором или коммутатором.

Мультиплексор принимает информацию по N входным каналам от конечных абонентов, каждый из которых передает данные по абонентскому каналу со скоростью 64 Кбит/с — 1 байт каждые 125 мкс. В каждом цикле мультиплексор выполняет следующие действия:

1. Прием от каждого канала очередного байта данных.
2. Составление из принятых байтов кадра.
3. Передача кадра на выходной канал с битовой скоростью, равной N × 64 Кбит/с.

Порядок следования байта в кадре соответствует номеру входного канала, от которого этот байт получен. Количество обслуживаемых мультиплексором абонентских каналов зависит от его быстродействия. Например, мультиплексор типа T1, представляющий собой первый промышленный мультиплексор, работавший по технологии TDM, поддерживает 24 входных абонентских канала, создавая на выходе *обоймы стандарта T1*, передаваемые с битовой скоростью 1,544 Мбит/с.

Демультиплексор решает обратную задачу — он разбирает байты кадра и распределяет их по своим нескольким выходным каналам, при этом он считает, что порядковый номер байта в обойме соответствует номеру выходного канала.

Коммутатор принимает кадр по скоростному каналу от мультиплексора и записывает каждый байт из него в отдельную ячейку своей буферной памяти, причем в том порядке, в котором эти байты были упакованы в уплотненный кадр. Для выполнения операции коммутации байты извлекаются из буферной памяти не в порядке поступления, а в том порядке, который соответствует поддерживаемым в сети соединениям абонентов. Так, например, если первый абонент левой части сети на рис. 9.12 должен соединиться со вторым абонентом в правой части сети, то байт, записанный в первую ячейку буферной памяти, будет извлекаться из нее вторым. «Перемешивая» нужным образом байты в кадре, коммутатор обеспечивает соединение конечных абонентов в сети.

Однажды выделенный номер тайм-слота остается в распоряжении соединения в течение всего времени существования этого соединения, даже если передаваемый трафик является пульсирующим и не всегда использует закрепленный за ним тайм-слот. Это означает, что соединение в TDM-сети всегда обладает известной и фиксированной пропускной способностью, кратной 64 Кбит/с.

Работа TDM-оборудования напоминает работу сетей с коммутацией пакетов, так как каждый байт данных можно считать некоторым элементарным пакетом. Однако в отличие от пакета компьютерной сети, «пакет» TDM-сети не имеет индивидуального адреса. Его адресом является порядковый номер в кадре или номер выделенного тайм-слота в мультиплексоре или коммутаторе. Сети, использующие технику TDM, требуют синхронной работы всего оборудования, что и определило второе название этой техники — **синхронный режим передачи** (Synchronous Transfer Mode, STM).

Нарушение синхронности разрушает требуемую коммутацию абонентов, так как при этом теряется адресная информация. Поэтому перераспределение тайм-слотов между различными каналами в TDM-оборудовании невозможно, даже если в каком-то цикле работы мультиплексора тайм-слот одного из каналов оказывается избыточным, поскольку на входе этого канала в данный момент нет данных для передачи (например, абонент телефонной сети молчит).

Существует модификация техники TDM, называемая **статистическим временным мультиплексированием** (Statistical TDM, STDM). Эта техника разработана специально для того, чтобы с помощью временно свободных тайм-слотов одного канала можно было увеличить пропускную способность остальных. Для решения этой задачи каждый байт данных дополняется полем адреса небольшой длины, например в 4 или 5 бит, что позволяет мультиплексировать 16 или 32 канала. Фактически, STDM представляет собой уже технику коммутации пакетов, но только с очень упрощенной адресацией и узкой областью применения. Техника STDM не стала популярной и используется в основном в нестандартном оборудовании подключения терминалов к мэйнфреймам. Развитием идей статистического мультиплексирования стала **технология асинхронного режима передачи** (Asynchronous Transfer Mode, ATM), которая относится уже к коммутации пакетов.

TDM-сети могут поддерживать либо режим динамической коммутации, либо режим постоянной коммутации, а иногда и оба эти режима. Основным режимом цифровых телефонных сетей, работающих на основе технологии TDM, является динамическая коммутация, но они поддерживают также и постоянную коммутацию, предоставляя своим абонентам выделенную линию.

Дуплексный режим работы канала

Дуплексный режим — наиболее универсальный и производительный способ работы канала. Самым простым вариантом организации дуплексного режима является использование двух независимых линий связи (двух пар проводников или двух оптических волокон) в кабеле, каждая из которых работает в симплексном режиме, то есть передает данные в одном направлении. Именно такая идея лежит в основе реализации дуплексного режима работы во многих сетевых технологиях, например Fast Ethernet или ATM.

Иногда такое простое решение оказывается недоступным или неэффективным, например, когда прокладка второй линии связи ведет к большим затратам. Так, при обмене данными с помощью модемов через телефонную сеть у пользователя имеется только одна линия связи с телефонной станцией — двухпроводная. В таких случаях дуплексный режим работы организуется на основе разделения линии связи на два логических канала с помощью техники FDM или TDM.

При использовании техники FDM для организации дуплексного канала диапазон частот делится на две части. Деление может быть симметричным и асимметричным, в последнем случае скорости передачи информации в каждом направлении отличаются (популярный пример такого подхода — технология ADSL, используемая для широкополосного доступа в Интернет). В случае когда техника FDM

обеспечивает дуплексный режим работы, ее называют **дуплексной связью с частотным разделением** (Frequency Division Duplex, FDD).

При цифровом кодировании дуплексный режим на двухпроводной линии организуется с помощью техники TDM. Часть тайм-слотов используется для передачи данных в одном направлении, а часть — в другом. Обычно тайм-слоты противоположных направлений чередуются, из-за чего такой способ иногда называют «пинг-понговой» передачей. Дуплексный режим TDM получил название **дуплексной связи с временны́м разделением** (Time Division Duplex, TDD).

В волоконно-оптических кабелях с одним оптическим волокном для организации дуплексного режима работы может применяться технология DWDM. Передача данных в одном направлении осуществляется с помощью светового пучка одной длины волны, а в обратном — другой длины волны. Собственно, решение частной задачи — создание двух независимых спектральных каналов в одном окне прозрачности оптического волокна — и привело к рождению технологии WDM, которая затем трансформировалась в DWDM.

Появление мощных процессоров DSP (Digital Signal Processor), которые могут выполнять сложные алгоритмы обработки сигналов в реальном времени, сделало возможным еще один вариант дуплексной работы. Два передатчика работают одновременно навстречу друг другу, создавая в канале суммарный аддитивный сигнал. Так как каждый передатчик знает спектр собственного сигнала, то он вычитает его из суммарного сигнала, получая в результате сигнал, посылаемый другим передатчиком.

Выводы

Для представления дискретной информации применяются сигналы двух типов: прямоугольные импульсы и синусоидальные волны. В первом случае используют термин «кодирование», во втором — «модуляция».

При модуляции дискретной информации единицы и нули кодируются изменением амплитуды, частоты или фазы синусоидального сигнала.

Аналоговая информация может передаваться по линиям связи в цифровой форме. Это повышает качество передачи, так как при этом могут применяться эффективные методы обнаружения и исправления ошибок, недоступные для систем аналоговой передачи. Для качественной передачи голоса в цифровой форме используется частота оцифровывания в 8 кГц, когда каждое значение амплитуды голоса представляется 8-битным числом. Это определяет скорость голосового канала в 64 Кбит/с.

При выборе способа кодирования нужно одновременно стремиться к достижению нескольких целей: минимизировать возможную ширину спектра результирующего сигнала, обеспечивать синхронизацию между передатчиком и приемником, обеспечивать устойчивость к шумам, обнаруживать и по возможности исправлять битовые ошибки, минимизировать мощность передатчика.

Спектр сигнала является одной из наиболее важных характеристик способа кодирования. Более узкий спектр сигналов позволяет добиваться более высокой скорости передачи данных при фиксированной полосе пропускания среды.

Код должен обладать свойством самосинхронизации, то есть сигналы кода должны содержать признаки, по которым приемник может определить, в какой момент времени нужно осуществлять распознавание очередного бита.

При дискретном кодировании двоичная информация представляется различными уровнями постоянного потенциала или полярностью импульса.

Наиболее простым потенциальным кодом является код без возвращения к нулю (NRZ), однако он не является самосинхронизирующимся.

Для улучшения свойств потенциального кода NRZ используются методы, основанные на введении избыточных битов в исходные данные и на скремблировании исходных данных.

Коды Хэмминга и сверточные коды позволяют не только обнаруживать, но и исправлять многократные ошибки. Эти коды являются наиболее часто используемыми средствами прямой коррекции ошибок (FEC).

Для повышения полезной скорости передачи данных в сетях применяется динамическая компрессия данных на основе различных алгоритмов. Коэффициент сжатия зависит от типа данных и применяемого алгоритма и может колебаться в пределах от 1:2 до 1:8.

Для образования нескольких каналов в линии связи используются различные методы мультиплексирования, включая частотное (FDM), временно́е (TDM) и волновое (WDM), а также множественный доступ с кодовым разделением (CDMA). Техника коммутации пакетов сочетается только с методом TDM, а техника коммутации каналов позволяет использовать любой тип мультиплексирования.

Вопросы и задания

1. Что можно отнести к достоинствам и недостаткам кода NRZ?
2. Какой тип информации передается с помощью амплитудной манипуляции?
3. Почему амплитудная манипуляция не применяется в широкополосных каналах?
4. Какие параметры синусоиды изменяются в методе QAM? Варианты ответов:
   - ○ амплитуда;
   - ○ амплитуда и фаза;
   - ○ амплитуда и частота;
   - ○ частота и фаза.
5. Сколько битов передает один символ кода, имеющий семь состояний?
6. Поясните, из каких соображений выбрана пропускная способность 64 Кбит/с элементарного канала цифровых телефонных сетей?
7. Какой способ применяется для повышения самосинхронизации кода B8ZS?
8. Чем логическое кодирование отличается от физического?
9. Какой принцип лежит в основе методов обнаружения и коррекции ошибок? Варианты ответов:
   - ○ самосинхронизация;
   - ○ избыточность;
   - ○ максимизация отношения мощности сигнала к мощности помех.

10. Назовите методы компрессии, наиболее подходящие для текстовой информации. Почему они неэффективны для сжатия двоичных данных?
11. Что подразумевается под расстоянием Хемминга?
12. Каково расстояние Хемминга в схемах контроля по паритету?
13. Можно ли использовать частотное мультиплексирование в сети Ethernet?
14. Какой режим временно́го мультиплексирования используется в сетях с коммутацией пакетов?
15. Можно ли сочетать различные методы мультиплексирования? Если да, то приведите соответствующие примеры.
16. Что общего в методах частотного и временно́го мультиплексирования?
17. На основании какой техники организуется дуплексный режим работы канала, если оба передатчика используют один и тот же диапазон частот в одно и то же время?
18. Найдите первые две гармоники спектра NRZ-сигнала при передаче последовательности 110011001100..., если тактовая частота передатчика равна 100 МГц.
19. Какие из 16-ти кодов 3B/4B вы выберете для передачи пользовательской информации?
20. Предложите избыточный код с расстоянием Хемминга, равным 3.
21. Могут ли данные надежно передаваться по каналу с полосой пропускания от 2,1 до 2,101 ГГц, если для их передачи используется несущая частота 2,1005 ГГц, амплитудная манипуляция с двумя значениями амплитуды и тактовая частота 5 МГц?
22. Предложите коды неравной длины для каждого из символов A, B, C, D, F и O, если нужно передать сообщение BDDACAAFOOOAOOOO. Будет ли достигнута компрессия данных по сравнению с использованием:
    - традиционных кодов ASCII?
    - кодов равной длины, учитывающих наличие только данных символов?
23. Во сколько раз увеличится ширина спектра кода NRZ при увеличении тактовой частоты передатчика в два раза?

ГЛАВА 10 Беспроводная передача данных

Беспроводная связь стала использоваться для общения между людьми ненамного позже, чем проводная. Уже в 90-х годах 19 века были проведены первые эксперименты по передаче телеграфных сообщений с помощью радиосигналов, а в 20-е годы 20 века началось применение радио для передачи голоса.

Сегодня существует большое число беспроводных телекоммуникационных систем, в том числе не только широковещательных, таких как радио или телевидение. Беспроводные системы также широко используются как транспортное средство для передачи дискретной информации. Для создания протяженных линий связи используются радиорелейные и спутниковые системы, существуют также беспроводные системы доступа к сетям операторов связи и беспроводные локальные сети.

Беспроводная среда, для которой сегодня в основном используется микроволновый диапазон, отличается высоким уровнем помех, которые создают внешние источники излучения, а также многократно отраженные от стен и других преград полезные сигналы. Поэтому в беспроводных системах связи применяют различные средства для снижения влияния помех. В арсенал таких средств входят уже рассмотренные нами коды прямой коррекции ошибок и протоколы с подтверждением доставки информации. Эффективным средством борьбы с помехами является техника расширенного спектра, разработанная специально для беспроводных систем.

В этой главе приводятся базовые сведения об элементах, принципах работы и методах кодирования беспроводных систем, которые используются для построения двухточечных и многоточечных линий связи.

Беспроводная среда передачи

Список ключевых слов: мобильная телефония, фиксированная беспроводная связь, мобильная компьютерная сеть, параболическая антенна, изотропная антенна, радиодиапазон, широковещательное радио, микроволновая система, система инфракрасных волн, система видимого света, дифракция, многолучевое распространение сигнала, межсимвольная интерференция, многолучевое замирание, лицензия, конкурс, лотерея, аукцион, ISM-диапазон.

Преимущества беспроводных коммуникаций

Возможность передавать информацию без проводов, привязывающих (в буквальном смысле этого слова) абонентов к определенной точке пространства, всегда была очень привлекательной. И как только технические возможности становились достаточными для того, чтобы новый вид беспроводных услуг приобрел две необходимые составляющие успеха — удобство использования и низкую стоимость, — успех ему был гарантирован.

Последнее тому доказательство — **мобильная телефония**. Первый мобильный телефон был изобретен еще в 1910 году Ларсом Магнусом Эрикссоном (Lars Magnus Ericsson). Этот телефон предназначался для автомобиля и был беспроводным только во время движения. Однако в движении им нельзя было пользоваться, для разговора нужно было остановиться, выйти из автомобиля и с помощью длинных жердей присоединить телефон к придорожным телефонным проводам (рис. 10.1). Понятно, что определенные неудобства и ограниченная мобильность воспрепятствовали коммерческому успеху этого вида телефонии.

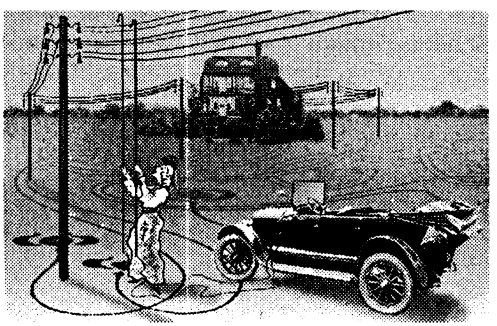


Рис. 10.1. Первый мобильный телефон

Прошло много лет, прежде чем технологии радиодоступа достигли определенной степени зрелости и в конце 70-х обеспечили производство сравнительно компактных и недорогих радиотелефонов. С этого времени начался бум мобильной телефонии, который продолжается в настоящее время.

Беспроводная связь не обязательно означает мобильность. Существует так называемая **фиксированная беспроводная связь**, когда взаимодействующие узлы

постоянно располагаются в пределах небольшой территории; например определенного здания. Фиксированная беспроводная связь применяется вместо проводной, когда по какой-то причине невозможно или невыгодно использовать кабельные линии связи. Причины могут быть разными. Например, малонаселенная или труднодоступная местность — болотистые районы и джунгли Бразилии, пустыни, крайний Север или Антарктида еще не скоро дождутся своих кабельных систем. Другой пример — здания, имеющие историческую ценность, стены которых непозволительно подвергать испытанию прокладкой кабеля. Еще один часто встречающийся случай использования фиксированной беспроводной связи — получение доступа к абонентам, дома которых уже подключены к точкам присутствия существующих уполномоченных операторов связи. Наконец, организация временной связи, например, при проведении конференции в здании, в котором отсутствует проводной канал, имеющий скорость, достаточную для качественного обслуживания многочисленных участников конференции.

Беспроводная связь уже достаточно давно используется для передачи данных. До недавнего времени большая часть применений беспроводной связи в компьютерных сетях была связана с ее фиксированным вариантом. Не всегда архитекторы и пользователи компьютерной сети знают о том, что на каком-то участке пути данные передаются не по проводам, а распространяются в виде электромагнитных колебаний через атмосферу или космическое пространство. Это может происходить в том случае, когда компьютерная сеть арендует линию связи у оператора первичной сети, и отдельный канал такой линии является спутниковым или наземным СВЧ-каналом.

Начиная с середины 90-х годов достигла необходимой зрелости и технология **мобильных компьютерных сетей**. С появлением стандарта IEEE 802.11 в 1997 году появилась возможность строить мобильные сети Ethernet, обеспечивающие взаимодействие пользователей независимо от того, в какой стране они находятся и оборудованием какого производителя они пользуются. Пока такие сети еще играют достаточно скромную роль по сравнению с мобильными телефонными сетями, но аналитики предсказывают их быстрый рост в ближайшие годы.

Беспроводные сети часто связывают с *радиосигналами*, однако это не всегда верно. Беспроводная связь использует широкий диапазон электромагнитного спектра, от радиоволн низкой частоты в несколько килогерц до видимого света, частота которого составляет примерно 8×10^{14} Гц.

Беспроводная линия связи

Беспроводная линия связи строится в соответствии с достаточно простой схемой (рис. 10.2).

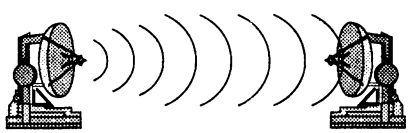


Рис. 10.2. Беспроводная линия связи

Каждый узел оснащается антенной, которая одновременно является передатчиком и приемником электромагнитных волн. Электромагнитные волны распространяются в атмосфере или вакууме со скоростью 3×10^8 м/с во всех направлениях или же в пределах определенного сектора.

Направленность или ненаправленность распространения зависит от типа антенны. На рис. 10.2 показана **параболическая антенна**, которая является *направленной*. Другой тип антенн — **изотропные антенны**, представляющие собой вертикальный проводник длиной в четверть волны излучения, являются *ненаправленными*. Они широко используются в автомобилях и портативных устройствах. Распространение излучения во всех направлениях можно также обеспечить несколькими направленными антеннами.

Так как при ненаправленном распространении электромагнитные волны заполняют все пространство (в пределах определенного радиуса, определяемого затуханием мощности сигнала), то это пространство может служить *разделяемой средой*. Разделение среды передачи порождает те же проблемы, что и в локальных сетях, однако здесь они усугубляются тем, что пространство в отличие от кабеля является общедоступным, а не принадлежит одной организации.

Кроме того, проводная среда строго определяет направление распространения сигнала в пространстве, а *беспроводная среда является ненаправленной*.

Для передачи дискретной информации с помощью беспроводной линии связи необходимо модулировать электромагнитные колебания передатчика в соответствии с потоком передаваемых битов. Эту функцию осуществляет DCE-устройство, располагаемое между антенной и DTE-устройством, которым может быть компьютер, коммутатор или маршрутизатор компьютерной сети.

Диапазоны электромагнитного спектра

Характеристики беспроводной линии связи — расстояние между узлами, территория охвата, скорость передачи информации и т. п. — во многом зависят от частоты используемого электромагнитного спектра (частота f и длина волны λ связаны соотношением $c = f \times \lambda$).

На рис. 10.3 показаны диапазоны электромагнитного спектра. Можно сказать, что они и соответствующие им беспроводные системы передачи информации делятся на четыре группы.

- Диапазон до 300 ГГц имеет общее стандартное название — **радиодиапазон**. Союз ITU разделил его на несколько поддиапазонов (они показаны на рисунке), начиная от сверхнизких частот (Extremely Low Frequency, ELF) и заканчивая сверхвысокими (Extra High Frequency, EHF). Привычные для нас радиостанции работают в диапазоне от 20 кГц до 300 МГц, и для этих диапазонов существует хотя и не определенное в стандартах, однако часто используемое название **широковещательное радио**. Сюда попадают низкоскоростные системы AM- и FM-диапазонов, предназначенные для передачи данных со скоростями от нескольких десятков до сотен килобит в секунду. Примером могут служить радиомодемы, которые соединяют два сегмента локальной сети на скоростях 2400, 9600 или 19 200 Кбит/с.

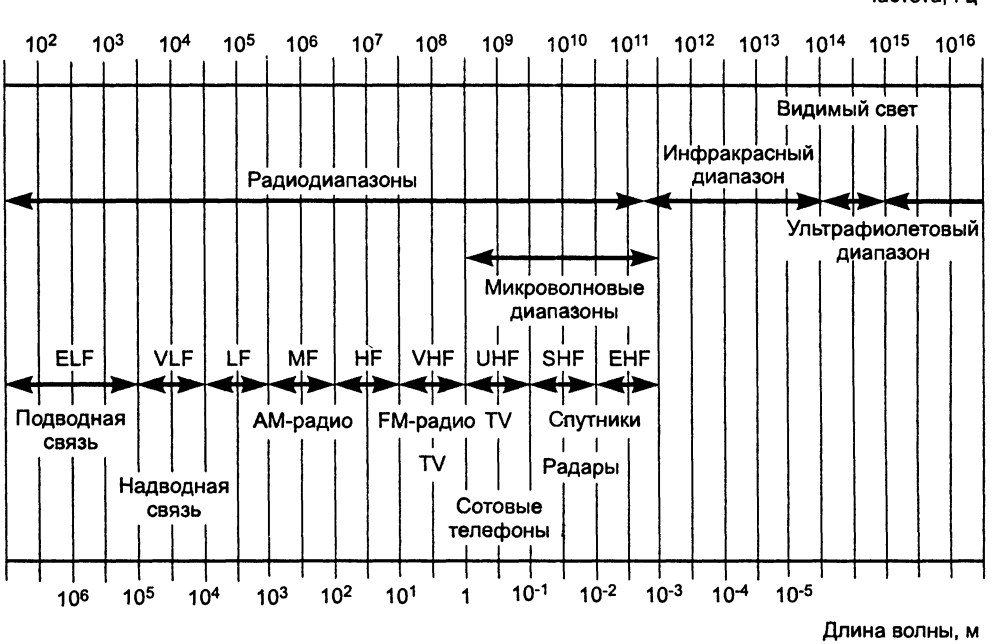


Рис. 10.3. Диапазоны электромагнитного спектра

- Несколько диапазонов от 300 МГц до 3000 ГГц имеют также нестандартное название микроволновых диапазонов. **Микроволновые системы** представляют наиболее широкий класс систем, объединяющий радиорелейные линии связи, спутниковые каналы, беспроводные локальные сети и системы фиксированного беспроводного доступа,, называемые также системами беспроводных абонентских окончаний (Wireless Local Loop, WLL).
- Выше микроволновых диапазонов располагается инфракрасный диапазон. Микроволновые и инфракрасный диапазоны также широко используются для беспроводной передачи информации. Так как инфракрасное излучение не может проникать через стены, то **системы инфракрасных волн** используются для образования небольших сегментов локальных сетей в пределах одного помещения.
- В последние годы видимый свет тоже стал применяться для передачи информации (с помощью лазеров). **Системы видимого света** используются как высокоскоростная альтернатива микроволновым двухточечным каналам для организации доступа на небольших расстояниях.

ПРИМЕЧАНИЕ

Справедливости ради нужно отметить, что свет был, очевидно, первой беспроводной средой передачи информации, так как он использовался в древних цивилизациях (например, в Древней Греции) для эстафетной передачи сигналов между цепочкой наблюдателей, располагавшихся на вершинах холмов.

Распространение электромагнитных волн

Перечислим некоторые общие закономерности распространения электромагнитных волн, связанные с частотой излучения.

- Чем выше несущая частота, тем выше возможная скорость передачи информации.
- Чем выше частота, тем хуже проникает сигнал через препятствия. Низкочастотные радиоволны АМ-диапазонов легко проникают в дома, позволяя обходиться комнатной антенной. Более высокочастотный сигнал телевидения требует, как правило, внешней антенны. И наконец, инфракрасный и видимый свет не проходят через стены, ограничивая передачу *прямой видимостью* (Line Of Sight, LOS).
- Чем выше частота, тем быстрее убывает энергия сигнала с расстоянием от источника. При распространении электромагнитных волн в свободном пространстве (без отражений) затухание мощности сигнала пропорционально произведению квадрата расстояния от источника сигнала на квадрат частоты сигнала.
- Низкие частоты (до 2 МГц) распространяются вдоль поверхности земли. Именно поэтому сигналы АМ-радио могут передаваться на расстояния в сотни километров.
- Сигналы частот от 2 до 30 МГц отражаются ионосферой земли, поэтому они могут распространяться даже на более значительные расстояния, в несколько тысяч километров (при достаточной мощности передатчика).
- Сигналы в диапазоне выше 30 МГц распространяются только по прямой, то есть являются сигналами прямой видимости. При частоте свыше 4 ГГц их подстерегает неприятность — они начинают поглощаться водой, а это означает, что не только дождь, но и туман может стать причиной резкого ухудшения качества передачи микроволновых систем. Недаром испытания лазерных систем передачи данных часто проводят в Сиэтле, городе, который известен своими туманами.

Потребность в скоростной передаче информации является превалирующей, поэтому все современные системы беспроводной передачи информации работают в высокочастотных диапазонах, начиная с 800 МГц, несмотря на преимущества, которые сулят низкочастотные диапазоны благодаря распространению сигнала вдоль поверхности земли или отражения от ионосферы.

Для успешного использования микроволнового диапазона необходимо также учитывать дополнительные проблемы, связанные с поведением сигналов, распространяющихся в режиме прямой видимости и встречающих на своем пути препятствия.

На рис. 10.4 показано, что сигнал, встретившись с препятствием, может распространяться в соответствии с тремя механизмами: отражением, дифракцией и рассеиванием.

Когда сигнал встречается с препятствием, которое частично прозрачно для данной длины волны и в то же время размеры которого намного превышают длину волны, то часть энергии сигнала *отражается* от такого препятствия. Волны микроволнового диапазона имеют длину несколько сантиметров, поэтому они частично отражаются от стен домов при передаче сигналов в городе. Если сигнал встречает непроницаемое для него препятствие (например, металлическую пластину) также намного большего размера, чем длина волны, то происходит **дифракция** — сигнал как бы огибает препятствие, так что такой сигнал можно получить, даже не находясь в зоне прямой видимости. И наконец, при встрече

с препятствием, размеры которого соизмеримы с длиной волны, сигнал *рассеивается*, распространяясь под различными углами.

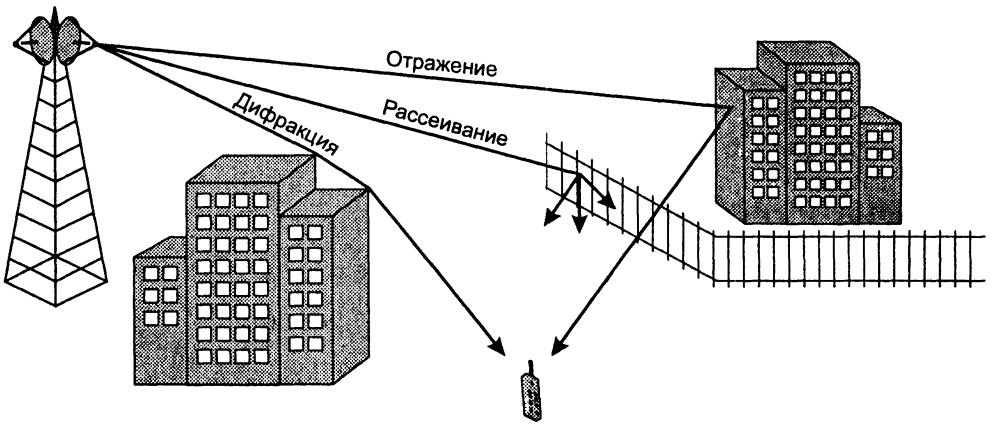


Рис. 10.4. Распространение электромагнитной волны

В результате подобных явлений, которые повсеместно встречаются при беспроводной связи в городе, приемник может получить несколько копий одного и того же сигнала. Такой эффект называется **многолучевым распространением сигнала**. Результат многолучевого распространения сигнала часто оказывается отрицательным, поскольку один из сигналов может прийти с обратной фазой и подавить основной сигнал.

Так как время распространения сигнала вдоль различных путей будет в общем случае различным, то может также наблюдаться и **межсимвольная интерференция**, ситуация, когда в результате задержки сигналы, кодирующие соседние биты данных, доходят до приемника одновременно.

Искажения из-за многолучевого распространения приводят к ослаблению сигнала, этот эффект называется **многолучевым замиранием**. В городах многолучевое замирание приводит к тому, что ослабление сигнала становится пропорциональным не квадрату расстояния, а его кубу или даже четвертой степени!

Все эти искажения сигнала складываются с внешними электромагнитными помехами, которых в городе довольно много. Достаточно сказать, что в диапазоне 2,4 ГГц работают микроволновые печи.

ВНИМАНИЕ

Отказ от проводов и обретение мобильности приводят к высокому уровню помех в беспроводных линиях связи. Если интенсивность битовых ошибок (BER) в проводных линиях связи равна 10^{-9} – 10^{-10}, то в беспроводных линиях связи она достигает величины 10^{-3}!

Проблема высокого уровня помех беспроводных каналов решается различными способами. Важную роль играют специальные методы кодирования, распреде-

ляющие энергию сигнала в широком диапазоне частот. Кроме того, передатчики сигнала (и приемники, если это возможно) стараются разместить на высоких башнях, чтобы избежать многократных отражений. Еще одним способом является применение протоколов с установлением соединений и повторными передачами кадров уже на *канальном* уровне стека протоколов. Эти протоколы позволяют быстрее корректировать ошибки, так как работают с меньшими значениями тайм-аутов, чем корректирующие протоколы *транспортного* уровня, такие как TCP.

Лицензирование

Итак, электромагнитные волны могут распространяться во всех направлениях на значительные расстояния и проходить через препятствия, такие как стены домов. Поэтому проблема совместного использования электромагнитного спектра является весьма острой и требует *централизованного* регулирования. В каждой стране есть специальный государственный орган, который (в соответствии с рекомендациями ITU) выдает **лицензии** операторам связи на использование определенной части спектра, достаточной для передачи информации по определенной технологии. Лицензия выдается на определенную территорию, в пределах которой оператор монопольно использует закрепленный за ним диапазон частот.

При выдаче лицензий правительственные органы руководствуются различными стратегиями. Наиболее популярными являются три: конкурс, лотерея, аукцион.

- Участники **конкурса** — операторы связи — разрабатывают детальные предложения. В них они описывают их будущие услуги, технологии, которые будут использоваться для реализации этих услуг, уровень цен для потенциальных клиентов и т. п. Затем комиссия рассматривает все предложения и выбирает оператора, который в наилучшей степени будет соответствовать общественным интересам. Сложность и неоднозначность критериев выбора победителя в прошлом часто приводили к значительным задержкам в принятии решений и коррупции среди государственных чиновников, поэтому некоторые страны, например США, отказались от такого метода. В то же время в других странах он все еще используется, чаще всего для наиболее значимых для страны услуг, например развертывания современных систем мобильной связи 3G.
- **Лотерея** — это наиболее простой способ, но он также не всегда приводит к справедливым результатам, поскольку в лотерее могут принимать участие и «подставные» операторы, которые не собираются вести операторскую деятельность, а хотят просто перепродать лицензию.
- **Аукционы** сегодня являются достаточно популярным способом выявления обладателя лицензии. Они отсекают недобросовестные компании и приносят немалые доходы государствам. Впервые аукцион был проведен в Новой Зеландии в 1989 году. В связи с бумом вокруг мобильных систем 3G многие государства хорошо пополнили свои бюджеты за счет подобных аукционов.

Существуют также три частотных диапазона, 900 МГц, 2,4 ГГц и 5 ГГц, которые рекомендованы ITU как диапазоны для международного использования *без ли-*

цензирования[1]. Эти диапазоны предназначены для использования промышленными товарами беспроводной связи общего назначения, например устройствами блокирования дверей автомобилей, научными и медицинскими приборами. В соответствии с назначением эти диапазоны получили название **ISM-диапазонов** (Industrial, Scientific, Medical — промышленность, наука, медицина). Диапазон 900 МГц является наиболее «населенным». Это и понятно, низкочастотная техника всегда стоила дешевле. Сегодня активно осваивается диапазон 2,4 ГГц, например, в технологиях IEEE 802.11 и Bluetooth. Диапазон 5 ГГц только начал осваиваться, несмотря на то, что он обеспечивает более высокие скорости передачи данных.

Обязательным условием использования этих диапазонов на совместной основе является ограничение максимальной мощности передаваемых сигналов уровнем 1 Ватт. Это условие ограничивает радиус действия устройств, чтобы их сигналы не стали помехами для других пользователей, которые, возможно, задействуют этот же диапазон частот в других районах города.

Существуют также специальные методы кодирования (они рассматриваются далее), которые уменьшают взаимное влияние устройств, работающих в ISM-диапазонах.

Беспроводные системы

Список ключевых слов: радиорелейная линия связи, базовая станция, точка доступа, сота, эстафетная передача, диффузный передатчик, геостационарный спутник, среднеорбитальный спутник, низкоорбитальный спутник.

Двухточечная связь

Типичная схема проводного двухточечного канала является популярной и для беспроводной связи. По двухточечной схеме могут работать беспроводные каналы различного назначения, использующие различные диапазоны частот.

В телекоммуникационных первичных сетях такая схема уже долгое время используется для создания так называемых **радиорелейных линий связи**. Такую линию образуют несколько башен, на которых установлены параболические направленные антенны (рис. 10.5). Каждая такая линия работает в микроволновом диапазоне на частотах в несколько гигагерц. Направленная антенна концентрирует энергию в узком пучке, что позволяет передавать информацию на значительные расстояния, обычно до 50 км. Высокие башни обеспечивают прямую видимость антенн.

Пропускная способность линии может быть достаточно высокой, обычно она находится в пределах от нескольких до сотен мегабит в секунду. Такие линии

[1] Диапазоны 900 МГц и 5 ГГц не во всех странах являются свободными от лицензирования.

могут быть как магистральными, так и линиями доступа (в последнем случае они имеют чаще всего один канал). Операторы связи часто используют такие линии, когда прокладка оптического волокна либо невозможна (из-за природных условий), либо экономически невыгодна.

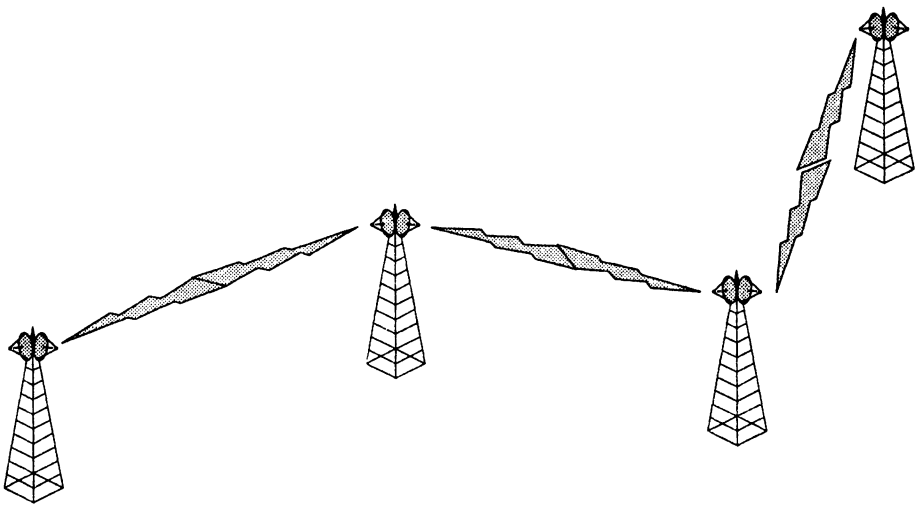


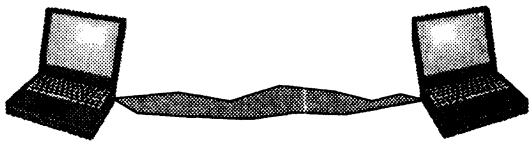
Рис. 10.5. Радиорелейная линия связи

Радиорелейная линия связи может использоваться в городе для соединения двух зданий. Так как высокая скорость в таком случае не всегда нужна (например, нужно соединить небольшой сегмент локальной сети с основной локальной сетью предприятия), то здесь могут применяться радиомодемы, работающие в АМ-диапазоне. Для связи двух зданий может также использоваться лазер, обеспечивая высокую информационную скорость (до 155 Мбит/с), но только при соответствующем состоянии атмосферы.

Другой пример беспроводной двухточечной линии связи показан на рис. 10.6. Здесь она используется для соединения двух компьютеров. Такая линия образует простейший сегмент локальной сети, поэтому расстояния и мощности сигнала здесь принципиально иные.

Для расстояний в пределах одного помещения могут использоваться как диапазон инфракрасных волн (рис. 10.6, *а*), так и микроволновый диапазон (рис. 10.6, *б*). Большинство современных ноутбуков оснащено встроенным инфракрасным портом, поэтому такое соединение может быть образовано автоматически, как только порты двух компьютеров окажутся в пределах прямой видимости (или видимости отраженного луча).

Микроволновый вариант работает в пределах нескольких десятков или сотен метров — предельное расстояние предсказать невозможно, так как при распространении микроволнового сигнала в помещении происходят многочисленные отражения, дифракции и рассеивания, к которым добавляются эффекты проникновения волн через стены и межэтажные перекрытия.



а. Инфракрасные волны

б. Микроволны

Рис. 10.6. Беспроводная связь двух компьютеров

Связь одного источника и нескольких приемников

Схема беспроводного канала с одним источником и несколькими приемниками характерна для такой организации доступа, при которой многочисленные пользовательские терминалы соединяются с **базовой станцией** (Base Station, BS).

Беспроводные линии связи для схемы одного источника и нескольких приемников используются как для фиксированного доступа, так и для мобильного.

На рис. 10.7 показан вариант фиксированного доступа с помощью микроволновых линий связи. Оператор связи использует высокую башню (возможно, телевизионную), чтобы обеспечить прямую видимость с антеннами, установленными на крышах зданий своих клиентов. Фактически такой вариант может представлять собой набор двухточечных линий связи — по количеству зданий, которые необходимо соединить с базовой станцией. Однако это достаточно расточительный вариант, так как для каждого нового клиента нужно устанавливать новую антенну на башне. Поэтому для экономии обычно применяют антенны, захватывающие определенный сектор, например в 45°. Тогда за счет нескольких антенн оператор может обеспечить связь в пределах полного сектора в 360°, конечно, на ограниченном расстоянии (обычно в несколько километров).

Пользователи линий доступа могут обмениваться информацией только с базовой станцией, а она, в свою очередь, транзитом обеспечивает взаимодействие между отдельными пользователями.

Базовая станция обычно соединяется проводной связью с проводной частью сети, обеспечивая взаимодействие с пользователями других базовых станций или пользователями проводных сетей. Поэтому базовая станция также называется **точкой доступа** (Access Point, AP). Точка доступа включает не только DCE-оборудование, необходимое для образования линии связи, но и чаще всего является коммутатором сети, доступ к которой она обеспечивает — телефонным коммутатором или коммутатором пакетов.

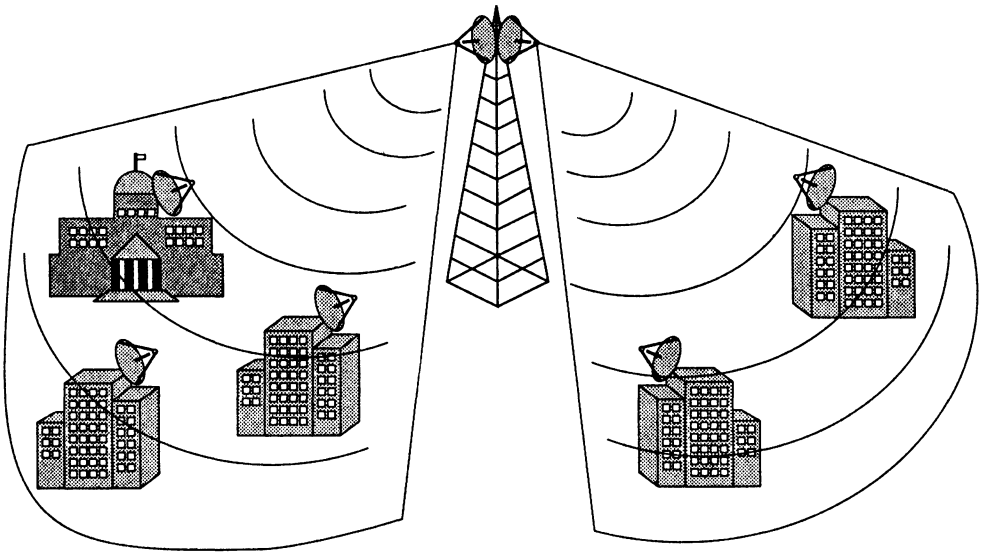


Рис. 10.7. Фиксированный беспроводный доступ

В большинстве схем мобильного доступа используется сегодня принцип **сот**, которые представляют собой небольшие по площади территории, обслуживаемые одной базовой станцией. Идея сот родилась не сразу, первые мобильные телефоны работали по другому принципу, обращаясь к одной базовой станции, покрывающей большую территорию. Идея небольших сот была впервые сформулирована еще в 1945 году, с тех пор прошло довольно много времени, пока заработали первые коммерческие сотовые телефонные сети — пробные участки появились в конце 60-х, а широкое коммерческое применение началось в начале 80-х.

Принцип разбиения всей области охвата сети на небольшие соты дополняется идеей многократного использования частоты. На рис. 10.8 показан вариант организации сот при наличии всего трех частот, при этом ни одна из соседних пар сот не задействует одну и ту же частоту. Многократное использование частот позволяет оператору экономно расходовать выделенный ему частотный диапазон, при этом абоненты и базовые станции соседних сот не испытывают проблем из-за интерференции сигналов. Конечно, базовая станция должна контролировать мощность излучаемого сигнала, чтобы две соты (несмежные), работающие на одной и той же частоте, не создавали друг другу помех.

При гексагональной форме сот количество повторяемых частот может быть больше, чем 3, например 4, 7, 9, 12, 13 и т. д.

Если известно минимальное расстояние D между центрами сот, работающих на одной и той же частоте, то число сот (N) можно выбрать по формуле:

$$N = D^2/3R^2,$$

где R — радиус соты.

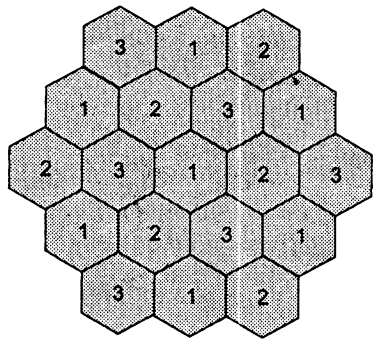


Рис. 10.8. Многократное использование частот в сотовой сети

Небольшие по величине соты обеспечивают небольшие габариты и мощность терминального устройства пользователя. Именно это обстоятельство (а также общий технологический прогресс) позволяет современным мобильным телефонам быть такими компактными.

Мобильные компьютерные сети пока не получили такого распространения, как телефонные, но принципы организации беспроводных линий связи в них остаются теми же.

Важной проблемой мобильной линии связи является переход терминального устройства из одной соты в другую. Эта процедура, которая называется **эстафетной передачей**, отсутствует при фиксированном доступе и относится к протоколам более высоких уровней, нежели физический.

Связь нескольких источников и нескольких приемников

В случае схемы с несколькими источниками и несколькими приемниками беспроводная линия связи представляет собой общую электромагнитную среду, разделяемую несколькими узлами. Каждый узел может использовать эту среду для взаимодействия с любым другим узлом без обращения к базовой станции. Так как базовая станция отсутствует, то необходим децентрализованный алгоритм доступа к среде.

Чаще всего такой вариант беспроводного канала применяется для соединения компьютеров (рис. 10.9). Для телефонного трафика неопределенность в доле пропускной способности, получаемой при разделении среды, может резко ухудшить качество передачи голоса. Поэтому они строятся по ранее рассмотренной схеме с одним источником (базовой станцией) для распределения полосы пропускания и несколькими приемниками.

Собственно, первая локальная сеть, созданная в 70-е годы на Гавайях, в точности соответствовала приведенной на рисунке схеме. Ее отличие от современных беспроводных локальных сетей состоит в низкой скорости передачи данных (9600 бит/с), а также в весьма неэффективном способе доступа, в результате использовалось только 18 % полосы пропускания.

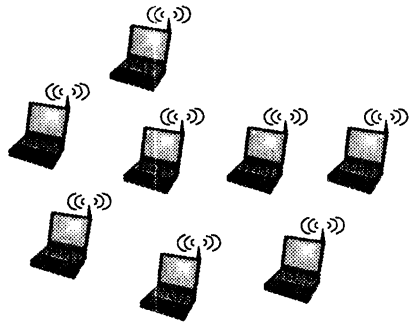


Рис. 10.9. Беспроводная многоточечная линия связи

Сегодня такие сети передают данные со скоростью до 52 Мбит/с в микроволновом или инфракрасном диапазоне. Для связи каждого с каждым используются ненаправленные антенны. Для того чтобы инфракрасный свет распространялся в разных направлениях, применяются **диффузные передатчики**, которые рассеивают лучи с помощью системы линз.

Типы спутниковых систем

Спутниковая связь используется для организации высокоскоростных микроволновых протяженных линий. Так как для таких линий связи нужна прямая видимость, которую из-за кривизны Земли невозможно обеспечить на больших расстояниях, то спутник как отражатель сигнала является естественным решением этой проблемы (рис. 10.10).

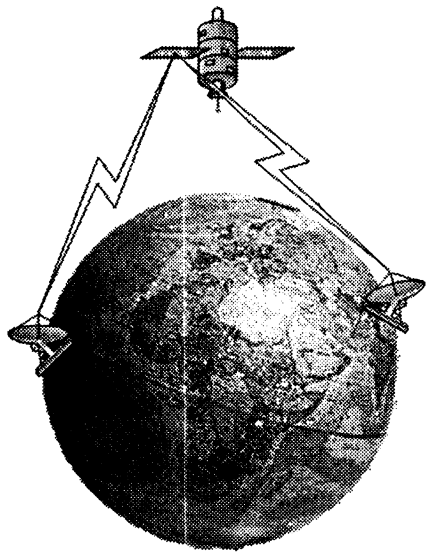


Рис. 10.10. Спутник как отражатель сигнала

Идея использовать искусственный спутник Земли для создания линий связи появилась задолго до запуска в 1957 году первого такого спутника Советским Союзом. Писатель-фантаст Артур Кларк продолжил дело Жюля Верна и Герберта Уэллса, которым удалось описать много технических изобретений до их появления. Кларк в 1945 году описал геостационарный спутник, который висит над одной точкой экватора и обеспечивает связью большую территорию Земли.

Первый спутник, запущенный Советским Союзом в годы холодной войны, обладал очень ограниченными телекоммуникационными возможностями — он только передавал радиосигнал «бип-бип», извещая мир о своем присутствии в космосе. Однако успех России в космосе подхлестнул усилия Америки, и в 1962 году она запустила первый телекоммуникационный спутник Telstar-1, который поддерживал 600 голосовых каналов.

Со времени запуска первого телекоммуникационного спутника прошло уже более 40 лет, и функции спутника как телекоммуникационного узла, естественно, усложнились. Сегодня спутник может играть роль узла первичной сети, а также телефонного коммутатора и коммутатора/маршрутизатора компьютерной сети. Для этого аппаратура спутников может взаимодействовать не только с наземными станциями, но и между собой, образуя прямые космические беспроводные линии связи. Принципиально техника передачи микроволновых сигналов в космосе и на Земле не отличается, однако у спутниковых линий связи есть и очевидная специфика — один из узлов такой линии постоянно находится в полете, причем на большом расстоянии от других узлов.

Для спутниковой связи союз ITU выделил несколько частотных диапазонов (табл. 10.1).

Таблица 10.1. Частотные диапазоны для спутниковой связи

Диапазон	Нисходящая частота, ГГц	Восходящая частота, ГГц
L	1,5	1,6
S	1,9	2,2
C	3,7–4,2	5,925–6,425
Ku	11,7–12,2	14,0–14,5
Ka	17,7–21,7	27,5–30,5

Исторически первым использовался диапазон **C**, в котором для каждого из дуплексных потоков Земля-спутник (восходящая частота) и спутник-Земля (нисходящая частота) выделяется по 500 МГц — достаточно для большого числа каналов. Диапазоны **L** и **S** предназначаются для организации мобильных услуг с помощью спутников. Они также часто используются наземными системами. Диапазоны **Ku** и **Ka** пока мало «населены» на Земле, их применению препятствует высокая стоимость оборудования, особенно для диапазона **Ka**.

Искусственные спутники Земли вращаются вокруг нее в соответствии с законами, открытыми Йоханесом Кеплером (Johannes Kepler). Орбита вращения спут-

ника в общем случае является эллиптической, но для сохранения постоянной высоты над Землей спутники могут переходить на почти круговую орбиту.

Сегодня используется три группы круговых орбит, отличающихся высотой над Землей (рис. 10.11):

- геостационарная орбита (Geostationary Orbit, GEO) — 35 863 км;
- средневысотная орбита (Medium Earth Orbit, MEO) — 5000–15 000 км;
- маловысотная орбита (Low Earth Orbit, LEO) — 100–1000 км.

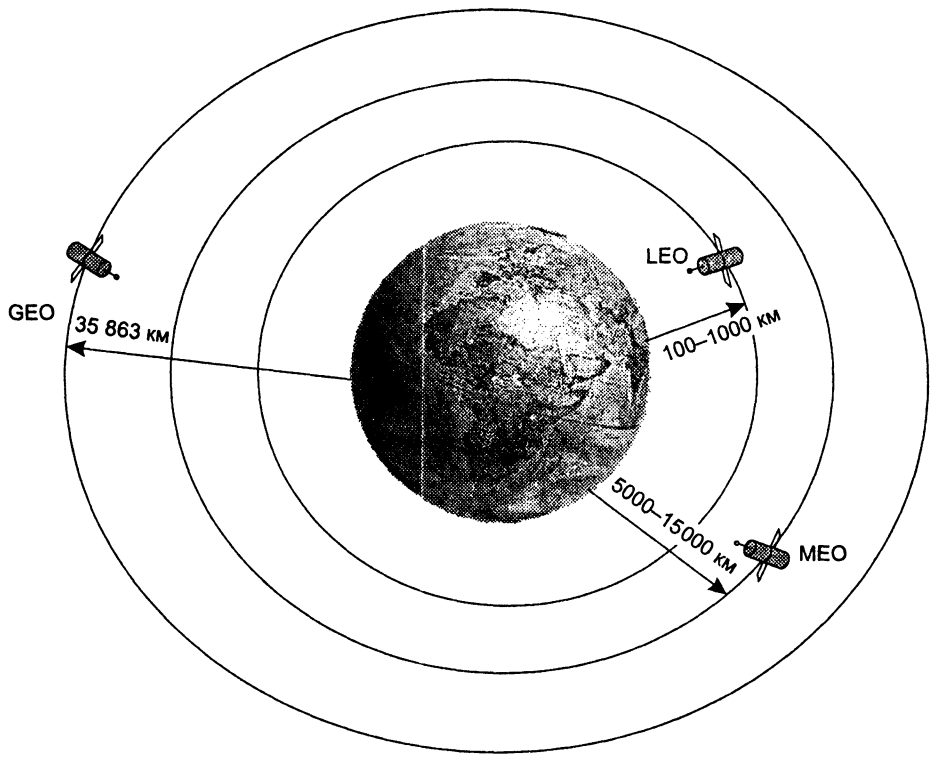


Рис. 10.11. Типы орбит спутников

Геостационарный спутник

Геостационарный спутник висит над определенной точкой экватора, в точности следуя скорости вращения Земли. Такое положение выгодно по следующим обстоятельствам.

Во-первых, четверть поверхности Земли оказывается с такой высоты в зоне прямой видимости, поэтому с помощью геостационарных спутников *просто организовать широковещание в пределах страны или даже континента.*

Во-вторых, сам *спутник неподвижен для наземных антенн*, что значительно облегчает организацию связи, так как не нужно автоматически корректировать

направление наземной антенны, как это приходится делать для низкоорбитальных и средневысотных спутников. Правда, с появлением в 1990 году небольших всенаправленных антенн ситуация изменилась — теперь уже не нужно следить за положением низкоорбитального спутника, достаточно, чтобы он находился в зоне прямой видимости.

В-третьих, геостационарный спутник находится за пределами земной атмосферы и *меньше «изнашивается»*, чем низкоорбитальные и средневысотные спутники. Низкоорбитальные спутники из-за трения о воздух постоянно теряют высоту и им приходится восстанавливать ее с помощью двигателей.

Геостационарные спутники обычно *поддерживают большое количество каналов* за счет наличия нескольких антенн. Раньше для работы с геостационарными спутниками в качестве антенн требовались очень большие тарелки (диаметром до 10 м). Это затрудняло использование геостационарных спутников для небольших организаций и личных целей. Однако ситуация изменилась с появлением направленных антенн, устанавливаемых на спутниках. Такие антенны создают сигнал, который можно принимать с помощью сравнительно небольших наземных антенн, так называемых миниатюрных апертурных терминалов (Very Small Aperture Terminals, VSAT). Диаметр антенны VSAT составляет около 1 м. Наземные станции, оснащенные VSAT, предоставляют сегодня большой набор услуг, к которым относятся телефония, передача данных, конференции.

Наряду с достоинствами у геостационарных спутников есть и недостатки. Наиболее очевидные связаны с *большим удалением спутника от поверхности Земли*. Это приводит к большим задержкам распространения сигнала — от 230 до 280 мс. При использовании спутника для передачи разговора или телевизионного диалога возникают неудобные паузы, мешающие нормальному общению.

Кроме того, на таких расстояниях *потери сигнала высоки*, что означает необходимость использования мощных передатчиков и тарелок больших размеров (это не относится к антеннам VSAT, но при их использовании уменьшается область охвата).

Принципиальным недостатком геостационарного спутника с его круговой орбитой является также *плохая связь для районов, близких к Северному и Южному полюсам*. Сигналы для таких районов проходят бóльшие расстояния, чем для районов, расположенных в экваториальных и умеренных широтах, и, естественно, больше ослабляются. Решением является спутник с ярко выраженной эллиптической орбитой, который приближается к Земле как раз в районе Северного и Южного полюсов. Примером такого спутника являются спутники серии «Молния», которые запускаются Россией, имеющей большие территории на Крайнем Севере.

Место на орбите геостационарного спутника также регулируется союзом ITU. Сегодня наблюдается определенный дефицит таких мест, так как геостационарные спутники не могут располагаться на орбите ближе, чем 2° друг к другу. Из этого следует, что на орбите может находиться не более 180 геостационарных спутников. Так как не все страны в состоянии (пока) запустить геостационарный спутник, то здесь наблюдается та же ситуация, что и в конкурсе на получе-

ние определенного диапазона частот, только еще усиленная политическими амбициями стран.

Средне- и низкоорбитальные спутники

Класс среднеорбитальных спутников пока не так популярен, как геостационарных и низкоорбитальных спутников. **Среднеорбитальные спутники** обеспечивают диаметр покрытия от 10 000 до 15 000 км и задержку распространения сигнала 50 мс. Наиболее известной услугой, предоставляемой спутниками этого класса, является *глобальная система навигации* (Global Positioning System, GPS). GPS — это всеобщая система определения текущих координат пользователя на поверхности Земли или в околоземном пространстве. GPS состоит из 24 спутников, сети наземных станций слежения за ними и неограниченного количества пользовательских приемников-вычислителей. По радиосигналам спутников GPS-приемники пользователей устойчиво и точно определяют координаты. Погрешности не превышают десятков метров. Этого вполне достаточно для решения задач навигации подвижных объектов (самолеты, корабли, космические аппараты, автомобили и т. д.).

Достоинства и недостатки **низкоорбитальных спутников** противоположны соответствующим качествам геостационарных спутников. Главное их достоинство — близость к Земле, а значит, пониженная мощность передатчиков, малые размеры антенн и небольшое время распространения сигнала (около 20–25 мс). Кроме того, их легче запускать. Основной недостаток — малая площадь покрытия, диаметр которой составляет всего около 8000 км. Период оборота вокруг Земли такого спутника составляет 1,5–2 часа, и время видимости спутника наземной станцией составляет всего 20 минут. Это значит, что постоянная связь с помощью низкоорбитальных спутников может быть обеспечена, только когда на орбите находится достаточно большое их количество. Кроме того, атмосферное трение снижает срок службы таких спутников до 8–10 лет.

Если основным назначением геостационарных спутников является широковещание и дальняя связь, то низкоорбитальные спутники рассматриваются как важное средство поддержания мобильной связи.

В начале 90-х годов достоинства компактных терминальных устройств для низкоорбитальных спутников показались руководителям компании Motorola более важными, чем их недостатки. Вместе с несколькими крупными партнерами эта компания начала проект *Iridium*, который имел весьма амбициозную цель — создать всемирную спутниковую сеть, обеспечивающую мобильную связь в любой точке земного шара. В конце 80-х еще не существовало такой плотной системы сот мобильной телефонии, как сегодня, так что коммерческий успех казался обеспеченным.

В 1997 году группа из 66 спутников была запущена, а в 1998 году началась коммерческая эксплуатация системы Iridium. Спутники Iridium действительно покрывают всю поверхность земного шара, вращаясь по 6 орбитам, проходящим через полюсы Земли. На каждой орбите находится по 11 спутников, которые имеют передатчики на частоте 1,6 ГГц с полосой пропускания 10 МГц. Эта полоса

расходуется 240 каналами по 41 кГц каждый. За счет многократного использования частот система Iridium поддерживает 253 440 каналов, организуя системы скользящих по поверхности Земли сот. Для пользователей системы Iridium основным видом услуги является телефонная связь (7 долларов в минуту) и передача данных со скоростью 2,4 Кбит/с.

Спутники Iridium обладают значительным интеллектом, они могут, пользуясь специальными межспутниковыми каналами, передавать друг другу информацию со скоростью 25 Мбит/с. Поэтому телефонный вызов идет от спутникового телефона Iridium прямо на спутник, находящийся в зоне видимости. Затем этот спутник маршрутизирует вызов через систему промежуточных спутников тому спутнику, который в данный момент ближе к вызываемому абоненту. Система Iridium представляет собой сеть с полным собственным стеком протоколов, который обеспечивает всемирный роуминг.

К сожалению, коммерческие успехи Iridium оказались очень скромными, и через два года своего существования компания обанкротилась. Расчет на мобильных телефонных абонентов оказался неверным — к моменту начала работы наземная сеть сотовой связи уже покрывала большую часть территории развитых стран. А услуги по передаче данных со скоростью 2,4 Кбит/с не соответствовали потребностям пользователей конца 20-го века.

Сегодня система Iridium снова работает, теперь уже с новым владельцем и новым именем — *Iridium Satellite*. У нее теперь более скромные планы, связанные с созданием местных систем связи в тех частях земного шара, где связь практически отсутствует. Программное обеспечение спутников модернизируется «на лету», что позволило повысить скорость передачи данных до 10 Кбит/с.

Другой известной системой низкоорбитальных спутников является *Globalstar*. В отличие от Iridium 48 низкоорбитальных спутников Globalstar выполняют традиционные для геостационарных спутников функции — принимают телефонные вызовы от мобильных абонентов и передают их ближайшей наземной базовой станции. Маршрутизацию вызовов выполняет базовая станция, которая передает вызов базовой станции, ближайшей к спутнику, в зоне видимости которого находится вызываемый абонент. Межспутниковые каналы не используются. Помимо телефонных разговоров Globalstar также передает данные со скоростью 4,8 Кбит/с.

Еще одна LEO-сеть — *Orbcomm* предоставляет сервис, ориентированный на передачу данных. К сожалению, доставка сообщений осуществляется не в режиме реального времени. Если спутник невидим, терминал Orbcomm просто хранит пакеты, пока космический аппарат не войдет в зону видимости. Это приводит к чрезвычайно значительной неравномерности в передаче данных. Вместо привычных для пользователей Интернета задержек в доли секунды, в этой сети паузы иногда измеряются минутами.

Сегодня, когда стало ясно, что мобильная телефония будет поддерживаться в основном наземными сотовыми сетями, ориентация многих спутниковых систем меняется, на первый план выходит предоставление скоростного доступа в Интернет. В число таких систем входит LEO-система Teledesic, одним из основателей которой является Билл Гейтс. В этой системе, создаваемой с начала 90-х го-

дов, спутники представляют собой полноценные маршрутизаторы, соединенные межспутниковыми каналами 64 Кбит/с.

Создаются также системы доступа в Интернет на основе геостационарных спутников. К таким системам относятся Spaceway, Astrolink, Euro Skyway. Они ориентированы на использование антенн VSAT и обещают предоставлять пользователям каналы 2–20 Мбит/с.

Технология широкополосного сигнала

Список ключевых слов: техника расширенного спектра, ортогональное частотное мультиплексирование, расширение спектра скачкообразной перестройкой частоты, начальное число, последовательность псевдослучайной перестройки частоты, медленное расширение спектра, быстрое расширение спектра, прямое последовательное расширение спектра, расширяющая последовательность, чип, чиповая скорость, коэффициент расширения, множественный доступ с кодовым разделением, пилотный сигнал.

Техника расширенного спектра разработана специально для беспроводной передачи. Она позволяет улучшить помехоустойчивость кода для сигналов малой мощности, что очень важно для мобильных применений. Однако нужно подчеркнуть, что техника расширенного спектра – не единственная техника кодирования, которая применяется для беспроводных линий связи микроволнового диапазона. Здесь также применяются частотная (FSK) и фазовая (PSK) манипуляции, описанные в предыдущей главе. Амплитудная манипуляция (ASK) не используется по той причине, что каналы микроволнового диапазона имеют широкую полосу пропускания, а усилители, которые обеспечивают одинаковый коэффициент усиления для широкого диапазона частот, очень дороги.

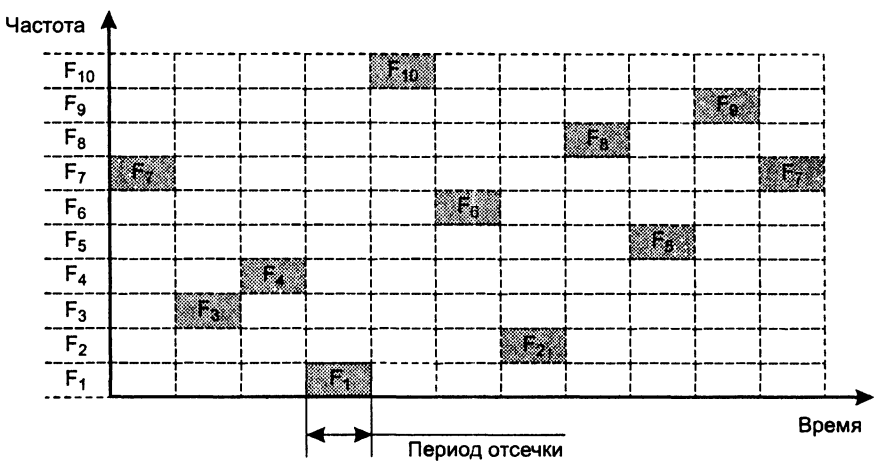
Широкая полоса пропускания позволяет также применять модуляцию с несколькими несущими, когда полоса делится на несколько подканалов, каждый из которых использует свою несущую частоту. Соответственно, битовый поток делится на несколько подпотоков, текущих с более низкой скоростью. Затем каждый подпоток модулируется с помощью определенной несущей частоты, которая обычно кратна основной несущей частоте, то есть f_0, $2f_0$, $3f_0$ и т. д. Модуляция выполняется с помощью обычных методов FSK или PSK. Такая техника называется **ортогональным частотным мультиплексированием** (Orthogonal Frequency Division Multiplexing, OFDM).

Перед передачей все несущие сворачиваются в общий сигнал путем быстрого преобразования Фурье. Спектр такого сигнала примерно равен спектру сигнала, кодируемого одной несущей. После передачи из общего сигнала путем обратного преобразования Фурье выделяются несущие подканалы, а затем из каждого канала выделяется битовый поток. Выигрыш в разделении исходного высокоскоростного битового потока на несколько низкоскоростных подпотоков проявляется в том, что увеличивается интервал между отдельными символами кода. Это означает, что снижается эффект межсимвольной интерференции, появляющийся из-за многолучевого распространения электромагнитных волн.

Расширение спектра скачкообразной перестройкой частоты

Идея метода **расширения спектра скачкообразной перестройкой частоты** (Frequency Hopping Spread Spectrum, FHSS) возникла во время Второй мировой войны, когда радио широко использовалось для секретных переговоров и для управления военными объектами, например торпедами. Для того чтобы радиообмен нельзя было перехватить или подавить узкополосным шумом, было предложено вести передачу с постоянной сменой несущей в пределах широкого диапазона частот. В результате мощность сигнала распределялась по всему диапазону, и прослушивание какой-то определенной частоты давало только небольшой шум. Последовательность несущих частот выбиралась псевдослучайной, известной только передатчику и приемнику. Попытка подавления сигнала в каком-то узком диапазоне также не слишком ухудшала сигнал, так как подавлялась только небольшая часть информации.

Идею этого метода иллюстрирует рис. 10.12.



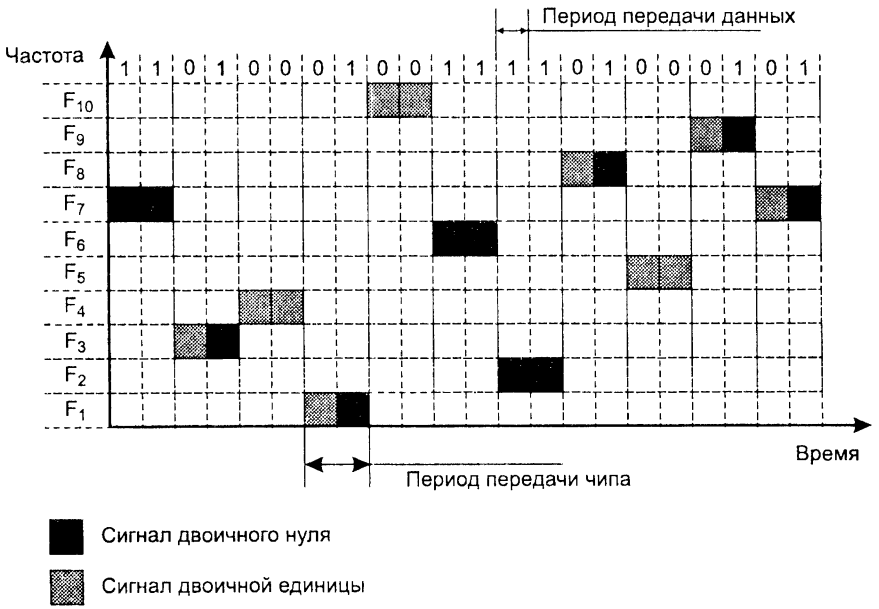
Последовательность перестройки частот: F_7-F_3-F_4-F_1-F_{10}-F_6-F_2-F_8-F_5-F_9

Рис. 10.12. Расширение спектра скачкообразной перестройкой частоты

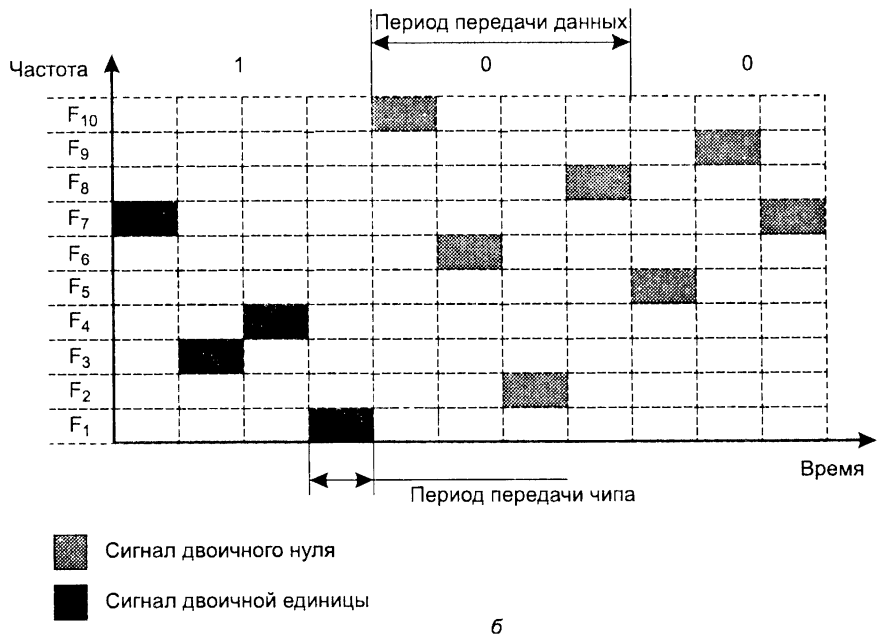
В течение определенного фиксированного интервала времени передача ведется на неизменной несущей частоте. На каждой несущей частоте для передачи дискретной информации применяются стандартные методы модуляции, такие как FSK или PSK. Для того чтобы приемник синхронизировался с передатчиком, для обозначения начала каждого периода передачи в течение некоторого времени передаются синхробиты. Так что полезная скорость этого метода кодирования оказывается меньше из-за постоянных накладных расходов на синхронизацию.

Несущая частота меняется в соответствии с номерами частотных подканалов, вырабатываемых алгоритмом псевдослучайных чисел. Псевдослучайная последовательность зависит от некоторого параметра, который называют **начальным числом**. Если приемнику и передатчику известны алгоритм и значение началь-

ного числа, то они меняют частоты в одинаковой последовательности, называемой **последовательностью псевдослучайной перестройки частоты.**



а



б

Рис. 10.13. Соотношение между скоростью передачи данных и частотой смены подканалов: *а* — скорость передачи данных выше чиповой скорости, *б* — скорость передачи данных ниже чиповой скорости

Если частота смены подканалов ниже, чем скорость передачи данных в канале, то такой режим называют **медленным расширением спектра** (рис. 10.13, *а*); в противном случае мы имеем дело с **быстрым расширением спектра** (рис. 10.13, *б*).

Метод быстрого расширения спектра более устойчив к помехам, поскольку узкополосная помеха, которая подавляет сигнал в определенном подканале, не приводит к потере бита, так как его значение повторяется несколько раз в различных частотных подканалах. В этом режиме не проявляется эффект межсимвольной интерференции, потому что ко времени прихода задержанного вдоль одного из путей сигнала система успевает перейти на другую частоту.

Метод медленного расширения спектра таким свойством не обладает, но зато он проще в реализации и имеет меньшие накладные расходы.

Методы FHSS используются в беспроводных технологиях IEEE 802.11 и Bluetooth.

В методах FHSS подход к использованию частотного диапазона не такой, как в других методах кодирования — вместо экономного расходования узкой полосы делается попытка занять весь доступный диапазон. На первый взгляд это кажется не очень эффективным — ведь в каждый момент времени в диапазоне работает только один канал. Однако последнее утверждение не всегда справедливо — коды расширенного спектра можно использовать также и для мультиплексирования *нескольких* каналов в широком диапазоне. В частности, методы FHSS позволяют организовать одновременную работу нескольких каналов путем выбора для каждого канала таких псевдослучайных последовательностей, чтобы в каждый момент времени каждый канал работал на своей частоте (конечно, это можно сделать, только если число каналов не превышает числа частотных подканалов).

Прямое последовательное расширение спектра

В методе **прямого последовательного расширения спектра** (Direct Sequence Spread Spectrum, DSSS) также используется весь частотный диапазон, выделенный для одной беспроводной линии связи. В отличие от метода FHSS весь частотный диапазон занимается не за счет постоянных переключений с частоты на частоту, а за счет того, что каждый бит информации заменяется N битами, так что тактовая скорость передачи сигналов увеличивается в N раз. А это, в свою очередь, означает, что спектр сигнала также расширяется в N раз. Достаточно соответствующим образом выбрать скорость передачи данных и значение N, чтобы спектр сигнала заполнил весь диапазон.

Цель кодирования методом DSSS та же, что методом FHSS — повышение устойчивости к помехам. Узкополосная помеха будет искажать только определенные частоты спектра сигнала, так что приемник с большой степенью вероятности сможет правильно распознать передаваемую информацию.

Код, которым заменяется двоичная единица исходной информации, называется **расширяющей последовательностью**, а каждый бит такой последовательности — **чипом**. Соответственно, скорость передачи результирующего кода называют **чиповой скоростью**. Двоичный нуль кодируется инверсным значением расширяющей

последовательности. Приемники должны знать расширяющую последовательность, которую использует передатчик, чтобы понять передаваемую информацию.

Количество битов в расширяющей последовательности определяет **коэффициент расширения** исходного кода. Как и в случае FHSS, для кодирования битов результирующего кода может использоваться любой вид модуляции, например BFSK.

Чем больше коэффициент расширения, тем шире спектр результирующего сигнала и тем больше степень подавления помех. Но при этом растет занимаемый каналом диапазон спектра. Обычно коэффициент расширения имеет значения от 10 до 100.

Примером значения расширяющей последовательности является *последовательность Баркера* (Barker), которая состоит из 11 бит: 10110111000. Если передатчик использует эту последовательность, то передача трех битов 110 ведет к передаче следующих битов:

10110111000 10110111000 01001000111.

Последовательность Баркера позволяет приемнику быстро синхронизироваться с передатчиком, то есть надежно выявлять начало последовательности. Приемник определяет такое событие, поочередно сравнивая получаемые биты с образцом последовательности. Действительно, если сравнить последовательность Баркера с такой же последовательностью, но сдвинутой на один бит влево или вправо, то мы получим меньше половины совпадений значений битов. Значит, даже при искажении нескольких битов с большой долей вероятности приемник правильно определит начало последовательности, а значит, сможет правильно интерпретировать получаемую информацию.

Метод DSSS в меньшей степени защищен от помех, чем метод быстрого расширения спектра, так как мощная узкополосная помеха влияет на часть спектра, а значит, и на результат распознавания единиц или нулей.

Множественный доступ с кодовым разделением

Как и в случае FHSS, кодирование методом DSSS позволяет мультиплексировать несколько каналов в одном диапазоне. Техника такого мультиплексирования называется **множественным доступом с кодовым разделением** (Code Division Multiple Access, CDMA). Она широко используется в сотовых сетях.

Техника CDMA может использоваться совместно с кодированием методом FHSS, однако на практике она чаще применяется в беспроводной сети с методом DSSS.

Каждый узел сети, работающий по методу CDMA, посылает данные в разделяемую среду в те моменты времени, когда это ему нужно, то есть синхронизация между узлами отсутствует. Идея CDMA заключается в том, что каждый узел сети использует собственное значение расширяющей последовательности. Эти значения выбираются так, чтобы принимающий узел, который знает значение расширяющей последовательности передающего узла, мог выделить данные передающего узла из суммарного сигнала, образующегося в результате одновременной передачи информации несколькими узлами.

Для того чтобы такую операцию демультиплексирования можно было выполнить, значения расширяющей последовательности выбираются определенным образом. Поясним идею CDMA на примере.

Пусть в сети работает четыре узла: A, B, C и D. Каждый узел использует такие значения расширяющей последовательности:

A: 0 0 0 0

B: 0 1 0 1

C: 0 0 1 1

D: 0 1 1 0

Предположим также, что при передаче единиц и нулей расширяющей последовательности (то есть уже преобразованного исходного кода) используются сигналы, которые являются аддитивными и инверсными. Инверсность означает, что двоичная единица кодируется, например, синусоидой с амплитудой +A, а двоичный ноль — синусоидой с амплитудой –A. Из условия аддитивности следует, что если фазы этих амплитуд совпадут, то при одновременной передаче единицы и нуля мы получим нулевой уровень сигнала. Для упрощения записи расширяющей последовательности обозначим синусоиду с положительной амплитудой значением +1, а синусоиду с отрицательной амплитудой — значением –1. Для простоты допустим также, что все узлы CDMA-сети синхронизированы.

Таким образом, при передаче единицы исходного кода 4 узла передают в среду такие последовательности:

A: –1 –1 –1 –1

B: –1 +1 –1 +1

C: –1 –1 +1 +1

D: –1 +1 +1 –1

При передаче нуля исходного кода сигналы расширяющей последовательности инвертируются.

Пусть теперь каждый из 4 узлов независимо от других передает в сеть один бит исходной информации: узел A — 1, узел B — 0, узел C — 0, узел D — 1.

В среде S сети наблюдается такая последовательность сигналов:

A: –1 –1 –1 –1

B: +1 –1 +1 –1

C: +1 +1 –1 –1

D: –1 +1 +1 –1

В соответствии со свойством аддитивности получаем:

S: 0 0 0 –4

Если, например, некоторый узел E хочет принимать информацию от узла A, то он должен использовать свой демодулятор CDMA, задав ему в качестве параметра значение расширяющей последовательности узла A.

Демодулятор CDMA работает следующим образом. Он последовательно складывает все четыре суммарных сигнала S_i, принятые в течение каждого такта работы.

При этом сигнал S_i принятый в такте, на котором код расширения станции A равен +1, учитывается в сумме со своим знаком, а сигнал, принятый в такте, на котором код расширения станции A равен −1, добавляется в сумму с противоположным знаком. Другими словами, демодулятор выполняет операцию скалярного умножения вектора принятых сигналов на вектор значения расширяющей последовательности нужной станции:

$$S \times A = (0\ 0\ 0\ -4) \times (-1\ -1\ -1\ -1) = 4.$$

Для того чтобы узнать, какой бит послала станция A, осталось нормализовать результат, то есть разделить его на количество станций сети: 4/4 = 1.

Если бы станция хотела принимать информацию от станции B, то ей нужно было бы при демодуляции использовать код расширения станции B (−1 +1 −1 +1):

$$S \times B = (0\ 0\ 0\ -4) \times (-1\ +1\ -1\ +1) = -4.$$

После нормализации мы получаем сигнал −1, который соответствует двоичному нулю исходной информации станции B.

Особенность расширяющих последовательностей, используемых в CDMA, состоит в том, что они являются взаимно ортогональными. Это значит, что если их рассматривать как векторы, то при попарном умножении они дают нулевой результат, например, взаимно ортогональными являются векторы координат пространства: (1 0 0), (0 1 0) и (0 0 1). Однако помимо взаимной ортогональности нужно, чтобы такие векторы были ортогональны с инверсиями членов набора векторов (так как инверсии применяются для кодирования нулей исходной информации).

Мы объяснили только основную идею CDMA, предельно упростив ситуацию. На практике CDMA является весьма сложной технологией, которая оперирует не условными значениями +1 и −1, а модулированными сигналами, например сигналами BPSK. Кроме того, узлы сети не синхронизированы между собой, а сигналы, которые приходят от удаленных на различные расстояния от приемника узлов, имеют разную мощность. Проблема синхронизации приемника и передатчика решается за счет передачи длинной последовательности определенного кода, называемого **пилотным сигналом**. Для того же, чтобы мощности всех передатчиков были примерно равны для базовой станции, в CDMA применяются специальные процедуры управления мощностью.

Выводы

Беспроводная связь делится на мобильную и фиксированную. Для организации мобильной связи беспроводная среда является единственной альтернативой. Фиксированная беспроводная связь обеспечивает доступ к узлам сети, расположенным в пределах небольшой территории, например здания.

Каждый узел беспроводной линии связи оснащается антенной, которая одновременно является передатчиком и приемником электромагнитных волн.

Электромагнитные волны могут распространяться во всех направлениях или же в пределах определенного сектора. Тип распространения зависит от типа антенны.

Беспроводные системы передачи данных делятся на четыре группы в зависимости от используемого диапазона электромагнитного спектра: широковещательные (радио-) системы, микроволновые системы, системы инфракрасных волн, системы видимого света.

Из-за отражения, дифракции и рассеивания электромагнитных волн возникает многолучевое распространение одного и того же сигнала. Это приводит к межсимвольной интерференции и многолучевому замиранию.

Передача данных в диапазонах 900 МГц, 2,4 ГГц и 5 ГГц, которые получили название ISM, не требует лицензирования, если мощность передатчика не превышает 1 Ватт.

Беспроводные двухточечные линии связи используются для создания радиорелейных линий, соединения зданий, а также двух компьютеров.

Беспроводные линии связи с одним источником и несколькими приемниками строятся на основе базовой станции. Такие линии используются в мобильных сотовых сетях, а также в системах фиксированного доступа.

Топология с несколькими источниками и несколькими приемниками характерна для беспроводных локальных сетей.

В системах спутниковой связи используются три группы спутников: геостационарный, среднеорбитальный и низкоорбитальный.

Для кодирования дискретной информации беспроводные системы используют манипуляцию (FSK и PSK), модуляцию с несколькими несущими частотами (OFDM) и методы расширения спектра (FHSS и DSSS).

В методах расширения спектра для представления информации используется широкий диапазон частот, это уменьшает влияние на сигналы узкополосных шумов.

На основе методов FHSS и DSSS можно мультиплексировать несколько каналов в одном диапазоне частот. Такая техника мультиплексирования называется множественным доступом с кодовым разделением (CDMA).

Вопросы и задания

1. Назовите основные области применения беспроводных линий связи.
2. В чем достоинства и недостатки беспроводной передачи информации по сравнению с проводной?
3. Каким образом можно организовать ненаправленное распространение радиоволн и микроволн?
4. За счет чего радиоволны с частотами от 2 до 30 МГц могут распространяться на сотни километров?
5. Какой спектр волн используется для спутниковой связи?
6. Какие атмосферные явления мешают распространению микроволн?
7. Что из нижеперечисленного используется для ненаправленного распространения инфракрасных волн:
   - ❍ лазерные диоды;
   - ❍ система линз;
   - ❍ отражение от потолка;
   - ❍ тепловые антенны.

8. Какие препятствия вызывают дифракцию, а какие рассеивание электромагнитных волн?
9. В каких случаях применяются эллиптические орбиты телекоммуникационных спутников?
10. Какими недостатками обладает геостационарный спутник?
11. В чем, по вашему мнению, заключается причина неудачи проекта Iridium?
12. При соблюдении какого условия технология FHSS является высокоскоростной?
13. Какое свойство последовательности Баркера определяет возможность ее использования в технологии DSSS?
14. Назовите основное свойство расширяющих последовательностей, используемых в технологии CDMA.
15. Можно ли в качестве расширяющих последовательностей узлов сети, поддерживающих множественный доступ с кодовым разделением на основе технологии DSSS, использовать значения 1 0 0 ... 0, 0 1 0 0 ... 0, 0 0 1 0 ... 0, 0 0 0 1 0 ...0 и т. д.?
16. Предложите 11-битную расширяющую последовательность, отличную от последовательности Баркера, которая, как и последовательность Баркера, позволяет надежно определять начало передачи очередного бита исходной информации.

ГЛАВА 11 Первичные сети

Первичные сети предназначены для создания коммутируемой инфраструктуры, с помощью которой можно достаточно быстро и гибко организовать постоянный канал с двухточечной топологией между двумя пользовательскими устройствами, подключенными к такой сети. В первичных сетях используется техника коммутации каналов. На основе каналов, образованных первичными сетями, работают наложенные компьютерные или телефонные сети. Каналы, предоставляемые первичными сетями своим пользователям, отличаются высокой пропускной способностью — обычно от 2 Мбит/с до 10 Гбит/с.

Существует три поколения технологий первичных сетей:

- плезиохронная цифровая иерархия (Plesiochronous Digital Hierarchy, PDH);
- синхронная цифровая иерархия (Synchronous Digital Hierarchy, SDH), которой в Америке соответствует стандарт SONET;
- уплотненное волновое мультиплексирование (DWDM).

Первые две технологии (PDH и SDH) для разделения высокоскоростного канала используют временное мультиплексирование (TDM) и передают данные в цифровой форме. Каждая из них поддерживает иерархию скоростей, так что пользователь может выбрать подходящую ему скорость для каналов, с помощью которых он будет строить наложенную сеть.

Технология SDH обеспечивает более высокие скорости, чем PDH, так что при построении крупной первичной сети ее магистраль строится на технологии SDH, а сеть доступа — на технологии PDH.

Сети DWDM представляют собой последнее достижение в области создания высокоскоростных каналов. Они уже не являются цифровыми, так как предоставляют своим пользователям выделенную волну для передачи информации, которую те могут задействовать по своему усмотрению — модулировать или кодировать. Технология DWDM вытесняет сегодня технологию SDH из протяженных магистралей на периферию сети, превращая SDH в технологию сетей доступа.

Три различные технологии коммутации и мультиплексирования позволяют создать гибкую и масштабируемую первичную сеть, способную обслуживать большое количество компьютерных и телефонных сетей.

Сети PDH

Список ключевых слов: мультиплексор T-1, система T-каналов, каналы типа E-1, E-2 и E-3, сигнальный протокол, «кража» бита.

Технология PDH была разработана в конце 60-х годов компанией AT&T для решения проблемы связи крупных коммутаторов телефонных сетей между собой. Линии связи FDM, применяемые ранее для решения этой задачи, исчерпали свои возможности по организации высокоскоростной многоканальной связи по одному кабелю. В технологии FDM для одновременной передачи данных 12 абонентских каналов использовалась витая пара, а для повышения скорости связи приходилось прокладывать кабели с большим количеством пар проводов или более дорогие коаксиальные кабели.

Иерархия скоростей

Начало технологии PDH было положено разработкой мультиплексора **T-1**, который позволял в цифровом виде мультиплексировать, передавать и коммутировать (на постоянной основе) голосовой трафик 24 абонентов. Так как абоненты по-прежнему пользовались обычными телефонными аппаратами, то есть передача голоса шла в аналоговой форме, то мультиплексоры T-1 сами осуществляли оцифровывание голоса с частотой 8000 Гц и кодировали голос методом импульсно-кодовой модуляции. В результате каждый абонентский канал образовывал цифровой поток данных 64 Кбит/с, а мультиплексор T-1 обеспечивал передачу 1,544 Мбит/с.

Для соединения крупных телефонных станций каналы T-1 представляли собой слишком слабые и негибкие средства мультиплексирования, поэтому была реализована идея образования каналов с *иерархией скоростей*. Четыре канала типа T-1 объединили в канал следующего уровня цифровой иерархии — T-2, передающий данные со скоростью 6,312 Мбит/с. Канал T-3, образованный путем объединения семи каналов T-2, имеет скорость 44,736 Мбит/с. Канал T-4 объединяет 6 каналов T-3, в результате его скорость равна 274 Мбит/с. Эта технология получила название **системы T-каналов**.

С середины 70-х годов выделенные каналы, построенные на основе систем T-каналов, стали сдаваться телефонными компаниями в аренду на коммерческих условиях, перестав быть внутренней технологией этих компаний. Системы T-каналов позволяют передавать не только голос, но и любые данные, представленные в цифровой форме, — компьютерные данные, телевизионное изображение, факсы и т. п.

Технология систем T-каналов была стандартизована Американским национальным институтом стандартов (ANSI), а позже — международным комитетом CCITT. При стандартизации она получила название Plesiochronous Digital Hierarchy (PDH). В результате внесенных комитетом CCITT изменений возникла несовместимость американской и международной версий стандарта PDH. Аналогом систем T-каналов в международном стандарте являются каналы типа **E-1**, **E-2** и **E-3** с отличающимися скоростями — соответственно 2,048 Мбит/с, 8,488 Мбит/с

и 34,368 Мбит/с. Американская версия сегодня помимо США распространена также в Канаде и Японии (с некоторыми различиями), а в Европе применяется международный стандарт CCITT.

Несмотря на различия, в американской и международной версиях технологии цифровой иерархии принято использовать одни и те же обозначения для иерархии скоростей — DSn (Digital Signal n). В табл. 11.1 приводятся значения для всех введенных стандартами уровней скоростей обеих технологий.

Таблица 11.1. Иерархия цифровых скоростей

Обозначение скорости	Америка			CCITT (Европа)		
	Количество голосовых каналов	Количество каналов предыдущего уровня	Скорость, Мбит/с	Количество голосовых каналов	Количество каналов предыдущего уровня	Скорость, Мбит/с
DS-0	1	1	64 Кбит/с	1	1	64 Кбит/с
DS-1	24	24	1,544	30	30	2,048
DS-2	96	4	6,312	120	4	8,488
DS-3	672	7	44,736	480	4	34,368
DS-4	4032	6	274,176	1920	4	139,264

На практике в основном используются каналы T-1/E-1 и T-3/E-3.

Методы мультиплексирования

Мультиплексор T-1 обеспечивает передачу данных 24-х абонентов со скоростью 1,544 Мбит/с в кадре, имеющем достаточно простой формат. В этом кадре последовательно передается по одному байту каждого абонента, а после 24 байт вставляется один *бит синхронизации*. Первоначально устройства T-1 (которые дали имя всей технологии, работающей на скорости 1,544 Мбит/с) работали только на внутренних тактовых генераторах, и каждый кадр с помощью битов синхронизации мог передаваться асинхронно. Аппаратура T-1 (а также более скоростная аппаратура T-2 и T-3) за долгие годы существования претерпела значительные изменения. Сегодня мультиплексоры и коммутаторы первичной сети работают на централизованной тактовой частоте, распределяемой из одной точки всей сети. Однако принцип формирования кадра остался, поэтому биты синхронизации в кадре по-прежнему присутствуют. Суммарная скорость пользовательских каналов составляет 24 × 64 = 1,536 Мбит/с, а еще 8 Кбит/с добавляют биты синхронизации, итого получается 1,544 Мбит/с.

Теперь рассмотрим еще одну особенность формата кадра T-1. В аппаратуре T-1 восьмой бит каждого байта в кадре имеет назначение, зависящее от типа передаваемых данных и поколения аппаратуры. При передаче *голоса* с помощью этого

бита передается служебная информация, переносящая номер вызываемого абонента и другие сведения, необходимые для установления соединения между абонентами сети. Протокол, обеспечивающий такое соединение, называется в телефонии **сигнальным протоколом**. Поэтому реальная скорость передачи пользовательских данных в этом случае составляет не 64, а 56 Кбит/с. Техника использования восьмого бита для служебных целей получила название **«кражи» бита**.

При передаче компьютерных данных канал Т-1 предоставляет для пользовательских данных только 23 канала, а 24-й канал отводится для служебных целей, в основном — для восстановления искаженных кадров. Компьютерные данные передаются со скоростью 64 Кбит/с, так как восьмой бит не «крадется».

При одновременной передаче как голосовых, так и компьютерных данных используются все 24 канала, причем и компьютерные, и голосовые данные передаются со скоростью 56 Кбит/с

При мультиплексировании 4-х каналов Т-1 в один канал Т-2 между кадрами DS-1 по-прежнему передается один бит синхронизации, а кадры DS-2 (которые состоят из 4-х последовательных кадров DS-1) разделяются 12 служебными битами, которые предназначены не только для разделения кадров, но и для их синхронизации. Соответственно, кадры DS-3 состоят из 7 кадров DS-2, разделенных служебными битами.

Версия технологии PDH, описанная в международных стандартах G.700–G.706 комитета CCITT, как уже отмечалось, имеет отличия от американской технологии систем Т-каналов. В частности, в ней не используется схема «кражи бита». При переходе к следующему уровню иерархии коэффициент кратности скорости имеет постоянное значение 4. Вместо восьмого бита в канале Е-1 на служебные цели отводятся 2 байта из 32, а именно нулевой (для целей синхронизации приемника и передатчика) и шестнадцатый (в нем передается служебная сигнальная информация). Для голосовых каналов или каналов данных остается 30 каналов со скоростью передачи 64 Кбит/с каждый.

Пользователь может арендовать несколько каналов 64 Кбит/с (56 Кбит/с) в канале Т-1/Е-1. Такой канал называется «дробным» каналом Т-1/Е-1. В этом случае пользователю отводится несколько тайм-слотов работы мультиплексора.

Физический уровень технологии PDH поддерживает различные виды кабелей: витую пару, коаксиальный кабель и волоконно-оптический кабель. Основным вариантом абонентского доступа к каналам Т-1/Е-1 является кабель из двух витых пар с разъемами RJ-48. Две пары требуются для организации дуплексного режима передачи данных со скоростью 1,544/2,048 Мбит/с. Для представления сигналов используются:

- в каналах Т-1 — биполярный потенциальный код B8ZS;
- в каналах Е-1 — биполярный потенциальный код HDB3.

Для усиления сигнала на линиях Т-1 через каждые 1800 м (одна миля) устанавливаются регенераторы и аппаратура контроля линии.

Коаксиальный кабель благодаря своей широкой полосе пропускания поддерживает канал Т-2/Е-2 или 4 канала Т-1/Е-1. Для работы каналов Т-3/Е-3 обычно

используется либо коаксиальный кабель, либо волоконно-оптический кабель, либо каналы СВЧ.

Физический уровень международного варианта технологии определяется стандартом G.703. Название этого стандарта используется также для обозначения типа интерфейса маршрутизатора или моста, подключаемого к каналу E-1. Американский вариант названия для интерфейса — T-1.

Ограничения технологии PDH

Как американский, так и международный варианты технологии PDH обладают недостатками, основным из которых является сложность и неэффективность операций мультиплексирования и демультиплексирования пользовательских данных. Сам термин «плезиохронный», то есть «почти синхронный», используемый для этой технологии, говорит о причине такого явления — отсутствии полной синхронности потоков данных при объединении низкоскоростных каналов в высокоскоростные. Изначально асинхронный подход к передаче кадров сделал необходимой вставку бита или нескольких битов синхронизации между кадрами.

В результате для извлечения пользовательских данных из объединенного канала необходимо полностью (!) демультиплексировать кадры объединенного канала. Например, чтобы получить данные одного абонентского канала 64 Кбит/с из кадров канала T-3, необходимо произвести демультиплексирование этих кадров до уровня кадров T-2, затем — до уровня кадров T-1, а в конце концов демультиплексировать и сами кадры T-1.

Если сеть PDH используется только в качестве транзитной магистрали между двумя крупными узлами, то операции мультиплексирования и демультиплексирования выполняются исключительно в конечных узлах, и проблем не возникает. Но если необходимо выделить один или несколько абонентских каналов в промежуточном узле сети PDH, то эта задача простого решения не имеет. Как вариант предлагается установка двух мультиплексоров уровня T3/E3 и выше в каждом узле сети (рис. 11.1). Первый выполняет полное демультиплексирование потока и отвод части низкоскоростных каналов абонентам, а второй опять собирает в выходной высокоскоростной поток оставшиеся каналы вместе с вновь вводимыми. При этом количество работающего оборудования удваивается.

Другой вариант — «обратная доставка». В промежуточном узле, где нужно выделить и отвести абонентский поток, устанавливается единственный высокоскоростной мультиплексор, который просто передает данные транзитом дальше по сети без их демультиплексирования. Эту операцию выполняет только мультиплексор конечного узла, после чего данные соответствующего абонента возвращаются по отдельной линии связи на промежуточный узел. Естественно, такие сложные взаимоотношения коммутаторов усложняют работу сети, требуют ее тонкого конфигурирования, что ведет к большому объему ручной работы и ошибкам.

Кроме этого, в технологии PDH не предусмотрены встроенные средства обеспечения отказоустойчивости и администрирования сети.

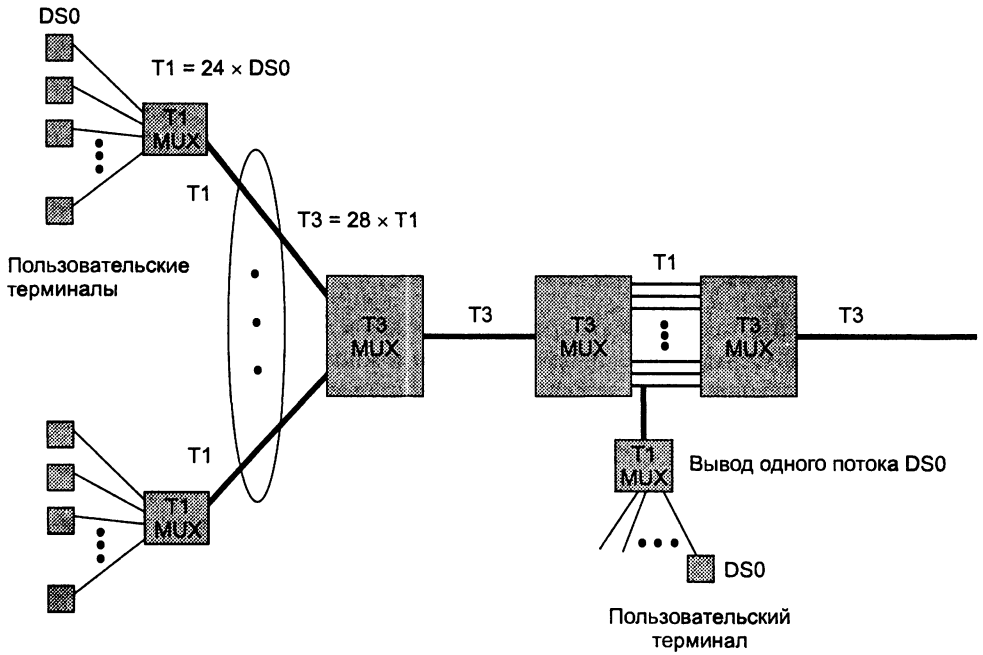


Рис. 11.1. Выделение низкоскоростного канала путем полного демультиплексирования

Наконец, недостатком PDH являются слишком низкие по современным понятиям скорости передачи данных. Волоконно-оптические кабели позволяют передавать данные со скоростями в несколько гигабит в секунду по одному волокну, что обеспечивает консолидацию в одном кабеле десятков тысяч пользовательских каналов, но эту возможность технология PDH не реализует — ее иерархия скоростей заканчивается уровнем 139 Мбит/с.

Сети SONET/SDH

Список ключевых слов: синхронная оптическая сеть, технология SONET, стандарт SDH, кадр STM-N, виртуальный контейнер, заголовок пути, таблица соединений (кросс-соединений), указатель, трибутарный блок, административный блок, мультиплексор, трибутарный порт, агрегатный порт, терминальный мультиплексор, мультиплексор ввода-вывода, цифровой кросс-коннектор, регенератор сигналов, стек протоколов SDH, фотонный уровень, уровень секции, уровень линии, уровень тракта, регенераторная секция, заголовок регенераторной секции, мультиплексная секция, заголовок мультиплексной секции, заголовок тракта, положительное выравнивание, отрицательное выравнивание, кольцо SDH, цепь, плоское кольцо, ячеистая топология, автоматическое защитное переключение, самовосстанавливающаяся сеть, защита 1+1, защита 1:1, защита 1:N, защитное переключение оборудования, защита карт, защита мультиплексной секции, защита сетевого соединения, разделяемая защита мультиплексной секции в кольцевой топологии.

Указанные выше недостатки были учтены и преодолены разработчиками технологии **синхронных оптических сетей** (Synchronous Optical NET, SONET), первый вариант стандарта которой появился в 1984 году. Затем она была стандартизована комитетом T-1 института ANSI. Международная стандартизация технологии проходила под эгидой Европейского института телекоммуникационных стандартов (European Telecommunications Standards Institute, ETSI) и сектором телекоммуникационной стандартизации союза ITU (ITU Telecommunication Standardization Sector, ITU-T) совместно с ANSI и ведущими телекоммуникационными компаниями Америки, Европы и Японии. Основной целью разработчиков международного стандарта было создание технологии, способной передавать трафик всех существующих цифровых каналов уровня PDH (как американских T1–T3, так и европейских E1–E4) по высокоскоростной магистральной сети на базе волоконно-оптических кабелей и обеспечить иерархию скоростей, продолжающую иерархию технологии PDH до скорости в несколько гигабит в секунду.

В результате длительной работы ITU-T и ETSI удалось разработать международный стандарт **SDH** (Synchronous Digital Hierarchy — синхронная цифровая иерархия). Кроме того, стандарт SONET был доработан так, чтобы аппаратура и сети SDH и SONET являлись совместимыми и могли мультиплексировать входные потоки практически любого стандарта PDH — и американского, и европейского.

Иерархия скоростей и методы мультиплексирования

Поддерживаемая технологией SONET/SDH иерархия скоростей представлена в табл. 11.2.

Таблица 11.2. Иерархия скоростей SONET/SDH

SDH	SONET	Скорость
	STS-1, OC-1	51,84 Мбит/с
STM-1	STS-3, OC-3	155,520 Мбит/с
STM-3	OC-9	466,560 Мбит/с
STM-4	OC-12	622,080 Мбит/с
STM-6	OC-18	933,120 Мбит/с
STM-8	OC-24	1,244 Гбит/с
STM-12	OC-36	1,866 Гбит/с
STM-16	OC-48	2,488 Гбит/с
STM-64	OC-192	9,953 Гбит/с
STM-256	OC-768	39,81 Гбит/с

В стандарте SDH все уровни скоростей (и, соответственно, форматы кадров для этих уровней) имеют общее название STM-N (Synchronous Transport Module level N — синхронный транспортный модуль уровня N). В технологии SONET существует два обозначения для уровней скоростей: STS-N (Synchronous Transport Signal level N — синхронный транспортный сигнал уровня N), употребляемое в случае передачи данных электрическим сигналом, и OC-N (Optical Carrier level N — оптоволоконная линия связи уровня N), употребляемое в случае передачи данных по волоконно-оптическому кабелю. Далее для упрощения изложения сосредоточимся на технологии SDH.

Кадры STM-N имеют достаточно сложную структуру, позволяющую агрегировать в общий магистральный поток потоки SDH и PDH различных скоростей, а также выполнять операции ввода-вывода без полного демультиплексирования магистрального потока.

Операции мультиплексирования и ввода-вывода выполняются при помощи **виртуальных контейнеров** (Virtual Container, VC), в которых блоки данных PDH можно транспортировать через сеть SDH. Помимо блоков данных PDH в виртуальный контейнер помещается еще некоторая служебная информация, в частности **заголовок пути** (Path OverHead, POH) контейнера, в котором размещается статистическая информация о процессе прохождении контейнера вдоль пути от его начальной до конечной точки (сообщения об ошибках), а также другие служебные данные, например индикатор установления соединения между конечными точками. В результате размер виртуального контейнера оказывается больше, чем соответствующая нагрузка в виде блоков данных PDH, которую он переносит. Например, виртуальный контейнер VC-12 помимо 32 байт данных потока E-1 содержит еще 3 байта служебной информации.

В технологии SDH (рис. 11.2) определено несколько типов виртуальных контейнеров, предназначенных для транспортировки основных типов блоков данных PDH: VC-11 (1,5 Мбит/с), VC-12 (2 Мбит/с), VC-2 (6 Мбит/с), VC3 (34/45 Мбит/с) и VC-4 (140 Мбит/с).

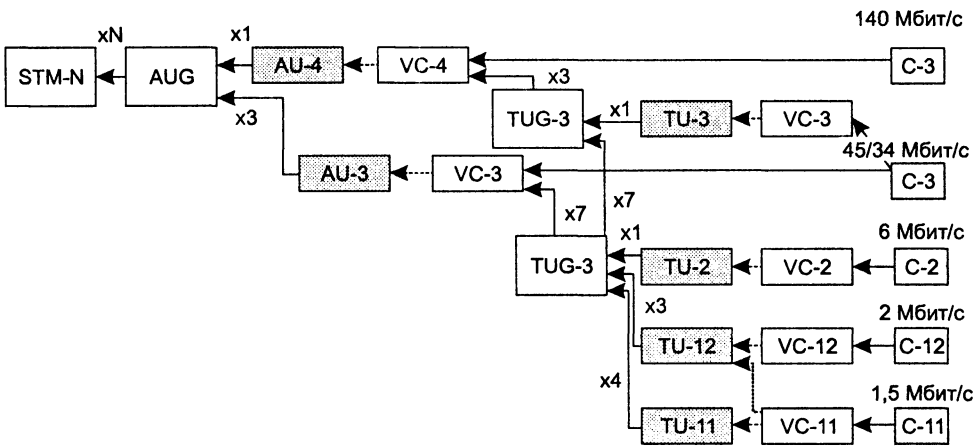


Рис. 11.2. Схема мультиплексирования данных в SDH

Виртуальные контейнеры являются *единицей коммутации* мультиплексоров SDH. В каждом мультиплексоре существует **таблица соединений** (называемая также **таблицей кросс-соединений**), в которой указано, например, что контейнер VC-12 порта P1 соединен с контейнером VC12 порта P5, а контейнер VC3 порта P8 — с контейнером VC3 порта P9. Таблицу соединений формирует администратор сети с помощью системы управления или управляющего терминала на каждом мультиплексоре так, чтобы обеспечить сквозной путь между конечными точками сети, к которым подключено пользовательское оборудование.

Для совмещения в рамках одной сети механизмов синхронной передачи кадров (STM-N) с асинхронным характером переносимых этими кадрами пользовательских данных PDH в технологии SDH применяются **указатели**. Концепция указателей — ключевая в технологии SDH, она заменяет принятое в PDH выравнивание скоростей асинхронных источников посредством дополнительных битов. Указатель определяет текущее положение виртуального контейнера в агригированной структуре более высокого уровня — **трибутарном блоке** (Tributary Unit, TU) или **административном блоке** (Administrative Unit, AU). Собственно, основное отличие этих блоков от виртуального контейнера заключается в наличии дополнительного поля указателя. С помощью этого указателя виртуальный контейнер может «смещаться» в определенных пределах внутри своего трибутарного или административного блока, положение которого, в свою очередь, в кадре фиксировано. Именно благодаря системе указателей мультиплексор находит положение пользовательских данных в синхронном потоке байтов кадров STM-N и «на лету» извлекает их оттуда, чего механизм мультиплексирования, применяемый в PDH, делать не позволяет.

Трибутарные блоки объединяются в группы, а те, в свою очередь, входят в административные блоки. Группа из N административных блоков (Administrative Unit Group, AUG) и образует полезную нагрузку кадра STM-N. Помимо этого в кадре имеется заголовок с общей для всех блоков AU служебной информацией. На каждом шаге преобразования к предыдущим данным добавляется несколько служебных байтов: они помогают распознать структуру блока или группы блоков и затем определить с помощью указателей начало пользовательских данных.

На рис. 11.2 структурные единицы кадра SDH, содержащие указатели, заштрихованы, а связь между контейнерами и блоками, допускающая сдвиг данных по фазе, показана пунктиром.

Схема мультиплексирования SDH предоставляет разнообразные возможности по объединению пользовательских потоков PDH. Например, для кадра STM-1 можно реализовать такие варианты:

- 1 поток E-4;
- 63 потока E-1;
- 1 поток E-3 и 42 потока E-1.

Другие варианты читатель может предложить сам.

Типы оборудования

Основным элементом сети SDH является **мультиплексор** (рис. 11.3). Обычно он оснащен некоторым количеством портов PDH и SDH: например, портами PDH на 2 и 34/45 Мбит/с и портами SDH STM-1 на 155 Мбит/с и STM-4 на 622 Мбит/с. Порты мультиплексора SDH делятся на агрегатные и трибутарные.

Трибутарные порты часто называют также портами ввода-вывода, а **агрегатные** — линейными портами. Эта терминология отражает типовые топологии сетей SDH, где имеется ярко выраженная магистраль в виде цепи или кольца, по которой передаются потоки данных, поступающие от пользователей сети через порты ввода-вывода (трибутарные порты), то есть втекающие в агрегированный поток («tributary» дословно означает «приток»).

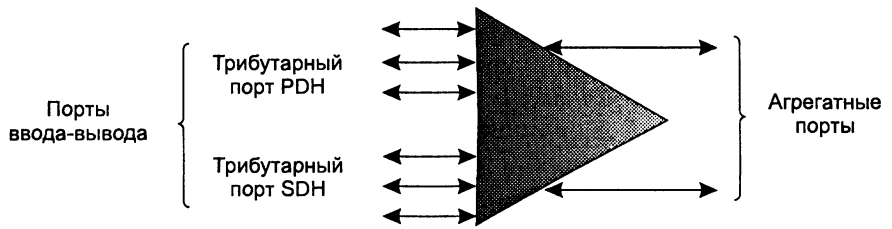


Рис. 11.3. Мультиплексор SDH

Мультиплексоры SDH обычно разделяют на два типа, разница между которыми определяется положением мультиплексора в сети SDH.

- **Терминальный мультиплексор** (Terminal Multiplexer, TM) *завершает* агрегатные каналы, мультиплексируя в них большое количество трибутарных каналов (рис. 11.4), поэтому он оснащен одним агрегатным портом и большим числом трибутарных портов.

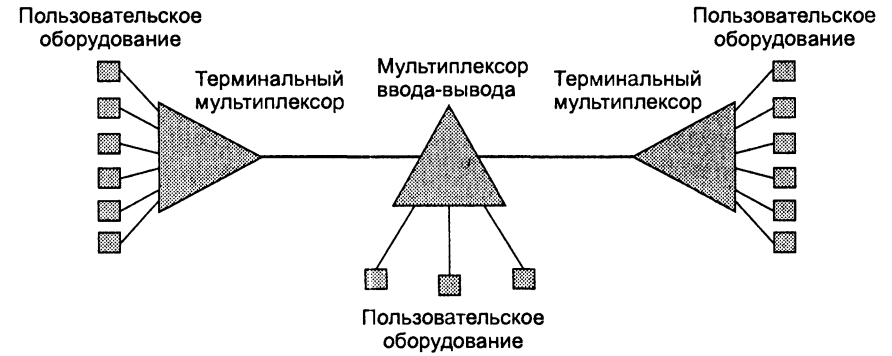


Рис. 11.4. Типы мультиплексоров SDH

- **Мультиплексор ввода-вывода** (Add-Drop Multiplexer, ADM) занимает промежуточное положение на магистрали (в кольце, цепи или смешанной топологии). Он имеет два агрегатных порта, транзитом передавая агрегатный по-

ток данных. С помощью небольшого количества трибутарных портов такой мультиплексор вводит в агрегатный поток или выводит из агрегатного потока данные трибутарных каналов.

Иногда различают так называемые **цифровые кросс-коннекторы** (Digital Cross-Connect, DXC) — мультиплексоры, которые выполняют операции коммутации над произвольными виртуальными контейнерами. В таких мультиплексорах не делается различий между агрегатными и трибутарными портами, так как они предназначены для работы в ячеистой топологии, где невозможно выделить агрегатные потоки.

Помимо мультиплексоров в состав сети SDH могут входить **регенераторы сигналов**, необходимые для преодоления ограничений по расстоянию между мультиплексорами. Эти ограничения зависят от мощности оптических передатчиков, чувствительности приемников и затухания волоконно-оптического кабеля. Регенератор преобразует оптический сигнал в электрический и обратно, при этом восстанавливается форма сигнала и его временны́е характеристики. В настоящее время регенераторы SDH применяются достаточно редко, так как стоимость их ненамного меньше стоимости мультиплексора, а функциональные возможности несоизмеримо беднее.

Стек протоколов

Стек протоколов SDH состоит из протоколов 4-х уровней. Эти уровни никак не соотносятся с уровнями модели OSI, для которой вся сеть SDH представляет собой оборудование физического уровня.

- **Фотонный уровень** имеет дело с кодированием битов информации путем модуляции света. Для кодирования оптического сигнала применяется потенциальный код NRZ, обладающий свойствами самосинхронизации.
- **Уровень секции** поддерживает физическую целостность сети. **Регенераторной секцией** в технологии SDH называется каждый непрерывный отрезок волоконно-оптического кабеля, который соединяет между собой такие, например, пары устройств SONET/SDH, как мультиплексор и регенератор, регенератор и регенератор, но не два мультиплексора. Компоненты регенераторной секции поддерживают протокол, который имеет дело с определенной частью заголовка кадра, называемой **заголовком регенераторной секции** (Regenerator Section OverHead, RSOH), на основе служебной информации может проводить тестирование секции и выполнять операции административного контроля.
- **Уровень линии** отвечает за передачу данных *по линии* между двумя мультиплексорами сети, поэтому линию также часто называют **мультиплексной секцией**. Протокол этого уровня работает с кадрами уровней STS-N для выполнения различных операций мультиплексирования и демультиплексирования, а также вставки и удаления пользовательских данных. Протокол линии также ответственен за реконфигурирование линии в случае отказа какого-либо ее элемента — оптического волокна, порта или соседнего мультиплексора. Служебная информация мультиплексной секции располагается в части заголовка

кадра, называемой **заголовком мультиплексной секции** (Multiplex Section OverHead, MSOH).

- **Уровень тракта** отвечает за доставку данных между двумя конечными пользователями сети. Тракт — это составное виртуальное соединение между пользователями. Протокол тракта должен принять данные, поступающие в пользовательском формате, например формате T-1, и преобразовать их в синхронные кадры STM-N.

На рис. 11.5 показано распределение протоколов SDH по типам оборудования SDH.

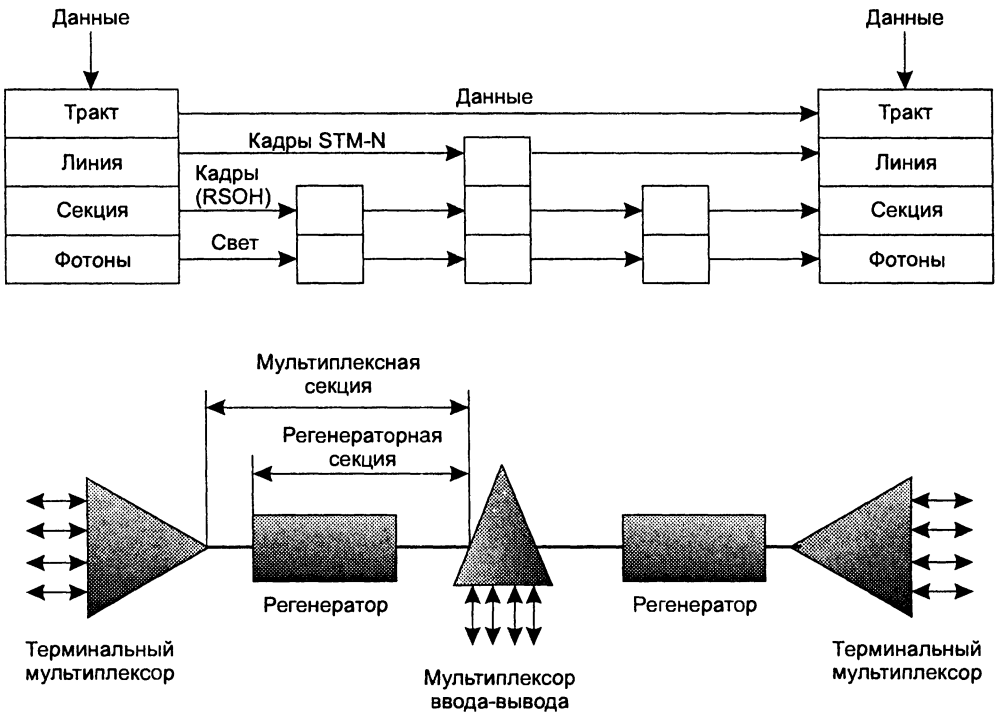


Рис. 11.5. Стек протоколов технологии SDH

Кадры STM-N

Основные элементы кадра STM-1 показаны на рис. 11.6, а в табл.11.3 приведена структура заголовков регенераторной и мультиплексной секций.

Кадр обычно представляют в виде матрицы, состоящей из 270 столбцов и 9 строк. Первые 9 байт каждой строки отводятся под служебные данные заголовков, из последующих 261 байт 260 отводятся под полезную нагрузку (данные таких структур, как AUG, AU, TUG, TU и VC), а один байт каждой строки отводится под **заголовок тракта**, что позволяет контролировать соединение «из конца в конец».

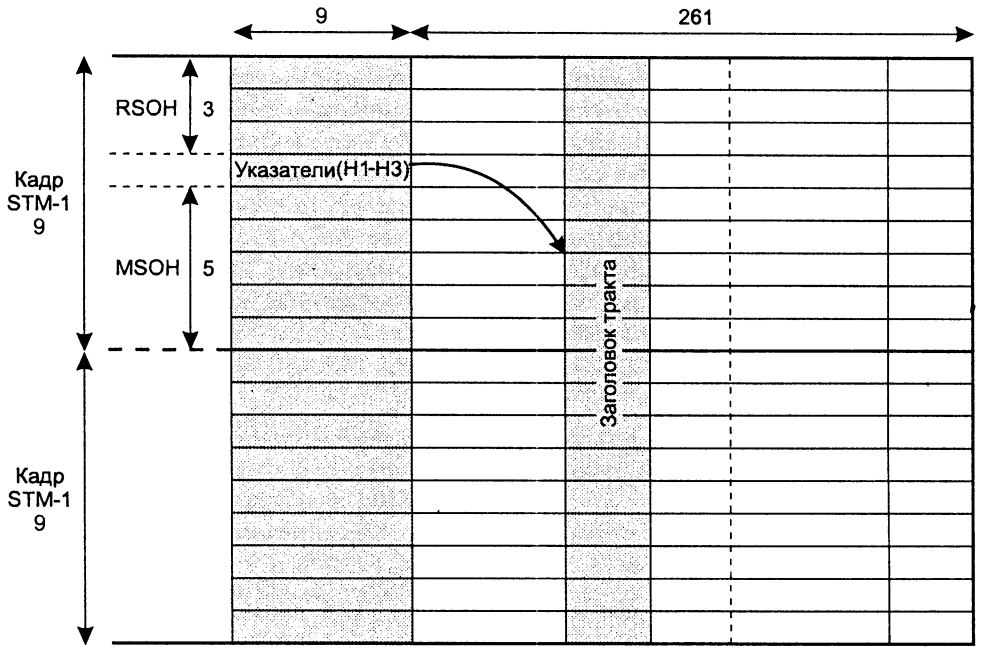


Рис. 11.6. Структура кадра STM-1

Таблица 11.3. Состав заголовков регенераторной и мультиплексной секций

Заголовок регенераторной секции	Заголовок мультиплексной секции
Синхробайты	Байты контроля ошибок для мультиплексной секции
Байты контроля ошибок для регенераторной секции	Шесть байтов канала передачи данных, работающего на скорости 576 Кбит/с
Один байт служебного аудиоканала (64 Кбит/с)	Два байта протокола автоматической защиты трафика (байты K1 и K2), обеспечивающего живучесть сети
Три байта канала передачи данных (Data Communication Channel, DCC), работающего на скорости 192 Кбит/с	Байт передачи сообщений статуса системы синхронизации
Байты, зарезервированные для национальных операторов связи	Остальные байты заголовка MSOH либо зарезервированы для национальных операторов связи, либо не используются
Поля указателей H1, H2, H3 задают положение начала виртуального контейнера VC-4 или трех виртуальных контейнеров VC-3 относительно поля указателей	

Рассмотрим механизм работы указателя H1-H2-H3 на примере кадра STM-1, переносящего контейнер VC-4. Указатель занимает 9 байт четвертого ряда кадра, причем под каждое из полей H1, H2 и H3 в этом случае отводится по 3 байта. Разрешенные значения указателя находятся в диапазоне 0–782, причем указатель отмечает начало контейнера VC-4 в трехбайтовых единицах. Например, если указатель имеет значение 27, то первый байт VC-4 находится на расстоянии 27 × 3 = 81 байт от последнего байта поля указателей, то есть является 90-м байтом (нумерация начинается с единицы) в 4-й строке кадра STM-1. Фиксированное значение указателя позволяет учесть фазовый сдвиг между данным мультиплексором и источником данных, в качестве которого может выступать мультиплексор PDH, оборудование пользователя с интерфейсом PDH или другой мультиплексор SDH. В результате виртуальный контейнер передается в двух последовательных кадрах STM-1, как и показано на рис. 11.6.

Указатель может отрабатывать не только фиксированный фазовый сдвиг, но и рассогласование тактовой частоты мультиплексора с тактовой частотой устройства, от которого принимаются пользовательские данные. Для компенсации этого эффекта значение указателя периодически наращивается или уменьшается на единицу.

Если скорость поступления данных контейнера VC-4 меньше, чем скорость отправки STM-1, то у мультиплексора периодически (этот период зависит от величины рассогласования частоты синхронизации) возникает нехватка пользовательских данных для заполнения соответствующих полей виртуального контейнера. Поэтому мультиплексор вставляет три «холостых» (незначащих) байта в данные виртуального контейнера, после чего продолжает заполнение VC-4 «подоспевшими» за время паузы пользовательскими данными. Указатель наращивается на единицу, что отражает запаздывание начала очередного контейнера VC-4 на три байта. Эта операция над указателем называется **положительным выравниванием**. В итоге средняя скорость отправляемых пользовательских данных становится равной скорости их поступления, причем без вставки дополнительных битов в стиле PDH.

Если же скорость поступления данных VC-4 выше, чем скорость отправки кадра STM-1, то у мультиплексора периодически возникает потребность вставки в кадр «лишних» (преждевременно пришедших) байтов, для которых в поле VC-4 нет места. Для их размещения используются три младших байта указателя, то есть поле H3 (само значение указателя умещается в поля H1 и H2). Указатель при этом уменьшается на единицу, поэтому такая операция носит название **отрицательного выравнивания**.

Тот факт, что выравнивание контейнера VC-4 происходит с дискретностью в три байта, объясняется достаточно просто. Дело в том, что в кадре STM-1 может переноситься либо один контейнер VC-4, либо три контейнера VC-3. Каждый из контейнеров VC-3 имеет в общем случае независимое значение фазы относительно начала кадра, а также собственную величину рассогласования частоты. Указатель VC-3 в отличие от указателя VC-4 состоит уже не из девяти, а из трех байтов: H1, H2, H3 (каждое из этих полей — однобайтовое). Эти три указателя помещаются в те же байты, что и указатель VC-4, но по схеме с чередованием

байтов, то есть в порядке H1-1, H1-2, H1-3, H2-1, H2-2, H2-3, H3-1, H3-2, H3-3 (второй индекс идентифицирует определенный контейнер VC-3). Значения указателей VC-3 интерпретируются в байтах, а не трехбайтовых единицах. При отрицательном выравнивании контейнера VC-3 лишний байт помещается в соответствующий байт H3-1, H3-2 или H3-3 — в зависимости от того, над каким из контейнеров VC-3 проводится операция.

Вот мы и дошли до размера смещения для контейнеров VC4 — этот размер был выбран для унификации этих операций над контейнерами любого типа, размещаемыми непосредственно в AUG кадра STM-1. Выравнивание контейнеров более низкого уровня всегда происходит с шагом в 1 байт.

При объединении блоков TU и AU в группы в соответствии с описанной схемой (см. рис. 11.6) выполняется их последовательное побайтное расслоение, так что период следования пользовательских данных в кадре STM-N совпадает с периодом их следования в трибутарных портах. Это исключает необходимость в их временной буферизации, поэтому говорят, что *мультиплексоры SDH передают данные в реальном масштабе времени.*

Типовые топологии

В сетях SDH применяются различные топологии связей. Наиболее часто используются кольца и линейные цепи мультиплексоров, также находит все большее применение ячеистая топология, близкая к полносвязной.

Кольцо SDH строится из мультиплексоров ввода-вывода, имеющих, по крайней мере, по два агрегатных порта (рис. 11.7, *а*). Пользовательские потоки вводятся в кольцо и выводятся из кольца через трибутарные порты, образуя двухточечные соединения (на рисунке показаны в качестве примера два таких соединения). Кольцо является классической регулярной топологией, обладающей потенциальной отказоустойчивостью — при однократном обрыве кабеля или выходе из строя мультиплексора соединение сохранится, если его направить по кольцу в противоположном направлении. Кольцо обычно строится на основе кабеля с двумя оптическими волокнами, но иногда для повышения надежности и пропускной способности применяют четыре волокна.

Цепь (рис. 11.7, *б*) — это линейная последовательность мультиплексоров, из которых два оконечных играют роль терминальных мультиплексоров, а остальные — мультиплексоров ввода-вывода. Обычно сеть с топологией цепи применяется в тех случаях, когда узлы имеют соответствующее географическое расположение, например вдоль магистрали железной дороги или трубопровода. Правда, в таких случаях может применяться и **плоское кольцо** (рис. 11.7, *в*), обеспечивающее более высокий уровень отказоустойчивости за счет двух дополнительных волокон в магистральном кабеле и по одному дополнительному агрегатному порту у терминальных мультиплексоров.

Эти базовые топологии могут комбинироваться при построении сложной и разветвленной сети SDH, образуя участки с радиально-кольцевой топологией, соединениями «кольцо-кольцо» и т. п. Наиболее общим случаем является **ячеистая топология сети** (рис. 11.7, *г*), при которой мультиплексоры соединяются

друг с другом большим количеством связей, за счет чего сеть может достичь очень высокой степени производительности и надежности.

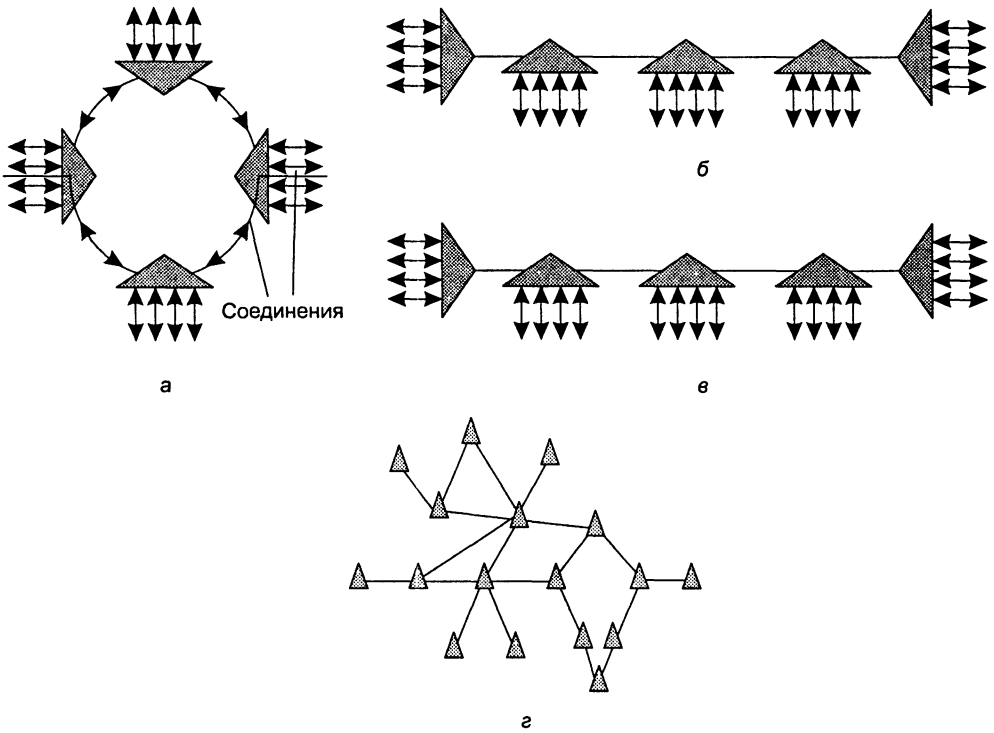


Рис. 11.7. Типовые топологии

Методы обеспечения живучести сети

Одной из сильных сторон первичных сетей SDH является разнообразный набор средств отказоустойчивости, который позволяет сети быстро (за десятки миллисекунд) восстановить работоспособность в случае отказа какого-либо элемента сети — линии связи, порта или карты мультиплексора, мультиплексора в целом.

В SDH в качестве общего названия механизмов отказоустойчивости используется термин **автоматическое защитное переключение** (Automatic Protection Switching, APS), отражающий факт перехода (переключения) на резервный путь или резервный элемент мультиплексора при отказе основного. Сети, поддерживающие такой механизм, в стандартах SDH названы **самовосстанавливающимися**.

В сетях SDH применяются три схемы защиты.

- **Защита 1+1** означает, что резервный элемент выполняет ту же работу, что и основной. Например, при защите трибутарной карты по схеме 1+1 трафик проходит как через рабочую карту (резервируемую), так и через защитную (резервную).

- **Защита 1:1** подразумевает, что защитный элемент в нормальном режиме не выполняет функции защищаемого элемента, а переключается на них только в случае отказа.
- **Защита 1:N** предусматривает выделение одного защитного элемента на N защищаемых. При отказе одного из защищаемых элементов его функции начинает выполнять защитный, при этом остальные элементы остаются без защиты — до тех пор, пока отказавший элемент не будет заменен.

В зависимости от типа защищаемого путем резервирования элемента сети в оборудовании и сетях SDH применяются следующие основные виды автоматической защиты: защитное переключение оборудования, защита карт, защита мультиплексной секции, защита сетевого соединения, разделяемая защита мультиплексной секции в кольцевой топологии.

- **Защитное переключение оборудования** (Equipment Protection Switching, EPS) — защита блоков и элементов оборудования SDH. Применяется для таких жизненно важных элементов мультиплексора, как процессорный блок, блок коммутации (кросс-коннектор), блок питания, блок ввода сигналов синхронизации и т. п. EPS обычно работает по схеме 1+1 или 1:1.
- **Защита карт** (Card Protection, CP) — защита агрегатных и трибутарных карт мультиплексора; позволяет мультиплексору автоматически продолжать работу в случае отказа одной из агрегатных или трибутарных карт. Используется защита по схемам 1+1, 1:1 и 1:N. Защита 1+1 обеспечивает непрерывность транспортного сервиса, так как трафик пользовательских соединений не прерывается при отказе карты. В приведенном на рис. 11.8 примере в мультиплексоре поддерживается защита CP трибутарных двухпортовых карт по схеме 1+1. Одна из трибутарных карт является основной, или рабочей, а другая — защитной. Режим работы пары связанных таким образом карт задается командой конфигурирования мультиплексора. В режиме, когда обе трибутарные карты являются работоспособными, трафик обрабатывается параллельно каждой картой.

Для переключения трафика между трибутарными картами используется дополнительная карта-переключатель. Входящий трафик каждого порта поступает на входной мост карты-переключателя, который разветвляет трафик и передает его на входы соответствующих портов трибутарных карт. Агрегатная карта получает оба сигнала STM-N от трибутарных карт и выбирает сигнал только от активной в данный момент карты. Исходящий трафик от агрегатной карты также обрабатывается обеими трибутарными картами, но карта-переключатель передает на выход только трафик от активной карты.

При отказе основной карты или другом событии, требующем перехода на защитную карту (деградация сигнала, ошибка сигнала, удаление карты), агрегатная карта по команде от блока управления мультиплексором переходит на прием сигнала от защитной трибутарной карты. Одновременно карта-переключатель также начинает передавать на выход сигналы выходящего трафика от защитной карты.

Данный способ обеспечивает автоматическую защиту всех соединений, проходящих через защищаемую карту. При установлении CP-защиты конфигурация соединений рабочей карты дублируется для защитной карты.

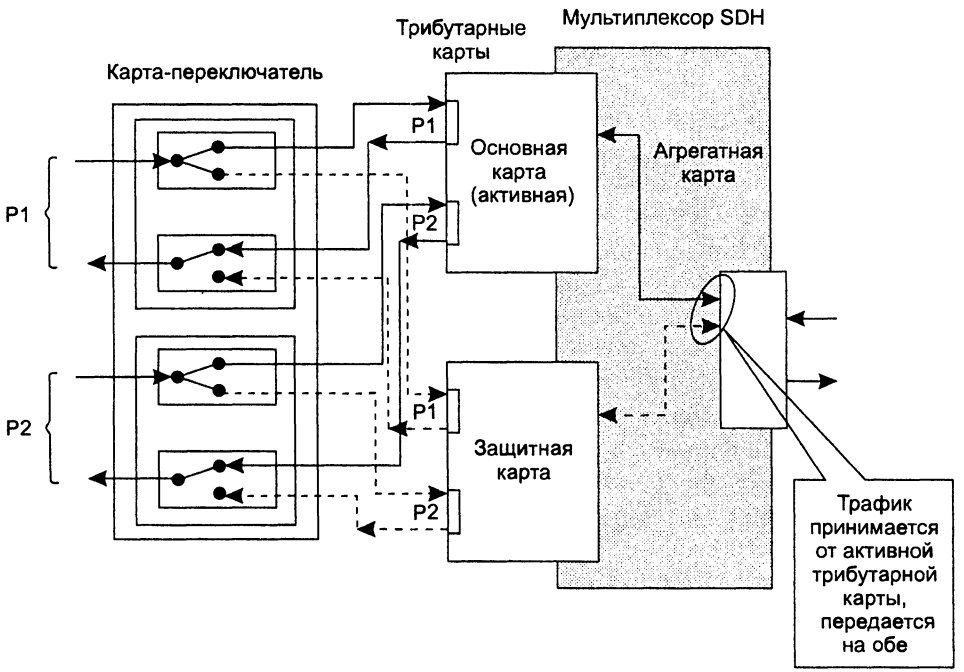


Рис. 11.8. Защита карт по схеме 1+1

Защита мультиплексной секции (Multiplex Section Protection, MSP), то есть участка сети между двумя смежными мультиплексорами SDH, действует более избирательно по сравнению с защитой карт. Защищается секция между двумя мультиплексорами, включающая два порта и линию связи (возможно, в свою очередь, включающую регенераторы, но не мультиплексоры). Обычно применяется схема защиты 1+1. При этом для рабочего канала (верхняя пара соединенных кабелем портов на рис. 11.9, *а*) конфигурируется защитный канал (нижняя пара портов). При установлении защиты MSP в каждом мультиплексоре необходимо выполнить конфигурирование, указав связь между рабочим и защитным портами. В исходном состоянии весь трафик передается по обоим каналам (как по рабочему, так и по защитному).

Существует однонаправленная и двунаправленная защита MSP. При однонаправленной защите (именно этот случай показан на рисунке) решение о переключении принимает только один из мультиплексоров — тот, который является приемным для отказавшего канала. Этот мультиплексор после обнаружения отказа (отказ порта, ошибка сигнала, деградация сигнала и т. п.) переходит на прием по защитному каналу. При этом передача и прием ведутся по разным портам (рис. 11.9, *б*).

При двунаправленной защите MSP при отказе рабочего канала в каком-либо направлении выполняется полное переключение на защитные порты мультиплексоров. Для уведомления передающего (по рабочему каналу) мультиплексора о необходимости переключения принимающий мультиплексор использует протокол, называемый протоколом «K-байт». Этот протокол указывает в двух байтах

заголовка кадра STM-N статус рабочего и защитного каналов, а также детализирует информацию об отказе. Механизм MSP обеспечивает защиту всех соединений, проходящих через защищаемую мультиплексную секцию. Время переключения защиты MSP, согласно требованиям стандарта, не должно превышать 50 мс.

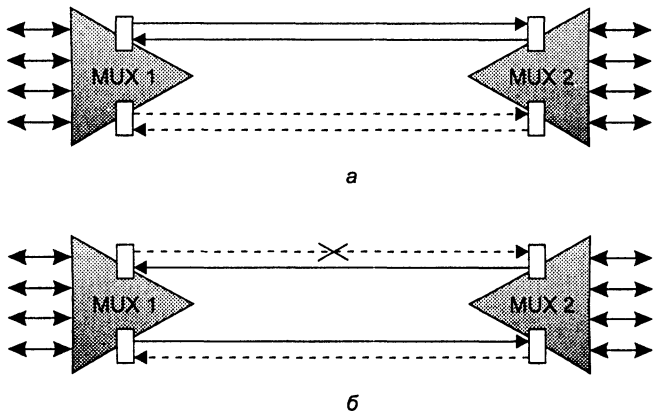


Рис. 11.9. Защита мультиплексной секции

Защита сетевого соединения (Sub-Network Connection Protection, SNC-P), то есть защита пути (соединения) через сеть для определенного виртуального контейнера, обеспечивает переключение определенного пользовательского соединения на альтернативный путь при отказе основного пути. Объектом защиты SNC-P является трибутарный трафик, помещенный в виртуальный контейнер определенного типа (например, в VC12, VC-3 или VC-4). Используется схема защиты 1+1.

Защита SNC-P конфигурируется в двух мультиплексорах — во входном, в котором трибутарный трафик, помещенный в виртуальный контейнер, разветвляется, и в выходном, в котором сходятся два альтернативных пути трафика. Пример защиты SNC-P показан на рис. 11.10. В мультиплексоре ADM1 для виртуального контейнераVC-4 трибутарного порта T-2 заданы два соединения: с одним из четырех контейнеров VC-4 агрегатного порта A1 и с одним из четырех контейнеров VC-4 агрегатного порта A2. Одно из соединений конфигурируется как рабочее, а второе — как защитное, при этом трафик передается по обоим соединениям. Промежуточные (для данных соединений) мультиплексоры конфигурируются обычным образом. В выходном мультиплексоре контейнер VC-4 трибутарного порта T-3 также соединяется с контейнерами — агрегатного порта A1 и агрегатного порта A2. Из двух поступающих на порт T3 потоков выбирается тот, качество которого выше (при равном нормальном качестве выбирается сигнал из агрегатного порта, выбранном при конфигурировании в качестве рабочего).

Защита SNC-P работает в любых топологиях сетей SDH, в которых имеются альтернативные пути следования трафика, то есть кольцевых и ячеистых.

Разделяемая защита мультиплексной секции в кольцевой топологии (Multiplex Section Shared Protection Ring, MS-SPRing) обеспечивает в некоторых случаях более экономичную защиту трафика в кольце. Хотя защита SNC-P вполне подходит для

кольцевой топологии сети SDH, в некоторых случаях ее применение снижает полезную пропускную способность кольца, так как каждое соединение потребляет удвоенную полосу пропускания вдоль всего кольца. Так, в кольце STM-16 можно установить только 16 защищенных по SNC-P соединений VC-4 (рис. 11.11).

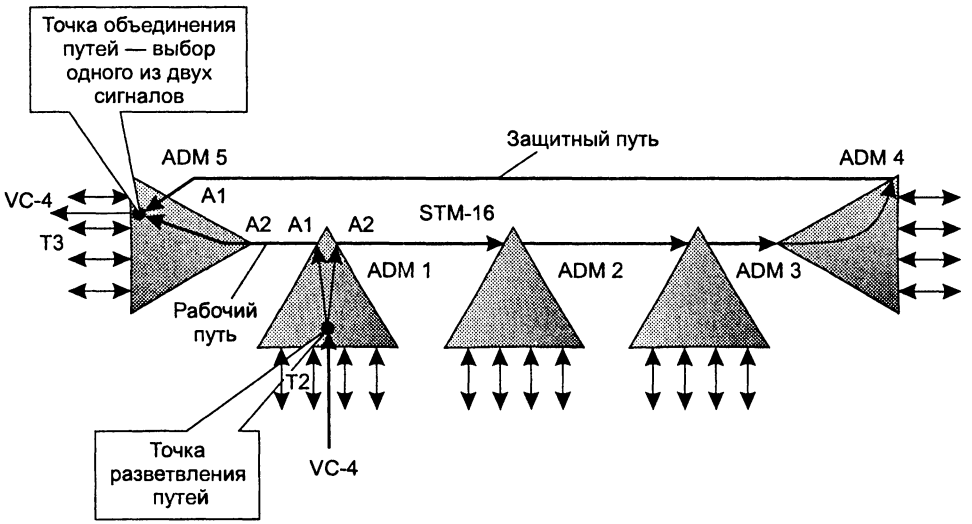


Рис. 11.10. Защита сетевого соединения

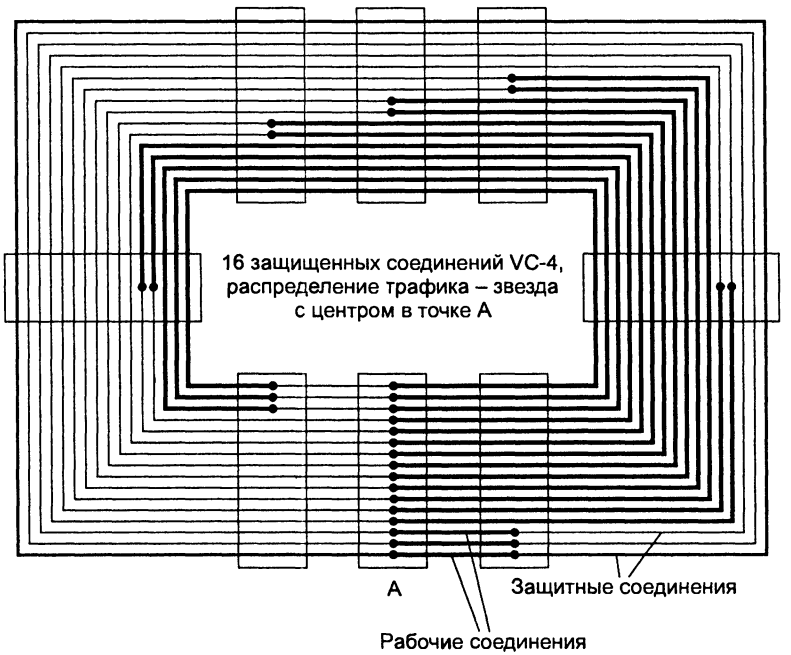


Рис. 11.11. Защита SNC-P в кольце

Защита MS-SPRing позволяет использовать пропускную способность кольца эффективно, так как полоса пропускания не резервируется заранее для каждого соединения. Вместо этого резервируется половина пропускной способности кольца, но эта резервная полоса выделяется для соединений динамически, по мере необходимости, то есть после обнаружения факта отказа линии или мультиплексора. Степень экономии полосы при применении защиты MS-SPRing зависит от распределения трафика.

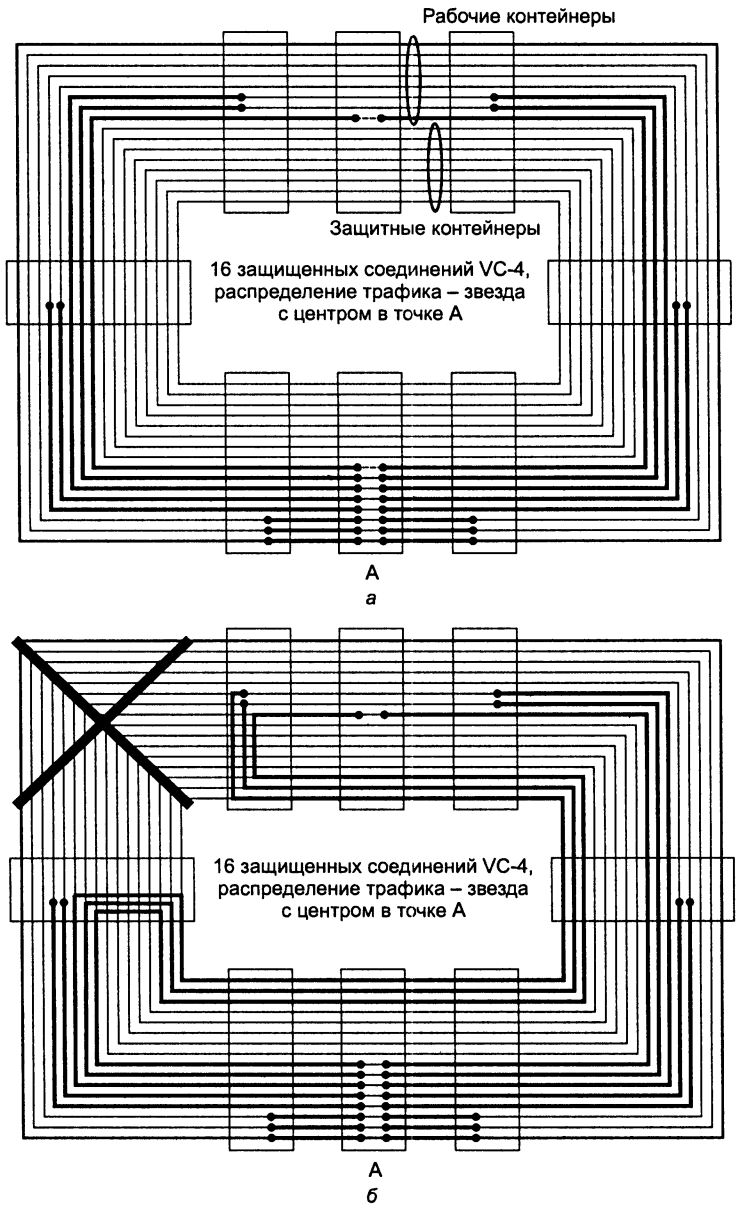


Рис. 11.12. Защита с разделением кольца

Если весь трафик сходится в один мультиплексор, то есть имеет распределение звезда, защита MS-SPRing экономии по сравнению с SNC-P вообще не дает. Пример такой ситуации представлен на рис. 11.12, *а*, где центром «тяготения» трафика является мультиплексор A, а в кольце установлены те же 16 защищенных соединений, что и в примере защиты SNC-P на рис. 11.11. Для защиты соединений резервируется 8 из 16 виртуальных контейнеров агрегатного потока STM-16.

При возникновении неисправности, например обрыве линии, как это показано на рис. 11.12 *б*), трафик в мультиплексорах, между которыми нарушилась связь, «разворачивается» в обратном направлении. Для этого используются резервные виртуальные контейнеры агрегатных портов, с которыми соединяются виртуальные контейнеры пострадавших соединений. В то же время соединения, на которые отказ не повлиял, работают в прежнем режиме, не подключая резервные контейнеры. Для уведомления мультиплексоров о реконфигурировании кольца используется протокол K-байт. Время переключения на защитные соединения MS-SPRing составляет около 50 мс. При смешанном распределении трафика экономия полосы в кольце MS-SPRing может быть еще более значительной.

Сети DWDM

Список ключевых слов: уплотненное волновое мультиплексирование, лямбда, технология волнового мультиплексирования, частотный план с шагом 100, 50 и 25 ГГц, высокоуплотненное волновое мультиплексирование, полностью оптическая сеть, двухточечная цепь, цепь с промежуточными подключениями, оптический мультиплексор ввода-вывода, кольцевая топология, ячеистая топология, оптический кросс-коннектор, тонкопленочный фильтр, дифракционная фазовая решетка (дифракционная структура), фазар, оптоэлектронный кросс-коннектор, полностью оптический кросс-коннектор (фотонный коммутатор), микроэлектронная механическая система.

Технология **уплотненного волнового мультиплексирования** (Dense Wave Division Multiplexing, DWDM) предназначена для создания оптических магистралей нового поколения, работающих на мультигигабитных и терабитных скоростях. Такой революционный скачок производительности обеспечивает принципиально иной, нежели у SDH, метод мультиплексирования — информация в оптическом волокне передается одновременно большим количеством световых волн — **лямбд** — термин возник в связи с традиционным для физики обозначением длины волны λ.

Сети DWDM работают по принципу коммутации каналов, при этом каждая световая волна представляет собой отдельный *спектральный канал* и несет собственную информацию.

Оборудование DWDM не занимается непосредственно проблемами передачи данных на каждой волне, то есть способом кодирования информации и протоколом ее передачи. Его основными функциями являются операции *мультиплексирования* и *демультиплексирования*, а именно — объединение различных волн в одном

световом пучке и выделение информации каждого спектрального канала из общего сигнала. Наиболее развитые устройства DWDM могут также *коммутировать* волны.

ВНИМАНИЕ

Технология DWDM является революционной не только потому, что в десятки раз повышает верхний предел скорости передачи данных по оптическому волокну. Она также открывает новую эру в технике мультиплексирования и коммутации, выполняя эти операции над световыми сигналами без преобразования их в электрическую форму. Все другие типы технологий, которые также используют световые сигналы для передачи информации по оптическим волокнам, например SDH и Gigabit Ethernet, обязательно преобразуют световые сигналы в электрические и только потом могут их мультиплексировать и коммутировать.

Первым применением технологии DWDM были протяженные магистрали, предназначенные для связи двух сетей SDH. При такой простейшей двухточечной топологии способность устройств DWDM выполнять коммутацию волн является излишней, однако по мере развития технологии и усложнения топологии сетей DWDM эта функция становится востребованной.

Принципы работы

Сегодня оборудование DWDM позволяет передавать по одному оптическому волокну 32 и более волн разной длины в окне прозрачности 1550 нм, при этом каждая волна может переносить информацию со скоростью до 10 Гбит/с (при применении протоколов технологий STM или 10 Gigabit Ethernet для передачи информации на каждой волне). В настоящее время ведутся работы по повышению скорости передачи информации на одной длине волны до 40–80 Гбит/с.

У технологии DWDM имеется предшественница — **технология волнового мультиплексирования** (Wave Division Multiplexing, WDM), которая использует четыре спектральных канала в окнах прозрачности 1310 нм и 1550 нм с разносом несущих в 800–400 ГГц. (Поскольку стандартной классификации WDM не существует, то встречаются системы WDM и с другими характеристиками.)

Мультиплексирование DWDM называется «уплотненным» из-за того, что в нем используется существенно меньшее расстояние между длинами волн, чем в WDM. На сегодня рекомендацией G.692 сектора ITU-T определены два *частотных плана* (то есть набора частот, отстоящих друг от друга на некоторую постоянную величину):

- частотный план с разнесением частот между соседними каналами **100 ГГц** ($\Delta\lambda \approx 0{,}8$ нм), в соответствии с которым для передачи данных применяется 41 волна в диапазоне от 1528,77 (196,1 ТГц) до 1560,61 нм (192,1 ТГц);
- частотный план с шагом **50 ГГц** ($\Delta\lambda \approx 0{,}4$ нм), позволяющий передавать в этом же диапазоне 81 длину волны.

Некоторыми компаниями выпускается также оборудование, называемое оборудованием **высокоуплотненного волнового мультиплексирования** (High-Dense

WDM, HDWDM), способное работать с частотным планом с шагом **25 ГГц** (сегодня это чаще всего экспериментальные образцы, а не серийная продукция).

Реализация частотных планов с шагом 50 ГГц и 25 ГГц предъявляет гораздо более жесткие требования к оборудованию DWDM, особенно в том случае, если каждая волна переносит сигналы со скоростью модуляции 10 Гбит/с и выше (STM-64, 10GE или STM-256). Еще раз подчеркнем, что сама технология DWDM (как и WDM) не занимается непосредственно кодированием переносимой на каждой волне информации — это проблема более высокоуровневой технологии, которая пользуется предоставленной ей волной по своему усмотрению и может передавать на этой волне как дискретную, так и аналоговую информацию. Такими технологиями могут быть SDH или 10 Gigabit Ethernet. Теоретически зазоры между соседними волнами в 50 ГГц и даже 25 ГГц позволяют передавать данные со скоростями 10 Гбит/с, но при этом нужно обеспечить высокую точность частоты и минимально возможную ширину спектра несущей волны, а также снизить уровень шумов, чтобы минимизировать эффект перекрытия спектра (рис. 11.13).

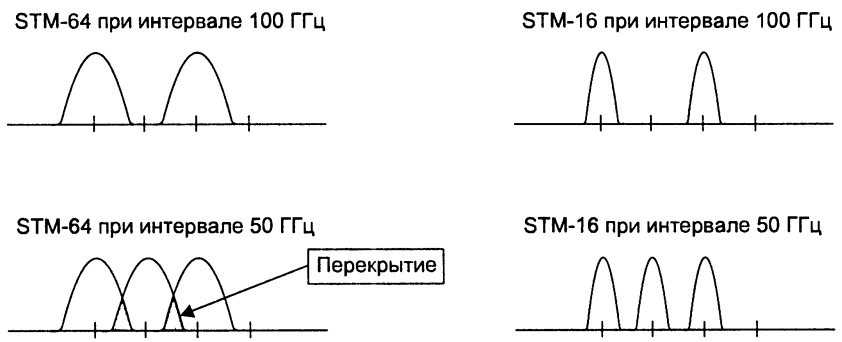


Рис. 11.13. Перекрытие спектра соседних волн для разных частотных планов и скоростей передачи данных

Волоконно-оптические усилители

Практический успех технологии DWDM, оборудование которой уже работает на магистралях многих ведущих мировых операторов связи, во многом определило появление волоконно-оптических усилителей. Эти оптические устройства непосредственно усиливают световые сигналы в диапазоне 1550 нм, исключая необходимость промежуточного преобразования их в электрическую форму, как это делают регенераторы, применяемые в сетях SDH. Системы электрической регенерации сигналов весьма дороги и, кроме того, зависят от протокола, так как они должны воспринимать определенный вид кодирования сигнала. Оптические усилители, «прозрачно» передающие информацию, позволяют наращивать скорость магистрали без необходимости модернизировать усилительные блоки.

Протяженность участка между оптическими усилителями может достигать 150 км и более, что обеспечивает экономичность создаваемых магистралей DWDM,

в которых длина мультиплексной секции составляет на сегодня 600–3000 км при применении от 1 до 7 промежуточных оптических усилителей.

В рекомендации ITU-T G.692 определены три типа усилительных участков, то есть участков между двумя соседними мультиплексорами DWDM:

- **L** (Long) — участок состоит максимум из 8 пролетов волоконно-оптических линий связи и 7 оптических усилителей, максимальное расстояние между усилителями — до 80 км при общей максимальной протяженности участка 640 км;
- **V** (Very long) — участок состоит максимум из 5 пролетов волоконно-оптических линий связи и 4 оптических усилителей, максимальное расстояние между усилителями — до 120 км при общей максимальной протяженности участка 600 км;
- **U** (Ultra long) — участок без промежуточных усилителей длиной до 160 км.

Ограничения на количество пассивных участков и их длину связаны с деградацией оптического сигнала при его оптическом усилении. Хотя оптический усилитель восстанавливает мощность сигнала, он не полностью компенсирует эффект хроматической дисперсии (то есть распространения волн разной длины с разной скоростью, из-за чего сигнал на приемном конце волокна «размазывается»), а также другие нелинейные эффекты. Поэтому для построения более протяженных магистралей необходимо между усилительными участками устанавливать DWDM-мультиплексоры, выполняющие регенерацию сигнала путем его преобразования в электрическую форму и обратно. Для уменьшения нелинейных эффектов в системах DWDM применяется также ограничение мощности сигнала.

Оптические усилители используются не только для увеличения расстояния между мультиплексорами, но и внутри самих мультиплексоров. Если мультиплексирование и кросс-коммутация выполняются исключительно оптическими средствами, без преобразования в электрическую форму, то сигнал при пассивных оптических преобразованиях теряет мощность и его нужно усиливать перед передачей в линию.

Новые исследования привели к появлению усилителей, работающих в так называемом L-диапазоне (4-е окно прозрачности), от 1570 до 1605 нм. Использование этого диапазона, а также сокращение расстояния между волнами до 50 ГГц и 25 ГГц позволяет нарастить количество одновременно передаваемых длин волн до 80–160 и более, то есть обеспечить передачу трафика со скоростями 800 Гбит/с–1,6 Тбит/с в одном направлении по одному оптическому волокну. С успехами DWDM связано еще одно перспективное технологическое направление — **полностью оптические сети**. В таких сетях все операции по мультиплексированию/демультиплексированию, вводу-выводу и кросс-коммутации (маршрутизации) пользовательской информации выполняются без преобразования сигнала из оптической формы в электрическую. Исключение преобразований в электрическую форму позволяет существенно удешевить сеть. Однако возможности оптических технологий пока еще недостаточны для создания полностью оптических масштабных сетей, поэтому их практическое применение ограничено фрагментами, между которыми выполняется электрическая регенерация сигнала.

Типовые топологии

Хронологически первой областью применения технологии DWDM (как и технологии SDH) стало создание сверхдальних высокоскоростных магистралей, имеющих топологию **двухточечной цепи** (рис. 11.14).

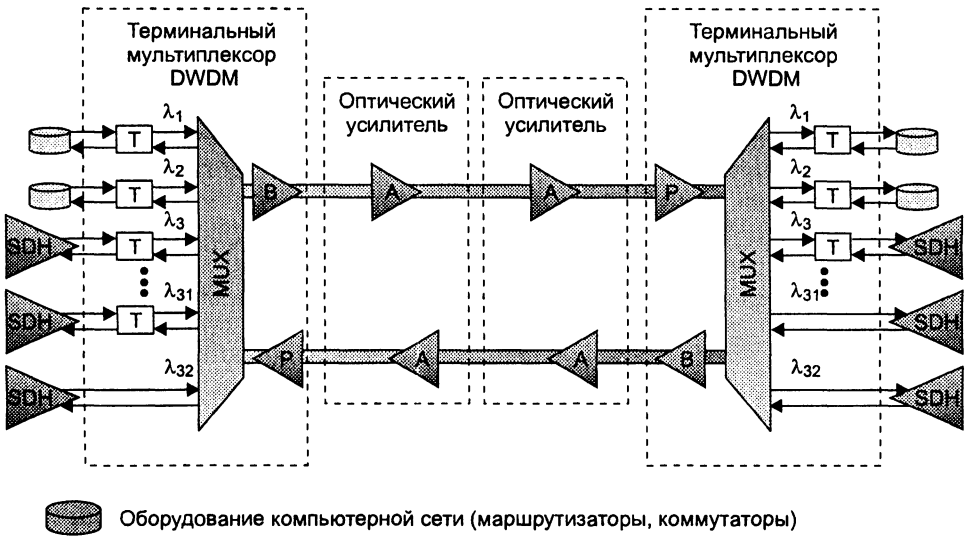


Рис. 11.14. Сверхдальняя двухточечная связь на основе терминальных мультиплексоров DWDM

Для организации такой магистрали достаточно в ее конечных точках установить терминальные мультиплексоры DWDM, а в промежуточных точках — оптические усилители, если этого требует расстояние между конечными точками.

В приведенной на рис. 11.4 схеме дуплексный обмен между абонентами сети происходит за счет однонаправленной передачи всего набора волн по двум волокнам. Существует и другой вариант работы сети DWDM, когда для связи узлов сети используется одно волокно. Дуплексный режим достигается путем двунаправленной передачи оптических сигналов по волокну — половина волн частотного плана передают информацию в одном направлении, половина — в обратном.

Естественным развитием топологии двухточечной цепи является **цепь с промежуточными подключениями**, в которой промежуточные узлы выполняют функции мультиплексоров ввода-вывода (рис. 11.15).

Оптические мультиплексоры ввода-вывода (Optical Add-Drop Multiplexer, OADM) могут вывести из общего оптического сигнала волну определенной длины и ввести туда сигнал этой же длины волны, так что спектр транзитного сигнала не изменится, а соединение будет выполнено с одним из абонентов, подключенных к промежуточному мультиплексору. OADM может выполнять операции ввода-вывода волн оптическими средствами или путем промежуточного преобразования в электрическую форму. Обычно полностью оптические (пассивные) мультиплексоры ввода-вывода могут отводить небольшое число волн, так как каждая операция вывода требует последовательного прохождения оптического сигнала через

оптический фильтр, который вносит дополнительное затухание. Если же мультиплексор выполняет электрическую регенерацию сигнала, то количество выводимых волн может быть любым в пределах имеющегося набора волн, так как транзитный оптический сигнал предварительно полностью демультиплексируется.

Кольцевая топология (рис. 11.16) обеспечивает живучесть сети DWDM за счет резервных путей. Методы защиты трафика, применяемые в DWDM, аналогичны методам SDH (хотя в DWDM они пока не стандартизованы). Для того чтобы какое-либо соединение было защищено, между его конечными точками устанавливаются два пути — основной и резервный. Мультиплексор конечной точки сравнивает два сигнала и выбирает сигнал лучшего качества (или сигнал, заданный по умолчанию).

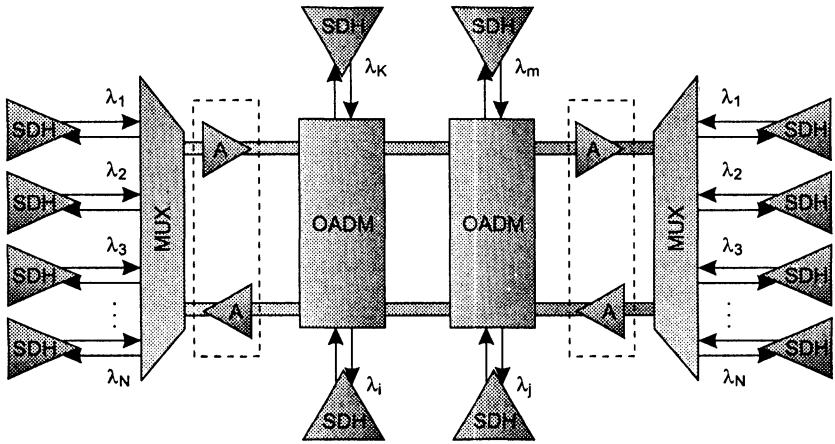


Рис. 11.15. Цепь DWDM с вводом-выводом в промежуточных узлах

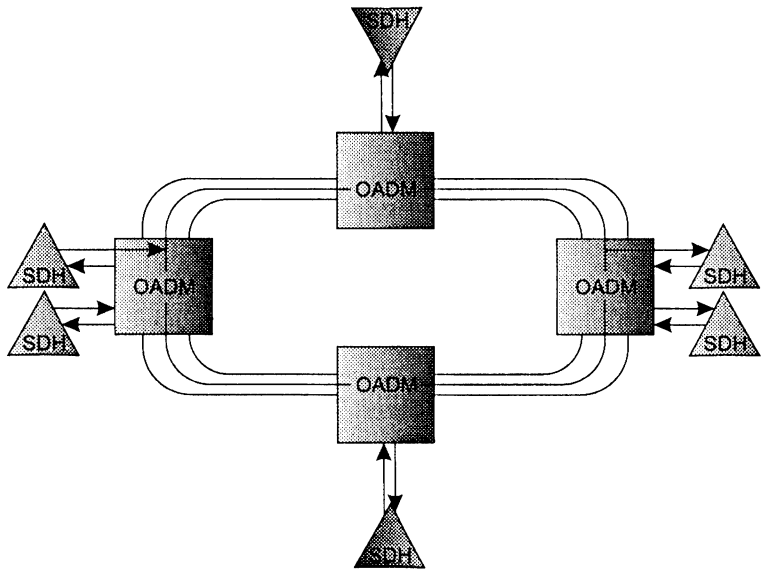


Рис. 11.16. Кольцо мультиплексоров DWDM

По мере развития сетей DWDM в них все чаще будет применяться **ячеистая топология** (рис. 11.17), которая обеспечивает бóльшую гибкость, производительность и отказоустойчивость, чем остальные топологии. Однако для реализации ячеистой топологии необходимо наличие **оптических кросс-коннекторов** (Optical Cross-Connect, OXC), которые не только добавляют волны в общий транзитный сигнал и выводят их оттуда, как это делают мультиплексоры ввода-вывода, но и поддерживают произвольную коммутацию между оптическими сигналами, передаваемыми волнами разной длины.

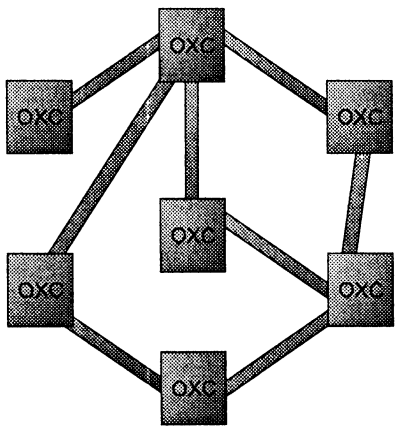


Рис. 11.17. Ячеистая топология сети DWDM

Оптические мультиплексоры ввода-вывода

Оптический мультиплексор выполняет операции смешения нескольких длин волн в общий сигнал, а также выделения волн различной длины из общего сигнала.

Для выделения волн в мультиплексоре могут использоваться разнообразные оптические механизмы. В оптических мультиплексорах, поддерживающих сравнительно небольшое количество длин волн в волокне, обычно 16 или 32, применяются **тонкопленочные фильтры**. Они состоят из пластин с многослойным покрытием, в качестве которых на практике используется торец оптического волокна, скошенный под углом 30–45°, с нанесенными на него слоями покрытия. Для систем с большим числом волн требуются другие принципы фильтрации и мультиплексирования.

В мультиплексорах DWDM применяются интегрально выполненные **дифракционные фазовые решетки**, или **дифракционные структуры** (Arrayed Waveguide Grating, AWG). Функции пластин в них выполняют оптические волноводы или волокна. Приходящий мультиплексный сигнал попадает на входной порт (рис. 11.18, *а*). Затем этот сигнал проходит через волновод-пластину и распределяется по множеству волноводов, представляющих дифракционную структуру AWG. По-прежнему сигнал в каждом из волноводов остается мультиплексным, а каждый канал (λ_1, λ_2, ..., λ_N) остается представленным во всех волноводах.

Далее происходит отражение сигналов от зеркальной поверхности, и в итоге световые потоки вновь собираются в волноводе-пластине, где происходит их фокусировка и интерференция — образуются пространственно разнесенные интерференционные максимумы интенсивности, соответствующие разным каналам. Геометрия волновода-пластины, в частности расположение выходных полюсов, и значения длины волноводов структуры AWG рассчитываются таким образом, чтобы интерференционные максимумы совпадали с выходными полюсами. Мультиплексирование происходит обратным путем.

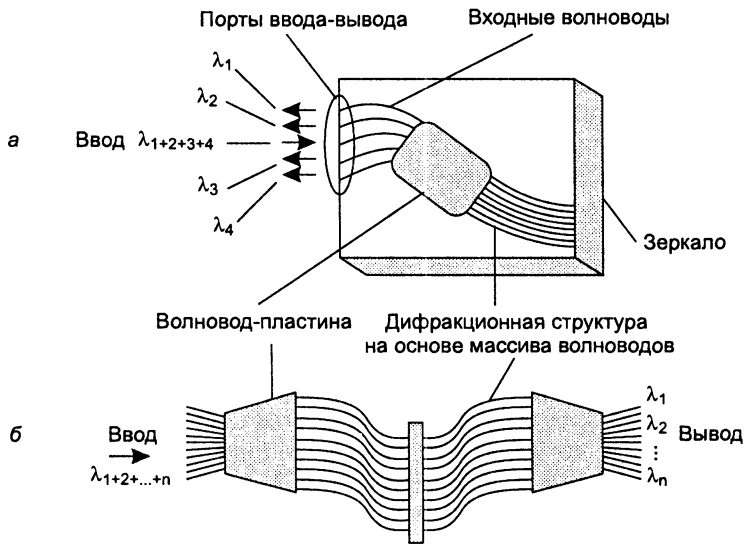


Рис. 11.18. Полное демультиплексирование сигнала с помощью дифракционной фазовой решетки

Другой способ построения мультиплексора базируется не на одной, а на паре волноводов-пластин (рис. 11.18, *б*). Принцип действия такого устройства аналогичен предыдущему случаю за исключением того, что здесь для фокусировки и интерференции используется дополнительная пластина.

Интегральные решетки AWG (называемые также **фазарами**) стали одними из ключевых элементов мультиплексоров DWDM. Они обычно применяются для полного демультиплексирования светового сигнала, так как хорошо масштабируются и потенциально могут успешно работать в системах с сотнями спектральных каналов.

Оптические кросс-коннекторы

В сетях с ячеистой топологией необходимо обеспечить гибкие возможности для изменения маршрута следования волновых соединений между абонентами сети. Такие возможности предоставляют оптические кросс-коннекторы, позволяющие направить любую из волн входного сигнала каждого порта в любой из выходных

портов (конечно, при условии, что никакой другой сигнал этого порта не использует эту волну, иначе необходимо выполнить трансляцию длины волны).

Существуют оптические кросс-коннекторы двух типов:

- **оптоэлектронные кросс-коннекторы** с промежуточным преобразованием в электрическую форму;
- **полностью оптические кросс-коннекторы**, или **фотонные коммутаторы**.

Исторически первыми появились оптоэлектронные кросс-коннекторы, за которыми и закрепилось название оптических кросс-коннекторов. Поэтому производители полностью оптических устройств этого типа стараются использовать для них отличающиеся названия — фотонные коммутаторы, маршрутизаторы волн, или лямбда-маршрутизаторы. У оптоэлектронных кросс-коннекторов имеется принципиальное ограничение — они хорошо справляются со своими обязанностями при работе на скоростях до 2,5 Гбит/с, но, начиная со скорости 10 Гбит/с и выше, габариты таких устройств и потребление энергии превышают допустимые пределы. Фотонные коммутаторы свободны от такого ограничения.

В фотонных коммутаторах используются различные оптические механизмы, в том числе дифракционные фазовые решетки и **микроэлектронные механические системы** (Micro-Electro Mechanical Systems, MEMS).

Система MEMS представляет собой набор подвижных зеркал очень маленького (с диаметром менее миллиметра) размера (рис. 11.19). Коммутатор на основе MEMS применяется после демультиплексора, когда исходный сигнал уже разделен на составляющие волны. За счет поворота микрозеркала на заданный угол исходный луч определенной волны направляется в соответствующее выходное волокно. Затем все лучи мультиплексируются в общий выходной сигнал.

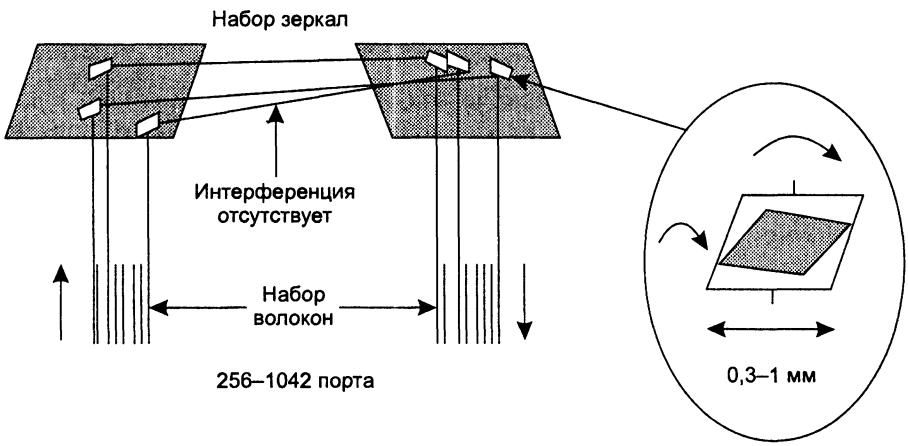


Рис. 11.19. Микроэлектронная механическая система кросс-коммутации

По сравнению с оптоэлектронными кросс-коннекторами фотонные коммутаторы занимают объем в 30 раз меньше и потребляют примерно в 100 раз меньше энергии. Однако этот тип устройств обладает низким быстродействием и чувст-

вительностью к вибрации. Тем не менее системы MEMS находят широкое применение в новых моделях фотонных коммутаторов. Сегодня подобные устройства могут обеспечивать коммутацию 256 × 256 спектральных каналов, и планируется выпуск устройств с возможностями коммутации 1024 × 1024 каналов и выше.

Пример первичной сети энергетической компании

В этом примере рассказывается о первичной сети крупной энергетической компании ABC-Power. Название компании вымышленное, однако в основе примера лежит подлинный проект.

Компания ABC-Power снабжает электрической энергией большой регион с территорией в несколько сот квадратных километров. В состав ABC-Pwer входит несколько крупных электростанций, генерирующих электроэнергию, а также распределительная сеть, с помощью которой энергия доставляется клиентам — крупным предприятиям и массовым потребителям.

Объекты ABC-Power — электростанции и распределительные станции — рассредоточены по 50 городам и деревням региона. Управляет энергетической сетью трехуровневая иерархическая система: центральный узел управления, районные узлы управления, электростанции и распределительные станции. Для управления выработкой и распределением энергии ABC-Power использует различные средства.

- *Системы телеметрии и автоматического управления* технологическими объектами (электростанциями и распределительными станциями). Эти системы состоят из датчиков, поставляющих оперативную информацию о состоянии энергетических объектов, а также исполнительных механизмов, с помощью которых объекты управляются, например, энергия перераспределяется из одной части сети в другую. Данные телеметрии передаются в режиме реального времени между объектами, а также на центральное табло, которое используют диспетчеры центрального и региональных узлов управления.
- *Специализированная диспетчерская связь.* Это система голосовой связи, подобная телефонной, но с большим количеством дополнительных функций, помогающих диспетчерам совместно решать возникающие проблемы.
- *Собственная телефонная сеть на PBX.* Эта сеть дополняет возможности диспетчерской системы связи и имеет соединение с национальной телефонной сетью.
- *Автоматизированная компьютерная система управления ресурсами предприятия собственной разработки.*

Каждая из перечисленных систем состоит из подсистем, расположенных во всех 50 точках присутствия ABC-Power. Очевидно, что для работы систем управления необходима качественная телекоммуникационная сеть, связывающая точки присутствия ABC-Power надежными и скоростными каналами.

Долгое время ABC-Power арендовала линии связи со скоростями от 64 Кбит/с до 2 Мбит/с у регионального оператора связи. Эти линии использовались для объединения PBX и маршрутизаторов/коммутаторов локальных сетей. Системы телеметрии и автоматического управления частично опирались на собственные медные линии связи ABC-Power, которые прокладывались параллельно линиям электропередач к объектам, не входившим в зону обслуживания регионального оператора связи.

Развитие бизнеса ABC-Power потребовало внедрения новых цифровых коммутаторов PBX, способных совмещать функции диспетчерской связи и стандартной телефонии; применения мощной интегрированной программной системы управления предприя-

тием SAP R/3 вместо разрозненных собственных подсистем аналогичного назначения; совершенствования систем телеметрии и автоматического управления.

Такая модернизация средств управления требовала, в свою очередь, модернизации линий связи — повышения их пропускной способности и надежности.

Анализ вариантов модернизации инфраструктуры линий связи показал, что аренда высокоскоростных каналов 34–155 Мбит/с экономически невыгодна. В результате ABC-Power решила пойти по пути создания собственной первичной сети, используя преимущество существующей сети линий электропередач. По такому пути пошли многие железнодорожные, энергетические и нефтегазовые компании. Прокладка оптического кабеля вдоль уже существующих путей следования железнодорожного полотна или газового трубопровода не требует больших затрат и обычно быстро окупается.

Первичная сеть ABC-Power была построена за два года. Волоконно-оптический кабель соединил мультиплексоры SDH во всех 50 точках присутствия ABC-Power (рис. 11.20).

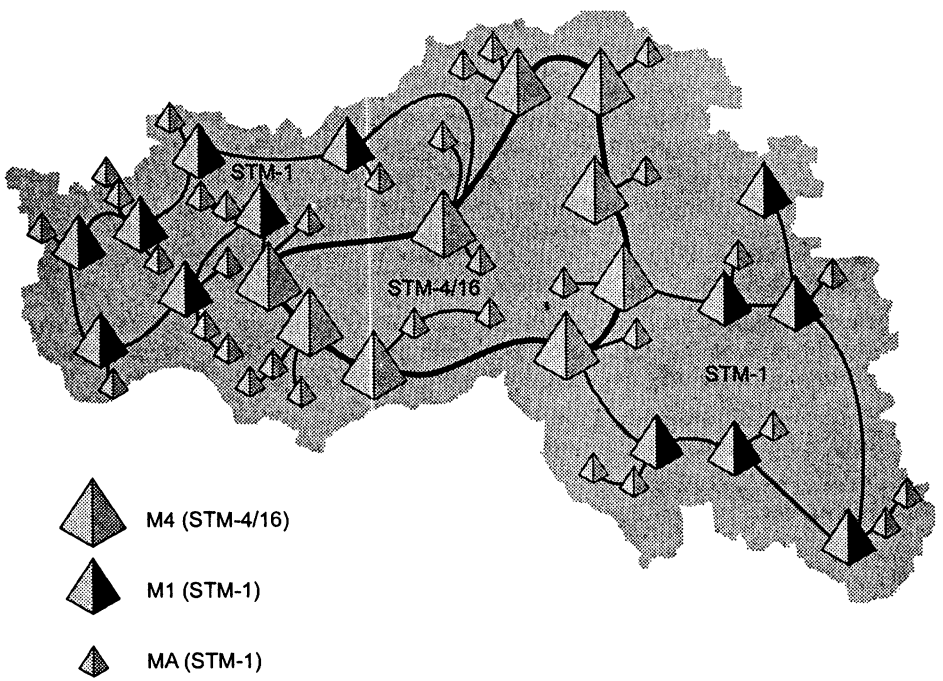


Рис. 11.20. Первичная сеть SDH компании ABC-Power

Сеть имеет ячеистую топологию, что позволяет применить методы защиты путей (соединений) технологии SDH и обеспечить высокую надежность. В сети используются мультиплексоры трех типов: M4, M1 и MA. Мультиплексоры M4 являются мультиплексорами ввода-ввода уровня STM-4, то есть их агрегатные порты работают на скорости STM-4 (622 Мбит/с). Эти мультиплексоры образуют магистральное кольцо, которое связывает крупные региональные узлы управления, а также центральный узел

управления. Мультиплексоры M4 допускают замену агрегатных портов STM-4 агрегатными портами STM-16 (2,5 Гбит/с), которые работают на одной из волн частотного плана DWDM. Это дает возможность дальнейшего наращивания скорости магистрали сети без замены оборудования, в том числе подключить в будущем сеть SDH к магистрали DWDM.

На мультиплексорах M1 и MA (агрегатные порты STM-1 155 Мбит/с) построена сеть доступа. Она охватывает все энергетические объекты и более мелкие региональные узлы управления. Сеть доступа сочетает ячеистую топологию с древовидной, обеспечивая защиту только наиболее ответственных путей. Мультиплексоры MA отличаются тем, что они оснащены большим количеством портов PDH для подключения оборудования наложенных сетей — телефонной, компьютерной и сети телеметрии/управления (рис. 11.21).

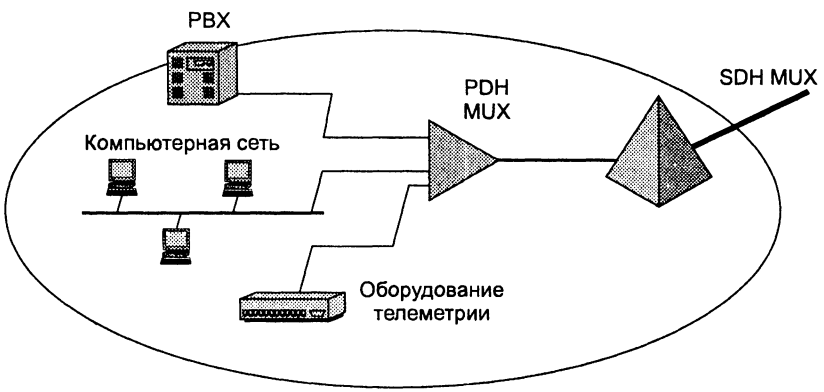


Рис. 11.21. Подключение оборудования к сети SDH

Создание собственной сети SDH позволило ABC-Power с большим запасом обеспечить потребности предприятия в высокоскоростных линиях связи. Этот запас компания собирается частично использовать для коммерческой деятельности в качестве ISP.

Выводы

Первичные сети предназначены для создания коммутируемой инфраструктуры, с помощью которой можно достаточно быстро создать постоянные каналы, организующие произвольную топологию.

В первичных сетях используют технику коммутации каналов различного типа: с частотным (FDM), временным (TDM) и волновым (WDM/DWDM) мультиплексированием.

В сетях FDM каждому абонентскому каналу выделяется полоса частот шириной 4 кГц. Существует иерархия каналов FDM, при этом 12 абонентских каналов образуют группу каналов первого уровня иерархии (базовую группу) с полосой 48 кГц, 5 каналов первого уровня объединяются в канал второго уровня иерархии (супергруппу) с полосой 240 кГц, а 10 каналов второго уровня составляют канал третьего уровня иерархии (главную группу) с полосой в 2,4 мГц.

Цифровые первичные сети PDH позволяют образовывать каналы с пропускной способностью от 64 Кбит/с до 140 Мбит/с, предоставляя своим абонентам скорости четырех уровней иерархии.

Недостатком сетей PDH является невозможность непосредственного выделения данных низкоскоростного канала из данных высокоскоростного канала, если каналы работают на несмежных уровнях иерархии скоростей.

Асинхронность ввода абонентских потоков в кадр SDH обеспечивается благодаря концепции виртуальных контейнеров и системы плавающих указателей, отмечающих начало пользовательских данных в виртуальном контейнере.

Мультиплексоры SDH могут работать в сетях с различной топологией: цепи, кольца, ячеистой топологией. Различают несколько специальных типов мультиплексоров, которые занимают специфическое место в сети: терминальные мультиплексоры, мультиплексоры ввода-вывода, кросс-коннекторы.

В сетях SDH поддерживается большое количество механизмов отказоустойчивости, которые защищают трафик данных на уровне отдельных блоков, портов или соединений: EPS, CP, MSP, SNC-P и MS-SPRing. Наиболее эффективная схема защиты выбирается в зависимости от логической топологии соединений в сети.

Технология WDM/DWDM реализует принципы частотного мультиплексирования для сигналов иной физической природы и на новом уровне иерархии скоростей. Каждый канал WDM/DWDM представляет собой определенный диапазон световых волн, позволяющих переносить данные в аналоговой и цифровой форме, при этом полоса пропускания канала в 25–50–100 ГГц обеспечивает скорости в несколько гигабит в секунду (при передаче дискретных данных).

В ранних системах WDM использовалось небольшое количество спектральных каналов, от 2 до 16. В системах DWDM задействовано уже от 32 до 160 каналов на одном оптическом волокне, что обеспечивает скорости передачи данных для одного волокна до нескольких терабит в секунду.

Современные оптические усилители позволяют удлинить оптический участок линии связи (без преобразования сигнала в электрическую форму) до 700–1000 км.

Для выделения нескольких каналов из общего светового сигнала разработаны сравнительно недорогие устройства, которые обычно объединяются с оптическими усилителями для организации мультиплексоров ввода-вывода в сетях дальней связи.

Для взаимодействия с традиционными оптическими сетями (SDH, Gigabit Ethernet, 10G Ethernet) в сетях DWDM применяются транспондеры и трансляторы длин волн, которые преобразуют длину волны входного сигнала в длину одной из волн стандартного частотного плана DWDM.

В полностью оптических сетях все операции мультиплексирования и коммутации каналов выполняются над световыми сигналами без их промежуточного преобразования в электрическую форму. Это упрощает и удешевляет сеть.

Вопросы и задания

1. Какие недостатки первичных сетей FDM привели к созданию цифровых первичных сетей?
2. Название T-1 обозначает:
   - аппаратуру мультиплексирования, разработанную компанией AT&T;
   - уровень скорости 1,544 Мбит/с;
   - международный стандарт линии связи;
   - способ мультиплексирования цифровых потоков 64 Кбит/с.

3. Какие функции выполняет младший бит каждого байта в канале T-1 при передаче голоса?
4. Можно ли в сети PDH выделить канал DS-0 непосредственно из канала DS-3?
5. Какие методы применяются на практике для решения предыдущей проблемы?
6. Какие механизмы в канале E-1 заменяют «кражу бита» канала T-1?
7. Почему первичные сети обеспечивают высокое качество обслуживания всех видов трафика?
8. Какое свойство технологии PDH отражает слово «плезиохронная»?
9. Каким образом компенсируется отсутствие синхронности трибутарных потоков в технологии SDH?
10. Какое максимальное количество каналов E-1 может мультиплексировать кадр STM-1?
11. Сколько каналов T-1 может мультиплексировать кадр STM-1, если в нем уже мультиплексировано 15 каналов E-1?
12. Какие уровни стека протоколов SDH отвечают за реконфигурирование сети в случае отказов оборудования?
13. Какова максимальная скорость канала передачи данных между регенераторами SDH?
14. По какой причине в кадре STM-1 используется три указателя?
15. С какой целью в технологиях PDH и SDH используется чередование байтов?
16. В чем отличие схем защиты 1+1 и 1:1? Варианты ответов:
    - в схеме 1+1 два потока мультиплексируются в один, а в схеме 1:1 нет;
    - схема 1+1 говорит о том, что резервный элемент выполняет те же функции, что и основной, а в схеме 1:1 резервный элемент простаивает до момента выхода из строя основного;
    - схема 1+1 используется для защиты портов, а схема 1:1 — для защиты путей трафика.
17. При каких условиях защита MS-SPRing более эффективна, чем SNC-P?
18. Что общего между первичными сетями FDM и DWDM?
19. К какому типу сетей относятся сети DWDM — аналоговым или цифровым?
20. С какой целью в сетях DWDM используются регенераторы, преобразующие оптический сигнал в электрический?
21. Назовите причины ухудшения качества оптического сигнала при передаче через большое количество пассивных участков DWDM?
22. Какие принципы коммутации световых сигналов используют оптические кросс-коннекторы?
23. С какой частотой будет выполняться операция отрицательного выравнивания указателя контейнера VC-4 в кадре STM-1, если относительная разница

между тактовыми частотами передающего и принимающего мультиплексоров SDH равна 10^{-5}?

24. Сеть SDH состоит из четырех мультиплексоров уровня STM-4: A, B, C и D. На рис. 11.22 показано распределение трафика между мультиплексорами, причем все потоки имеют скорость STM-1. Мультиплексоры соединены в кольцо STM-4. Какую схему защиты нужно выбрать, чтобы все соединения были защищены?

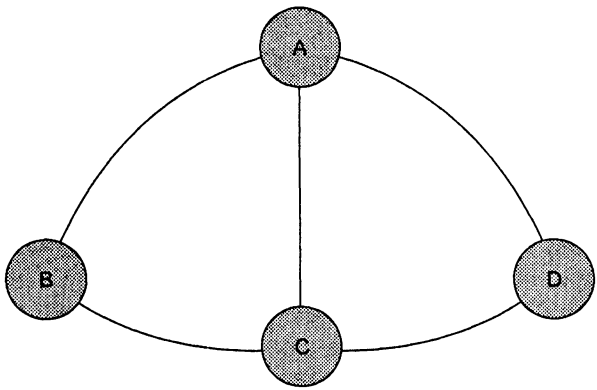


Рис. 11.22. Распределение трафика

Часть III

Локальные сети

Локальные сети являются неотъемлемой частью любой современной компьютерной сети. Если мы рассмотрим структуру глобальной сети, например Интернета или крупной корпоративной сети, то обнаружим, что практически все информационные ресурсы этой сети сосредоточены в локальных сетях, а глобальная сеть является транспортом, который соединяет многочисленные локальные сети.

Одним из основных назначений локальной сети является объединение компьютеров в пределах одного здания или нескольких близко стоящих зданий для предоставления пользователям сети доступа к информационным услугам локальных серверов. Локальные сети также являются удобным средством группирования компьютеров для объединения их в глобальную сеть, так как глобальной сети проще маршрутизировать данные между сетями, а не отдельными компьютерами. Примером могут служить беспроводные локальные сети, обслуживающие аэропорты или вокзалы. Они используются, как правило, не для обмена информацией между временными пользователями такой сети, а для доступа этих пользователей в Интернет — и доступ организуется не для каждого отдельного пользователя, а для локальной сети в целом. Локальные сети применяются также и в телекоммуникационных сетях другого типа, например в телефонных и первичных сетях. Так, системы управления телефонными коммутаторами или первичными сетями обычно строятся на основе локальной сети, которая объединяет компьютеры ее операторов и обеспечивает им доступ к устройствам управления, встроенным в оборудование телекоммуникационной сети.

Технологии локальных сетей прошли большой путь. Практически во всех технологиях 80-х годов использовалась *разделяемая среда* как удобное и экономичное средство объединения компьютеров на физическом уровне. В главе 2 уже обсуждались принципиальные подходы к разделению среды, в этой части книги мы возвращаемся к этому вопросу, чтобы рассмотреть его более подробно, на уровне конкретных алгоритмов и стандартов.

С середины 90-х в локальных сетях стали также применяться *коммутируемые* версии технологий. Отказ от разделяемой среды позволил повысить производительность и масштабируемость локальных сетей. В коммутируемых локальных сетях применяются те же протоколы, что и в локальных сетях на разделяемой среде, но в дуплексном режиме. Преимуществом коммутируемых локальных сетей является также возможность применения разнообразных методов обеспечения QoS, что важно, когда в локальной сети передается трафик реального времени, например, трафик IP-телефонов. Несмотря на популярность коммутируемых локальных сетей, мы по-прежнему часто встречаемся с использованием разделяемых сред в традиционных и новых технологиях. Они эффективны в небольших сегментах проводных локальных сетей, а также в беспроводных локальных сетях, где среда является разделяемой по своей природе.

В локальных сетях меняются не только принципы использования среды. Быстро растет верхний предел информационной скорости LAN-протоколов. С принятием в 2002 году стандарта 10G Ethernet технологии локальных сетей стали поддерживать иерархию скоростей, не уступающую иерархии скоростей первичных сетей — от 10 Мбит/с до 10 Гбит/с. Это дает возможность строить на этих технологиях не только локальные сети, но и сети мегаполисов.

Развитие локальных сетей идет и в направлении «миниатюризации» — появился новый тип сетей — *персональные сети* (Personal Area Networks, PAN), которые объединяют электронные устройства одного пользователя в радиусе нескольких десятков метров.

Современные локальные сети существуют в условиях одной победившей технологии — Ethernet, точнее — семейства технологий Ethernet. Естественно, что этой технологии в книге уделяется значительно большее внимание, чем остальным. В главе 12 рассматривается классическая технология Ethernet 10 Мбит/с на разделяемой среде, а в главе 13 — высокоскоростные версии Ethernet на разделяемой среде — Fast Ethernet и Gigabit Ethernet. В главе 14 описаны другие технологии LAN, также использующие разделяемую среду, — Token Ring, FDDI, а также две беспроводные технологии — IEEE 802.11 LAN и Bluetooth PAN.

Последние две главы этой части (главы 15 и 16) посвящены коммутируемым локальным сетям. В первой из них рассматриваются основные принципы работы таких сетей: алгоритм работы коммутатора локальной сети, дуплексные версии LAN-протоколов, особенности реализации LAN-коммутаторов. В главе 16 изучаются расширенные возможности сетей этого типа: резервные связи на основе алгоритма связующего дерева, агрегирование каналов, а также техника виртуальных локальных сетей.

- Глава 12. Технология Ethernet
- Глава 13. Высокоскоростной стандарт Ethernet
- Глава 14. Локальные сети на основе разделяемой среды
- Глава 15. Коммутируемые локальные сети
- Глава 16. Интеллектуальные функции коммутаторов

ГЛАВА 12 Технология Ethernet

Ethernet — это самый распространенный сегодня стандарт локальных сетей. Общее количество сетей, работающих по протоколу Ethernet в настоящее время, оценивается в несколько миллионов.

Когда говорят Ethernet, то под этим обычно понимают любой из вариантов этой технологии, в которую входят сегодня также Fast Ethernet, Gigabit Ethernet и 10G Ethernet.

В более узком смысле Ethernet — это сетевой стандарт передачи данных со скоростью 10 Мбит/с, который появился в конце 70-х годов как стандарт трех компаний — Digital, Intel и Xerox. В начале 80-х Ethernet был стандартизован рабочей группой IEEE 802.3, и с тех пор он является международным стандартом. Технология Ethernet была первой технологией, которая предложила использовать разделяемую среду для доступа к сети.

Локальные сети, являясь пакетными сетями, используют принцип временно́го мультиплексирования, то есть разделяют передающую среду во времени. Алгоритм *управления доступом к среде* является одной из важнейших характеристик любой технологии LAN, в значительно большей степени определяющей ее облик, чем метод кодирования сигналов или формат кадра. В технологии Ethernet в качестве алгоритма разделения среды применяется *метод случайного доступа*. И хотя его трудно назвать совершенным — при росте нагрузки полезная пропускная способность сети резко падает, — он благодаря своей простоте послужил основной причиной успеха технологии Ethernet.

Популярность стандарта Ethernet 10 Мбит/с послужила мощным стимулом его развития. В 1995 году был принят стандарт Fast Ethernet, в 1998 — Gigabit Ethernet, а в 2002 году — 10G Ethernet. Каждый из новых стандартов превышал скорость своего предшественника в 10 раз, образуя впечатляющую иерархию скоростей 10 Мбит/с — 100 Мбит/с — 1000 Мбит/с — 10 Гбит/с.

В этой главе мы подробно рассмотрим классическую технологию Ethernet 10 Мбит/с, большинство механизмов которой используется и на более высоких скоростях.

Общая характеристика протоколов локальных сетей

Список ключевых слов: общая среда передачи данных, стандартные топологии физических связей, метод случайного доступа, коллизия, слот, детерминированный доступ, передача токена, алгоритмы опроса, транспортировка кадров, дейтаграммный полудуплексный режим передачи, интерфейсные функции LLC, точка входа службы приемника, точка входа службы источника, доставка кадров с заданной степенью надежности, услуги LLC1, LLC2 и LLC3, стандарты межсетевого взаимодействия.

Технология Ethernet принадлежит к семейству технологий локальных сетей, в которое входят также такие технологии, как Token Ring, FDDI, IEEE 802.11 и 100VG-AnyLAN[1]. Несмотря на определенную специфику, все эти технологии имеют единое назначение — создание локальных сетей. Поэтому полезно начать изучение Ethernet с рассмотрения общих принципов, использованных при разработке технологий LAN.

Стандартная топология и разделяемая среда

Основная цель, которую ставили перед собой разработчики первых локальных сетей во второй половине 70-х годов, заключалась в нахождении простого и дешевого решения для объединения в вычислительную сеть нескольких десятков компьютеров, находящихся в пределах одного здания. Решение должно было быть недорогим, поскольку в сеть объединялись недорогие компьютеры — появившиеся и быстро распространявшиеся тогда мини-компьютеры стоимостью в 10 000–20 000 долларов. Количество их в одной организации было небольшим, поэтому предел в несколько десятков компьютеров представлялся вполне достаточным для практически любой локальной сети. Задача связи локальных сетей в глобальные не была первоочередной, поэтому практически все технологии локальных сетей ее игнорировали.

Для упрощения и, соответственно, удешевления аппаратных и программных решений разработчики первых локальных сетей остановились на совместном использовании **общей среды передачи данных**.

Этот метод связи компьютеров впервые был опробован при создании радиосети ALOHA Гавайского университета в начале 70-х под руководством Нормана Абрамсона (Norman Abramson). Радиоканал определенного диапазона частот естественным образом является общей средой для всех передатчиков, использующих частоты этого диапазона для кодирования данных. Сеть ALOHA работала по методу случайного доступа, когда каждый узел мог начать передачу пакета в любой момент времени. Если после этого он не дожидался подтверждения приема в те-

[1] Технология 100VG-AnyLAN сегодня практически не применяется, однако теоретический интерес представляет использованная в ней оригинальная концепция разделения среды.

чение определенного тайм-аута, он посылал этот пакет снова. Общим был радиоканал с несущей частотой 400 МГц и полосой 40 кГц, что обеспечивало передачу данных со скоростью 9600 бит/с.

Немного позже Роберт Меткалф (R. Metcalfe) повторил идею разделяемой среды уже для проводного варианта технологии LAN. Непрерывный сегмент коаксиального кабеля стал аналогом общей радиосреды. Все компьютеры присоединялись к этому сегменту кабеля по схеме монтажного ИЛИ (рис. 12.1), поэтому при передаче сигналов одним из передатчиков все приемники получали один и тот же сигнал, как и при использовании радиоволн.

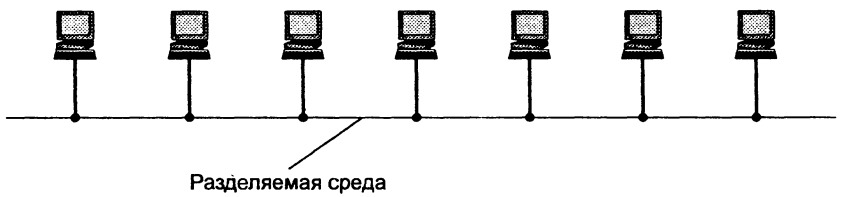


Рис. 12.1. Разделяемая среда на коаксиальном кабеле

В технологиях Token Ring и FDDI тот факт, что компьютеры используют разделяемую среду, не так очевиден, как в случае Ethernet. Физическая топология этих сетей — кольцо, каждый узел соединяется кабелем с двумя соседними узлами (рис. 12.2). Однако эти отрезки кабеля также являются разделяемыми, так как в каждый момент времени только один компьютер может использовать кольцо для передачи своих пакетов.

Простые стандартные топологии физических связей (звезда у коаксиального кабеля Ethernet и кольцо у Token Ring и FDDI) обеспечивают простоту разделения кабельной среды.

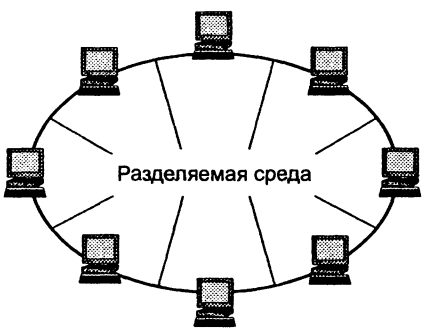


Рис. 12.2. Разделяемая среда в кольцевых топологиях

Использование разделяемых сред позволяет *упростить* логику работы узлов сети. Действительно, поскольку в каждый момент времени выполняется только одна передача, отпадает необходимость в буферизации кадров в транзитных узлах. Транзитных узлов также нет (мы уже рассматривали этот необычный вари-

ант сети с коммутацией пакетов в главе 2). Соответственно, отпадает необходимость в сложных процедурах управления потоком и борьбы с перегрузками.

Основной недостаток разделяемой среды — плохая масштабируемость. Этот недостаток является принципиальным, так как независимо от метода доступа к среде ее пропускная способность делится между всеми узлами сети. Здесь применимо положение теории очередей, которое мы изучали в главе 7: как только коэффициент использования общей среды превышает определенный порог, очереди к среде начинают расти нелинейно, и сеть становится практически неработоспособной. Значение порога зависит от метода доступа. Так, в сетях ALOHA это значение является крайне низким — всего около 18 %, в сетях Ethernet — около 30 %, а в сетях Token Ring и FDDI оно выросло до 60–70 %.

Стек протоколов локальных сетей

Технологии локальных сетей реализуют, как правило, функции только двух нижних уровней модели OSI — *физического* и *канального* (рис. 12.3). Функциональности этих уровней достаточно для доставки кадров в пределах стандартных топологий, которые поддерживают LAN — звезда (общая шина), кольцо и дерево.

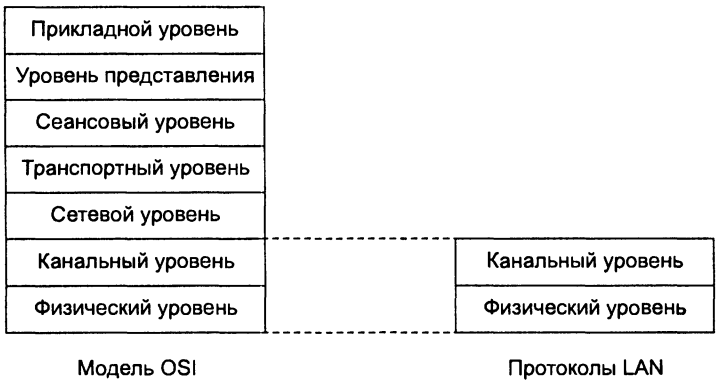


Рис. 12.3. Соответствие протоколов LAN уровням модели OSI

Однако из этого не следует, что компьютеры, связанные в локальную сеть, не поддерживают протоколы уровней, расположенных выше канального. Эти протоколы также устанавливаются и работают на узлах локальной сети, но выполняемые ими функции *не относятся* к технологии LAN. Сетевой и транспортный протоколы нужны узлу локальной сети для того, чтобы взаимодействовать с компьютерами, подключенными к другим локальным сетям, путь к которым проходит, возможно, через глобальные сети. Если бы нужно было обеспечить взаимодействие компьютеров только в пределах одной локальной сети, то прикладные протоколы могли бы работать непосредственно над канальным уровнем. Но так как такое ограниченное взаимодействие не устраивает пользователей, то каждый компьютер локальной сети поддерживает полный стек протоколов, так что над канальным уровнем работает один из сетевых протоколов, например IP или IPX.

Кроме того, установка на конечных узлах LAN полных стеков протоколов, а не только физического и канального, необходима для обеспечения совместимости приложений — приложения должны корректно исполняться в любой сетевой среде, во всяком случае, не зависеть от того, является сеть односегментной локальной сетью или крупной локальной сетью, построенной на маршрутизаторах.

Канальный уровень локальных сетей делится на два подуровня, которые часто также называют уровнями:

- уровень управления логическим каналом (Logical Link Control, LLC);
- уровень управления доступом к среде (Media Access Control, MAC).

Функции уровня LLC обычно реализуются программно, соответствующим модулем операционной системы, а функции уровня MAC реализуются программно аппаратно: сетевым адаптером и его драйвером.

Уровень MAC

Основными функциями уровня MAC являются:

- обеспечение доступа к разделяемой среде;
- передача кадров между конечными узлами, используя функции и устройства физического уровня.

Метод случайного доступа является одним из основных методов захвата разделяемой среды. Он основан на том, что узел, у которого есть кадр для передачи, пытается его отправить без какой бы то ни было предварительной процедуры согласования времени использования разделяемой среды с другими узлами сети.

Метод случайного доступа является *децентрализованным*, он не требует наличия в сети специального узла, который играл бы роль арбитра, регулирующего доступ к среде. Результатом этого является высокая вероятность **коллизий**, то есть случаев одновременной передачи кадра несколькими станциями[1]. Во время коллизии происходит наложение сигналов нескольких передатчиков, из-за чего информация всех передаваемых на периоде коллизии кадров искажается. Поскольку в локальных сетях применяются достаточно простые методы кодирования, то они не позволяют выделить нужный сигнал из суммарного, как это, например, может делать технология CDMA.

Существует большое количество алгоритмов случайного доступа, которые снижают вероятность коллизий и тем самым повышают производительность сети. Например, существует класс алгоритмов, которые разрешают начать передачу кадров только в начале очередного временного интервала, обычно называемого **слотом**. Впервые такое улучшение было предложено для сети ALOHA. В этой сети метод случайного доступа разрешает узлу передавать кадр в любой момент времени без всяких предварительных условий. *Синхронизация передачи кадров* с началом очередного слота позволила снизить вероятность коллизий в модифицированном алгоритме ALOHA по сравнению с первоначальным вариантом

[1] Термины «станция» и «узел» используются здесь и далее как синонимы.

ALOHA в два раза, обеспечив нормальную работу сети с коэффициентом использования среды до 36 %.

Еще одним способом улучшения случайного доступа является введение *процедуры прослушивания среды* перед передачей. Узел не имеет права передавать кадр, если он обнаруживает, что среда уже занята передачей другого кадра. Это снижает вероятность коллизий (хотя, как увидим позже, и не исключает их).

Алгоритмы случайного доступа не гарантируют узлу, что он получит доступ к разделяемой среде в течение определенного времени. Какое бы большое время ожидания мы ни выбрали, всегда есть ненулевая вероятность, что реальное время ожидания превысит этот предел. Алгоритмы случайного доступа также не предоставляют никаких возможностей для дифференцированной поддержки характеристик QoS для разных типов трафика — все кадры получают одинаковый уровень доступа к среде.

Детерминированный доступ — это другой популярный подход к обеспечению доступа к разделяемой среде. Он получил свое название благодаря тому, что максимальное время ожидания доступа к среде всегда известно.

Алгоритмы детерминированного доступа используют два механизма – передачу токена и опрос.

Передача токена обычно реализуется *децентрализовано*. Каждый компьютер, получивший токен, имеет право на использование разделяемой среды в течение фиксированного промежутка времени — *времени удержания токена*. В это время компьютер передает свои кадры. После истечения этого промежутка компьютер обязан передать токен другому компьютеру. Таким образом, если мы знаем количество компьютеров в сети, то максимальное время ожидания доступа равно произведению времени удержания токена на это число. Время ожидания может быть и меньше, поскольку, если компьютер, получивший токен, не имеет кадров для передачи, то он передает его следующему компьютеру, не дожидаясь истечения времени удержания. Последовательность передачи токена от компьютера к компьютеру может определяться разными способами. В сетях Token Ring и FDDI она определяется топологией связей. Компьютер в кольце получает токен от предыдущего соседа, а передает токен следующему. Алгоритм передачи токена можно реализовать не только в кольце. Например, в прекратившей свое существование технологии *ArcNet* использовался общий коаксиальный кабель для физического подключения компьютеров, а в качестве метода доступа — передача токена. При этом токен передавался между компьютерами в заранее определенной последовательности, не зависящей от мест подключения компьютеров к кабелю.

Алгоритмы опроса чаще всего основаны на *централизованной* схеме. В сети существует выделенный узел, который играет роль *арбитра* в споре узлов за разделяемую среду. Арбитр периодически опрашивает остальные узлы сети, есть ли у них кадры для передачи. Собрав заявки на передачу, арбитр решает, какому узлу он предоставит право использования разделяемой среды. Затем он сообщает свое

решение выбранному узлу, и тот передает свой кадр, захватывая разделяемую среду. После завершения передачи кадра фаза опроса повторяется.

Алгоритм опроса может быть также *децентрализованным*. В этом случае все узлы должны предварительно сообщить друг другу с помощью разделяемой среды свои потребности в передаче кадров. Затем на основе этой информации и в соответствии с определенным критерием каждый из узлов, желающих передать кадр, независимо от других узлов определяет свою очередь в последовательности передач.

Алгоритмы детерминированного доступа отличаются от алгоритмов случайного доступа тем, что они более эффективно работают при большой загрузке сети, когда коэффициент использования приближается к единице. В то же время при небольшой загрузке сети более эффективными являются алгоритмы случайного доступа, так как они позволяют передать кадр немедленно, не тратя время на процедуры определения права доступа к среде.

Достоинство детерминированных методов доступа также заключается в том, что они могут приоритезировать трафик, а значит, поддерживать требования QoS.

Транспортировка кадров осуществляется уровнем MAC в несколько этапов, которые в общем случае не зависят от выбранного метода доступа.

1. *Формирование кадра.* На этом этапе осуществляется заполнение полей кадра на основании информации, получаемой от протокола верхнего уровня, такой как адреса источника и назначения, пользовательские данные, признак протокола верхнего уровня, отсылающего эти данные. После того как кадр сформирован, уровень MAC подсчитывает контрольную сумму кадра и помещает ее в соответствующее поле.
2. *Передача кадра через среду.* Когда кадр сформирован и доступ к разделяемой среде получен, уровень MAC передает кадр на физический уровень, который побитно передает все поля кадра в среду. Функции физического уровня выполняет передатчик сетевого адаптера, который преобразует байты кадра в последовательность битов и кодирует их соответствующими электрическими или оптическими сигналами. После прохождения сигналов по среде они поступают в приемники сетевых адаптеров, подключенных к разделяемой среде, которые выполняют обратное преобразование сигналов в байты кадра.
3. *Прием кадра.* Уровень MAC каждого узла сети, подключенного к разделяемой среде, проверяет адрес назначения поступившего кадра, и если он совпадает с его собственным адресом, то продолжает его обработку, в противном случае кадр отбрасывается. Продолжение обработки заключается в проверке корректности контрольной суммы кадра. Кадр с корректной контрольной суммой передается уровнем MAC вверх по стеку, на чем функции уровня MAC заканчиваются. Если же контрольная сумма кадра говорит о том, что информация при передаче через среду была искажена, то кадр отбрасывается.

Из этого описания следует, что Ethernet реализует **дейтаграммный полудуплексный** режим передачи данных.

Уровень LLC

Уровень LLC выполняет две функции:

- организует интерфейс с прилегающим к нему сетевым уровнем;
- обеспечивает доставку кадров с заданной степенью надежности.

Интерфейсные функции LLC заключаются в передаче пользовательских и служебных данных между уровнем MAC и сетевым уровнем. При передаче данных *сверху вниз* уровень LLC принимает от протокола сетевого уровня пакет (например, IP- или IPX-пакет), в котором уже находятся пользовательские данные. Помимо пакета сверху также передается адрес узла назначения в формате той технологии LAN, которая будет использована для доставки кадра в пределах данной локальной сети. Напомним, что в терминах стека TCP/IP такой адрес называется аппаратным. Полученные от сетевого уровня пакет и аппаратный адрес уровень LLC передает далее вниз — уровню MAC. Кроме того, LLC при необходимости решает *задачу мультиплексирования*, передавая данные от нескольких протоколов сетевого уровня единственному протоколу уровня MAC.

При передаче данных *снизу вверх* LLC принимает от уровня MAC пакет сетевого уровня, пришедший из сети. Теперь ему нужно выполнить еще одну интерфейсную функцию — *демультиплексирование*, то есть решить, какому из сетевых протоколов передать полученные от MAC данные (рис. 12.4).

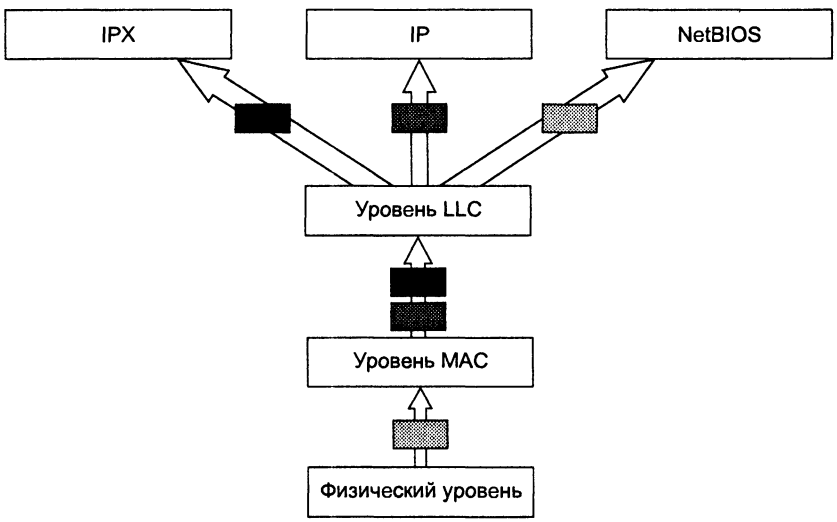


Рис. 12.4. Демультиплексирование кадров протоколом LLC

Задачи мультиплексирования и демультиплексирования свойственны не только LLC, но и любому протоколу, над которым может работать несколько протоколов. Для демультиплексирования данных LLC использует в своем заголовке специальные поля (рис. 12.5). Поле **DSAP** (Destination Service Access Point — точка входа службы приемника) используется для хранения кода протокола, которому адресовано содержимое поля данных. Соответственно, поле **SSAP** (Source Service

Access Point — точка входа службы источника) используется для указания кода протокола, от которого посылаются данные. Применение двух полей для целей демультиплексирования является нетипичным, обычно протоколы обходятся одним полем, например, протокол IP всегда посылает свои пакеты протоколу IP, а протокол IPX — протоколу IPX. Два поля полезны в тех случаях, когда вышележащий протокол поддерживает несколько режимов работы, так что протокол на узле-отправителе может использовать различные значения DSAP и SSAP для уведомления узла-получателя о переходе в новый режим работы. Этим свойством протокола LLC часто пользуется протокол NetBEUI.

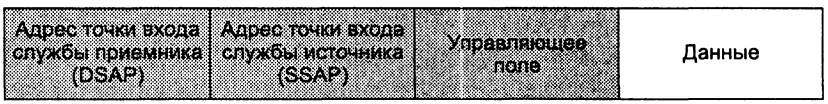


Рис. 12.5. Формат LLC-кадра

Обеспечение **доставки кадров с заданной степенью надежности** — вторая основная функция уровня LLC. Протокол LLC поддерживает несколько режимов работы, отличающихся наличием или отсутствием процедур восстановления кадров в случае их потери или искажения, то есть отличающихся надежностью доставки. Уровень LLC, непосредственно прилегающий к сетевому уровню, принимает от него запрос на выполнение транспортной операции канального уровня с тем или иным качеством.

ВНИМАНИЕ

Возможно, функции LLC по обеспечению надежной передачи данных в LAN напомнят читателю функции транспортного уровня моделей OSI и TCP/IP. Действительно, соотношение функций протоколов LLC и MAC во многом подобно соотношению функций протоколов UDP/TCP и IP. Как и протоколы транспортного уровня UDP/TCP, протокол LLC не занимается непосредственно доставкой кадров узлам сети. Передачу данных между узлами, подобно IP, выполняет после получения доступа к разделяемой среде уровень MAC. MAC, так же как и IP, обеспечивает доставку в дейтаграммном режиме, то есть без установления соединения и без восстановления потерянных или поврежденных кадров. В том случае, когда протоколы верхних уровней запрашивают у LLC надежный транспортный сервис, LLC устанавливает соединение с узлом назначения и организует повторную доставку кадров.

Уровень LLC предоставляет верхним уровням три типа транспортных услуг.

- **Услуга LLC1** — услуга *без установления соединения и без подтверждения получения данных*. LLC1 дает пользователю средства для передачи данных с минимумом издержек. В этом случае LLC поддерживает дейтаграммный режим работы, как и MAC, так что и технология LAN в целом работает в дейтаграммном режиме. Обычно эта процедура используется, когда такие функции, как восстановление данных после ошибок и упорядочивание данных, выполняются протоколами вышележащих уровней, поэтому нет нужды дублировать их на уровне LLC.
- **Услуга LLC2** — дает пользователю возможность установить *логическое соединение* перед началом передачи любого блока данных и, если это требуется,

выполнить *процедуры восстановления* после ошибок и упорядочивание потока блоков в рамках установленного соединения. Для надежной доставки данных протокол LLC2 использует алгоритм скользящего окна.

- **Услуга LLC3** — услуга *без установления соединения, но с подтверждением получения данных*. В некоторых случаях (например, при использовании сетей в системах реального времени, управляющих промышленными объектами), с одной стороны, временны́е издержки установления логического соединения перед отправкой данных неприемлемы, а, с другой стороны, подтверждение о корректности приема переданных данных необходимо. Для такого рода ситуаций и предусмотрена дополнительная услуга LLC3, которая является компромиссом между LLC1 и LLC2, так как она не предусматривает установление логического соединения, но обеспечивает подтверждение получения данных.

Какой из трех режимов работы уровня LLC будет использован, зависит от требований протокола верхнего уровня. Информация о требуемой от LLC транспортной услуге передается через межуровневый интерфейс уровню LLC вместе с аппаратным адресом и пакетом с пользовательскими данными. Например, когда поверх LLC работает протокол IP, он всегда запрашивает режим LLC1, поскольку в стеке TCP/IP задачу обеспечения надежной доставки решает протокол TCP.

Из протоколов, применяющихся на практике, только стек Microsoft/IBM, основанный на протоколе NetBIOS/NetBEUI, использует режим LLC2. Это происходит тогда, когда сам протокол NetBIOS/NetBEUI должен работать в режиме с восстановлением потерянных и искаженных данных. В этом случае эта работа перепоручается уровню LLC2. Если же протокол NetBIOS/NetBEUI работает в дейтаграммном режиме, то он пользуется услугой LLC1.

Структура стандартов IEEE 802.x

В 1980 году в институте IEEE был организован комитет 802 по стандартизации технологий LAN, в результате работы которого было принято семейство стандартов IEEE 802.x, содержащих рекомендации по проектированию нижних уровней локальных сетей. Эти стандарты базируются на популярных фирменных стандартах, в частности Ethernet, ArcNet и Token Ring.

Результаты работы комитета 802 легли в основу комплекса международных стандартов ISO 8802-1...5. Комитет IEEE 802 и сегодня является основным международным органом, разрабатывающим стандарты технологий локальных сетей.

Помимо IEEE в работе по стандартизации протоколов LAN принимали и принимают участие и другие организации. Так, для сетей, работающих на оптоволокне, институтом ANSI был разработан стандарт FDDI, обеспечивающий скорость передачи данных 100 Мбит/с. Это был первый протокол LAN, который достиг такой скорости, в 10 раз превысив скорость технологии Ethernet.

Структуру стандартов IEEE 802 иллюстрирует рис. 12.6.

На рисунке над уровнем MAC, специфичным для разных технологий, показан общий для них уровень LLC. Стандарт LLC курирует рабочая группа 802.2. Даже технологии, стандартизованные не в рамках комитета 802 (например, стан-

дартизованный ANSI протокол FDDI), ориентируются на использование протокола LLC, определенного стандартом 802.2.

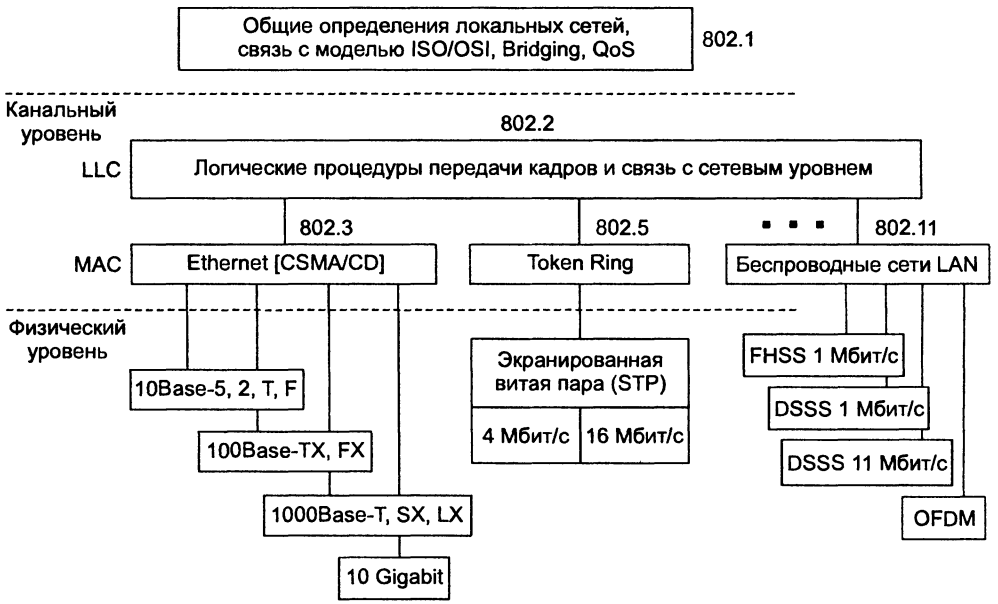


Рис. 12.6. Структура стандартов IEEE 802.x

Описание каждой технологии в стандарте разделено на две части: описание уровня MAC и описание физического уровня. Как видно из рисунка, практически у каждой технологии единственному протоколу уровня MAC соответствуют несколько вариантов протоколов физического уровня (на рисунке в целях экономии места приведены только технологии Ethernet и Token Ring, но все сказанное справедливо также и для других технологий, таких как ArcNet, FDDI, Fast-Ethernet, GigabitEthernet, 10G Ethernet).

Особняком стоят стандарты, разрабатываемые подкомитетом 802.1. Эти стандарты носят общий для всех технологий характер. В подкомитете 802.1 были даны общие определения локальных сетей и их свойств, показана связь трех уровней модели IEEE 802 с моделью OSI. Наиболее практически важными являются те стандарты подкомитета 802.1, которые описывают взаимодействие различных технологий, а также стандарты по построению более сложных сетей на основе базовых топологий. Эта группа стандартов носит общее название **стандартов межсетевого взаимодействия**. Сюда входят такие важные стандарты, как стандарт 802.1D, описывающий логику работы прозрачного моста/коммутатора, стандарт 802.1H, определяющий функционирование транслирующего моста и т. п. Набор стандартов, разработанных подкомитетом 802.1, продолжает расти. Например, сравнительно недавно он пополнился двумя важными стандартами: стандартом 802.1Q, определяющим способ построения виртуальных локальных сетей (Virtual LAN, VLAN) в сетях на основе коммутаторов, и стандартом 802.1p, описывающим способ приоритезации трафика на канальном уровне, то есть поддержку механизмов QoS.

Стандарты комитетов 802.3, 802.4, 802.5 и 802.12 описывают технологии локальных сетей, которые появились в результате улучшений фирменных технологий, легших в их основу.

Основу стандарта 802.3 составила технология экспериментальной сети Ethernet Network, которую фирма Xerox разработала и реализовала в 1975 году. В 1980 году фирмы DEC, Intel и Xerox (сокращенно — DIX) совместно разработали и опубликовали стандарт Ethernet версии II для сети, построенной на основе коаксиального кабеля. Эту последнюю версию фирменного стандарта Ethernet называют стандартом Ethernet DIX, или Ethernet II. На базе стандарта Ethernet DIX был разработан стандарт IEEE 802.3, который во многом совпадает со своим предшественником.

Стандарт 802.4 появился как обобщение технологии ArcNet компании Datapoint Corporation, а стандарт 802.5 в основном соответствует технологии Token Ring компании IBM.

Комитет 802.11 занимается разработкой локальных беспроводных сетей с методами доступа к среде, близкими к тем, которые используются в сетях Ethernet. Поэтому стандарты 802.11 также называют стандартами радиоEthernet (хотя само название Ethernet в тексте стандартов 802.11 не фигурирует).

Исходные фирменные технологии и их модифицированные варианты — стандарты 802.x в ряде случаев долгие годы сосуществовали параллельно. Например, технология ArcNet так до конца не была приведена в соответствие со стандартом 802.4 (теперь это делать поздно, так как где-то примерно с 1993 года производство оборудования ArcNet было свернуто). Исключение составляет технология Ethernet. Последний фирменный стандарт Ethernet DIX версии II был принят в 1980 году, и с тех пор никто больше не предпринимал попыток фирменного развития Ethernet. Все новшества в семействе технологий Ethernet появляются только в результате принятия открытых стандартов комитетом 802.3.

Более поздние стандарты изначально разрабатывались не одной компанией, а группой заинтересованных компаний, а потом передавались в соответствующий подкомитет IEEE 802 для утверждения. Так произошло с технологиями Fast Ethernet, 100VG-AnyLAN, Gigabit Ethernet. Сначала группа компаний образовывала объединение, а затем по мере развития работ к нему присоединялись другие компании, так что процесс принятия стандарта носил открытый характер.

Метод доступа CSMA/CD

Список ключевых слов: метод коллективного доступа с опознаванием несущей и обнаружением коллизий, MAC-адрес, индивидуальный MAC-адрес, групповой MAC-адрес, широковещательный MAC-адрес, централизованный способ назначения адресов, локальный способ назначения адресов, организационно уникальный идентификатор, коллективный доступ, несущая частота, преамбула, ограничитель начала кадра, межпакетный интервал, коллизия, обнаружение коллизии, jam-последовательность, интервал отсрочки, усеченный экспоненциальный двоичный алгоритм отсрочки, время оборота, максимальный диаметр сети.

Метод CSMA/CD (Carrier Sense Multiple Access with Collision Detection — коллективный доступ с опознаванием несущей и обнаружением коллизий) используется для доступа к среде передачи данных в сетях Ethernet.

MAC-адреса

На уровне MAC, который обеспечивает доступ к среде и передачу кадра, для идентификации сетевых интерфейсов узлов сети используются регламентированные стандартом IEEE 802.3 уникальные 6-байтовые адреса, называемые **MAC-адресами**. Обычно MAC-адрес записывают в виде шести пар шестнадцатеричных цифр, разделенных тире или двоеточиями, например 11-A0-17-3D-BC-01. Каждый сетевой адаптер имеет, по крайней мере, один MAC-адрес.

Помимо отдельных интерфейсов, MAC-адрес может определять группу интерфейсов или даже все интерфейсы сети. Первый (младший) бит старшего байта адреса назначения является признаком того, является адрес индивидуальным или групповым. Если он равен 0, то адрес является **индивидуальным**, то есть идентифицирует один сетевой интерфейс, а если 1, то **групповым**. Групповой адрес связан только с интерфейсами, сконфигурированными (например, вручную или автоматически по запросу вышележащего уровня) как члены группы, номер которой указан в групповом адресе. Если сетевой интерфейс включен в группу, то наряду с уникальным MAC-адресом с ним ассоциируется еще один адрес — групповой. В частном случае, если групповой адрес состоит из всех единиц, то есть имеет шестнадцатеричное представление 0xFFFFFFFFFFFF, он идентифицирует все узлы сети и называется **широковещательным**.

Второй бит старшего байта адреса определяет способ назначения адреса — **централизованный** или **локальный**. Если этот бит равен 0 (что бывает почти всегда в стандартной аппаратуре Ethernet), то адрес назначен централизованно по правилам IEEE 802.

ВНИМАНИЕ

В стандартах IEEE Ethernet младший бит байта изображается в самой левой позиции поля, а старший бит — в самой правой. Этот нестандартный способ отображения порядка следования битов в байте соответствует порядку передачи битов в линию связи передатчиком Ethernet (первым передается младший бит). В стандартах других организаций, например RFC IETF, ITU-T, ISO, используется традиционное представление байта, когда младший бит считается самым правым битом байта, а старший — самым левым. При этом порядок следования байтов остается традиционным. Поэтому при чтении стандартов, опубликованных этими организациями, а также чтении данных, отображаемых на экране операционной системой или анализатором протоколов, значения каждого байта кадра Ethernet нужно зеркально отобразить, чтобы получить представление о значении разрядов этого байта в соответствии с документами IEEE. Например, групповой адрес, имеющий в нотации IEEE вид 1000 0000 0000 0000 1010 0111 1111 0000 0000 0000 0000 0000 или в шестнадцатеричной записи 80-00-A7-F0-00-00, будет, скорее всего, отображен анализатором протоколов в традиционном виде как 01-00-5E-0F-00-00.

Комитет IEEE распределяет между производителями оборудования так называемые **организационно уникальные идентификаторы** (Organizationally Unique

Identifier, OUI). Каждый производитель помещает выделенный ему идентификатор в три старших байта адреса (например, идентификатор 0x0020AF определяет компанию 3COM, а 0x00000C — Cisco). За уникальность младших трех байтов адреса отвечает производитель оборудования. Двадцать четыре бита, отводимые производителю для адресации интерфейсов его продукции, позволяют выпустить примерно 16 миллионов интерфейсов под одним идентификатором организации. Уникальность централизованно распределяемых адресов распространяется на все основные технологии локальных сетей — Ethernet, Token Ring, FDDI и т. д. Локальные адреса назначаются администратором сети, в обязанности которого входит обеспечение их уникальности.

Доступ к среде и передача данных

Предполагая для простоты изложения, что каждый узел (станция) имеет только один сетевой интерфейс, рассмотрим, как на основе алгоритма CSMA/CD происходит передача данных в сети Ethernet.

Все компьютеры в сети с разделяемой средой имеют возможность немедленно (с учетом задержки распространения сигнала по физической среде) получить данные, которые любой из компьютеров начал передавать в общую среду. Говорят, что среда, к которой подключены все станции, работает в режиме **коллективного доступа** (Multiply Access, MA).

Чтобы получить возможность передавать кадр, интерфейс-отправитель должен убедиться, что разделяемая среда свободна. Это достигается прослушиванием основной гармоники сигнала, которая также называется **несущей частотой** (Carrier Sense, CS).

Признаком «незанятости» среды является отсутствие на ней несущей частоты, которая при манчестерском способе кодирования, принятом для всех вариантов Ethernet 10 Мбит/с, равна 5–10 МГц в зависимости от последовательности единиц и нулей, передаваемых в данный момент.

Если среда свободна, то узел имеет право начать передачу кадра. В примере, показанном на рис. 12.7, узел 1 обнаружил, что среда свободна, и начал передавать свой кадр. В классической сети Ethernet на коаксиальном кабеле сигналы передатчика узла 1 распространяются в обе стороны, так что их получают все узлы сети. Кадр данных всегда сопровождается **преамбулой**, которая состоит из 7 байт, каждый из которых имеет значение 10101010, и 8-го байта, равного 10101011. Последний байт носит название **ограничителя начала кадра**. Преамбула нужна для вхождения приемника в побитовую и побайтовую синхронизацию с передатчиком. Наличие двух единиц, идущих подряд, говорит приемнику о том, что преамбула закончилась и следующий бит является началом кадра.

Все станции, подключенные к кабелю, начинают записывать байты передаваемого кадра в свои внутренние буферы. Первые 6 байт кадра содержат адрес назначения. Та станция, которая узнает собственный адрес в заголовке кадра, продолжает записывать его содержимое в свой внутренний буфер, а остальные станции на этом прием кадра прекращают. Станция назначения обрабатывает полученные данные, передает их вверх по своему стеку. Кадр Ethernet содержит не только

адрес назначения, но и адрес источника данных, поэтому станция-получатель знает, кому нужно послать ответ.

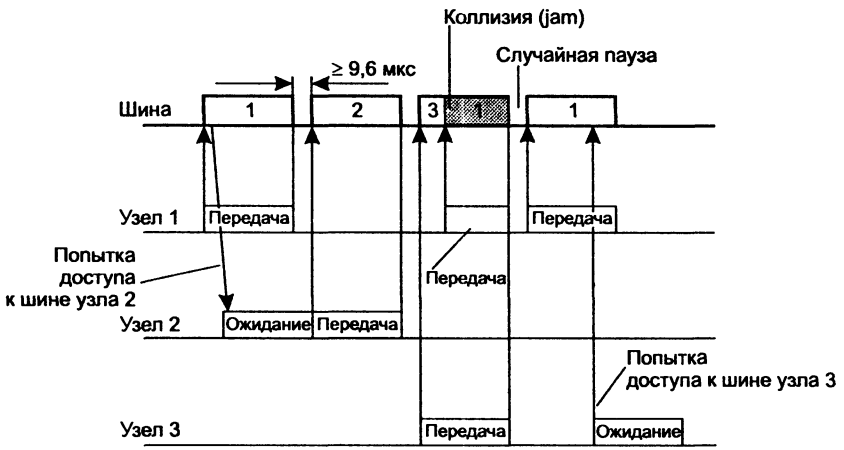


Рис. 12.7. Метод случайного доступа CSMA/CD

Узел 2 во время передачи кадра узлом 1 также пытался начать передачу своего кадра, однако обнаруживает, что среда занята — на ней присутствует несущая частота, — поэтому узел 2 вынужден ждать, пока узел 1 не прекратит передачу кадра.

После окончания передачи кадра все узлы сети обязаны выдержать технологическую паузу, равную **межпакетному интервалу** (Inter Packet Gap, IPG) в 9,6 мкс. Эта пауза нужна для приведения сетевых адаптеров в исходное состояние, а также для предотвращения монопольного захвата среды одной станцией. После окончания технологической паузы узлы имеют право начать передачу своего кадра, так как среда свободна. В приведенном примере узел 2 дождался окончания передачи кадра узлом 1, сделал паузу в 9,6 мкс и начал передачу своего кадра.

Возникновение коллизии

Механизм прослушивания среды и пауза между кадрами не гарантируют исключения такой ситуации, когда две или более станции одновременно решают, что среда свободна, и начинают передавать свои кадры. Говорят, что при этом происходит **коллизия**, так как содержимое обоих кадров сталкивается на общем кабеле и происходит искажение информации.

Коллизия — это нормальная ситуация в работе сетей Ethernet. В примере на рис. 12.8 коллизию породила одновременная передача данных узлами 3 и 1. Для возникновения коллизии не обязательно, чтобы несколько станций начали передачу *абсолютно* одновременно, такая ситуация маловероятна. Более вероятна ситуация, когда один узел начинает передачу, а через некоторое (короткое) время другой узел, проверив среду и не обнаружив несущую (сигналы первого узла

еще не успели до него дойти), начинает передачу своего кадра. Таким образом, возникновение коллизии является следствием распределения узлов сети в пространстве.

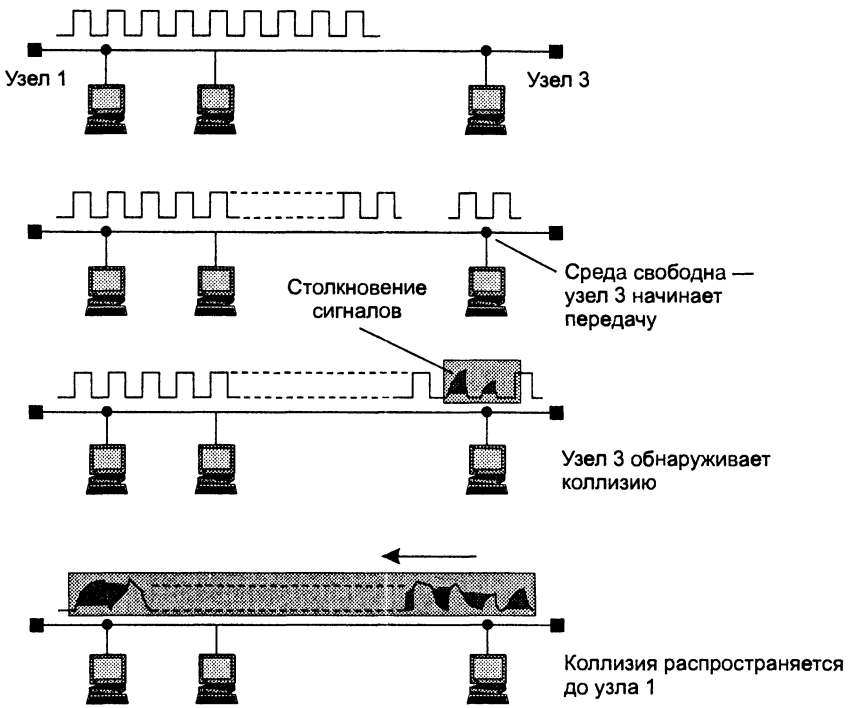


Рис. 12.8. Схема возникновения и распространения коллизии

Чтобы корректно обработать коллизию, все станции одновременно наблюдают за возникающими на кабеле сигналами. Если передаваемые и наблюдаемые сигналы отличаются, то фиксируется факт **обнаружения коллизии** (Collision Detection, CD). Для увеличения вероятности скорейшего обнаружения коллизии всеми станциями сети станция, которая обнаружила коллизию, прерывает передачу своего кадра (в произвольном месте, возможно, и не на границе байта) и усиливает ситуацию коллизии посылкой в сеть специальной последовательности из 32 бит, называемой **jam-последовательностью**.

После этого обнаружившая коллизию передающая станция обязана прекратить передачу и сделать паузу в течение короткого случайного интервала времени. Затем она может снова предпринять попытку захвата среды и передачи кадра. Случайная пауза выбирается по следующему алгоритму:

$$\text{Пауза} = L \times (\text{интервал отсрочки}).$$

В технологии Ethernet **интервал отсрочки** выбран равным значению 512 битовых интервалов. Битовый интервал соответствует времени между появлением двух последовательных битов данных на кабеле; для скорости 10 Мбит/с величина битового интервала равна 0,1 мкс, или 100 нс.

L представляет собой целое число, выбранное с равной вероятностью из диапазона [0, 2^N], где N — номер повторной попытки передачи данного кадра: 1, 2, ..., 10. После 10-й попытки интервал, из которого выбирается пауза, не увеличивается.

Таким образом, случайная пауза в технологии Ethernet может принимать значения от 0 до 52,4 мс.

Если 16 последовательных попыток передачи кадра вызывают коллизию, то передатчик должен прекратить попытки и отбросить этот кадр. Описанный алгоритм носит название **усеченного экспоненциального двоичного алгоритма отсрочки**.

Поведение сети Ethernet при значительной нагрузке, когда коэффициент использования среды растет и начинает приближаться к 1, в целом соответствует графикам, которые были приведены в главе 6 при анализе модели теории очередей M/M/1. Однако рост времени ожидания освобождения среды в сетях Ethernet начинается раньше, чем в модели M/M1. Это происходит из-за того, что модель M/M/1 является очень простой и не учитывает такой важной особенности Ethernet, как коллизии.

Администраторы сетей Ethernet с разделяемой средой руководствуются простым эмпирическим правилом — коэффициент использования среды не должен превышать 30 %. Для поддержки чувствительного к задержкам трафика сети Ethernet (и другие сети с разделяемой средой) могут применять только один метод поддержания характеристик QoS — *недогруженный режим работы*.

Время оборота и распознавание коллизий

Надежное распознавание коллизий всеми станциями сети является необходимым условием корректной работы сети Ethernet. Если какая-либо передающая станция не распознает коллизию и решит, что кадр данных передан ею верно, этот кадр будет утерян. Из-за наложения сигналов при коллизии информация кадра исказится, и он будет отбракован принимающей станцией из-за несовпадения контрольной суммы. Скорее всего, недошедшие до получателя данные будут повторно переданы каким-либо протоколом верхнего уровня, например транспортным или прикладным, работающим с установлением соединения, или протоколом LLC, если он работает в режиме LLC2. Но повторная передача сообщения протоколами верхних уровней произойдет гораздо позже (иногда по прошествии нескольких секунд), чем повторная передача средствами сети Ethernet, работающей с микросекундными интервалами. Поэтому если коллизии не будут надежно распознаваться узлами сети Ethernet, то это приведет к заметному снижению полезной пропускной способности сети.

Для надежного распознавания коллизий должно выполняться следующее соотношение:

$$T_{min} \geq PDV.$$

Здесь T_{min} — время передачи кадра минимальной длины, а PDV (Path Delay Value) — **время оборота**[1], то есть время, за которое сигнал коллизии успевает распространиться до самого дальнего узла сети. В худшем случае сигнал должен пройти дважды между наиболее удаленными друг от друга станциями сети (в одну сторону проходит неискаженный сигнал, а в обратном направлении распространяется уже искаженный коллизией сигнал).

При выполнении этого условия передающая станция должна успеть обнаружить коллизию, которую вызвал переданный ее кадр, еще до того, как она закончит передачу этого кадра.

Очевидно, что выполнение этого условия зависит, с одной стороны, от длины минимального кадра и скорости передачи данных протокола, а с другой стороны, от длины кабельной системы сети и скорости распространения сигнала в кабеле (для разных типов кабеля эта скорость несколько отличается).

Все параметры протокола Ethernet подобраны таким образом, чтобы при нормальной работе сети коллизии четко распознавались.

Так, стандарт Ethernet определяет минимальную длину поля данных кадра в 46 байт (что вместе со служебными полями дает минимальную длину кадра 64 байт, а вместе с преамбулой — 72 байт, или 576 бит). Отсюда может быть вычислено ограничение на расстояние между станциями. В стандарте Ethernet 10 Мбит/с время передачи кадра минимальной длины равно 575 битовых интервалов, следовательно, время оборота должно быть меньше 57,5 мкс. Расстояние, которое сигнал может пройти за это время, зависит от типа кабеля и для толстого коаксиального кабеля равно примерно 13 280 метров. Учитывая, что за время 57,5 мкс сигнал должен пройти по линии связи дважды, расстояние между двумя узлами не должно быть больше 6635 м. В стандарте величина этого расстояния выбрана существенно меньше с учетом других, более строгих ограничений.

Одно из таких ограничений связано с предельно допустимым затуханием сигнала. Для обеспечения необходимой мощности сигнала при его прохождении между наиболее удаленными друг от друга станциями максимальная длина непрерывного сегмента толстого коаксиального кабеля с учетом вносимого им затухания выбрана в 500 м. Очевидно, что на кабеле в 500 м условия распознавания коллизий будут выполняться с большим запасом для кадров любой стандартной длины, в том числе и 72 байт (время оборота по кабелю 500 м составляет всего 43,3 битовых интервала). Поэтому минимальная длина кадра могла бы быть установлена еще меньше. Однако разработчики технологии не стали уменьшать минимальную длину кадра, имея в виду сети, которые строятся из нескольких сегментов, соединенных повторителями.

Повторители увеличивают мощность передаваемых с сегмента на сегмент сигналов, что позволяет использовать сеть гораздо большей длины. В коаксиальных реализациях Ethernet разработчики ограничили максимальное количество сег-

[1] Для обозначения времени оборота используют также аббревиатуру RTT (Round Trip Time). Обе аббревиатуры (PDV и RTT) означают одно и то же, однако в некоторых стандартах Ethernet обозначение PVD предпочтительнее.

ментов в сети *пятью*, что, в свою очередь, ограничивает общую длину сети 2500 метрами. Даже в такой многосегментной сети условие обнаружения коллизий по-прежнему выполняется с большим запасом (сравним полученное из условия допустимого затухания расстояние в 2500 м с вычисленным выше максимально возможным по времени распространения сигнала расстоянием 6635 м). Однако в действительности временной запас существенно меньше, поскольку в многосегментных сетях сами повторители вносят в распространение сигнала дополнительную задержку в несколько десятков битовых интервалов. Небольшой запас был сделан также для компенсации отклонений параметров кабеля и повторителей.

В результате учета всех факторов было тщательно подобрано соотношение между минимальной длиной кадра и максимально возможным расстоянием между станциями сети, которое обеспечивает надежное распознавание коллизий. Это расстояние называют максимальным диаметром сети. Для всех типов сетей Ethernet оно не должно превышать *2500 м*.

С увеличением скорости передачи кадров, что имеет место в новых стандартах, базирующихся на том же методе доступа CSMA/CD, например в Fast Ethernet, стали пропорционально уменьшать максимальное расстояние между станциями сети. В стандарте Fast Ethernet оно составляет около *210 м*, а в стандарте Gigabit Ethernet оно было бы ограничено значением 25 м, если бы разработчики стандарта не увеличили минимальный размер пакета.

В табл. 12.1 приведены значения основных параметров передачи кадра стандарта 802.3, которые не зависят от физической среды. Важно отметить, что каждый вариант физической среды технологии Ethernet добавляет к этим ограничениям свои, часто более строгие ограничения, которые также должны выполняться и которые будут рассмотрены ниже.

Таблица 12.1. Параметры уровня MAC Ethernet

Параметры	Значения
Битовая скорость	10 Мбит/с
Интервал отсрочки	512 битовых интервала
Межкадровый, или межпакетный, интервал (IPG)	9,6 мкс
Максимальное число попыток передачи	16
Максимальное число возрастания диапазона паузы	10
Длина jam-последовательности	32 бит
Максимальная длина кадра (без преамбулы)	1518 байт
Минимальная длина кадра (без преамбулы)	64 байт (512 бит)
Длина преамбулы	64 бит
Минимальная длина случайной паузы после коллизии	0 битовых интервалов

продолжение

Таблица 12.1 (*продолжение*)

Параметры	Значения
Максимальная длина случайной паузы после коллизии	524 000 битовых интервала
Максимальное расстояние между станциями сети	2500 м
Максимальное число станций в сети	1024

Форматы кадров технологии Ethernet

Список ключевых слов: кадр Ethernet DIX, или Ethernet II, кадр 802.3/LLC, 802.3/802.2, или Novell 802.2, кадр Raw 802.3, или Novell 802.3, кадр Ethernet SNAP, поле преамбулы, начальный ограничитель кадра, адрес назначения, адрес источника, поле длины, поле данных, поле заполнения, поле контрольной последовательности кадра.

Стандарт технологии Ethernet, определенный в документе IEEE 802.3, дает описание единственного формата кадра уровня MAC. Так как в кадр уровня MAC должен вкладываться кадр уровня LLC, описанный в документе IEEE 802.2, то по стандартам IEEE в сети Ethernet может использоваться только единственный вариант кадра канального уровня, заголовок которого является комбинацией заголовков подуровней MAC и LLC.

Тем не менее на практике в сетях Ethernet на канальном уровне используются кадры 4-х различных форматов (типов). Один и тот же тип кадра может иметь разные названия, поэтому далее для каждого типа кадров приведено несколько наиболее употребительных названий.

- Кадр **Ethernet DIX**, или **Ethernet II**, появился в результате работы консорциума трех фирм Digital, Intel и Xerox в 1980 году, который представил на рассмотрение комитету 802.3 свою фирменную версию стандарта Ethernet в качестве проекта международного стандарта.
- Однако комитет 802.3 принял стандарт, отличающийся в некоторых деталях от предложения DIX, причем отличия касались и формата кадра. Так возник формат кадра **802.3/LLC**, **802.3/802.2**, или **Novell 802.2**.
- Кадр **Raw 802.3**, или **Novell 802.3**, — появился в результате усилий компании Novell по ускорению работы своего стека протоколов в сетях Ethernet.
- Кадр **Ethernet SNAP** стал результатом деятельности комитета 802.2 по приведению предыдущих форматов кадров к некоторому общему стандарту и приданию кадру необходимой гибкости для учета в будущем возможностей добавления полей или изменения их назначения.

Различия в форматах кадров могут приводить к несовместимости в работе аппаратуры и сетевого программного обеспечения, рассчитанного на функционирование только с одним стандартом кадра Ethernet. Однако сегодня практически все сетевые адаптеры, драйверы сетевых адаптеров, мосты/коммутаторы и маршрутизаторы умеют работать со всеми используемыми на практике форматами кадров технологии Ethernet, причем распознавание типа кадра выполняется автоматически.

Форматы всех этих четырех типов кадров Ethernet приведены на рис. 12.9.

Кадр 802.3/LLC

6	6	2	1	1	1(2)	46–1497 (1496)	4
DA	SA	L	DSAP	SSAP	Управляющее поле	Данные	FCS
			Заголовок LLC				

Кадр Raw 802.3/Novell 802.3

6	6	2	46–1500	4
DA	SA	L	Данные	FCS

Кадр Ethernet DIX (II)

6	6	2	46–1500	4
DA	SA	T	Данные	FCS

Кадр Ethernet SNAP

6	6	2	1	1	1	3	2	46–1492	4
DA	SA	L	DSAP	SSAP	Управляющее поле	OUI	T	Данные	FCS
			AA	AA	03	000000			
			Заголовок LLC			Заголовок SNAP			

Рис. 12.9. Форматы кадров Ethernet

Кадр 802.3/LLC

Заголовок кадра 802.3/LLC является результатом объединения полей заголовков кадров, определенных в стандартах IEEE 802.3 и 802.2.

Стандарт 802.3 определяет восемь полей заголовка (на рис. 12.9 поле преамбулы и начальный ограничитель кадра не показаны).

- **Поле преамбулы** состоит из семи синхронизирующих байтов — 10101010. При манчестерском кодировании эта комбинация представляется в физической среде периодическим волновым сигналом с частотой 5 МГц.
- **Начальный ограничитель кадра** (Start-of-Frame-Delimiter, SFD) состоит из одного байта 10101011. Появление этой комбинации битов является указанием на то, что следующий байт — это первый байт заголовка кадра.
- **Адрес назначения** (Destination Address, DA) может быть длиной 2 или 6 байт. На практике всегда используются MAC-адреса из 6 байт.
- **Адрес источника** (Source Address, SA) — это 2- или 6-байтовое поле, содержащее MAC-адрес узла — отправителя кадра. Первый бит адреса всегда имеет значение 0.

- **Длина** (Length, L) — 2-байтовое поле, которое определяет длину поля данных в кадре.
- **Поле данных** может содержать от 0 до 1500 байт. Но если длина поля меньше 46 байт, то используется следующее поле — *поле заполнителя*, дополняющее кадр до минимально допустимого значения в 46 байт.
- **Поле заполнителя** состоит из такого количества байтов заполнителя, которое обеспечивает минимальную длину поля данных в 46 байт. Это позволяет корректно работать механизму обнаружения коллизий. Если длина поля данных больше или равна минимальной, то поле заполнителя в кадре не появляется.
- **Поле контрольной последовательности кадра** (Frame Check Sequence, FCS) состоит из 4 байт контрольной суммы. Это значение вычисляется по алгоритму CRC-32.

Кадр 802.3 является кадром подуровня MAC, поэтому в соответствии со стандартом 802.2 в его поле данных вкладывается кадр подуровня LLC с удаленными флагами начала и конца кадра. Формат кадра LLC был описан выше. Так как кадр LLC имеет заголовок длиной 3 (в режиме LLC1) или 4 байт (в режиме LLC2), то максимальный размер поля данных уменьшается до 1497 или 1496 байт.

Кадр Raw 802.3/Novell 802.3

Кадр Raw 802.3, называемый еще кадром *Novell 802.3*, также представлен на рис. 12.9. Из рисунка видно, что он представляет собой кадр подуровня MAC стандарта 802.3, но без вложенного кадра подуровня LLC. Компания Novell долгое время не использовала служебные поля кадра LLC в своей операционной системе NetWare благодаря отсутствию необходимости идентифицировать тип информации, вложенной в поле данных, — там всегда находился пакет протокола IPX, долгое время бывшего единственным протоколом сетевого уровня в ОС NetWare.

Теперь, когда необходимость идентификации протокола верхнего уровня появилась, компания Novell стала использовать возможность инкапсуляции в кадр подуровня MAC кадра LLC, то есть возможность применять стандартные кадры 802.3/LLC. Такой кадр компания обозначает теперь в своих операционных системах как кадр 802.2, хотя он является комбинацией заголовков 802.3 и 802.2.

Кадр Ethernet DIX/Ethernet II

Кадр Ethernet DIX, называемый также *кадром Ethernet II*, имеет структуру, совпадающую со структурой кадра Raw 802.3 (см. рис. 12.9). Однако 2-байтовое поле длины (L) кадра Raw 802.3 в кадре *Ethernet DIX* используется в качестве поля типа (Type, T) протокола. Это поле предназначено для тех же целей, что и поля DSAP и SSAP кадра LLC — для указания типа протокола верхнего уровня, вложившего свой пакет в поле данных этого кадра.

В то время как коды протоколов в полях SAP имеют длину один байт, в поле типа для кода протокола отводятся 2 байта. Поэтому один и тот же протокол в поле SAP и поле типа будет кодироваться в общем случае разными числовыми значениями. Например, протокол IP имеет код 2048_{10} (0x0800) для поля типа

и значение 6 для поля SAP. Значения кодов протоколов для поля типа появились раньше значений для поля SAP, так как фирменная версия Ethernet DIX существовала до появления стандарта 802.3, и ко времени распространения оборудования 802.3 эти значения уже стали стандартами де-факто для многих аппаратных и программных продуктов. Так как структуры кадров Ethernet DIX и Raw 802.3 совпадают, то поле длины/типа часто в документации обозначают как поле L/T. При этом числовое значение этого поля определяет его смысл: если значение меньше 1500, то это поле длины, а если больше — то типа.

Кадр Ethernet SNAP

Для устранения разнобоя в кодировках типов протоколов, сообщения которых вложены в поле данных кадров Ethernet, комитетом 802.2 была проведена работа по дальнейшей стандартизации кадров Ethernet. В результате появился кадр Ethernet SNAP (SubNetwork Access Protocol — протокол доступа к подсетям). Кадр Ethernet SNAP (см. рис. 12.9) представляет собой расширение кадра 802.3/LLC за счет введения дополнительного заголовка протокола SNAP, состоящего из двух полей: OUI и типа. Поле типа состоит из 2 байт и повторяет по формату и назначению поле типа кадра Ethernet II (то есть в нем используются те же значения кодов протоколов). Поле OUI определяет уже знакомый нам организационно уникальный идентификатор — то есть идентификатор организации, которая контролирует коды протоколов в поле типа. С помощью заголовка SNAP достигнута совместимость с кодами протоколов в кадрах Ethernet II, а также создана универсальная схема кодирования протоколов. Коды протоколов для технологий 802 контролирует организация IEEE, идентификатор OUI которой равен 000000. Если в будущем потребуются другие коды протоколов для какой-либо новой технологии, для этого достаточно будет указать другой идентификатор организации, назначающей эти коды, а старые значения кодов останутся в силе (в сочетании с другим идентификатором OUI).

Так как SNAP представляет собой протокол, вложенный в протокол LLC, то в полях DSAP и SSAP записывается код 0xAA, отведенный для протокола SNAP. В управляющем поле заголовка LLC устанавливается значение 0x03, что соответствует использованию ненумерованных кадров.

Заголовок SNAP является дополнением к заголовку LLC, поэтому он допустим не только в кадрах Ethernet, но и в кадрах протоколов других технологий комитета 802. Например, протокол IP всегда использует структуру заголовков LLC/SNAP при инкапсуляции в кадры всех протоколов локальных сетей: FDDI, Token Ring, 100VG-AnyLAN, Ethernet, Fast Ethernet, Gigabit Ethernet. Правда, при передаче IP-пакетов через сети Ethernet, Fast Ethernet и Gigabit Ethernet протокол IP использует кадры Ethernet DIX.

Использование различных типов кадров Ethernet

Из-за того что существует четыре типа кадров Ethernet, для протоколов сетевого уровня возникает проблема — пользоваться ли всегда одним типом кадра, применять все четыре или же отдавать предпочтение только некоторым из них.

Протокол IP может использовать два типа кадров: оригинальный кадр Ethernet II и наиболее структурно сложный кадр Ethernet SNAP. Предпочтительным типом кадра для протокола IP является кадр Ethernet II.

Современные сетевые адаптеры автоматически распознают тип кадра Ethernet, используя значения полей кадров. Например, кадры Ethernet II легко отличить от других типов кадров по значению поля L/T: если оно больше 1500, это означает, что поле является полем типа протокола (T), так как значения кодов протоколов выбраны так, что они всегда больше 1500. В свою очередь, наличие поля T говорит о том, что это кадр Ethernet II, который единственный использует это поле в данной позиции кадра.

Протокол IPX «является максималистом», он может работать со всеми четырьмя типами кадров Ethernet. Он распознает кадры Ethernet II описанным выше способом, а если кадр принадлежит к другому типу (поле L/T имеет значение меньшее или равное 1500), то выполняется дальнейшая проверка по наличию или отсутствию полей LLC. Поля LLC могут отсутствовать только в том случае, если за полем длины идет начало пакета IPX, а именно 2-байтовое поле, которое всегда заполняется единицами, что дает значение 0xFFFF, или два байта по 255. Ситуация, когда поля DSAP и SSAP одновременно содержат такие значения, возникнуть не может, поэтому наличие двух байтов 255 говорит о том, что это кадр Raw 802.3.

В остальных случаях дальнейший анализ проводится в зависимости от значений полей DSAP и SSAP. Если они равны 0xAA, то это кадр Ethernet SNAP, а если нет, то 802.3/LLC.

Максимальная производительность сети Ethernet

Список ключевых слов: номинальная скорость протокола, полезная пропускная способность протокола, Ethernet, битовая скорость, скорость в кадрах в секунду, кадр минимальной длины, кадр максимальной длины, коэффициент использования сети.

Производительность сети зависит от скорости передачи кадров по линиям связи и скорости обработки этих кадров коммуникационными устройствами, передающими кадры между своими портами, к которым эти линии связи подключены. Скорость передачи кадров по линиям связи зависит от используемых протоколов физического и канального уровней, например Ethernet 10 Мбит/с, Ethernet 100 Мбит/с, Token Ring или FDDI.

Скорость, с которой протокол передает биты по линии связи, называется **номинальной скоростью протокола**.

Скорость обработки кадров коммуникационным устройством зависит от производительности его процессоров, внутренней архитектуры и других параметров.

Очевидно, что скорость коммуникационного устройства должна соответствовать скорости работы линии. Если она меньше скорости работы линии, то кадры будут стоять в очередях и отбрасываться при их переполнении. С другой стороны, нет смысла применять устройство, которое в сотни раз производительнее, чем того требует скорость подключаемых к нему линий.

Для оценки требуемой производительности коммуникационных устройств, имеющих порты Ethernet, необходимо оценить производительность *сегмента Ethernet*, но не в битах в секунду (ее мы знаем — это 10 Мбит/с), а в кадрах в секунду, так как именно этот показатель помогает оценить требования к производительности коммуникационных устройств. Это объясняется тем, что на обработку каждого кадра, независимо от его длины, мост, коммутатор или маршрутизатор тратит примерно равное время, которое уходит на просмотр таблицы продвижения пакета, формирование нового кадра (для маршрутизатора) и т. п.

При постоянной битовой скорости количество кадров, поступающих на коммуникационное устройство в единицу времени, является, естественно, максимальным при их минимальной длине. Поэтому для коммуникационного оборудования наиболее тяжелым режимом является обработка потока кадров *минимальной длины*.

Используя параметры, приведенные в табл. 12.1, рассчитаем максимальную производительность сегмента Ethernet в таких единицах, как число переданных кадров (пакетов) минимальной длины в секунду.

ПРИМЕЧАНИЕ

При указании производительности сетей термины «кадр» и «пакет» обычно используются как синонимы. Соответственно, аналогичными являются и единицы измерения производительности кадры в секунду (кадр/с) и пакеты в секунду (пакет/с).

Для расчета максимального количества кадров минимальной длины, проходящих по сегменту Ethernet, вспомним, что подсчитанное нами ранее время, затрачиваемое на передачу кадра минимальной длины (576 бит), составляет 57,5 мкс. Прибавив межкадровый интервал в 9,6 мкс, получаем, что период следования кадров минимальной длины составляет 67,1 мкс (рис. 12.10). Отсюда *максимально возможная пропускная способность сегмента Ethernet составляет 14 880 кадр/с.* (Естественно, что наличие в сегменте нескольких узлов снижает эту величину за счет ожидания доступа к среде, а также за счет коллизий.)

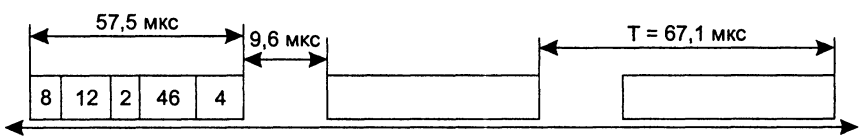


Рис. 12.10. К расчету пропускной способности протокола Ethernet

Кадры максимальной длины технологии Ethernet имеют поле данных 1500 байт, что вместе со служебной информацией дает 1518 байт, а с преамбулой составляет 1526 байт, или 12 208 бит. *Максимально возможная пропускная способность*

сегмента Ethernet для кадров максимальной длины составляет 813 кадр/с. Очевидно, что при работе с большими кадрами нагрузка на мосты, коммутаторы и маршрутизаторы довольно ощутимо снижается.

Теперь рассчитаем, какой максимально полезной пропускной способностью, измеряемой в битах в секунду, обладают сегменты Ethernet при использовании кадров разного размера.

Полезной пропускной способностью протокола называется максимальная скорость передачи *пользовательских* данных, которые переносятся полем данных кадра.

Эта пропускная способность всегда меньше номинальной битовой скорости протокола Ethernet за счет нескольких факторов:

- служебной информации кадра;
- межкадровых интервалов (IPG);
- ожидания доступа к среде.

Для кадров минимальной длины полезная пропускная способность равна:

$$B = 14\,880 \times 46 \times 8 = 5{,}48 \text{ Мбит/с}.$$

Это несколько меньше, чем 10 Мбит/с, но следует учесть, что кадры минимальной длины используются в основном для передачи квитанций, так что к передаче собственно данных файлов эта скорость имеет небольшое отношение.

Для кадров максимальной длины полезная пропускная способность равна:

$$B_{п} = 813 \times 1500 \times 8 = 9{,}76 \text{ Мбит/с}.$$

При использовании кадров среднего размера с полем данных в 512 байт пропускная способность протокола составит 9,29 Мбит/с.

В двух последних случаях пропускная способность протокола оказалась достаточно близкой к предельной пропускной способности в 10 Мбит/с, однако следует учесть, что при расчете мы предполагали, что двум взаимодействующим станциям «не мешают» никакие другие станции сети, то есть отсутствуют коллизии и ожидание доступа.

Таким образом, при отсутствии коллизий коэффициент использования сети зависит от размера поля данных кадра и имеет максимальное значение 0,976 при передаче кадров максимальной длины.

Спецификации физической среды Ethernet

Список ключевых слов: стандарт 10Base-5, стандарт 10Base-2, стандарт 10Base-T, стандарт 10Base-F, трансивер, jabber-контроль, детектор коллизий, развязывающий элемент, повторитель, правило 5-4-3, Т-коннектор, концентратор, или хаб, правило 4-х хабов, стандарт FOIRL, стандарт 10Base-FL, стандарт 10Base-FB, синхронный стандарт Ethernet, домен коллизий.

Исторически первые сети технологии Ethernet были созданы на коаксиальном кабеле диаметром 0,5 дюйма. В дальнейшем были определены и другие спецификации физического уровня для стандарта Ethernet, позволяющие задействовать различные среды передачи данных. Метод доступа CSMA/CD и все временные параметры остаются одними и теми же для любой спецификации физической среды технологии Ethernet 10 Мбит/с.

Физические спецификации технологии Ethernet на сегодня включают следующие среды передачи данных.

- **10Base-5** — коаксиальный кабель диаметром 0,5 дюйма, называемый «толстым» коаксиалом. Имеет волновое сопротивление 50 Ом. Максимальная длина сегмента — 500 метров (без повторителей).
- **10Base-2** — коаксиальный кабель диаметром 0,25 дюйма, называемый «тонким» коаксиалом. Имеет волновое сопротивление 50 Ом. Максимальная длина сегмента — 185 метров (без повторителей).
- **10Base-T** — кабель на основе неэкранированной витой пары (UTP). Образует звездообразную топологию на основе концентратора. Расстояние между концентратором и конечным узлом — не более 100 м.
- **10Base-F** — волоконно-оптический кабель. Топология аналогична топологии стандарта 10Base-T. Имеется несколько вариантов этой спецификации — FOIRL (расстояние до 1000 м), 10Base-FL (расстояние до 2000 м), 10Base-FB (расстояние до 2000 м).

Число 10 в указанных выше названиях обозначает номинальную битовую скорость передачи данных этих стандартов — 10 Мбит/с, а слово «Base» — метод передачи на одной базовой частоте 10 МГц — в отличие от методов, использующих несколько несущих частот (они называются широкополосными и имеют в своем составе слово «Broadband»). Последний символ в названии стандарта физического уровня обозначает тип кабеля.

Стандарт 10Base-5

Стандарт 10Base-5 в основном соответствует экспериментальной сети Ethernet фирмы Xerox и может считаться классическим стандартом Ethernet. Различные компоненты сети, выполненной на толстом коаксиале и состоящей из трех сегментов, соединенных повторителями, показаны на рис. 12.11.

Кабель используется как моноканал для всех станций. Сегмент кабеля максимальной длины в 500 м (без повторителей) должен иметь на концах согласующие *терминаторы* («заглушки») сопротивлением 50 Ом, поглощающие распространяющиеся по кабелю сигналы и препятствующие возникновению отраженных сигналов. При отсутствии терминаторов в кабеле возникают стоячие волны, так что одни узлы получают мощные сигналы, а другие — настолько слабые, что их прием становится невозможным.

Станция должна подключаться к кабелю при помощи приемопередатчика — **трансивера**. Трансивер — это часть сетевого адаптера; он устанавливается непосредственно на кабеле и питается от сетевого адаптера компьютера. Трансивер

может подсоединяться к кабелю как методом прокалывания, обеспечивающим непосредственный физический контакт, так и бесконтактным методом.

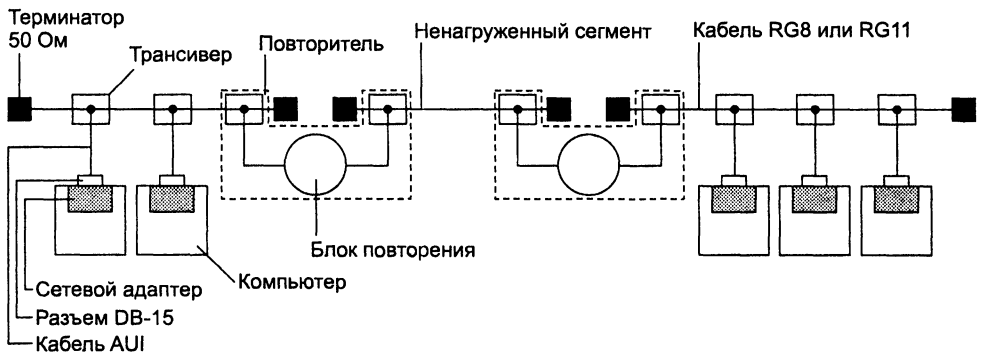


Рис. 12.11. Компоненты физического уровня сети стандарта 10 Base-5, состоящей из трех сегментов

Трансивер соединяется с сетевым адаптером интерфейсным кабелем *AUI* (Attachment Unit Interface — интерфейс подключаемых устройств) длиной до 50 м, состоящим из 4 витых пар (адаптер должен иметь разъем AUI). Наличие стандартного интерфейса между трансивером и остальной частью сетевого адаптера очень полезно при переходе с одного типа кабеля на другой. Для этого достаточно только заменить трансивер, а остальная часть сетевого адаптера остается неизменной, так как она отрабатывает протокол уровня MAC. При этом необходимо только, чтобы новый трансивер (например, трансивер для витой пары) поддерживал стандартный интерфейс AUI.

Допускается подключение к одному сегменту не более 100 трансиверов, причем расстояние между подключениями трансиверов не должно быть меньше 2,5 м. На кабеле имеется разметка через каждые 2,5 м, обозначающая точки подключения трансиверов. При подсоединении компьютеров в соответствии с разметкой влияние стоячих волн в кабеле на сетевые адаптеры сводится к минимуму.

Упрощенная структурная схема трансивера показана на рис. 12.12. Передатчик и приемник присоединяются к одной точке кабеля с помощью специальной схемы, например трансформаторной, позволяющей организовать одновременную передачу и прием сигналов с кабеля.

При неисправностях в адаптере может возникнуть ситуация, когда в кабель будет непрерывно выдаваться последовательность случайных сигналов. Так как кабель — это общая среда для всех станций, то работа сети будет заблокирована одним неисправным адаптером. Чтобы этого не случилось, на выходе трансивера ставится схема, которая проверяет время передачи кадра. Если максимально возможное время передачи пакета превышается (с некоторым запасом), то эта схема просто отсоединяет выход передатчика от кабеля. Максимальное время передачи кадра (вместе с преамбулой) равно 1221 мкс, а время затянувшейся передачи устанавливается равным 4000 мкс (4 мс). Эту функцию трансивера иногда называют проверкой затянувшейся передачи, или **jabber-контролем**.

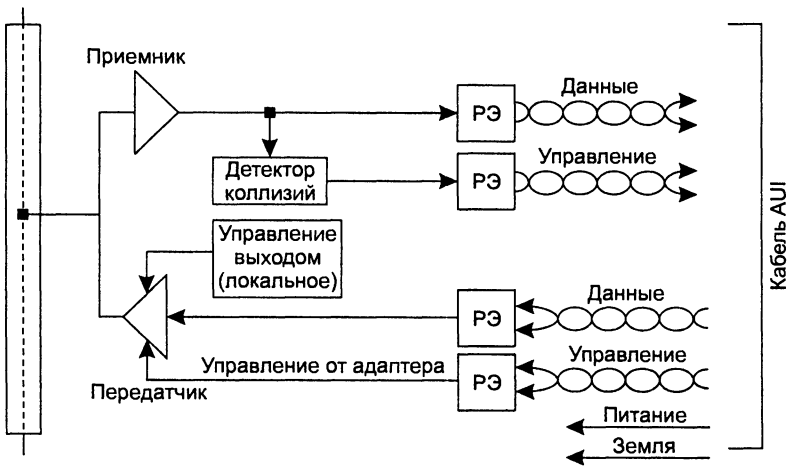


Рис. 12.12. Структурная схема трансивера

Детектор коллизий определяет наличие коллизии в коаксиальном кабеле по повышенному уровню постоянной составляющей сигналов. Если постоянная составляющая превышает определенный порог (около 1,5 В), значит, на кабель работает более одного передатчика.

Развязывающие элементы (РЭ) обеспечивают гальваническую развязку трансивера от остальной части сетевого адаптера и тем самым защищают адаптер и компьютер от значительных перепадов напряжения, возникающих на кабеле при его повреждении.

Стандарт 10Base-5 определяет возможность использования в сети **повторителя**. Повторитель служит для объединения в одну сеть нескольких сегментов кабеля и увеличения тем самым общей длины сети. Повторитель принимает сигналы из одного сегмента кабеля и побитно синхронно повторяет их в другом сегменте, улучшая форму и мощность импульсов, а также синхронизируя импульсы. Повторитель состоит из двух (или нескольких) трансиверов, которые присоединяются к сегментам кабеля, а также блока повторения со своим тактовым генератором. Для лучшей синхронизации повторитель задерживает передачу нескольких первых битов преамбулы кадра, за счет чего увеличивается задержка передачи кадра с сегмента на сегмент, а также несколько уменьшается межкадровый интервал (IPG).

Стандарт разрешает использование в сети не более 4 повторителей и, соответственно, не более 5 сегментов кабеля. При максимальной длине сегмента кабеля в 500 м это дает максимальную длину сети 10Base-5 в 2500 м. Это в точности соответствует общему ограничению стандарта на максимальный диаметр Ethernet.

Только 3 сегмента из 5 могут быть нагруженными, то есть такими, к которым подключаются конечные узлы. Между нагруженными сегментами должны быть ненагруженные сегменты, так что максимальная конфигурация сети представляет собой два нагруженных крайних сегмента, которые соединяются ненагруженными сегментами еще с одним центральным нагруженным сегментом. Ранее на

рис. 12.11 был приведен пример сети Ethernet, состоящей из трех сегментов, объединенных двумя повторителями. Крайние сегменты являются нагруженными, а промежуточный — ненагруженным.

Правило применения повторителей в сети Ethernet 10Base-5 носит название **правила 5-4-3**: 5 сегментов, 4 повторителя, 3 нагруженных сегмента.

Ограниченное число повторителей объясняется дополнительными задержками распространения сигнала, которые они вносят. Применение повторителей увеличивает время оборота сигнала, которое для надежного распознавания коллизий не должно превышать время передачи кадра минимальной длины, то есть кадра в 72 байт или 576 бит. Каждый повторитель подключается к сегменту одним своим трансивером, поэтому к нагруженным сегментам можно подключить не более 99 узлов (а не 100). Максимальное число конечных узлов в сети 10Base-5 таким образом составляет 99 × 3 = 297 узлов.

Стандарт 10Base-2

В стандарте 10Base-2 в качестве передающей среды используется «тонкий» коаксиал Ethernet. Максимальная длина сегмента без повторителей составляет 185 м, сегмент должен иметь на концах согласующие терминаторы 50 Ом. Тонкий коаксиальный кабель дешевле толстого, поэтому сети 10Base-2 иногда называют Cheapernet (дословно — дешевая сеть). Но за дешевизну кабеля приходится расплачиваться качеством — «тонкий» коаксиал обладает худшей помехозащищенностью, худшей механической прочностью и более узкой полосой пропускания.

Станции подключаются к кабелю с помощью высокочастотного **Т-коннектора**, который представляет собой тройник, один отвод которого соединяется с сетевым адаптером, а два других — с двумя концами разрыва кабеля. Максимальное количество станций, подключаемых к одному сегменту, — 30. Минимальное расстояние между станциями — 1 м. Кабель «тонкого» коаксиала имеет разметку для подключения узлов с шагом в 1 м.

Стандарт 10Base-2 также предусматривает использование повторителей по правилу 5-4-3.

В этом случае сеть будет иметь максимальную длину в 5 × 185 = 925 м. Очевидно, что это ограничение является более сильным, чем общее ограничение стандарта Ethernet в 2500 м.

ВНИМАНИЕ

Для построения корректной сети Ethernet нужно соблюсти много ограничений, причем некоторые из них относятся к одним и тем же параметрам сети, например, максимальная длина или максимальное количество компьютеров в сети должны удовлетворять одновременно нескольким разным условиям. Для того чтобы сеть была корректна, достаточно соблюсти только наиболее жесткие требования. Так, если в сети Ethernet не должно быть более 1024 узлов, а стандарт 10Base-2 ограничивает максимальное число станций, подключаемых к одному сегменту, значением 30, а число нагруженных сегментов — значением 3, то общее количество узлов в сети 10Base-2 не должно превышать 29 × 3 = 87.

Стандарт 10Base-2 очень близок к стандарту 10Base-5, но трансиверы в нем объединены с сетевыми адаптерами за счет того, что более гибкий тонкий коаксиальный кабель может быть подведен непосредственно к выходному разъему платы сетевого адаптера, установленной в шасси компьютера. Кабель в данном случае «висит» на сетевом адаптере, что затрудняет физическое перемещение компьютеров.

Типичный состав сети стандарта 10Base-2, состоящей из одного сегмента кабеля, показан на рис. 12.13.

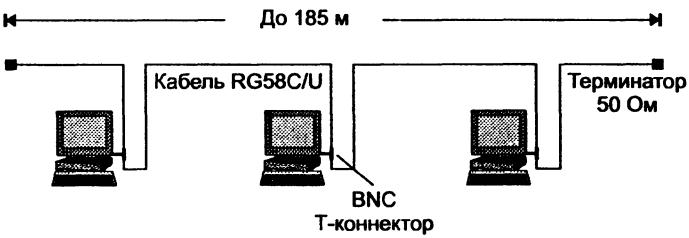


Рис. 12.13. Сеть стандарта 10Base-2

Реализация этого стандарта на практике приводит к наиболее простому решению для кабельной сети, так как для соединения компьютеров требуются только сетевые адаптеры, Т-коннекторы и терминаторы 50 Ом. Однако этот вид кабельных соединений наиболее сильно подвержен авариям и сбоям. Кабель более восприимчив к помехам, чем «толстый» коаксиал. В моноканале имеется большое количество механических соединений: каждый Т-коннектор дает три механических соединения, два из которых имеют жизненное значение для всей сети. Пользователи имеют доступ к разъемам и могут нарушить целостность моноканала. Кроме того, эстетика и эргономичность этого решения оставляют желать лучшего, так как от каждой станции через Т-коннектор отходят два довольно заметных провода, которые под столом часто образуют моток кабеля — запас, необходимый на случай даже небольшого перемещения рабочего места.

Общим недостатком стандартов 10Base-5 и 10Base-2 является отсутствие оперативной информации о состоянии моноканала. Повреждение кабеля обнаруживается сразу же (сеть перестает работать), но для поиска отказавшего отрезка кабеля необходим специальный прибор — кабельный тестер.

Стандарт 10Base-T

В сетях 10Base-T в качестве среды используются две *неэкранированные витые пары*. Многопарный кабель на основе неэкранированной витой пары категории 3 телефонные компании уже достаточно давно применяли для подключения телефонных аппаратов внутри зданий. Этот кабель носит также название Voice Grade, говорящее о том, что он предназначен для передачи голоса.

Идея приспособить этот популярный вид кабеля для локальных сетей оказалась очень плодотворной, так как многие здания уже были оснащены нужной кабельной системой. Оставалось разработать способ подключения сетевых адаптеров

и прочего коммуникационного оборудования к витой паре таким образом, чтобы изменения в сетевых адаптерах и программном обеспечении сетевых операционных систем были минимальными по сравнению с сетями Ethernet на коаксиале. Эта попытка оказалась успешной — переход на витую пару требует только замены трансивера сетевого адаптера или порта маршрутизатора, а метод доступа и все протоколы канального уровня остаются теми же, что и в сетях Ethernet на коаксиале.

Конечные узлы соединяются с помощью двух витых пар по двухточечной топологии со специальным устройством — многопортовым повторителем. Одна витая пара требуется для передачи данных от станции к повторителю (выход T_X сетевого адаптера), а другая — для передачи данных от повторителя к станции (вход R_X сетевого адаптера). На рис. 12.14 показан пример трехпортового повторителя. Повторитель принимает сигналы от одного из конечных узлов и синхронно передает их на все свои остальные порты, кроме того, с которого поступили сигналы.

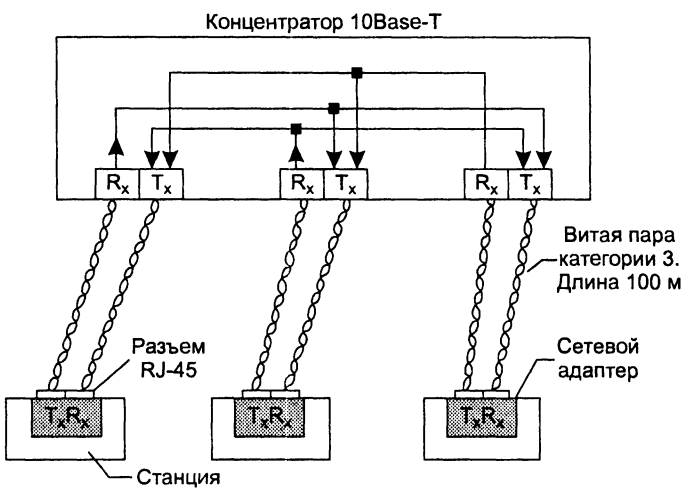


Рис. 12.14. Сеть стандарта 10Base-T

Многопортовые повторители в данном случае обычно называются **концентраторами**, или, на инженерном жаргоне, **хабами**. Концентратор осуществляет функции повторителя сигналов на всех отрезках витых пар, подключенных к его портам, так что образуется единая среда передачи данных — логический моноканал (логическая общая шина). Концентратор обнаруживает коллизию в сегменте в случае одновременной передачи сигналов по нескольким своим R_X-входам и посылает jam-последовательность на все свои T_X-выходы. Стандарт определяет битовую скорость передачи данных 10 Мбит/с и максимальное расстояние отрезка витой пары между двумя непосредственно связанными узлами (станциями и концентраторами) не более 100 м при наличии витой пары качества не ниже категории 3. Это расстояние определяется полосой пропускания витой пары — на длине 100 м она позволяет передавать данные со скоростью 10 Мбит/с при использовании манчестерского кода.

Концентраторы 10Base-T можно соединять друг с другом с помощью тех же портов, которые предназначены для подключения конечных узлов. При этом нужно позаботиться о том, чтобы передатчик и приемник одного порта были соединены соответственно с приемником и передатчиком другого порта.

В стандарте10Base-T определено максимальное число концентраторов между любыми двумя станциями сети, а именно 4. Это правило носит название **правила 4-х хабов**.

Правило 4-х хабов подобно правилу 5-4-3, применяемому к коаксиальным сетям, служит для гарантированной синхронизации станций при реализации процедур доступа CSMA/CD и надежного распознавания станциями коллизий.

При создании сети 10Base-T с большим числом станций концентраторы можно соединять друг с другом иерархическим способом, образуя древовидную структуру (рис. 12.15).

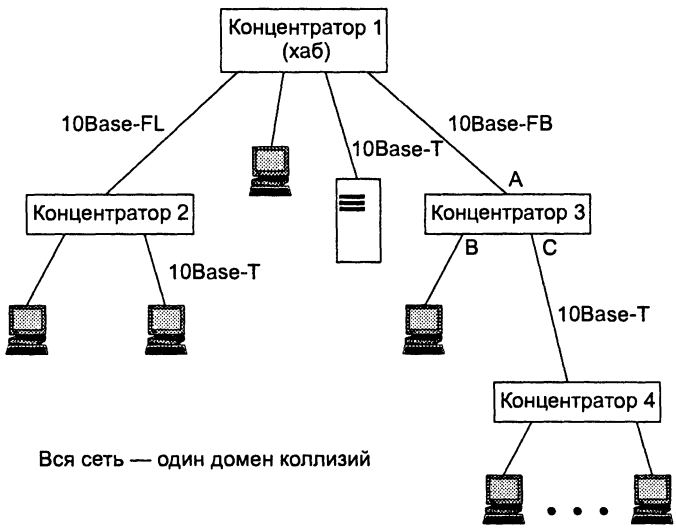


Рис. 12.15. Иерархическое соединение концентраторов Ethernet

ВНИМАНИЕ

Петлевидное соединение концентраторов в стандарте 10Base-T запрещено, так как оно приводит к некорректной работе сети. Это требование означает, что в сети 10Base-T не разрешается создавать параллельные каналы связи между критически важными концентраторами для резервирования связей на случай отказа порта, концентратора или кабеля. Резервирование связей возможно только за счет перевода одной из параллельных связей в неактивное (заблокированное) состояние.

Общее количество станций в сети 10Base-T не должно превышать общего предела в 1024, и для данного типа физического уровня это количество действительно достижимо. Для этого достаточно создать двухуровневую иерархию концентраторов, расположив на нижнем уровне достаточное количество концентраторов

с общим количеством портов 1024 (рис. 12.16). Конечные узлы нужно подключить к портам концентраторов нижнего уровня. Правило 4-х хабов при этом выполняется — между любыми конечными узлами будет ровно 3 концентратора.

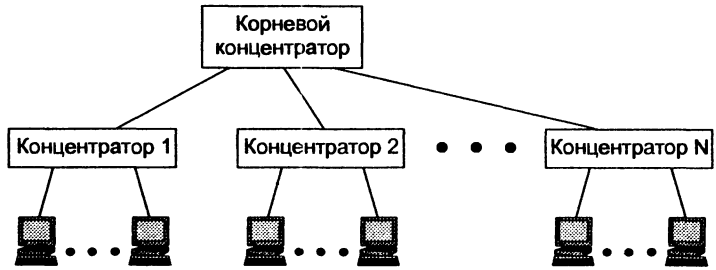


Рис. 12.16. Схема с максимальным количеством станций

Очевидно, что если между любыми двумя узлами сети не должно быть больше 4-х повторителей, то, учитывая, что максимальная длина кабеля между повторителями равна 100 м, получаем, что *максимальный диаметр сети 10Base-T составляет 5 × 100 = 500 м*. Заметим, что это ограничение строже общего ограничения стандартов Ethernet в 2500 м.

Сети, построенные на основе стандарта 10Base-T, обладают по сравнению с коаксиальными вариантами Ethernet многими преимуществами. Эти преимущества связаны с разделением общего физического кабеля на отдельные кабельные отрезки, подключенные к центральному коммуникационному устройству. И хотя логически эти отрезки по-прежнему образуют общую разделяемую среду, их физическое разделение позволяет контролировать их состояние и отключать в случае обрыва, короткого замыкания или неисправности сетевого адаптера на индивидуальной основе. Это обстоятельство существенно облегчает эксплуатацию больших сетей Ethernet, так как концентратор обычно автоматически выполняет такие функции, уведомляя при этом администратора сети о возникшей проблеме.

В стандарте 10Base-T определена процедура тестирования физической работоспособности двух отрезков витой пары, соединяющих трансивер конечного узла и порт повторителя. Эта процедура называется *тестом связности* и основана на передаче каждые 16 мс специальных сигналов J и K манчестерского кода между передатчиком и приемником каждой витой пары. Напомним, что информационные сигналы манчестерского кода всегда изменяют потенциал в середине такта. Коды J и K отличаются тем, что они нарушают это правило, сохраняя потенциал в середине такта неизменным. Коду J соответствует одно из двух значений потенциала, а коду K — другое. Так как коды J и K являются запрещенными при передаче кадров, то тестовые последовательности не влияют на работу алгоритма доступа к среде.

Появление между конечными узлами активного устройства, которое может контролировать работу узлов и изолировать от сети некорректно работающие узлы, является *главным преимуществом* технологии 10Base-T по сравнению со сложными в эксплуатации коаксиальными сетями.

Волоконно-оптическая сеть Ethernet

В качестве среды передачи данных 10-мегабитная сеть Ethernet использует оптическое волокно. Оптоволоконные стандарты в качестве основного типа кабеля рекомендуют достаточно дешевое многомодовое оптическое волокно, обладающее полосой пропускания 500–800 МГц при длине кабеля 1 км. Допустимо и более дорогое одномодовое оптическое волокно с полосой пропускания в несколько гигагерц, но при этом нужно применять специальный тип трансивера.

Функционально сеть Ethernet на оптическом кабеле состоит из тех же элементов, что и сеть стандарта 10Base-T — сетевых адаптеров, многопортового повторителя и отрезков кабеля, соединяющих адаптер с портом повторителя. Как и в случае витой пары, для соединения адаптера с повторителем используются два оптоволокна — одно соединяет выход T_X адаптера с входом R_X повторителя, а другое — вход R_X адаптера с выходом T_X повторителя.

Стандарт FOIRL (Fiber Optic Inter-Repeater Link — волоконно-оптический канал между повторителями) представляет собой первый стандарт комитета 802.3 для использования оптоволокна в сетях Ethernet. Он гарантирует длину оптоволоконной связи между повторителями до 1 км. Максимальное число повторителей между любыми узлами сети — 4. Как и в стандарте 10Base-5, максимального диаметра в 2500 м здесь достичь можно, однако отрезки кабеля предельного размера между *всеми* 4 повторителями, а также между повторителями и конечными узлами недопустимы — иначе получится сеть длиной 5000 м.

Стандарт 10Base-FL представляет собой незначительное улучшение стандарта FOIRL. Увеличена мощность передатчиков, поэтому максимальное расстояние между узлом и концентратором увеличилось до 2000 м. Максимальное число повторителей между узлами осталось равным 4, и стандартная максимальная длина сети 2500 м достижима.

Стандарт 10Base-FB предназначен только для соединения повторителей. Конечные узлы не могут использовать этот стандарт для присоединения к портам концентратора. Между узлами сети можно установить до 5 повторителей 10Base-FB при максимальной длине одного сегмента 2000 м и максимальной длине сети 2740 м.

Повторители, соединенные по стандарту 10Base-FB, при отсутствии кадров для передачи в целях синхронизации постоянно обмениваются специальными последовательностями сигналов, отличающимися от сигналов кадров данных. Поэтому они вносят меньшие задержки при передаче данных из одного сегмента в другой, и это является главной причиной, по которой количество повторителей удалось увеличить до 5. В качестве специальных сигналов используются манчестерские коды J и K в следующей последовательности: J-J-K-K-J-J-... Эта последовательность порождает импульсы частоты 2,5 МГц, которые и поддерживают синхронизацию приемника одного концентратора с передатчиком другого. Поэтому стандарт 10Base-FB имеет также название **синхронный стандарт Ethernet**.

Как и во всех стандартах Ethernet, оптоволоконные стандарты разрешают соединять концентраторы только в древовидные иерархические структуры. Любые петли между портами концентраторов не допускаются.

Домен коллизий

Домен коллизий — это часть сети Ethernet, все узлы которой конкурируют за общую разделяемую среду передачи и, следовательно, каждый узел которой может создать коллизию с любым другим узлом этой части сети.

Сеть Ethernet, построенная на повторителях, всегда образует один домен коллизий. Мосты, коммутаторы и маршрутизаторы делят сеть Ethernet на несколько доменов коллизий.

Приведенная на рис. 12.15 сеть представляет собой один домен коллизий. Если, например, столкновение кадров произошло в концентраторе 4, то в соответствии с логикой работы концентраторов 10Base-T сигнал коллизии распространится по всем портам всех концентраторов.

Если же вместо концентратора 3 поставить в сеть мост, то его порт С, связанный с концентратором 4, воспримет сигнал коллизии, но не передаст его на свои остальные порты, так как это не входит в его обязанности. Мост просто отработает ситуацию коллизии средствами порта С, который подключен к общей среде, где эта коллизия возникла. Если коллизия возникла из-за того, что *мост* пытался передать через порт С кадр в концентратор 4, то, зафиксировав сигнал коллизии, порт С приостановит передачу кадра и попытается передать его повторно через случайный интервал времени. Если порт С принимал в момент возникновения коллизии кадр, то он просто отбросит полученное начало кадра и будет ожидать, когда узел, передававший кадр через концентратор 4, сделает повторную попытку передачи. После успешного принятия данного кадра в свой буфер мост передаст его на другой порт в соответствии с таблицей продвижения, например на порт А. Все события, связанные с обработкой коллизий портом С, для остальных сегментов сети, которые подключены к другим портам моста, просто останутся неизвестными.

Общие характеристики стандартов Ethernet 10 Мбит/с

В табл. 12.2 и 12.3 сведены основные ограничения и характеристики стандартов Ethernet.

Таблица 12.2. Общие ограничения для всех стандартов Ethernet

Характеристика	Значение
Номинальная пропускная способность	10 Мбит/с
Максимальное число станций в сети	1024
Максимальное расстояние между узлами в сети	2500 м (в 10Base-FB 2750 м)
Максимальное число коаксиальных сегментов в сети	5

Таблица 12.3. Параметры спецификаций физического уровня для стандарта Ethernet

Параметр	10Base-5	10Base-2	10Base-T	10Base-F
Кабель	Толстый коаксиальный кабель RG-8 или RG-11	Тонкий коаксиальный кабель RG-58	Неэкранированная витая пара категорий 3, 4, 5	Многомодовый волоконно-оптический кабель
Максимальная длина сегмента, м	500	185	100	2000
Максимальное расстояние между узлами сети (при использовании повторителей), м	2500	925	500	2500 (2740 для 10Base-FB)
Максимальное число станций в сегменте	100	30	1024	1024
Максимальное число повторителей между любыми станциями сети	4	4	4	4 (5 для 10 Base-FB)

Сеть Ethernet завода «Трансмаш»

В начале 90-х годов крупный завод «Трансмаш» использовал сеть Ethernet 10 Мбит/с с разделяемой средой для объединения всех своих мини-компьютеров и персональных компьютеров (рис. 12.17). Компьютеры в основном использовались для решения автономных задач, а обмен данных между ними происходил сравнительно редко. Сеть передавала небольшие объемы алфавитно-цифровой информации, поэтому общая разделяемая среда вполне справлялась с потребностями завода. Для взаимодействия центрального сегмента сети с сегментами удаленных цехов использовались оптоволоконные линии связи стандартов 10Base-FB и 10Base-FL. Сеть удовлетворяла всем требованиям многосегментной конфигурации Ethernet: все отрезки кабелей не превышали предельной длины, между любыми двумя узлами находилось не более 4-х хабов, максимальное расстояние между узлами сети не превышало 1800 метров (компьютеры A и C на рисунке).

Через некоторое время к сети понадобилось присоединить компьютеры еще одного здания, а именно здания 4. Это здание находилось в пределах досягаемости оптоволоконных стандартов Ethernet (10 Base-FB или 10Base-FL), но его присоединение к сети привело бы к некорректной конфигурации, так как между компьютерами зданий 1 и 4 данные проходили бы уже через 5 хабов. Кроме того, диаметр сети достиг бы 2800 метров — еще одно нарушение ограничений Ethernet. Но архитектору сети «Трансмаш» в то время не хотелось коренным образом менять структуру сети и устанавливать мост или маршрутизатор для подключения нового сегмента.

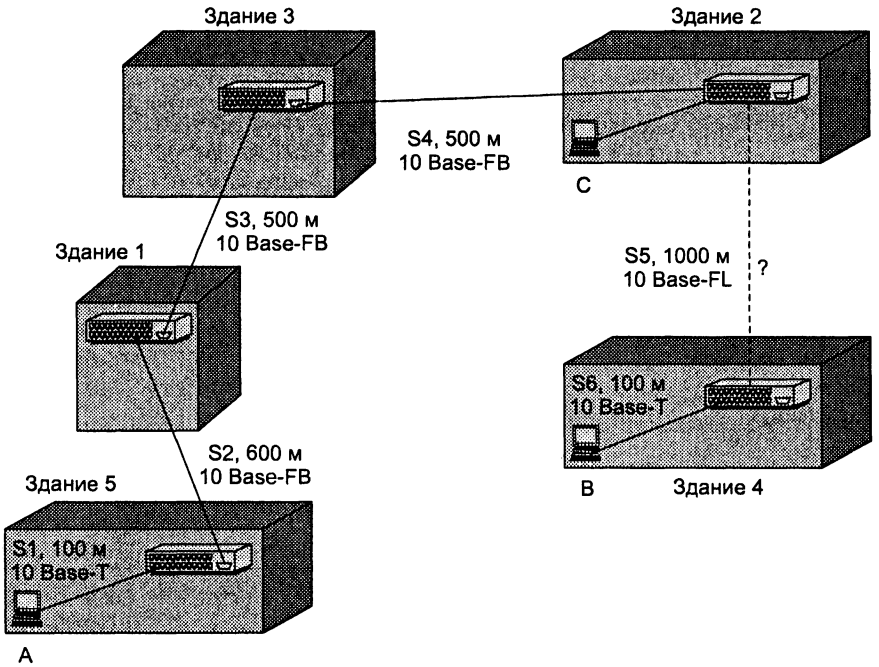


Рис. 12.17. Многосегментная сеть Ethernet завода «Трансмаш»

Он знал, что в разделе 13 стандарта IEEE 802.3 приведена методика расчета корректности конфигурации сети. Эта методика позволяет количественно определить, будет ли та или иная конфигурация сети работать нормально или нет. Расчеты показывают, что иногда можно нарушить правило 4-х хабов и ограничения на максимальный диаметр сети, и все равно конфигурация будет корректной. Дело в том, что эти ограничения выбраны так, чтобы сеть работала с большим запасом «прочности». Например, мы знаем, что для надежного распознавания коллизий любым узлом сети максимальное время оборота не должно превышать 575 битовых интервалов. Если посчитать по приведенной методике время оборота в сети 10Base-5, состоящей из 4-х повторителей 10Base-5 и 5-ти сегментов максимальный длины 500 м, то окажется, что оно составляет 537 битовых интервала. Это значит, что максимальная конфигурация сети 10Base-5 (4 хаба, диаметр сети 2500 м) обладает запасом в 38 битовых интервала. В то же время методика раздела 13 говорит о том, что даже при запасе 4 битовых интервала сеть будет работать корректно.

Поэтому архитектор сети «Трансмаш» выполнил расчет возможной конфигурации сети завода с учетом нового сегмента. Оказалось, что даже при присоединении сегмента здания 4 у сети имеется запас в 6,6 битовых интервала! После перепроверки расчета к зданию 4 был проложен волоконно-оптический кабель, и сеть начала работать в новой конфигурации. Практика подтвердила правильность расчета — сеть продолжала работать нормально. В такой конфигурации она оставалась несколько лет, пока возросшие потребности новых приложений не привели к разделению общей среды на коммутируемые сегменты.

Для того чтобы проверить расчет, который проделал архитектор сети «Трансмаш», нужно предварительно познакомиться с деталями методики, приведенной в разделе 13 стандарта 802.3.

В нем сказано, что сеть Ethernet будет работать корректно, если:

- время оборота (PDV) сигнала между двумя самыми удаленными друг от друга станциями сети не превышает 575 битовых интервала. Повторители и среда сегментов вносят задержки в распространение сигнала, данные о предельных уровнях этих задержек приведены в таблицах стандарта;
- сокращение межпакетного интервала IPG при прохождении последовательности кадров через все повторители будет не больше, чем 49 битовых интервала. Каждый повторитель сокращает значение IPG на определенную величину, которая также приводится в стандарте.

В таблицах стандарта 802.3 приводятся минимальные и максимальные значения возможных задержек распространения сигналов и сокращений IPG, их более определенные значения зависят от производителя повторителей. Архитектор сети «Трансмаш» использовал для расчета более точные данные, которые ему предоставил производитель сетевого оборудования. Эти данные приведены далее.

Рассмотрим сначала, как с помощью данных табл. 12.4 можно оценить значение PDV.

Таблица 12.4. Данные для расчета значения PDV

Тип сегмента	База левого сегмента, битовых интервалов	База промежуточного сегмента, битовых интервалов	База правого сегмента, битовых интервалов	Задержка среды на 1 м, битовых интервалов	Максимальная длина сегмента, м
10Base-5	11,8	46,5	169,5	0,0866	500
10Base-2	11,8	46,5	169,5	0,1026	185
10Base-T	15,3	42,0	165,0	0,113	100
10Base-FB	–	24,0	–	0,1	2000
10Base-FL	12,3	33,5	156,5	0,1	2000
FOIRL	7,8	29,0	152,0	0,1	1000
AUI (>2 м)	0	0	0	0,1026	2+48

Разработчики стандарта 802.3 старались максимально упростить выполнение расчетов, поэтому приведенные данные включают сразу несколько этапов прохождения сигнала. Например, задержки, вносимые повторителем, состоят из задержки входного трансивера, задержки блока повторения и задержки выходного трансивера. Тем не менее в таблице все эти задержки представлены одной величиной, названной базой сегмента.

Чтобы не нужно было два раза складывать задержки, вносимые кабелем, в таблице даются удвоенные величины задержек для каждого типа кабеля.

В таблице используются также такие понятия, как левый сегмент, правый сегмент и промежуточный сегмент. Поясним эти термины на примере сети завода «Трансмаш»

(см. рис. 12.17). Мы хотим рассчитать PDV для худшего случая. Поэтому мы выбрали для расчета узлы A и B, между которыми находятся 5 повторителей, а общая длина сети равна 2800 м.

Левым сегментом в терминологии 802.3 называется сегмент, в котором начинается путь сигнала от выхода передатчика конечного узла. Сам термин «левый» не имеет отношения к расположению сегментов в пространстве (и, конечно, на рисунке). Это просто условное название сегмента, с которого мы начинаем расчет. Для определенности мы выбрали в качестве левого сегмента сегмент S1, к которому подключен узел A.

Затем сигнал проходит через промежуточные сегменты S2–S5 и доходит до приемника (узел B), который подключен к сегменту S6. Именно здесь в худшем случае происходит столкновение кадров и возникает коллизия, что и подразумевается в таблице. Конечный сегмент, в котором может возникнуть коллизия, называется правым сегментом.

С каждым сегментом связана постоянная задержка, названная базой, которая зависит только от типа сегмента и от положения сегмента на пути сигнала (левый, промежуточный или правый). База правого сегмента, в котором возникает коллизия, намного превышает базу левого и промежуточных сегментов.

Кроме того, с каждым сегментом связана задержка распространения сигнала вдоль кабеля сегмента, которая зависит от длины сегмента и вычисляется путем умножения времени распространения сигнала по одному метру кабеля (в битовых интервалах) на длину кабеля в метрах.

Расчет заключается в вычислении задержек, вносимых каждым отрезком кабеля (приведенная в таблице задержка сигнала на 1 м кабеля умножается на длину сегмента), а затем суммировании этих задержек с базами левого, промежуточных и правого сегментов.

Так как левый и правый сегменты имеют разные величины базовой задержки, то в случае различных типов сегментов на удаленных краях сети необходимо выполнить расчеты дважды: один раз принять в качестве левого сегмента сегмент одного типа, а во второй — сегмент другого типа. Результатом можно считать максимальное из полученных значений PDV. В нашем примере крайние сегменты сети принадлежат одному типу — стандарту 10Base-T, поэтому двойной расчет не требуется.

Теперь мы можем рассчитать значение PDV для нашего примера.

- Левый сегмент S1:

$$15{,}3 \text{ (база)} + 100 \times 0{,}113 = 26{,}6.$$

- Промежуточный сегмент S2:

$$24 + 600 \times 0{,}1 = 84{,}0.$$

- Промежуточный сегмент S3:

$$24 + 500 \times 0{,}1 = 74{,}0.$$

- Промежуточный сегмент S4:

$$24 + 500 \times 0{,}1 = 74{,}0.$$

- Промежуточный сегмент S5:

$$33{,}5 + 1000 \times 0{,}1 = 133{,}5.$$

- Правый сегмент S6:

$$165 + 100 \times 0{,}113 = 176{,}3.$$

Сумма всех составляющих дает значение PDV, равное 568,4.

Так как значение PDV меньше максимально допустимой величины 575 на 6,6 битовых интервала, то эта сеть обладает корректной конфигурацией, несмотря на то, что ее общая длина превышает 2500 м, а количество повторителей больше 4-х.

Но проверки PDV еще недостаточно для общего положительного заключения. Нужно также оценить значение уменьшения межкадрового интервала. Исходные данные для этого расчета приведены в табл. 12.5.

Таблица 12.5. Уменьшение межкадрового интервала повторителями

Тип сегмента	Передающий сегмент, битовых интервалов	Промежуточный сегмент, битовых интервалов
10Base-5 или 10Base-2	16	11
10Base-FB	—	2
10Base-FL	10,5	8
10Base-T	10,5	8

В соответствии с этими данными рассчитаем значение уменьшения межкадрового интервала для нашего примера.

- Левый сегмент 1 10Base-T — сокращение в 10,5 битового интервала.
- Промежуточный сегмент 2 10Base-FL — 8.
- Промежуточный сегмент 3 10Base-FB — 2.
- Промежуточный сегмент 4 10Base-FB — 2.
- Промежуточный сегмент 5 10Base-FB — 2.

Сумма этих величин дает значение уменьшения межкадрового интервала, равное 24,5, что меньше предельного значения в 49 битовых интервала. В результате приведенная в примере сеть соответствует стандартам Ethernet по всем параметрам, связанным и с длинами сегментов, и с количеством повторителей.

Выводы

Разделяемые локальные сети представляют собой наиболее простой и дешевый в реализации тип локальных сетей. Основной недостаток разделяемых локальных сетей состоит в плохой масштабируемости, так как при увеличении узлов сети уменьшается доля пропускной способности, приходящаяся на каждый узел.

Комитет IEEE 802 разрабатывает стандарты, которые содержат рекомендации для проектирования нижних уровней локальных сетей — физического и канального. Специфика локальных сетей нашла свое отражение в разделении канального уровня на два подуровня — LLC и MAC.

Уровень MAC отвечает за доступ к разделяемой среде и отправку через нее кадров. В стандарты 802 определяют различные методы доступа, которые делятся на две категории: случайные и детерминированные. Случайные методы доступа обеспечивают минимальную задержку доступа к среде при низкой загрузке среды. Детерминированные методы доступа могут работать при большей загруженности сетей.

Стандарты рабочей группы 802.1 носят общий для всех технологий характер, они определяют типы локальных сетей, их свойства, процедуры межсетевого взаимодействия, логику работы мостов/коммутаторов.

Протокол LLC обеспечивает для протоколов верхних уровней нужное качество транспортных услуг, передавая кадры либо дейтаграммным способом, либо с помощью процедур с установлением соединения и восстановлением кадров.

Ethernet — самая распространенная на сегодняшний день технология локальных сетей. В широком смысле Ethernet — это семейство технологий, в которое входит фирменный стандарт Ethernet DIX, а также стандарты IEEE 802.3 Ethernet 10 Мбит/с, Fast Ethernet, Gigabit Ethernet и 10G Ethernet. Все виды технологий Ethernet, кроме 10G Ethernet, используют один и тот же метод доступа CSMA/CD.

Коллизия — ситуация, когда две станции одновременно пытаются передать кадр данных по общей среде. Наличие коллизий — это неотъемлемое свойство сетей Ethernet, являющееся следствием принятого случайного метода доступа. Возможность четкого распознавания коллизий обусловлена соблюдением соотношения между минимальной длиной кадра и максимально возможным диаметром сети.

Максимально возможная пропускная способность сегмента Ethernet 10 Мбит/с в кадрах в секунду достигается при передаче кадров *минимальной длины* и составляет 14 880 кадр/с. При этом полезная пропускная способность сети составляет всего 5,48 Мбит/с, что лишь ненамного превышает половину номинальной пропускной способности — 10 Мбит/с.

Максимально возможная полезная пропускная способность сети Ethernet при передаче кадров *максимальной длины* в 1518 байт составляет 513 кадр/с. Эти кадры передаются по сети со скоростью 9,75 Мбит/с, которая близка к номинальной.

Технология Ethernet поддерживает 4 разных типа кадров, которые имеют общий формат адресов узлов. Существуют формальные признаки, по которым сетевые адаптеры автоматически распознают тип кадра.

В зависимости от типа физической среды стандарт IEEE 802.3 определяет различные спецификации: 10Base-5, 10Base-2, 10Base-T, FOIRL, 10Base-FL, 10Base-FB. Для каждой спецификации определяются тип кабеля, максимальные длины непрерывных отрезков кабеля, а также правила использования повторителей для увеличения диаметра сети: правило 5-4-3 для коаксиальных вариантов сетей и правило 4-х хабов для витой пары и оптоволокна.

Вопросы и задания

1. Поясните разницу между расширяемостью и масштабируемостью на примере технологии Ethernet.
2. Сравните случайные и детерминированные методы доступа к разделяемой среде.
3. Почему протоколы канального уровня технологий глобальных сетей не делятся на подуровни MAC и LLC?
4. Какие функции выполняет уровень LLC?
5. Что такое коллизия?
6. В чем состоят функции преамбулы и начального ограничителя кадра в стандарте Ethernet?
7. Какие сетевые средства осуществляют jabber-контроль?

8. Зачем в технологии Ethernet введен межпакетный (межкадровый) интервал?
9. Чему равны значения следующих характеристик стандарта 10Base-5:
   - номинальная пропускная способность (бит/с);
   - эффективная пропускная способность (бит/с);
   - пропускная способность (кадр/с);
   - внутрипакетная скорость передачи (бит/с);
   - межбитовый интервал (с).
10. Чем объясняется, что минимальный размер кадра в стандарте 10Base-5 выбран равным 64 байт?
11. Почему стандарты 10Base-T и 10Base-FL/FB вытеснили стандарты Ethernet на коаксиальном кабеле?
12. Поясните смысл каждого поля кадра Ethernet.
13. Как известно, имеются 4 стандарта на формат кадров Ethernet. Выберите из нижеприведенного списка названия для каждого из этих стандартов. Учтите, что некоторые стандарты имеют несколько названий:
   - Novell 802.2;
   - Ethernet II;
   - 802.3/802.2
   - Novell 802.3;
   - Raw 802.3;
   - Ethernet DIX;
   - 802.3/LLC;
   - Ethernet SNAP.
14. Что может произойти в сети, в которой передаются кадры Ethernet разных форматов?
15. Как длина кадра влияет на работу сети? Какие проблемы связаны со слишком длинными кадрами? В чем состоит неэффективность коротких кадров?
16. Как коэффициент использования влияет на производительность сети Ethernet?
17. Как скорость передачи данных технологии Ethernet на разделяемой среде влияет на максимальный диаметр сети?
18. Из каких соображений выбрана максимальная длина физического сегмента в стандартах Ethernet?
19. За счет чего была увеличена максимальная длина сегмента при переходе от стандарта FOIRL к стандарту 10Base-FL?
20. С чем связано ограничение, известное как «правило 4-х хабов»?
21. Почему дуплексный режим Ethernet не поддерживается в концентраторах?
22. Являются ли доменами коллизий показанные на рис. 12.18 фрагменты сети?

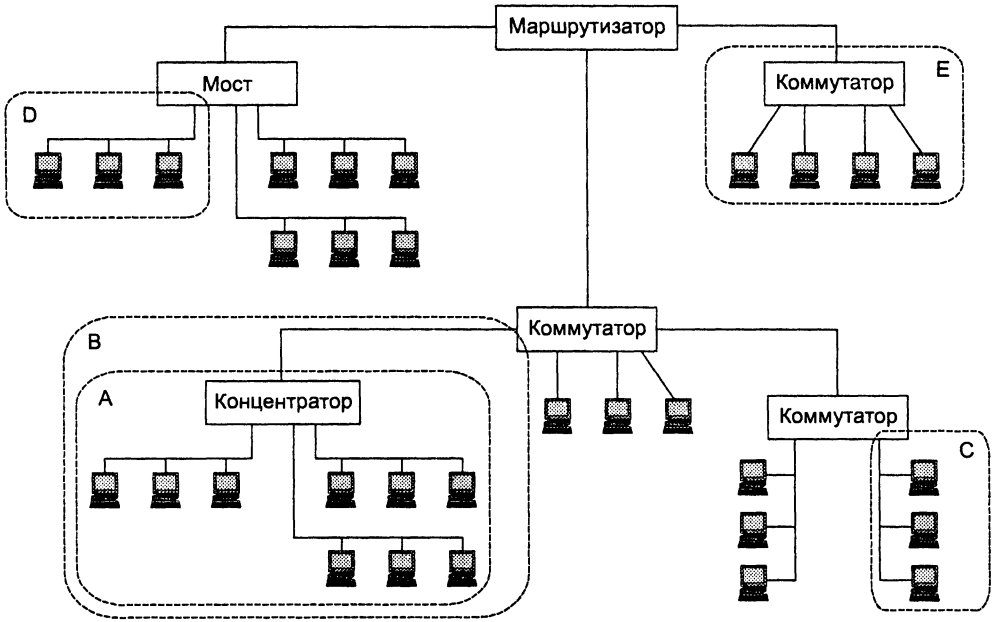


Рис. 12.18. Домены коллизий

23. Какое максимальное время может ожидать станция до того момента, когда ее кадр будет отброшен адаптером?
24. Что произойдет, если в сети, построенной на концентраторах, имеются замкнутые контуры (например, такие, как на рис. 12.19)?

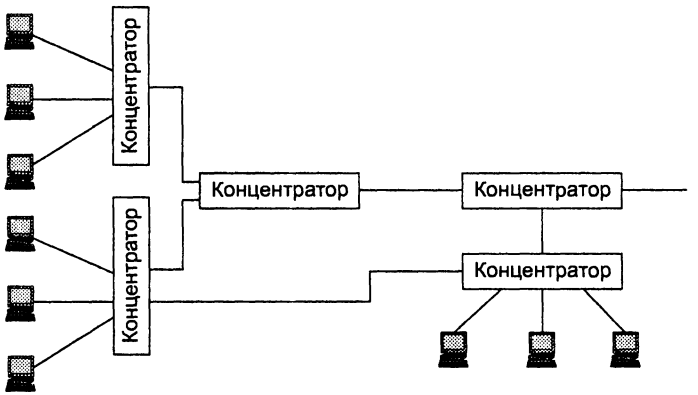


Рис. 12.19. Замкнутый контур в сети Ethernet на концентраторах

25. Оцените снижение производительности сети Ethernet при передаче файла размером в 240 000 байт при повышении уровня потерянных и искаженных кадров с 0 до 3 %. Работу сети иллюстрирует рис. 12.20.

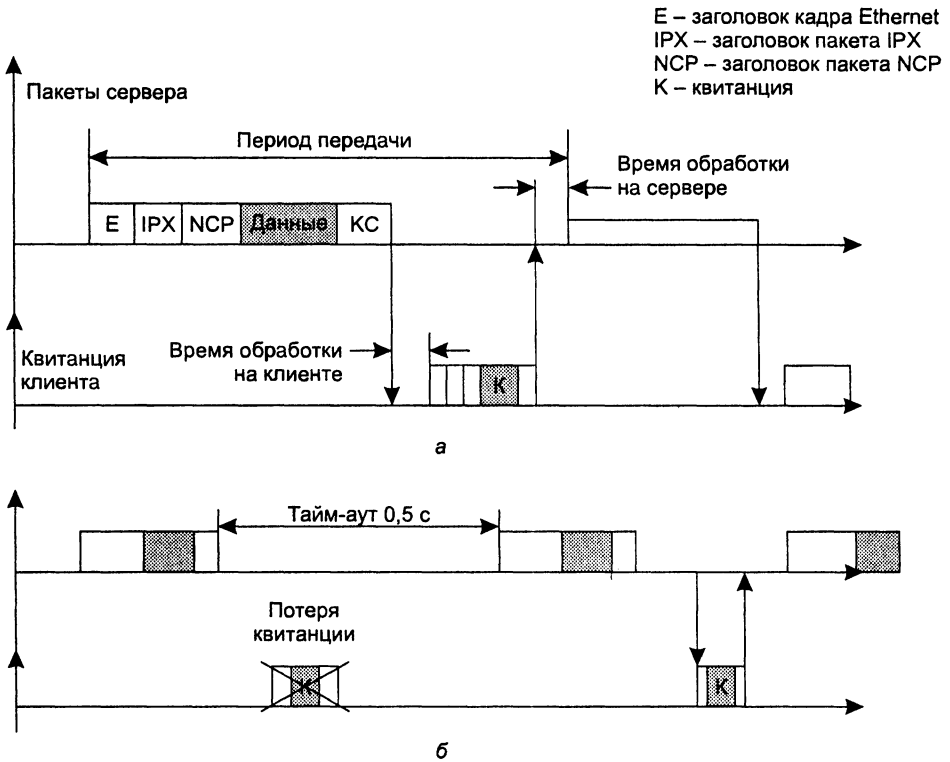


Рис. 12.20. Работа сети Ethernet при передаче файла: *а* — нормальный режим; *б* — потеря пакетов

Файл передается с помощью протоколов: Ethernet, IPX (сетевой уровень) и NCP (прикладной протокол файлового сервиса). Размеры заголовков протоколов:

- Ethernet — 26 байт (с учетом преамбулы и поля FCS);
- IPX — 30 байт;
- NCP — 20 байт.

Файл передается сегментами по 1000 байт. Восстанавливает потерянные и искаженные кадры только протокол NCP, который работает по методу простоя источника. Тайм-аут ожидания положительных квитанций фиксированный и равный 500 мс (это не единственный режим работы NCP, этот протокол может также работать в режиме скользящего окна, но в рассматриваемом случае режим скользящего окна не используется). Размер квитанции равен 10 байтам. Время обработки одного пакета на клиентской стороне составляет 650 мкс, а на сервере — 50 мкс.

Подсказка. Задача состоит из двух частей. Сначала нужно найти фактическую скорость передачи файла при идеальной работе сети, когда процент искаженных и потерянных кадров Ethernet равен нулю. Вторая часть задачи состоит в определении скорости передачи файла уже в условиях, когда по каким-то

причинам кадры начали искажаться и теряться. Выполним первую часть задачи вместе.

Всего для передачи файла потребуется 240 пакетов Размер кадра Ethernet, переносящего 1000 байт передаваемого файла, составит:

$$1000 + 20 + 30 + 26 = 1076 \text{ байт, или } 8608 \text{ бит.}$$

Размер кадра Ethernet, переносящего квитанцию, равен 86 байт (вместе с преамбулой), или 688 бит.

В этих условиях время одного цикла передачи очередной части файла в идеальной сети составит:

$$860{,}8 + 68{,}8 + 650 + 50 = 1629{,}6 \text{ мкс.}$$

При этом общее время передачи файла в 240 000 байт составит:

$$240 \times 1629{,}6 = 0{,}391 \text{ с,}$$

а фактическая скорость передачи данных:

$$240\,000/0{,}391 = 613810 \text{ байт/с, или } 4{,}92 \text{ Мбит/с.}$$

Остается найти скорость передачи данных в условиях потерь и искажений кадров.

ГЛАВА 13 Высокоскоростной стандарт Ethernet

Классическая 10-мегабитная сеть Ethernet устраивала большинство пользователей на протяжении около 15 лет. Однако в начале 90-х годов начала ощущаться ее недостаточная пропускная способность. Скорость обмена с сетью в 10 Мбит/с стала существенно меньше скорости внутренней шины компьютера, которая к тому времени превысила порог 1000 Мбит/с (шина PCI обеспечивала передачу данных со скоростью 133 Мбайт/с). Это приводило к замедлению работы в сети не только серверов, но и рабочих станций, которые также стали использовать шину PCI.

Назрела необходимость в разработке «новой» технологии Ethernet, то есть технологии, которая была бы такой же эффективной по соотношению цена/качество при производительности 100 Мбит/с. В результате поисков и исследований специалисты разделились на два лагеря, что, в конце концов, привело к появлению двух новых технологий — Fast Ethernet и 100VG-AnyLAN. В результате «выжила» только технология Fast Ethernet, сохранившая больше свойств классического стандарта Ethernet, в том числе метод доступа CSMA/CD.

Успех Fast Ethernet еще больше повысил интерес к высокоскоростным вариантам Ethernet. Следующий вариант — Gigabit Ethernet — был стандартизован через три года. Он также отличается высокой степенью преемственности с Ethernet 10 Мбит/с, сохранив возможность работы на разделяемой среде с применением метода CSMA/CD.

Однако последний вариант Ethernet — 10G Ethernet — уже существенно отличается от своего прародителя: в частности, он не поддерживает разделяемую среду.

Поэтому в данной главе мы рассматриваем только технологии Fast Ethernet и Gigabit Ethernet, а технология 10G Ethernet рассматривается в главе 15 вместе с другими технологиями, работающими в дуплексном режиме и позволяющими строить коммутируемые локальные сети.

Технология Fast Ethernet

Список ключевых слов: приоритетный доступ по требованию, спецификации 100Base-TX, 100Base-T4 и 100Base-FX, независимый от среды интерфейс, уровень согласования, устройство физического уровня, схема автопереговоров, импульсы FLP, повторители класса I и класса II, правило одного или двух хабов, технология 100VG-AnyLAN.

История стандартизации

В 1992 году группа производителей сетевого оборудования, включая таких лидеров технологии Ethernet, как SynOptics, 3Com и ряд других, образовала некоммерческое объединение *Fast Ethernet Alliance* для разработки стандарта новой технологии, которая должна была обеспечить резкое повышение производительности при максимально возможном сохранении особенностей технологии Ethernet.

В комитете 802 института IEEE в это же время была сформирована исследовательская группа для изучения технического потенциала новых высокоскоростных технологий. За период с конца 1992 года и по конец 1993 года группа IEEE изучила 100-мегабитные решения, предложенные различными производителями. Наряду с предложениями Fast Ethernet Alliance группа рассмотрела также и высокоскоростную технологию, предложенную компаниями Hewlett-Packard и AT&T.

Обсуждалась проблема сохранения метода случайного доступа CSMA/CD. Предложение Fast Ethernet Alliance сохраняло этот метод и тем самым обеспечивало преемственность и согласованность сетей 10 Мбит/с и 100 Мбит/с. Коалиция HP и AT&T, которая имела поддержку значительно меньшего числа производителей в сетевой индустрии, чем Fast Ethernet Alliance, предложила совершенно новый метод доступа, названный **приоритетным доступом по требованию** (demand priority). Он существенно менял картину поведения узлов в сети, поэтому не смог вписаться в технологию Ethernet и стандарт 802.3, и для его стандартизации был организован новый комитет IEEE 802.12.

Осенью 1995 года обе технологии стали стандартами IEEE. Комитет IEEE 802.3 принял спецификацию Fast Ethernet в качестве стандарта 802.3u, который не является самостоятельным стандартом, а представляет собой дополнение к существующему стандарту 802.3 в виде глав с 21 по 30. Комитет 802.12 принял технологию 100VG-AnyLAN, которая использовала принципиально новый метод доступа Demand Priority и поддерживала кадры двух форматов — Ethernet и Token Ring.

Физический уровень технологии Fast Ethernet

Все отличия технологий Fast Ethernet и Ethernet сосредоточены на физическом уровне (рис. 13.1). Уровни MAC и LLC в Fast Ethernet остались абсолютно теми же, и их описывают прежние главы стандартов 802.3 и 802.2. Поэтому, рассматривая технологию Fast Ethernet, мы будем изучать только несколько вариантов ее физического уровня.

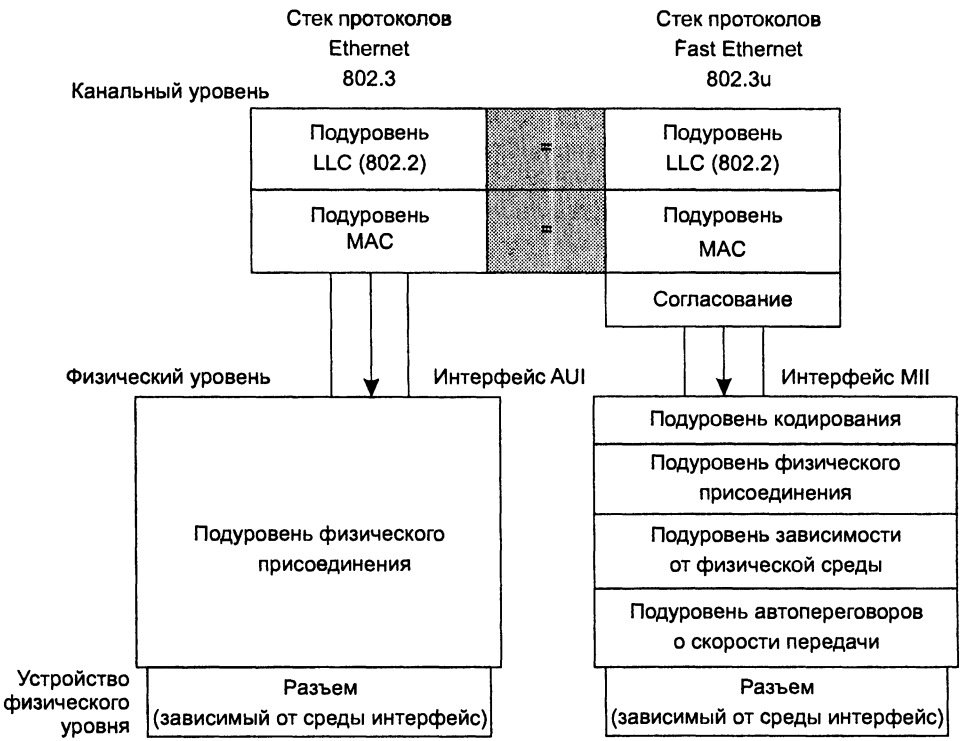


Рис. 13.1. Отличия технологий Fast Ethernet и Ethernet

Организация физического уровня технологии Fast Ethernet является более сложной, поскольку в ней используются *три* варианта кабельных систем:

- волоконно-оптический многомодовый кабель (два волокна);
- витая пара категории 5 (две пары);
- витая пара категории 3 (четыре пары).

Коаксиальный кабель, давший миру первую сеть Ethernet, в число разрешенных сред передачи данных новой технологии Fast Ethernet не попал. Это общая тенденция многих новых технологий, поскольку на небольших расстояниях витая пара категории 5 позволяет передавать данные с той же скоростью, что и коаксиальный кабель, но сеть получается более дешевой и удобной в эксплуатации. На больших расстояниях оптическое волокно обладает гораздо более широкой полосой пропускания, чем коаксиал, а стоимость сети получается ненамного выше, особенно если учесть высокие затраты на поиск и устранение неисправностей в крупной кабельной коаксиальной системе.

Сети Fast Ethernet на разделяемой среде подобно сетям 10Base-T/10Base-F имеют иерархическую древовидную структуру, построенную на концентраторах. Основным отличием конфигураций сетей Fast Ethernet является сокращение диаметра сети примерно до 200 м, что объясняется сокращением времени передачи

кадра минимальной длины в 10 раз за счет увеличения скорости передачи в 10 раз по сравнению с 10-мегабитной сетью Ethernet.

Тем не менее это обстоятельство не очень препятствует построению крупных сетей на технологии Fast Ethernet. Дело в том, что середина 90-х годов отмечена не только широким распространением недорогих высокоскоростных технологий, но и бурным развитием локальных сетей на основе *коммутаторов*. При использовании коммутаторов протокол Fast Ethernet может работать в дуплексном режиме, в котором нет ограничений на общую длину сети, а остаются только ограничения на длину физических сегментов, соединяющих соседние устройства (адаптер-коммутатор и коммутатор-коммутатор). В данном разделе рассматривается классический полудуплексный вариант работы технологии Fast Ethernet. Особенности дуплексного режима Fast Ethernet описаны в главе 15.

Физические варианты Fast Ethernet отличаются друг от друга в большей степени нежели варианты физической реализации Ethernet. Здесь меняется как количество проводников, так и методы кодирования. А так как физические варианты Fast Ethernet создавались одновременно, а не эволюционно, как для сетей Ethernet, то имелась возможность детально определить те подуровни физического уровня, которые не изменяются от варианта к варианту, и те подуровни, которые специфичны для каждого варианта физической среды.

Официальный стандарт 802.3 установил три различных спецификации для физического уровня Fast Ethernet и дал им следующие названия (рис. 13.2):

- **100Base-TX** для двухпарного кабеля на неэкранированной витой паре UTP категории 5 или экранированной витой паре STP типа 1;
- **100Base-T4** для четырехпарного кабеля на неэкранированной витой паре UTP категории 3, 4 или 5;
- **100Base-FX** для многомодового оптоволоконного кабеля с двумя волокнами.

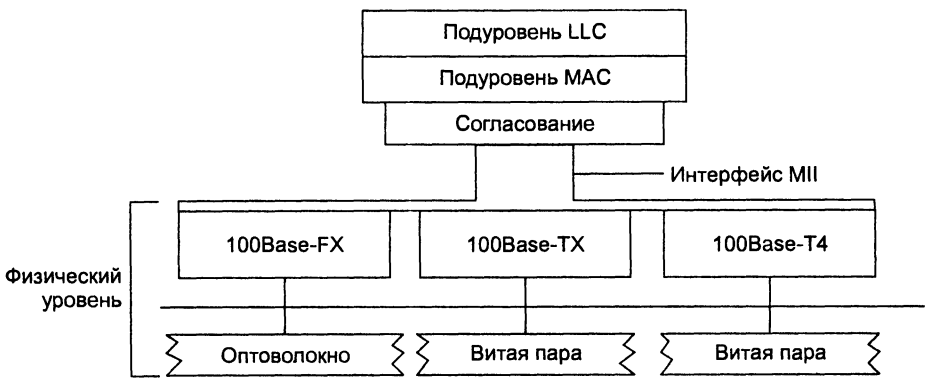


Рис. 13.2. Структура физического уровня Fast Ethernet

Для всех трех стандартов справедливы перечисленные ниже утверждения и характеристики.

Форматы кадров технологии Fast Ethernet не отличаются от форматов кадров технологий 10-мегабитной сети Ethernet.

Межкадровый интервал равен 0,96 мкс, а битовый интервал — 10 нс. Все временные параметры алгоритма доступа (интервал отсрочки, время передачи кадра минимальной длины и т. п.), измеренные в битовых интервалах, остались прежними.

Признаком свободного состояния среды является передача по ней символа простоя источника — соответствующего избыточного кода (а не отсутствие сигналов, как в стандартах Ethernet 10 Мбит/с).

Физический уровень включает три элемента.

- **Независимый от среды интерфейс** (Media Independent Interface, MII).
- **Уровень согласования** нужен для того, чтобы уровень MAC, рассчитанный на интерфейс AUI , мог работать с физическим уровнем через интерфейс MII.
- **Устройство физического уровня** (Physical Layer Device, PHY) состоит, в свою очередь, из нескольких подуровней (см. рис. 13.1):
  - подуровня логического кодирования данных, преобразующего поступающие от уровня MAC байты в символы кода 4B/5B или 8B/6T (оба кода используются в технологии Fast Ethernet);
  - подуровней физического присоединения и зависимости от физической среды (PMD), которые обеспечивают формирование сигналов в соответствии с методом физического кодирования, например NRZI или MLT-3;
  - подуровня автопереговоров, который позволяет двум взаимодействующим портам автоматически выбрать наиболее эффективный режим работы, например полудуплексный или дуплексный (этот подуровень является факультативным).

Интерфейс MII поддерживает независимый от физической среды способ обмена данными между подуровнем MAC и подуровнем PHY. Этот интерфейс аналогичен по назначению интерфейсу AUI классического стандарта Ethernet за исключением того, что интерфейс AUI располагался между подуровнем физического кодирования сигнала (для любых вариантов кабеля использовался одинаковый метод физического кодирования — манчестерский код) и подуровнем физического присоединения к среде, а интерфейс MII располагается между подуровнем MAC и подуровнями кодирования сигнала, которых в стандарте Fast Ethernet три — FX, TX и T4.

Спецификации 100Base-FX/TX/T4

Между спецификациями 100Base-FX, 100Base-TX и 100Base-T4 есть много общего, поэтому одинаковые для спецификаций свойства будут даваться под обобщенным названием, например 100Base-FX/TX или 100Base-TX/T4.

Спецификация 100Base-FX определяет работу протокола Fast Ethernet по многомодовому оптоволокну в полудуплексном и дуплексном режимах. В то время как в Ethernet со скоростью передачи 10 Мбит/с используется манчестерское кодирование для представления данных, в стандарте Fast Ethernet определен другой метод кодирования — 4B/5B. Мы рассмотрели детали кодирования 4B/5B в главе 9. Этот метод к моменту разработки технологии Fast Ethernet уже показал

свою эффективность в сетях FDDI, поэтому он без изменений был перенесен в спецификацию 100Base-FX/TX. Напомним, что в этом методе каждые 4 бита данных подуровня MAC (называемых символами) представляются 5 битами. Избыточный бит позволяет применить потенциальные коды при представлении каждого из пяти битов в виде электрических или оптических импульсов.

Существование запрещенных комбинаций символов позволяет отбраковывать ошибочные символы, что повышает устойчивость работы сетей с 100Base-FX/TX. Так, в Fast Ethernet признаком того, что среда свободна, стала повторяющаяся передача одного из запрещенных для кодирования пользовательских данных символа, а именно символа простоя источника *Idle (11111)*. Такой способ позволяет приемнику всегда находиться в синхронизме с передатчиком.

Для отделения кадра Ethernet от символов простоя источника используется комбинация символов начального ограничителя кадра — пара символов *J (11000)* и *K (10001)* кода 4B/5B, а после завершения кадра перед первым символом простоя источника вставляется символ T (рис. 13.3).

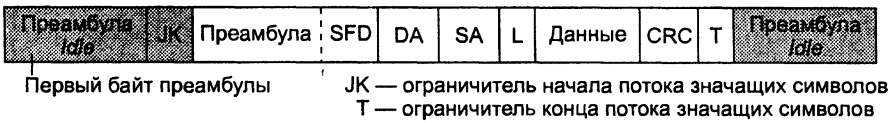


Рис. 13.3. Непрерывный поток данных спецификаций 100Base-FX/TX

После преобразования 4-битовых порций кодов MAC в 5-битовые порции физического уровня их необходимо представить в виде оптических или электрических сигналов в кабеле, соединяющем узлы сети. В спецификациях 100Base-FX и 100Base-TX для этого используются различные методы физического кодирования — NRZI и MLT-3 соответственно.

В спецификации 100Base-TX в качестве среды передачи данных используется витая пара UTP категории 5 или STP типа 1. Основным отличием от спецификации 100Base-FX — наряду с использованием метода кодирования MLT-3 — является наличие функции автопереговоров для выбора режима работы порта.

Схема автопереговоров позволяет двум физически соединенным устройствам, которые поддерживают несколько стандартов физического уровня, отличающихся битовой скоростью и количеством витых пар, согласовать наиболее выгодный режим работы. Обычно процедура автопереговоров происходит при подсоединении сетевого адаптера, который может работать на скоростях 10 и 100 Мбит/с, к концентратору или коммутатору.

Всего в настоящее время определено 5 различных режимов работы, которые могут поддерживать устройства 100Base-TX/T4 на витых парах:

- 10Base-T;
- дуплексный режим 10Base-T;
- 100Base-TX;
- 100Base-T4;
- дуплексный режим 100Base-TX.

Режим 10Base-T имеет самый низкий приоритет при переговорном процессе, а дуплексный режим 100Base-T4 — самый высокий.

Переговорный процесс происходит при включении питания устройства, а также может быть инициирован в любой момент модулем управления устройства. Устройство, начавшее процесс автопереговоров, посылает своему партнеру пачку специальных импульсов **FLP** (Fast Link Pulse), в которой содержится 8-битное слово, кодирующее предлагаемый режим взаимодействия, начиная с самого приоритетного, поддерживаемого данным узлом.

Если узел-партнер имеет функцию автопереговоров и также может поддерживать предложенный режим, он отвечает пачкой импульсов FLP, в которой подтверждает данный режим, и на этом переговоры заканчиваются. Если же узел-партнер не может поддерживать запрошенный режим, то он указывает в своем ответе имеющийся в его распоряжении следующий по степени приоритетности режим, и этот режим выбирается в качестве рабочего.

Спецификация 100Base-T4 (витая пара UTP категории 3, четыре пары) появилась позже других спецификаций физического уровня Fast Ethernet. Разработчики технологий 100Base-TX/FX, появившихся ранее, в первую очередь хотели создать физические спецификации, наиболее близкие к спецификациям 10Base-T и 10Base-F, которые работали на двух линиях передачи данных: двух парах или двух волокнах. Для реализации работы по двум витым парам пришлось перейти на более качественный кабель категории 5.

В то же время разработчики конкурирующей технологии 100VG-AnyLAN изначально сделали ставку на работу по витой паре категории 3; самое главное преимущество состояло не столько в стоимости, а в том, что она была уже проложена в подавляющем числе зданий. Конкуренция заставила разработчиков технологии Fast Ethernet также создать свой вариант физического уровня для витой пары категории 3, которым и стала спецификация 100Base-T4.

Вместо кодирования 4B/5B в этом методе используется кодирование 8B/6T, которое обладает более узким спектром сигнала и при скорости 33 Мбит/с укладывается в полосу 16 МГц витой пары категории 3 (при кодировании 4B/5B спектр сигнала в эту полосу не укладывается). Каждые 8 бит информации уровня MAC кодируются 6-ю троичными цифрами, то есть цифрами, имеющими три состояния. Каждая троичная цифра имеет длительность 40 нс. Группа из 6-ти троичных цифр затем передается на одну из трех передающих витых пар независимо и последовательно.

Четвертая пара всегда используется для прослушивания несущей частоты в целях обнаружения коллизий. Скорость передачи данных по каждой из трех передающих пар равна 33,3 Мбит/с, поэтому общая скорость протокола 100Base-T4 составляет 100 Мбит/с. В то же время из-за принятого способа кодирования скорость изменения сигнала на каждой паре равна всего 25 Мбод, что, собственно, и позволяет использовать витую пару категории 3.

На рис. 13.4 показано соединение порта MDI (Media Dependant Interface) сетевого адаптера 100Base-T4 с портом MDI-X концентратора (приставка X говорит о том, что у этого разъема присоединения приемника и передатчика меняются парами кабели по сравнению с разъемом сетевого адаптера, что позволяет проще соединять пары проводов в кабеле — без перекрещивания). Пара 1–2 всегда тре-

буется для передачи данных от порта MDI к порту MDI-X, пара 3–6 — для приема данных портом MDI от порта MDI-X, а пары 4–5 и 7–8 являются двунаправленными и используются как для приема, так и для передачи, в зависимости от потребности.

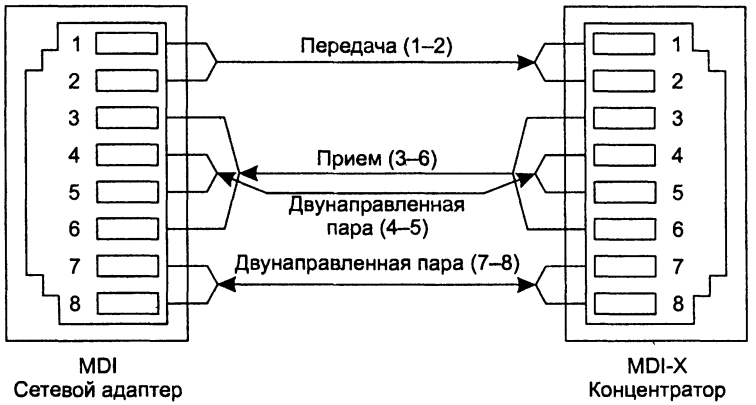


Рис. 13.4. Соединение узлов по спецификации 100Base-T4

Правила построения сегментов Fast Ethernet при наличии повторителей

Технология Fast Ethernet, как и все варианты Ethernet, рассчитана на использование концентраторов-повторителей для образования связей в сети. Правила корректного построения сегментов сетей Fast Ethernet включают:

- ограничения на максимальную длину сегментов между устройствами — источниками новых кадров;
- ограничения на максимальную длину сегментов, соединяющих устройство — источник новых кадров с портом повторителя;
- ограничения на максимальный диаметр сети;
- ограничения на максимальное число повторителей и максимальную длину сегмента, соединяющего повторители.

Ограничения длины сегментов между устройствами — источниками новых кадров. В качестве таких устройств могут выступать сетевой адаптер, порт моста, порт маршрутизатора, модуль управления сетью и другие подобные устройства. Можно возразить, что мост или коммутатор передает через выходной порт существовавшие ранее кадры, выработанные в свое время сетевым адаптером, но для сегмента сети, к которому подключен выходной порт устройства, этот кадр является новым. А вот порт повторителя не является источником новых кадров, так как он побитно повторяет уже появившийся в сегменте кадр.

Рассматриваемое ограничение касается случаев, когда сетевой адаптер непосредственно соединяется с портом моста/коммутатора или маршрутизатора, а также когда порты мостов, коммутаторов и маршрутизаторов соединяются между собой. Максимальные значения длины сегментов в соответствие со спецификацией IEEE 802.3u приведены в табл. 13.1.

Таблица 13.1. Максимальные значения длины сегментов между устройствами — источниками новых кадров

Стандарт	Тип кабеля	Максимальная длина сегмента
100Base-TX	Категория 5 UTP	100 м
100Base-FX	Многомодовое оптоволокно 62,5/125 мкм	412 м (полудуплекс), 2 км (дуплекс)
100Base-T4	Категория 3, 4 или 5 UTP	100 м

Ограничения сетей Fast Ethernet на повторителях. Повторители Fast Ethernet делят на два класса.

- **Повторители класса I** поддерживают оба типа кодирования данных: как 4B/5B, так и 8B/6T. Они способны выполнять трансляцию логических кодов с битовой скоростью 100 Мбит/с. Поэтому повторители класса I могут иметь порты всех трех типов физического уровня: 100Base-TX, 100Base-FX и 100Base-T4.
- **Повторители класса II** поддерживают только какой-либо один тип кодирования — либо 4B/5B, либо 8B/6T. Поэтому они имеют либо все порты 100Base-T4, либо порты 100Base-TX и 100Base-FX, так как последние используют один логический код 4B/5B.

В одном домене коллизий допускается наличие только одного повторителя класса I. Это связано с тем, что такой повторитель вносит большую задержку — 70 битовых интервалов — из-за необходимости трансляции различных систем кодирования.

Повторители класса II вносят меньшую задержку при передаче сигналов: 46 битовых интервалов для портов TX/FX и 33,5 битовых интервала для портов T4. Поэтому максимальное число повторителей класса II в домене коллизий — 2.

Небольшое количество повторителей Fast Ethernet не является серьезным препятствием при построении больших сетей, так как применение коммутаторов и маршрутизаторов делит сеть на несколько доменов коллизий, каждый из которых может строиться на одном или двух повторителях. В этом случае никаких ограничений на общую длину сети нет.

В табл. 13.2 приведены правила построения сети на основе повторителей класса I.

Таблица 13.2. Параметры сетей на основе повторителей класса I

Тип кабелей	Максимальный диаметр сети, м	Максимальная длина сегмента, м
Только витая пара (TX)	200	100
Только оптоволокно (FX)	272	136
Несколько сегментов на витой паре и один на оптоволокне	260	100 (TX) 160 (FX)
Несколько сегментов на витой паре и несколько сегментов на оптоволокне	272	100 (TX) 136 (FX)

Эти ограничения проиллюстрированы типовыми конфигурациями сетей, показанными на рис. 13.5.

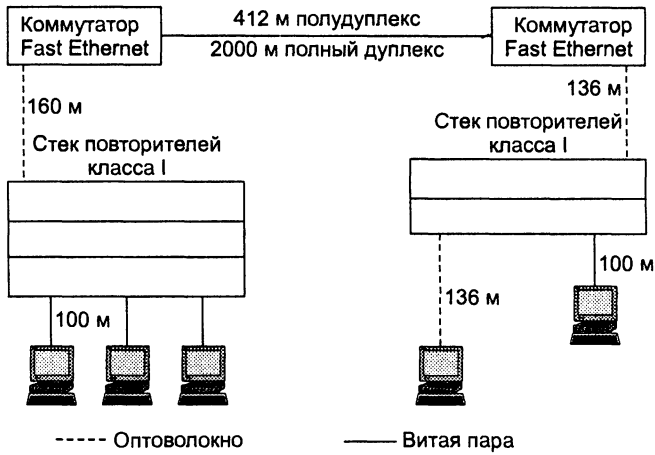


Рис. 13.5. Примеры построения сети Fast Ethernet с помощью повторителей класса I

Таким образом, правило 4-х хабов превратилось для технологии Fast Ethernet в *правило одного или двух хабов*, в зависимости от класса хаба.

При определении корректности конфигурации сети можно не руководствоваться правилами одного или двух хабов, а рассчитывать время оборота сети, как это было показано для сети Ethernet 10 Мбит/с. Как и для технологии Ethernet 10 Мбит/с, стандарт 802.3 дает исходные данные для расчета времени оборота сигнала.

Особенности технологии 100VG-AnyLAN

Несмотря на множество интересных технических решений, технология **100VG-AnyLAN** не имела большого количества сторонников и прекратила свое существование. Она не нашла своей ниши, оказавшись слишком сложной по сравнению с более традиционной и привычной технологией Fast Ethernet.

Технология 100VG-AnyLAN отличается от классической технологии Ethernet в значительно большей степени, чем Fast Ethernet.

Доступ к разделяемой среде осуществляется по принципиально иному методу — методу **приоритетного доступа по требованию**. Он основан на передаче концентратору функций арбитра, решающего проблему доступа к разделяемой среде. Сеть 100VG-AnyLAN состоит из центрального концентратора, называемого также корневым, и соединенных с ним конечных узлов и других концентраторов (рис. 13.6).

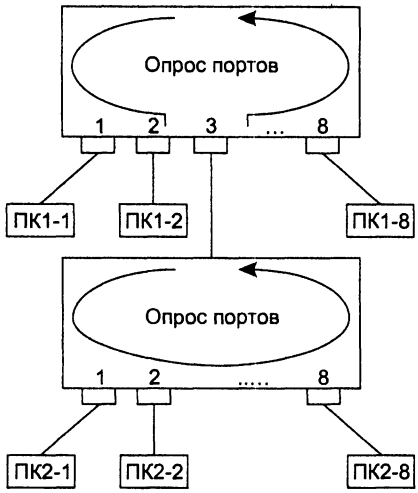


Рис. 13.6. Сеть 100VG-AnyLAN

Допускаются три уровня каскадирования. Каждый концентратор и сетевой адаптер 100VG-AnyLAN должен быть настроен на работу либо с кадрами Ethernet, либо с кадрами Token Ring, причем одновременно циркуляция обоих типов кадров не допускается.

Концентратор циклически выполняет опрос портов. Станция, желающая передать пакет, посылает специальный низкочастотный сигнал концентратору, запрашивая передачу кадра и указывая его приоритет. В сети 100VG-AnyLAN используются два уровня приоритетов — низкий и высокий. Низкий приоритет соответствует обычным данным (файловая служба, служба печати и т. п.), а высокий приоритет — данным, чувствительным к временны́м задержкам (например, мультимедиа). Приоритеты запросов имеют статическую и динамическую составляющие, то есть станция с низким уровнем приоритета, долго не имеющая доступа к сети, получает высокий приоритет.

Если сеть свободна, то концентратор разрешает передачу пакета. После анализа адреса получателя в принятом пакете концентратор автоматически отправляет пакет станции назначения. Если сеть занята, концентратор ставит полученный запрос в очередь, которая обрабатывается в соответствии с порядком поступления запросов и учетом приоритетов. Если к порту подключен другой концентратор, то опрос приостанавливается до завершения опроса концентратором нижнего уровня. Станции, подключенные к концентраторам различного уровня иерархии, не имеют преимуществ по доступу к разделяемой среде, так как решение о предоставлении доступа принимается после проведения опроса всеми концентраторами опроса всех своих портов.

Остается неясным вопрос — каким образом концентратор узнает, к какому порту подключена станция назначения? Во всех других технологиях кадр просто передавался всем станциям сети, а станция назначения, распознав свой адрес, копировала кадр в буфер. Для решения этой задачи концентратор узнает MAC-адрес станции в момент физического присоединения ее к сети кабелем. Если в других

технологиях процедура физического соединения выясняет связность кабеля (тест связности в технологии 10Base-T) или скорость работы порта (процедура автопереговоров в Fast Ethernet), то в технологии 100VG-AnyLAN концентратор при установлении физического соединения выясняет MAC-адрес станции и запоминает его в таблице MAC-адресов, аналогичной таблице моста/коммутатора.

Отличие концентратора 100VG-AnyLAN от моста/коммутатора в том, что у него нет внутреннего буфера для хранения кадров. Поэтому он принимает от станций сети только один кадр, отправляет его в порт назначения и пока этот кадр не будет полностью принят станцией назначения, новые кадры концентратор не принимает. Так что эффект разделяемой среды сохраняется. Улучшается только безопасность сети — кадры не попадают в чужие порты и их труднее перехватить.

Технология Gigabit Ethernet

Список ключевых слов: расширение, режим пульсаций, длина пульсации, группа 802.3z, Gigabit Ethernet, Fibre Channel, SDH, метод кодирования 8B/10B, витая пара категории 5, оптический кабель, диаметр сети, разделяемая среда, удлиненный кадр.

История

Достаточно быстро после появления на рынке продуктов Fast Ethernet сетевые интеграторы и администраторы при построении корпоративных сетей почувствовали определенные ограничения. Во многих случаях серверы, подключенные по 100-мегабитному каналу, перегружали магистрали сетей, работающие также на скорости 100 Мбит/с — магистрали FDDI и Fast Ethernet. Ощущалась потребность в следующем уровне иерархии скоростей. В 1995 году более высокий уровень скорости могли предоставить только коммутаторы ATM, которые из-за высокой стоимости, а также больших отличий от классических технологий применялись в локальных сетях достаточно редко.

Поэтому логичным выглядел следующий шаг, сделанный IEEE. Летом 1996 года было объявлено о создании группы 802.3z для разработки протокола, в максимальной степени подобного Ethernet, но с битовой скоростью 1000 Мбит/с. Как и в случае Fast Ethernet, сообщение было воспринято сторонниками Ethernet с большим энтузиазмом.

Основной причиной энтузиазма была перспектива плавного перевода сетевых магистралей на Gigabit Ethernet, подобно тому, как были переведены на Fast Ethernet перегруженные сегменты Ethernet, расположенные на нижних уровнях иерархии сети. К тому же опыт передачи данных на гигабитных скоростях уже имелся. В территориальных сетях такую скорость обеспечивала технология SDH, а в локальных — технология Fibre Channel. Последняя технология используется в основном для подключения высокоскоростной периферии к крупным компью-

терам и передает данные по волоконно-оптическому кабелю со скоростью, близкой к гигабитной. (Именно метод кодирования 8B/10B, используемый в технологии Fiber Channel, был принят в качестве первого варианта физического уровня Gigabit Ethernet.)

Стандарт 802.3z был окончательно принят в 1998 году. Работы по реализации Gigabit Ethernet на витой паре категории 5 были переданы проблемной группе 802.3ab ввиду сложности обеспечения гигабитной скорости на этом типе кабеля, который был создан для поддержки скоростей 100 Мбит/с. Проблемная группа 802.3ab успешно справилась со своей задачей, и версия Gigabit Ethernet для витой пары категории 5 была также принята.

Проблемы

Основная идея разработчиков стандарта Gigabit Ethernet состояла в максимальном сохранении идей классической технологии Ethernet при достижении битовой скорости в 1000 Мбит/с.

Так как при разработке новой технологии естественно ожидать некоторых технических новинок, идущих в общем русле развития сетевых технологий, то важно отметить, что стандарт Gigabit Ethernet, так же как и его менее скоростные собратья, на уровне протокола *не* поддерживает:

- качество обслуживания;
- избыточные связи;
- тестирование работоспособности узлов и оборудования (за исключением тестирования связи порт-порт, как это делается в Ethernet 10Base-T, 10Base-F и Fast Ethernet).

Все три названных свойства считаются весьма перспективными и полезными в современных сетях, а особенно в сетях ближайшего будущего. Почему же авторы Gigabit Ethernet отказались от них?

Ответ состоит в том, что этими полезными свойствами в локальных сетях сегодня обладают коммутаторы, в которых работают дуплексные версии протоколов семейства Ethernet. Поэтому разработчики технологии решили, что базовый протокол просто должен быстро передавать данные, а более сложные и не всегда востребованные функции (например, поддержка параметров QoS) должны быть переданы протоколам верхних уровней, которые работают в коммутаторах.

Что же общего имеется в технологии Gigabit Ethernet по сравнению с технологиями Ethernet и Fast Ethernet?

Сохраняются все форматы кадров Ethernet.

По-прежнему существует полудуплексная версия протокола, поддерживающая метод доступа CSMA/CD. Сохранение недорогого решения на основе разделяемой среды позволяет применять Gigabit Ethernet в небольших рабочих группах, имеющих быстрые серверы и рабочие станции.

Поддерживаются все основные виды кабелей, используемых в Ethernet и Fast Ethernet: волоконно-оптический, витая пара категории 5, экранированная витая пара.

Несмотря на то что в Gigabit Ethernet не стали встраиваться новые функции, обеспечение даже достаточно простых функций классического стандарта Ethernet на скорости 1 Гбит/с потребовало решения нескольких сложных задач.

- *Обеспечение приемлемого диаметра сети для работы на разделяемой среде.* В связи с ограничениями, накладываемыми методом CSMA/CD на длину кабеля, версия Gigabit Ethernet для разделяемой среды допускала бы длину сегмента всего в 25 м при сохранении размера кадров и всех параметров метода CSMA/CD неизменными. Так как существует большое количество применений, требующих диаметра сети хотя бы 200 м, необходимо было каким-то образом решить эту задачу за счет минимальных изменений в технологии Fast Ethernet.
- *Достижение битовой скорости 1000 Мбит/с на оптическом кабеле.* Технология Fibre Channel, физический уровень которой был взят за основу для оптоволоконной версии Gigabit Ethernet, обеспечивает скорость передачи данных всего в 800 Мбит/с.
- *Использование в качестве кабеля витой пары.* Такая задача на первый взгляд кажется неразрешимой — ведь даже для 100-мегабитных протоколов требуются достаточно сложные методы кодирования, чтобы уложить спектр сигнала в полосу пропускания кабеля.

Для решения этих задач разработчикам технологии Gigabit Ethernet пришлось внести изменения не только в физический уровень, как это было в случае Fast Ethernet, но и в уровень MAC.

Средства обеспечения диаметра сети в 200 м на разделяемой среде

Для расширения максимального диаметра сети Gigabit Ethernet до 200 м в полудуплексном режиме разработчики технологии предприняли достаточно естественные меры. Они основаны на известном соотношении времени передачи кадра минимальной длины и времени оборота (PDV).

Минимальный размер кадра был увеличен (без учета преамбулы) с 64 до 512 байт или до 4096 бит. Соответственно, время оборота также можно было увеличить до 4095 битовых интервалов, что делает допустимым диаметр сети около 200 м при использовании одного повторителя.

Для увеличения длины кадра до требуемой в новой технологии величины сетевой адаптер должен дополнить поле данных до длины 448 байт так называемым **расширением**, представляющим собой поле, заполненное нулями. Формально минимальный размер кадра не изменился, он по-прежнему равен 64 байт или 512 бит, но это объясняется тем, что поле расширения помещается после поля контрольной суммы кадра FCS. Соответственно, значение этого поля не включается в контрольную сумму и не учитывается при указании длины поля данных в поле длины. Поле расширения является просто расширением сигнала несущей частоты, необходимым для корректного обнаружения коллизий.

Для сокращения накладных расходов при использовании слишком длинных кадров для передачи коротких квитанций разработчики стандарта разрешили конечным узлам *передавать несколько кадров подряд без передачи среды* другим станциям. Такой режим получил название **режима пульсаций**. Станция может передать подряд несколько кадров с общей длиной не более 65 536 бит или 8192 байт. Если станции нужно передать несколько небольших кадров, то она может не дополнять первый кадр до размера в 512 байт за счет поля расширения, а передавать несколько кадров подряд до исчерпания предела в 8192 байт (в этот предел входят все байты кадра, в том числе преамбула, заголовок, данные и контрольная сумма). Предел 8192 байт называется **длиной пульсации**. Если станция начала передавать кадр и предел длины пульсации достигнут в середине кадра, то кадр разрешается передать до конца. Увеличение «совмещенного» кадра до 8192 байт несколько задерживает доступ к разделяемой среде других станций, но при скорости 1000 Мбит/с эта задержка не столь существенна.

Спецификации физической среды стандарта 802.3z

В стандарте 802.3z определены следующие типы физической среды:

- одномодовый волоконно-оптический кабель;
- многомодовый волоконно-оптический кабель 62,5/125;
- многомодовый волоконно-оптический кабель 50/125;
- экранированный сбалансированный медный кабель.

Для передачи данных по традиционному для компьютерных сетей многомодовому волоконно-оптическому кабелю стандарт определяет применение излучателей, работающих на двух длинах волн: 1300 и 850 нм. Применение светодиодов с длиной волны 850 нм объясняется тем, что они намного дешевле, чем светодиоды, работающие на волне 1300 нм, хотя при этом максимальная длина кабеля уменьшается, так как затухание многомодового оптоволокна на волне 850 м более чем в два раза выше, чем на волне 1300 нм. Однако возможность удешевления чрезвычайно важна для такой в целом дорогой технологии, как Gigabit Ethernet.

Для многомодового оптоволокна стандарт 802.3z определяет спецификации 1000Base-SX и 1000Base-LX. В первом случае используется длина волны 850 нм (S означает Short Wavelength), а во втором — 1300 нм (L — Long Wavelength). Спецификация 1000Base-SX может использовать только многомодовый кабель, при этом его максимальная длина составляет около 500 м.

Для спецификации 1000Base-LX в качестве источника излучения всегда применяется полупроводниковый лазерный диод с длиной волны 1300 нм. Спецификация 1000Base-LX может работать как с многомодовым (максимальное расстояние до 500 м), так и с одномодовым кабелем (максимальное расстояние зависит от мощности передатчика и качества кабеля и может доходить до нескольких десятков километров).

В качестве среды передачи данных в спецификации 1000-CX определен экранированный сбалансированный медный кабель с волновым сопротивлением 150 Ом. Максимальная длина сегмента составляет всего 25 м, поэтому это решение подходит для соединения оборудования, расположенного в одной комнате.

Gigabit Ethernet на витой паре категории 5

Как известно, каждая пара кабеля категории 5 имеет гарантированную полосу пропускания до 100 МГц. Для передачи по такому кабелю данных со скоростью 1000 Мбит/с было решено организовать параллельную передачу одновременно по всем 4 парам кабеля.

Это сразу снизило скорость передачи данных по каждой паре до 250 Мбит/с. Однако и для такой скорости необходимо было придумать метод кодирования, который имел бы спектр не выше 100 МГц. Например, применение кода 4B/5B не может решить поставленную задачу, так как основной вклад в спектр сигнала на такой скорости у него вносит частота 155 МГц. Кроме того, не нужно забывать, что каждая новая технология должна поддерживать не только классический полудуплексный режим, рассматриваемый в этой главе, но и дуплексный режим, о котором мы будем подробно говорить в главе 15. На первый взгляд кажется, что одновременное использование четырех пар лишает сеть возможность работы в дуплексном режиме, так как не остается свободных пар для одновременной передачи данных в двух направлениях — от узла и к узлу.

На оба эти вопроса проблемная группа 802.3ab нашла ответы.

Для кодирования данных был применен код PAM5, в котором 5 уровней потенциала: −2, −1, 0, +1, +2. Поэтому за один такт по одной паре передается 2,322 бит информации ($\log_2 5$). Следовательно, для достижения скорости 250 Мбит/с тактовую частоту 250 МГц можно уменьшить в 2,322 раза. Разработчики стандарта решили использовать несколько более высокую частоту, а именно 125 МГц. При этой тактовой частоте код PAM5 имеет спектр уже, чем 100 МГц, то есть он может быть передан без искажений по кабелю категории 5.

В каждом такте передается не 2,322 × 4 = 9,288 бит информации, а 8. Это и дает искомую суммарную скорость 1000 Мбит/с. Передача ровно 8 бит в каждом такте достигается за счет того, что при кодировании информации используются не все 625 ($5^4 = 625$) комбинаций кода PAM5, а только 256 ($2^8 = 256$). Оставшиеся комбинации приемник использует для контроля принимаемой информации и выделения правильных комбинаций на фоне шума.

Для организации дуплексного режима разработчики спецификации 802.3ab применили технику выделения принимаемого сигнала из суммарного. Два передатчика работают навстречу друг другу по каждой из 4-х пар в одном и том же диапазоне частот (рис. 13.7). H-образная схема гибридной развязки позволяет приемнику и передатчику одного и того же узла использовать одновременно витую пару и для приема, и для передачи (так же, как и в трансиверах Ethernet на коаксиале).

Для отделения принимаемого сигнала от собственного приемник вычитает из результирующего сигнала известный ему свой сигнал. Естественно, что это не

простая операция и для ее выполнения используются специальные процессоры цифрового сигнала (Digital Signal Processor, DSP).

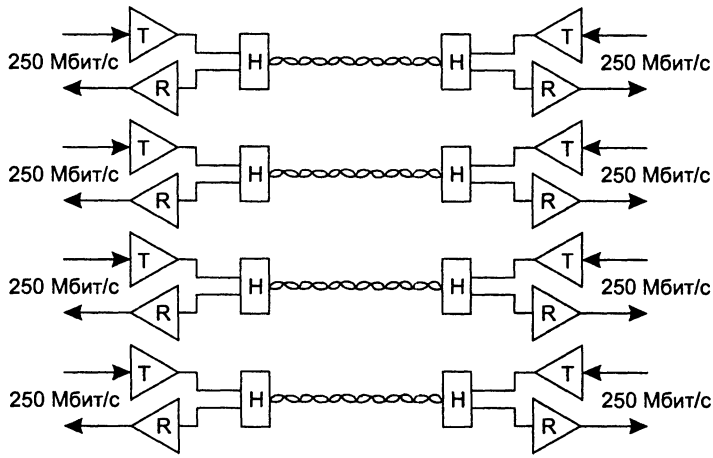


Рис. 13.7. Двунаправленная передача по четырем парам UTP категории 5

Выводы

Потребности в высокоскоростной и в то же время недорогой технологии для подключения к сети мощных рабочих станций привели в начале 90-х годов к созданию инициативной группы, которая занялась поисками такой же простой и эффективной технологии, как Ethernet, но работающей на скорости 100 Мбит/с.

Специалисты разбились на два лагеря, что, в конце концов, привело к появлению двух стандартов, принятых осенью 1995 года: комитет 802.3 утвердил стандарт Fast Ethernet, почти полностью повторяющий технологию Ethernet 10 Мбит/с, а специально созданный комитет 802.12 утвердил стандарт технологии 100VG-AnyLAN, которая сохраняла формат кадра Ethernet, но существенно изменяла метод доступа.

Технология Fast Ethernet сохранила в неприкосновенности метод доступа CSMA/CD, оставив в нем тот же алгоритм и те же временны́е параметры в битовых интервалах (сам битовый интервал уменьшился в 10 раз). Все отличия Fast Ethernet от Ethernet проявляются на физическом уровне. В стандарте Fast Ethernet определены три спецификации физического уровня: 100Base-TX, 100Base-FX и 100Base-T4.

Максимальный диаметр сети Fast Ethernet равен приблизительно 200 м, а более точные значения зависят от спецификации физической среды. В домене коллизий Fast Ethernet допускается не более одного повторителя класса I и не более двух повторителей класса II.

Технология Fast Ethernet при работе на витой паре позволяет за счет процедуры автопереговоров двум портам выбрать наиболее эффективный режим работы — скорость 10 Мбит/с или 100 Мбит/с, а также полудуплексный или дуплексный режим.

В технологии 100VG-AnyLAN арбитром, решающим вопрос о предоставлении станциям доступа к разделяемой среде, является концентратор, поддерживающий приоритетный доступ по требованию.

Технология Gigabit Ethernet добавляет в иерархию скоростей семейства Ethernet новую ступень в 1000 Мбит/с. Эта ступень позволяет эффективно строить крупные локальные сети, в которых серверы и магистрали нижних уровней сети работают на скорости 100 Мбит/с, а магистраль Gigabit Ethernet объединяет их, обеспечивая достаточно большой запас пропускной способности.

Разработчики технологии Gigabit Ethernet сохранили большую степень преемственности с технологиями Ethernet и Fast Ethernet. В Gigabit Ethernet те же форматы кадров, что и в предыдущих версиях Ethernet; Gigabit Ethernet работает в дуплексном и полудуплексном режимах, поддерживая на разделяемой среде тот же метод доступа CSMA/CD с минимальными изменениями.

Специальная рабочая группа 802.3ab разработала вариант Gigabit Ethernet на UTP категории 5. Для обеспечения скорости в 1000 Мбит/с используются: одновременная передача данных по 4 неэкранированным витым парам; метод кодирования PAM-5, передача информации в дуплексном режиме с выделением принимаемого сигнала из общего с помощью процессоров DSP.

Вопросы и задания

1. Какие недостатки метода CSMA/CD устраняет приоритетный доступ по требованию?
2. Почему разработчики технологии Fast Ethernet решили сохранить метод CSMA/CD? Какие топологии поддерживает сеть Fast Ethernet на разделяемой среде? Каков максимальный диаметр сети Fast Ethernet?
3. Сколько пар кабеля используется для передачи данных в версии 100Base-T4?
4. Чем отличаются повторители Fast Ethernet класса I и класса II?
5. Почему в сети Fast Ethernet разрешается использование не более одного повторителя класса I?
6. Чему равен минимальный межкадровый интервал (IPG) в технологии Gigabit Ethernet?
7. Из-за увеличения пропускной способности минимальный размер кадра в Gigabit Ethernet пришлось увеличить до 512 байт. В тех случаях, когда передаваемые данные не могут полностью заполнить поле данных кадра, оно дополняется до необходимой длины неким «заполнителем», который не несет полезной информации. Что предпринято в Gigabit Ethernet для сокращения накладных расходов, возникающих при передаче коротких данных?
8. Какие меры предприняли разработчики технологии Gigabit Ethernet в плане обеспечения передачи данных со скоростью 1000 Мбит/с по витой паре?
9. Почему в технологии Gigabit Ethernet наряду с многомодовым используется и одномодовое оптическое волокно?
10. Пользуясь информацией, представленной в табл. 13.3 и 13.4, определите, какой запас устойчивости имеет конфигурация сети Fast Ethernet с одним повторителем класса I.

    Подсказка. При определении корректности конфигурации сети Fast Ethernet можно не руководствоваться правилами одного или двух хабов, а рассчитать

время оборота сети, как это было сделано в примере сети завода «Трансмаш» (см. главу 12).

Таблица 13.3. Задержки, вносимые кабелем

Тип кабеля	Удвоенная задержка в битовых интервалах	
	на 1 м	на кабеле максимальной длины
UTP категории 3	1,14	114 (100 м)
UTP категории 4	1,14	114 (100 м)
UTP категории 5	1,112	111,2 (100 м)
STP	1,112	111,2 (100 м)
Оптоволокно	1,0	412 (412 м)

Таблица 13.4. Задержки, вносимые сетевыми адаптерами

Тип сетевых адаптеров	Максимальная задержка при полном обороте в битовых интервалах
Два адаптера TX/FX	100
Два адаптера T4	138
Один адаптер TX/FX и один адаптер T4	127

Как и для технологии Ethernet 10 Мбит/с, стандарт Fast Ethernet дает исходные данные для расчета времени оборота сигнала. Однако при этом сама форма представления этих данных и методика расчета меняются. Стандарт Fast Ethernet предоставляет данные об удвоенных задержках, вносимых каждым сегментом сети, не разделяя сегменты сети на левый, правый и промежуточный. Кроме того, задержки, вносимые сетевыми адаптерами, учитывают преамбулы кадров, поэтому время оборота нужно сравнивать с величиной 512 битовых интервала, то есть со временем передачи кадра минимальной длины без преамбулы.

Для повторителей класса I время оборота можно рассчитать следующим образом.

Задержки, вносимые прохождением сигналов по кабелю, рассчитываются на основании данных табл. 13.3, в которой учитывается удвоенное прохождение сигнала по кабелю. Задержки, которые вносят два взаимодействующих через повторитель сетевых адаптера (или порта коммутатора), берутся из табл. 13.4.

Учитывая, что удвоенная задержка, вносимая повторителем класса I, равна 140 битовых интервалов, можно рассчитать время оборота для произвольной конфигурации сети, естественно, учитывая максимально возможные длины

непрерывных сегментов кабелей, приведенные в табл. 13.2. Если получившееся значение меньше 512, значит, по критерию распознавания коллизий сеть является корректной. Стандарт 802.3 рекомендует оставлять запас в 4 битовых интервала для устойчиво работающей сети, но разрешает выбирать эту величину из диапазона от 0 до 5 битовых интервалов.

ГЛАВА 14 Локальные сети на основе разделяемой среды

В этой главе мы рассмотрим отличные от Ethernet технологии, которые также применяются в локальных сетях с разделяемой средой. К ним относятся технологии Token Ring и FDDI, которые долгое время успешно работали в локальных сетях, требующих повышенных показателей надежности, производительности и области охвата. До появления коммутируемых локальных сетей эти технологии превосходили Ethernet по указанным показателям. Поэтому именно им отдавалось предпочтение при построении магистралей локальных сетей или сетей финансовых и правительственных организаций, то есть там, где надежность и производительность особенно необходимы. В технологиях Token Ring и FDDI применяется детерминированный метод доступа, позволяющий разделять среду более эффективно и даже обеспечивать параметры QoS для трафика реального времени.

Технологии Token Ring и FDDI, используя кольцевую топологию физических связей, способны автоматически контролировать работоспособности сети. Сети FDDI, кроме того, обеспечивают автоматическое восстановление сети после отказа, для чего, как и в сетях SDH, применяется двойное кольцо для соединения узлов.

Беспроводная среда является разделяемой по своей физической природе. В этой главе мы рассмотрим две беспроводные технологии, IEEE 802.11 и Bluetooth (IEEE 802.15.1). Первая позволяет создавать беспроводные локальные сети, а вторая относится к персональным сетям. В каждой из них используются собственные методы доступа к среде.

Технология Token Ring

Список ключевых слов: активный монитор, время удержания токена, алгоритм раннего освобождения токена, устройство многостанционного доступа, пассивный концентратор, активный концентратор.

Технология **Token Ring** была разработана компанией IBM в 1984 году, а затем передана в качестве проекта стандарта в комитет IEEE 802, который на ее основе принял в 1985 году стандарт 802.5. Компания IBM в течение долгого времени использовала технологию Token Ring как свою основную сетевую технологию построения локальных сетей на основе компьютеров различных классов — мэйнфреймов, мини-компьютеров и персональных компьютеров. Однако в последнее время даже в продукции компании IBM доминируют представители семейства Ethernet.

Сети Token Ring работают с двумя битовыми скоростями — 4 и 16 Мбит/с. Смешение в одном кольце станций, работающих на разных скоростях, не допускается. Сети Token Ring, работающие со скоростью 16 Мбит/с, имеют некоторые усовершенствования в алгоритме доступа по сравнению со стандартом 4 Мбит/с.

Технология Token Ring сложнее, чем Ethernet. Она обладает некоторыми начальными свойствами отказоустойчивости. В сети Token Ring определены процедуры контроля работы сети, которые опираются на свойство обратной связи, изначально присущее кольцеобразной структуре — посланный кадр всегда возвращается к станции-отправителю. В некоторых случаях обнаруженные ошибки в работе сети устраняются автоматически, например, может быть восстановлен потерянный токен. В других случаях ошибки только фиксируются, а их устранение выполняется вручную обслуживающим персоналом.

Для контроля сети одна из станций исполняет роль так называемого **активного монитора**. Активный монитор выбирается во время инициализации кольца, критерием выбора служит максимальное значение MAC-адреса. Если активный монитор выходит из строя, процедура инициализации кольца повторяется и выбирается новый активный монитор. Чтобы сеть могла обнаружить отказ активного монитора, последний в работоспособном состоянии каждые 3 секунды генерирует специальный кадр, обозначающий его присутствие. Если этот кадр не появляется в сети более 7 секунд, то остальные станции сети начинают процедуру выборов нового активного монитора.

Доступ с передачей токена

Сети Token Ring пользуются разделяемой средой путем передачи токена, принципы которого были рассмотрены в главе 12 при описании функций уровня MAC. Давайте остановимся более детально на некоторых особенностях этого метода, присущих технологии Token Ring 4 Мбит/с, описанной в стандарте 802.5.

В сети Token Ring любая станция всегда непосредственно получает данные только от одной станции — той, которая является предыдущей в кольце. А передает данные своему ближайшему соседу вниз по потоку данных.

Получив токен, станция анализирует его и при отсутствии у нее данных для передачи продвигает токен к следующей станции. Станция, которая имеет данные для передачи, при получении токена изымает его из кольца, что дает ей право доступа к физической среде для передачи своих данных. Затем эта станция выдает в кольцо кадр данных установленного формата последовательно по битам. Кадр снабжается адресами приемника и источника.

Переданные данные проходят по кольцу всегда в одном направлении от одной станции к другой. Все станции кольца ретранслируют кадр побитно, как повторители. Если кадр проходит через станцию назначения, то, распознав свой адрес, эта станция копирует кадр в свой внутренний буфер и вставляет в кадр признак подтверждения приема. Станция, выдавшая кадр данных в кольцо, получив его с подтверждением приема, изымает свой кадр из кольца и передает в сеть новый токен, давая другим станциям сети возможность передавать данные.

На рис. 14.1 описанный алгоритм доступа к среде иллюстрируется временной диаграммой. Здесь показана передача пакета А в кольце, состоящем из 6 станций, от станции 1 к станции 3. После прохождения станции назначения 3 в пакете А устанавливаются два признака — признак А распознавания адреса и признак С копирования пакета в буфер (что на рисунке отмечено звездочкой внутри пакета). После возвращения пакета в станцию 1 отправитель распознает свой пакет по адресу источника и удаляет пакет из кольца. Установленные станцией 3 признаки говорят станции-отправителю о том, что пакет дошел до адресата и был успешно скопирован в его буфер.

Время владения разделяемой средой в сети Token Ring ограничивается фиксированной величиной, называемой **временем удержания токена**. После истечения этого времени станция обязана прекратить передачу собственных данных (текущий кадр разрешается завершить) и передать токен далее по кольцу. Станция может успеть передать за время удержания токена один или несколько кадров в зависимости от размера кадров и величины времени удержания токена.

Обычно время удержания токена по умолчанию равно 10 мс, а максимальный размер кадра в стандарте 802.5 не определен. Для сетей 4 Мбит/с он, как правило, равен 4 Кбайт, а для сетей 16 Мбит/с — 16 Кбайт. Это связано с тем, что за время удержания токена станция должна успеть передать хотя бы один кадр. При скорости 4 Мбит/с за время 10 мс можно передать 5000 байт, а при скорости 16 Мбит/с — 20 000 байт. Максимальные размеры кадра выбраны с некоторым запасом.

В сетях Token Ring 16 Мбит/с используется модернизированный вариант алгоритма доступа к кольцу, называемый **алгоритмом раннего освобождения токена**. В соответствии с ним станция передает токен доступа следующей станции *сразу же после окончания передачи последнего бита кадра*, не дожидаясь возвращения по кольцу этого кадра с установленными битами А и С. В этом случае пропускная способность кольца используется более эффективно, так как по кольцу одновременно продвигаются кадры нескольких станций. Тем не менее свои кадры в каждый момент времени может генерировать только одна станция — та, которая в данный момент владеет токеном. Остальные станции в это время только повторяют чужие кадры, так что принцип разделения кольца во времени сохраняется, ускоряется только процедура передачи владения кольцом.

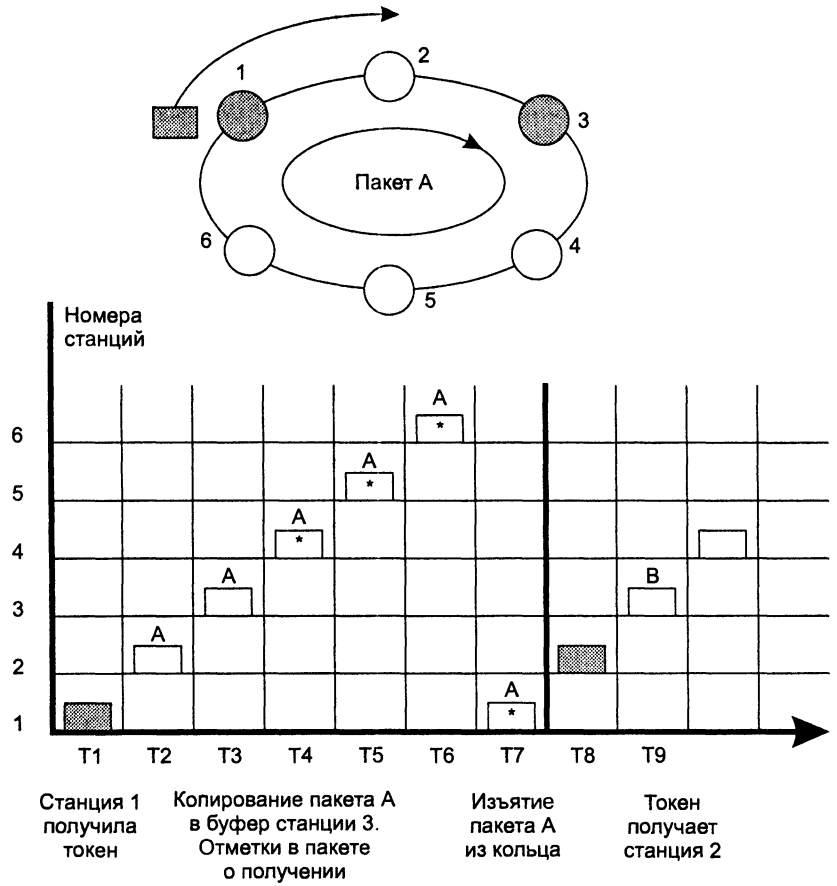


Рис. 14.1. Доступ с передачей токена

Для различных видов сообщений, передаваемых кадрами, могут назначаться различные *приоритеты*: от 0 (низший) до 7 (высший). Решение о приоритете конкретного кадра принимает передающая станция (протокол Token Ring получает этот параметр через межуровневые интерфейсы от протоколов верхнего уровня, например прикладного). Токен также всегда имеет некоторый уровень текущего приоритета. Станция имеет право захватить переданный ей токен только в том случае, если приоритет кадра, который она хочет передать, выше приоритета токена (или равен ему). В противном случае станция обязана передать токен следующей по кольцу станции.

За наличие в сети токена, причем единственной его копии, отвечает активный монитор. Если активный монитор не получает токен в течение длительного времени (например, 2,6 с), то он порождает новый токен.

Приоритетный доступ в технологии Token Ring предназначен для поддержки требований QoS приложений. Однако разработчики приложений для локальных сетей практически им не пользовались.

Физический уровень технологии Token Ring

Стандарт Token Ring фирмы IBM изначально предусматривал построение связей в сети с помощью концентраторов (рис. 14.2), называемых **устройствами многостанционного доступа** (Multi-station Access Unit, MAU, или MSAU). Сеть Token Ring может включать до 260 узлов. Использование концентраторов приводит к тому, что сети Token Ring имеют физическую топологию звезда, а логическую — кольцо.

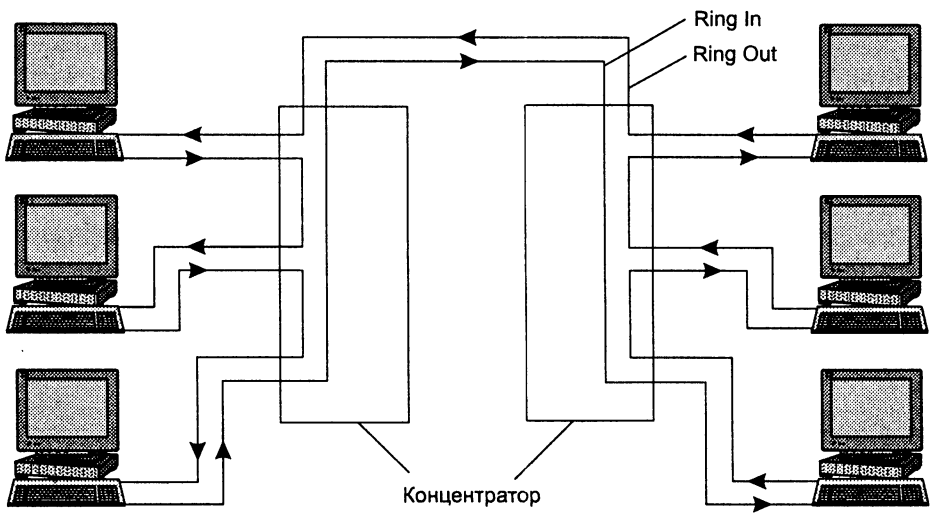


Рис. 14.2. Физическая конфигурация сети Token Ring

Концентратор Token Ring может быть активным или пассивным. **Пассивный концентратор** просто соединяет порты внутренними связями так, чтобы станции, подключаемые к этим портам, образовали кольцо. Ни усиление сигналов, ни их ресинхронизацию пассивный концентратор не выполняет. Такое MSAU-устройство можно считать простым кроссовым блоком за одним исключением — MSAU обеспечивает обход какого-либо порта, когда присоединенный к этому порту компьютер выключают. Такая функция необходима для обеспечения связности кольца вне зависимости от состояния подключенных компьютеров. Обычно обход порта выполняется за счет релейных схем, которые питаются постоянным током от сетевого адаптера, а при выключении сетевого адаптера нормально замкнутые контакты реле соединяют вход порта с его выходом.

Активный концентратор выполняет функции регенерации сигналов и поэтому его можно назвать повторителем.

Возникает вопрос — если концентратор является пассивным устройством, то каким образом обеспечивается качественная передача сигналов на большие расстояния, которые возникают при включении в сеть нескольких сот компьютеров? Ответ состоит в том, что роль усилителя сигналов в этом случае берет на себя каждый сетевой адаптер, а роль синхронизирующего блока — сетевой адаптер активного монитора кольца. Каждый сетевой адаптер Token Ring имеет блок по-

вторения, который умеет регенерировать и синхронизировать сигналы, однако последнюю функцию выполняет в кольце только блок повторения активного монитора.

В общем случае сеть Token Ring имеет комбинированную звездно-кольцевую конфигурацию. Конечные узлы подключаются к MSAU-устройствам по топологии звезды, а сами концентраторы объединяются через специальные порты Ring In (RI) и Ring Out (RO), образуя магистральное физическое кольцо.

Технология Token Ring позволяет использовать для соединения конечных станций и концентраторов различные типы кабеля: STP типа 1, UTP типа 3, UTP типа 6, а также волоконно-оптический кабель. При использовании экранированной витой пары STP типа 1 из номенклатуры кабельной системы IBM в кольцо допускается объединять до 260 станций при длине ответвительных кабелей до 100 метров, а при использовании неэкранированной витой пары максимальное количество станций сокращается до 72 при длине ответвительных кабелей до 45 метров. Расстояние между пассивными концентраторами может достигать 100 м при использовании кабеля STP типа 1 и 45 м при использовании кабеля UTP типа 3. Между активными концентраторами максимальное расстояние увеличивается соответственно до 730 или 365 м в зависимости от типа кабеля.

Максимальная длина кольца Token Ring составляет 4000 м.

ПРИМЕЧАНИЕ

Ограничения на максимальную длину кольца и количество станций в кольце в технологии Token Ring не являются такими жесткими, как в технологии Ethernet. Здесь эти ограничения во многом связаны со временем оборота токена по кольцу. Есть и другие соображения, диктующие выбор ограничений. Так, если кольцо состоит из 260 станций, то при времени удержания токена в 10 мс токен вернется в активный монитор в худшем случае через 2,6 с, а это время как раз составляет тайм-аут контроля оборота токена. В принципе, все значения тайм-аутов в сетевых адаптерах узлов сети Token Ring можно настраивать, поэтому можно построить сеть Token Ring с бóльшим количеством станций и с бóльшей длиной кольца.

Технология FDDI

Список ключевых слов: сквозной, или транзитный, режим, уровень администрирования станции, двойное подключение к кольцу, одиночное подключение к кольцу, двухпортовое подключение к кольцу.

Технология **FDDI** (Fiber Distributed Data Interface — распределенный интерфейс передачи данных по оптоволокну) — это первая технология локальных сетей, в которой в качестве среды передачи данных стал применяться волоконно-оптический кабель. Работы по созданию технологий и устройств локальных сетей, использующих волоконно-оптические каналы, начались в 80-е годы, вскоре после начала промышленной эксплуатации подобных каналов в территориальных сетях. Проблемная группа X3T9.5 института ANSI разработала в период с 1986 по 1988 гг. начальные версии стандарта FDDI, который описывает передачу кад-

ров со скоростью 100 Мбит/с по двойному волоконно-оптическому кольцу длиной до 100 км.

Основные характеристики технологии FDDI

Технология FDDI во многом основывается на технологии Token Ring, развивая и совершенствуя ее основные идеи. Разработчики технологии FDDI ставили перед собой следующие цели:

- повысить битовую скорость передачи данных до 100 Мбит/с;
- повысить отказоустойчивость сети за счет стандартных процедур восстановления ее после отказов различного рода — повреждения кабеля, некорректной работы узла, концентратора, возникновения высокого уровня помех на линии и т. п.;
- максимально эффективно использовать потенциальную пропускную способность сети как для асинхронного, так и для синхронного (чувствительного к задержкам) трафиков.

Сеть FDDI строится на основе двух оптоволоконных колец, которые образуют основной и резервный пути передачи данных между узлами сети. Наличие двух колец — это основное средство повышения отказоустойчивости в сети FDDI.

Узлы, которые хотят воспользоваться этим повышенным потенциалом надежности, должны быть подключены к обоим кольцам. В технологии FDDI для передачи световых сигналов по оптическим волокнам реализовано кодирование 4B/5B в сочетании с кодированием NRZI. Эта схема приводит к передаче по линии связи сигналов с тактовой частотой 125 МГц.

В *нормальном* режиме работы сети данные проходят через все узлы и все участки кабеля только первичного кольца, этот режим назван **сквозным**, или **транзитным**. Вторичное кольцо в этом режиме не используется.

В случае какого-либо вида отказа, когда часть первичного кольца не может передавать данные (например, обрыв кабеля или отказ узла), первичное кольцо объединяется со вторичным (рис. 14.3), вновь образуя единое кольцо. Этот режим работы сети называется режимом *свертывания* колец. Операция свертывания производится средствами концентраторов и/или сетевых адаптеров FDDI. Для упрощения этой процедуры данные по первичному кольцу всегда передаются в одном направлении (на диаграммах это направление изображается против часовой стрелки), а по вторичному — в обратном (изображается по часовой стрелке). Поэтому при образовании общего кольца из двух колец передатчики станций попрежнему остаются подключенными к приемникам соседних станций, что позволяет правильно передавать и принимать информацию соседними станциями.

В стандартах FDDI много внимания отводится различным процедурам, которые позволяют определить наличие отказа в сети, а затем произвести необходимое реконфигурацию. Технология FDDI дополняет механизмы обнаружения отказов технологии Token Ring механизмами реконфигурирования пути передачи данных в сети, основанными на наличии резервных связей, которые предоставляет второе кольцо.

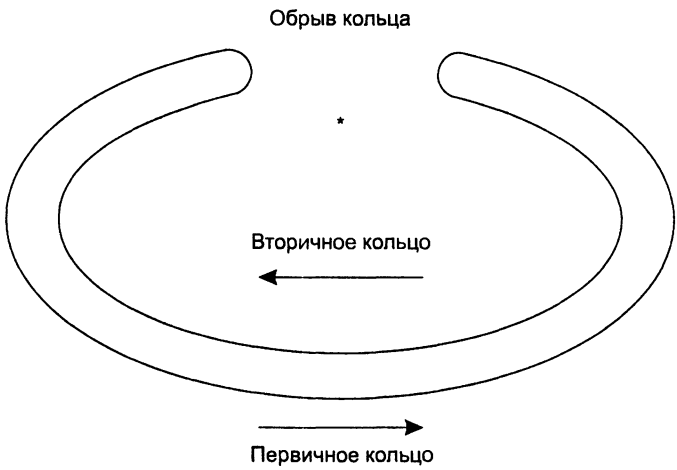


Рис. 14.3. Реконфигурация колец FDDI при отказе

Сеть FDDI может полностью восстанавливать свою работоспособность в случае единичных отказов ее элементов. При множественных отказах сеть распадается на несколько не связанных сетей.

Кольца в сетях FDDI рассматриваются как общая разделяемая среда передачи данных. Метод доступа к этой среде очень близок к методу доступа сетей Token Ring. Станции FDDI применяют *алгоритм раннего освобождения токена*, как и сети Token Ring 16 Мбит/с.

Отличия в методах доступа заключаются в следующем:

- Время удержания токена в сети FDDI не является постоянной величиной, как в сети Token Ring. Это время зависит от загрузки кольца — при небольшой загрузке оно растет, а при перегрузках может снижаться до нуля. Однако эти изменения касаются только асинхронного трафика, который не критичен к небольшим задержкам передачи кадров. Для синхронного трафика время удержания токена по-прежнему остается фиксированной величиной.
- Механизм приоритетов кадров, принятый в Token Ring, в технологии FDDI отсутствует. Разработчики технологии решили, что деление трафика на 8 уровней приоритетов избыточно, достаточно разделить трафик на два класса — асинхронный и синхронный, последний из которых обслуживается всегда, даже при перегрузках кольца.

В остальном пересылка кадров между станциями кольца на уровне MAC полностью соответствует технологии Token Ring.

Рисунок 14.4 иллюстрирует соответствие стека протоколов технологии FDDI семиуровневой модели OSI. FDDI определяет протокол физического уровня и протокол подуровня доступа к среде (MAC) канального уровня. Как и во многих других технологиях локальных сетей, в технологии FDDI используется протокол подуровня управления логическим каналом LLC.

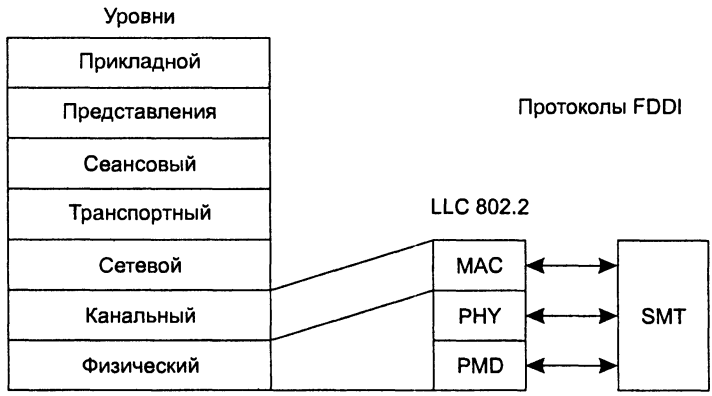


Рис. 14.4. Стек протоколов технологии FDDI

Специфической особенностью технологии FDDI является **уровень администрирования станции** (Station ManagemenT, SMT). Именно уровень SMT выполняет все функции по администрированию и мониторингу всех остальных уровней стека протоколов FDDI. В управлении кольцом принимает участие каждый узел сети FDDI. Поэтому все узлы обмениваются специальными SMT-кадрами для управления сетью.

Отказоустойчивость сетей FDDI обеспечивается протоколами и других уровней: с помощью физического уровня устраняются отказы сети по физическим причинам, например из-за обрыва кабеля, а с помощью уровня MAC — логические отказы сети, например потеря нужного внутреннего пути передачи токена и кадров данных между портами концентратора.

Отказоустойчивость технологии FDDI

Как уже отмечалось, для обеспечения отказоустойчивости в стандарте FDDI предусмотрено создание двух оптоволоконных колец — *первичного* и *вторичного*. В стандарте FDDI определены два типа конечных узлов — *станции* и *концентраторы*. Для подключения станций и концентраторов к сети может быть использован один из двух возможных способов.

- **Двойное подключение** (Dual Attachment, DA) — одновременное подключение к первичному и вторичному кольцам; станция и концентратор, подключенные таким способом, называются соответственно станцией двойного подключения (Dual Attachment Station, DAS) и концентратором двойного подключения (Dual Attachment Concentrator, DAC).
- **Одиночное подключение** (Single Attachment, SA) — подключение только к первичному кольцу; станция и концентратор, подключенные данным способом, называются соответственно станцией одиночного подключения (Single Attachment Station, SAS) и концентратором одиночного подключения (Single Attachment Concentrator, SAC).

Обычно, хотя и не обязательно, концентраторы имеют двойное подключение, а станции — одиночное, как показано на рис. 14.5. Чтобы устройства легче было правильно присоединять к сети, их разъемы маркируются. Разъемы типа A и B должны быть у устройств с двойным подключением; разъем M (Master) имеется у концентратора для одиночного подключения станции, у которой ответный разъем должен иметь тип S (Slave).

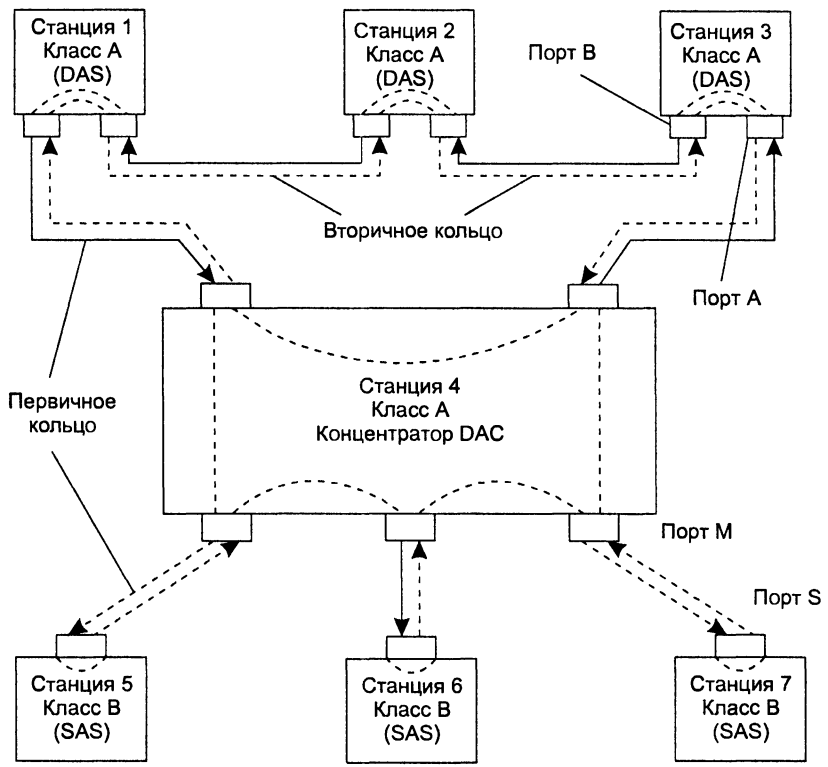


Рис. 14.5. Подключение узлов к кольцам FDDI

В случае однократного обрыва кабеля между устройствами с двойным подключением сеть FDDI сможет продолжить нормальную работу за счет автоматической реконфигурации внутренних путей передачи кадров между портами концентратора (рис. 14.6).

Двукратный обрыв кабеля приведет к образованию двух изолированных сетей FDDI. При обрыве кабеля, идущего к станции с одиночным подключением, она оказывается отрезанной от сети, а кольцо продолжает работать за счет реконфигурации внутреннего пути в концентраторе — порт M, к которому была подключена данная станция, исключается из общего пути.

Для сохранения работоспособности сети при отключении питания в станции с двойным подключением (например, просто при ее выключении) она должна

быть оснащена оптическим обходным переключателем, который создаст резервный путь для световых потоков.

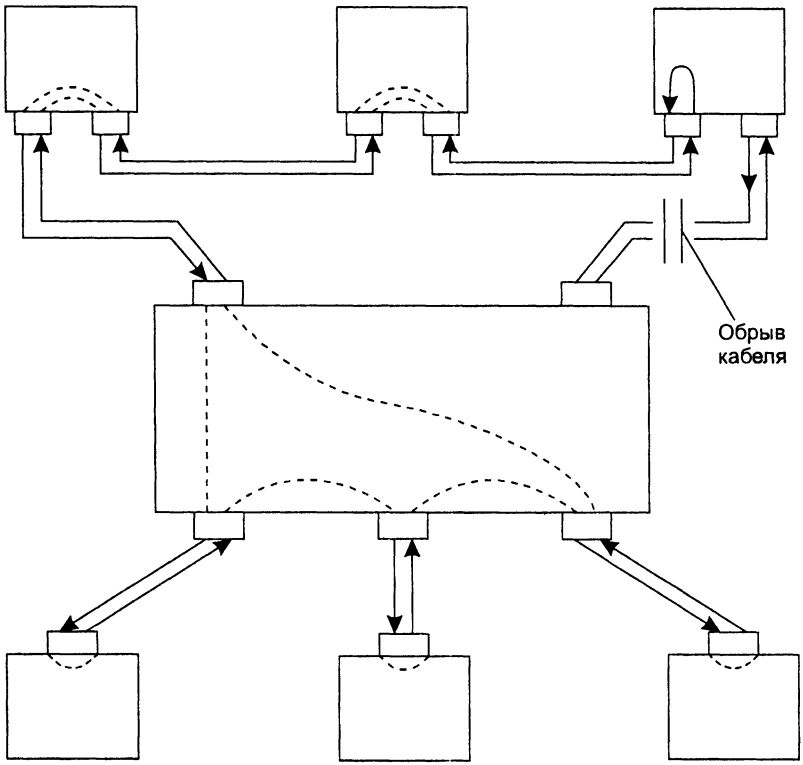


Рис. 14.6. Реконфигурация сети FDDI при обрыве

И, наконец, станции DAS или концентраторы DAC можно подключать к двум портам M одного или двух концентраторов, создавая древовидную структуру с основными и резервными связями. По умолчанию порт B поддерживает основную связь, а порт A — резервную. Такая конфигурация называется **двухпортовым подключением**.

Отказоустойчивость поддерживается за счет постоянного слежения концентраторов и станций уровня SMT за временны́ми интервалами циркуляции токена и кадров, а также за наличием физического соединения между соседними портами в сети. В сети FDDI нет выделенного активного монитора — все станции и концентраторы равноправны, и при обнаружении отклонений от нормы они начинают процесс повторной инициализации сети, а затем и ее реконфигурации.

Реконфигурация внутренних путей в концентраторах и сетевых адаптерах выполняется специальными оптическими переключателями, которые перенаправляют световой луч и имеют достаточно сложную конструкцию.

Максимальная общая длина кольца FDDI составляет 100 километров, максимальное число станций с двойным подключением в кольце — 500.

Технология FDDI разрабатывалась для ответственных участков сетей — на магистральных соединениях между крупными сетями, например сетями зданий, а также для подключения к сети высокопроизводительных серверов. Поэтому главным для разработчиков было обеспечить высокую скорость передачи данных, отказоустойчивость на уровне протокола и большие расстояния между узлами сети. Все эти цели были достигнуты. В результате технология FDDI получилась качественной, но весьма дорогой. Даже появление более дешевого варианта для витой пары не намного снизило стоимость подключения одного узла к сети FDDI. Основной областью применения технологии FDDI стали магистрали сетей, состоящих из нескольких зданий, а также сети масштаба крупного города, то есть класса MAN.

Беспроводные локальные сети

Список ключевых слов: скрытый терминал, методы опроса, спецификации 802.11, 802.11a, 802.11b, 802.11g, сеть с базовым набором услуг, точка доступа, служба распределенной системы, сеть с расширенным набором услуг, распределенный режим DCF, централизованный режим PCF, конкурентное окно, таймер отсрочки, межкадровые интервалы SIFS, PIFS и DIFS, контролируемый период, протокол WEP.

Беспроводные локальные сети сегодня рассматриваются как дополнение к проводным сетям, а не как конкурентное решение. Отношение к беспроводным локальным сетям не всегда было таковым, в середине 90-х было популярно мнение, в соответствии с которым все большее число локальных сетей будет переходить на беспроводные технологии. Преимущество беспроводных локальных сетей очевидно — их проще и дешевле разворачивать и модифицировать, так как вся громоздкая кабельная и инфраструктура оказывается излишней. Еще одно преимущество — обеспечение мобильности пользователей. Однако за эти преимущества беспроводные сети расплачиваются большим перечнем проблем, которые несет с собой неустойчивая и непредсказуемая беспроводная среда. Мы уже рассматривали особенности распространения сигналов в такой среде в главе 10.

Помехи от разнообразных бытовых приборов и других телекоммуникационных систем, атмосферные помехи и отражения сигнала создают большие трудности для надежного приема информации. Локальные сети — это, прежде всего, сети зданий, а распространение радиосигнала внутри здания еще сложнее, чем вне его. В стандарте IEEE 802.11 приводится изображение распределения интенсивности сигнала (рис. 14.7). В стандарте подчеркивается, что это статическое изображение, в действительности картина является динамической, и при перемещении объектов в комнате распределение сигнала может существенно измениться.

Методы *расширения спектра* помогают снизить влияние помех на полезный сигнал, кроме того, в беспроводных сетях широко используются *прямая коррекция ошибок* (Forward Error Control, FEC) и протоколы с повторной передачей потерянных кадров. Тем не менее практика показала, что в тех случаях, когда ничего

не мешает применению проводной локальной сети, организации предпочитают именно этот вид LAN, несмотря на то, что при этом нельзя обойтись без кабельной системы.

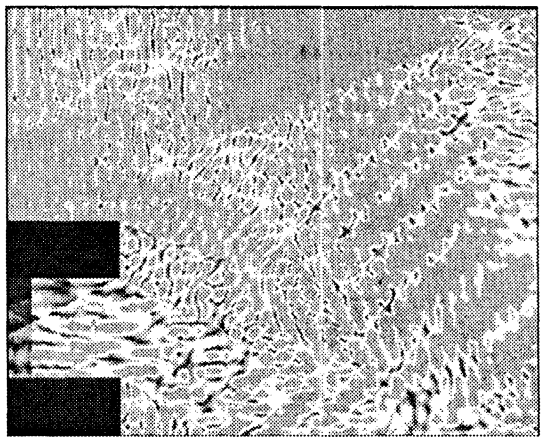


Рис. 14.7. Распределение интенсивности радиосигнала

Неравномерное распределение интенсивности сигнала приводит не только к битовым ошибкам передаваемой информации, но и к *неопределенности зоны покрытия* беспроводной локальной сети. В проводных локальных сетях такой проблемы нет, те и только те устройства, которые подключены к кабельной системе здания или кампуса, получают сигналы и участвуют в работе LAN. Беспроводная локальная сеть не имеет точной области покрытия. Часто используемое изображение такой области в форме шестиугольника или круга является не чем иным, как абстракцией. В действительности, сигнал может быть настолько ослаблен, что устройства, находящиеся в предполагаемых пределах зоны покрытия, вообще не могут принимать и передавать информацию.

Рисунок 14.7 хорошо иллюстрирует такую ситуацию. Подчеркнем, что с течением времени ситуация с распределением сигнала может измениться и состав LAN также изменится. По этой причине даже технологии, рассчитанные на фиксированные (не мобильные) узлы сети, должны учитывать то, что беспроводная локальная сеть является неполносвязной. Даже если считать, что сигнал распространяется идеально во все стороны, образованию полносвязной топологии может мешать то, что радиосигнал затухает пропорционально квадрату расстояния от источника. Поэтому при отсутствии базовой станции некоторые пары узлов не смогут взаимодействовать из-за того, что расположены за пределами зоны покрытия передатчиков партнера.

В примере на рис. 14.8, *а* показана такая фрагментированная локальная сеть. Неполносвязность беспроводной сети порождает проблему доступа к разделяемой среде, известную под названием **скрытого терминала**. Проблема возникает в том случае, когда два узла находятся вне зон досягаемости друг друга (узлы A и C на рис. 14.8, *а*), но существует третий узел B, который принимает сигналы как от A, так и от C. Предположим, что в радиосети используется традиционный метод

доступа, основанный на прослушивании несущей, например CSMA/CD. В данном случае коллизии будут возникать значительно чаще, чем в проводных сетях. Пусть, например, узел В занят обменом с узлом А. Узлу С сложно определить, что среда занята, он может посчитать ее свободной и начать передавать свой кадр. В результате сигналы в районе узла В будут искажены, то есть произойдет коллизия, вероятность возникновения которой в проводной сети была бы неизмеримо ниже.

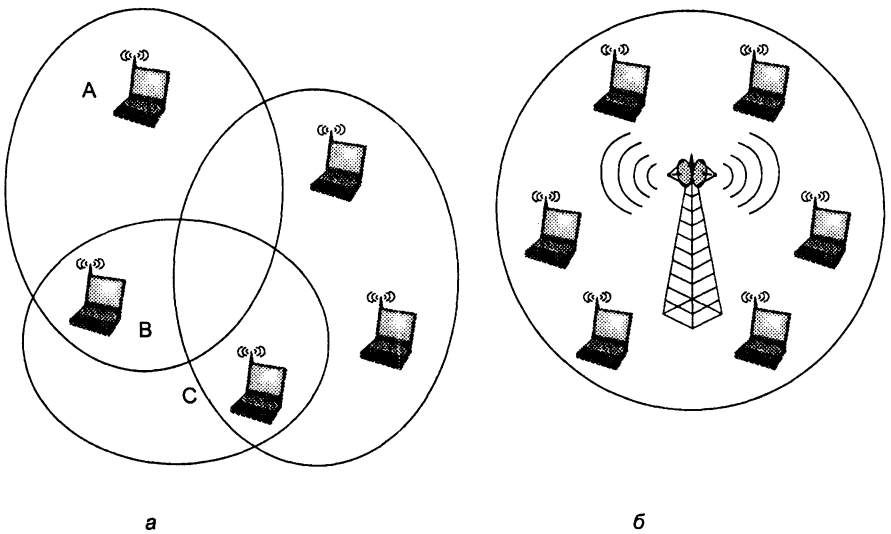


Рис. 14.8. Связность беспроводной локальной сети: *а* — специализированная беспроводная сеть, *б* — беспроводная сеть с базовой станцией

Распознавание коллизий затруднено в радиосети еще и потому, что сигнал собственного передатчика существенно подавляет сигнал от удаленного передатчика и распознать искажение сигнала чаще всего невозможно.

В методах доступа в беспроводных сетях не только отказываются от прослушивания несущей, но и от распознавания коллизий.

Вместо этого в них используют методы предотвращения коллизий, в том числе и **методы опроса**.

Применение базовой станции может улучшить связность сети (рис. 14.8, *б*). Базовая станция обычно обладает большей мощностью, а ее антенна устанавливается так, чтобы более равномерно и беспрепятственно покрывать нужную территорию. В результате все узлы беспроводной локальной сети получают возможность обмениваться данными с базовой станцией, которая транзитом передает данные между узлами.

Беспроводные локальные сети считаются перспективными для таких применений, в которых сложно или невозможно использовать проводные сети. Ниже перечислены основные области применения беспроводных локальных сетей.

- *Резидентный доступ альтернативных операторов связи*, у которых нет проводного доступа к клиентам, проживающим в многоквартирных домах.
- Так называемый *«кочевой» доступ* в аэропортах, железнодорожных вокзалах и т. п.
- Организация локальных сетей в зданиях, где нет возможности установить современную кабельную систему, например, *в исторических зданиях* с оригинальным интерьером.
- Организация *временных локальных сетей*, например, при проведении конференций.
- *Расширения локальных сетей.* Иногда одно здание предприятия, например испытательная лаборатория или цех, может быть расположено изолированно от других. Небольшое число рабочих мест в таком здании делает крайне невыгодным прокладку к нему отдельного кабеля, поэтому беспроводная связь оказывается более рациональным вариантом.
- *Мобильные локальные сети.* Если пользователь хочет пользоваться услугами сети, перемещаясь из помещения в помещение или из здания в здание, то здесь конкурентов у беспроводной локальной сети просто нет. Классическим примером такого пользователя является врач, совершающий обход и пользующийся своим ноутбуком для связи с базой данных больницы.

Пока что мобильные локальные сети не претендуют на полное покрытие крупных территорий, как это сделали мобильные сотовые телефонные сети, но перспективы такого развития имеются. В области построения территориальных сотовых мобильных сетей передачи данных технологиям беспроводных локальных сетей предстоит выдержать конкуренцию с мобильными сотовыми сетями третьего поколения. Мобильные сотовые сети второго поколения не представляют собой серьезных конкурентов, так как они разрабатывались в первую очередь для передачи голоса, их возможности по передаче данных ограничиваются скоростями в несколько килобит в секунду, в то время как беспроводные локальные сети обеспечивают скорости в несколько десятков мегабит в секунду. Однако в системах третьего поколения скорость передачи данных будет находиться в диапазоне от 144 Кбит/с до 2 Мбит/с (скорость 2 Мбит/с обеспечивается на небольших расстояниях от базовой станции). Так что конкуренция может оказаться жесткой.

Далее будет рассмотрен самый популярный стандарт беспроводных локальных сетей — IEEE 802.11. Заметим, что в этой области существуют и другие стандарты (в частности, институт ETSI разработал стандарт HIPERLAN 1), однако большинство производителей выпускают оборудование в соответствии со спецификациями IEEE 802.11.

Стек протоколов IEEE 802.11

Естественно, что стек протоколов стандарта IEEE 802.11 соответствует общей структуре стандартов комитета 802, то есть состоит из физического уровня и уровня MAC, над которыми работает уровень LLC. Как и у всех технологий семейства 802, технология 802.11 определяется нижними двумя уровнями, то есть физическим уровнем и уровнем MAC, а уровень LLC выполняет свои стандарт-

ные общие для всех технологий LAN функции. Так как искажения кадров в беспроводной среде более вероятны, чем в проводной, уровень LLC должен, скорее всего, использоваться в режиме LLC2. Но это уже не зависит от технологии 802.11, режим работы уровня LLC выбирается протоколами верхних уровней.

Структура стека протоколов IEEE 802.11 показана на рис. 14.9.

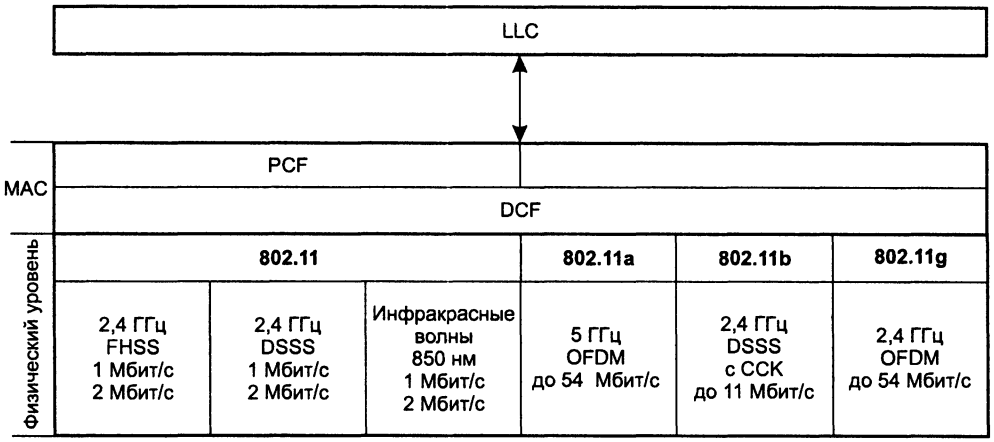


Рис. 14.9. Стек протоколов IEEE 802.11

На физическом уровне существует несколько вариантов спецификаций, которые отличаются используемым частотным диапазоном, методом кодирования и как следствие — скоростью передачи данных. Все варианты физического уровня работают с одним и тем же алгоритмом уровня MAC, но некоторые временны́е параметры уровня MAC зависят от используемого физического уровня.

В 1997 году комитетом **802.11** был принят стандарт, который определял функции уровня MAC вместе с *тремя вариантами физического уровня*, которые обеспечивают передачу данных со скоростями 1 и 2 Мбит/с.

- В первом варианте средой являются *инфракрасные волны* диапазона 850 нм, которые генерируются либо полупроводниковым лазерным диодом, либо светодиодом (LED). Так как инфракрасные волны не проникают через стены, область покрытия LAN ограничивается зоной прямой видимости. Стандарт предусматривает три варианта распространения излучения: ненаправленную антенну, отражение от потолка и фокусное направленное излучение. В первом случае узкий луч рассеивается с помощью системы линз. Фокусное направленное излучение предназначено для организации двухточечной связи, например, между двумя зданиями.
- Во втором варианте в качестве передающей среды используется *микроволновый диапазон* 2,4 ГГц, который в соответствии с рекомендациями ITU в большинстве стран не лицензируется. Этот вариант основан на методе FHSS, (см. главу 10). В методе FHSS каждый узкий канал имеет ширину 1 МГц. Частотная манипуляция (FSK) с двумя состояниями сигнала (частотами) дает скорость 1 Мбит/с, с четырьмя состояниями — 2 Мбит/с. В случае FHSS

сеть может состоять из сот, причем для исключения взаимного влияния в соседних сотах могут применяться ортогональные последовательности частот. Количество каналов и частота переключения между каналами настраиваются, так что при развертывании беспроводной локальной сети можно учитывать особенности регулирования спектра частот конкретной страны. Так, в США в диапазоне 2,4 ГГц может быть до 79 каналов, причем максимальное время нахождения на каждом канале не должно превышать 400 мс.

- Третий вариант, в котором используется тот же *микроволновый диапазон*, основан на методе DSSS, где в качестве последовательности чипов применяется 11-битный код 10110111000. Каждый бит кодируется путем двоичной фазовой (1 Мбит/с) или квадратурной фазовой (2 Мбит/с) манипуляции.

В 1999 году были приняты еще два варианта физического уровня: **802.11a** и **802.11b**.

- *Спецификация 802.11a* обеспечивает повышение скорости за счет более высокого диапазона частот (5 ГГц). Для этого задействуются 300 МГц из этого диапазона, ортогональное частотное мультиплексирование (OFDM) и прямая коррекция ошибок (FEC). Скорости передачи данных составляют 6, 9, 12, 18, 24, 36, 48 и 54 Мбит/с. Диапазон 5 ГГц спецификации 802.11a пока мало «населен» и обеспечивает высокие скорости передачи данных. Однако его использование связано с двумя проблемами. Во-первых, оборудование для этих частот пока еще слишком дорогое, во-вторых, в некоторых странах частоты этого диапазона подлежат лицензированию.
- В *спецификации 802.11b института IEEE* по-прежнему используется диапазон 2,4 ГГц, что позволяет задействовать более дешевое оборудование. Для повышения скорости до 11 Мбит/с, которая сопоставима со скоростью классического стандарта Ethernet, здесь применяется более эффективный метод DSSS, использующий технику Complementary Code Keying (CCK).

Еще один стандарт для физического уровня разработан группой **802.11g** института IEEE летом 2003 года. В нем также задействован диапазон 2,4 ГГц, но со скоростью передачи данных до 54 Мбит/с. В этой спецификации используется ортогональное частотное мультиплексирование (OFDM). До недавнего времени в США в диапазоне 2,4 ГГц разрешалось работать только за счет расширения спектра. Снятие этого ограничения дало импульс новым разработкам, в результате появилась новая высокоскоростная беспроводная технология. Для обратной совместимости с 802.11b поддерживается также техника CCK.

Диаметр сети 802.11 зависит от многих параметров, в том числе и от диапазона частот. Обычно диаметр беспроводной локальной сети находится в пределах от 100 до 300 м.

Уровень MAC выполняет в беспроводных сетях больше функций, чем в проводных сетях. Функции уровня MAC в стандарте 802.11 включают:

- доступ к разделяемой среде;
- обеспечение мобильности станций при наличии нескольких базовых станций;
- обеспечение безопасности, эквивалентной безопасности проводных локальных сетей.

Топологии локальных сетей стандарта 802.11

Станции могут использовать разделяемую среду для того, чтобы передавать данные:

- непосредственно друг другу в пределах одной BSS-сети;
- в пределах одной BSS-сети транзитом через точку доступа;
- между разными BSS-сетями через две точки доступа и распределенную систему;
- между BSS-сетью и проводной локальной сетью через точку доступа, распределенную систему и портал[1].

Стандарт 802.11 поддерживает два типа топологий локальных сетей: с базовым и с расширенным наборами услуг.

Сеть с базовым набором услуг (Basic Service Set, BSS) образуется отдельными станциями, базовая станция отсутствует, узлы взаимодействуют друг с другом непосредственно (рис. 14.10). Для того чтобы войти в BSS-сеть, станция должна выполнить процедуру присоединения.

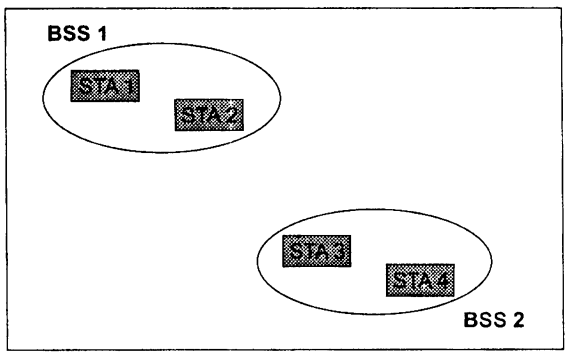


Рис. 14.10. Сети с базовым набором услуг

BSS-сети не являются традиционными сотами в отношении зон покрытия, они могут находиться друг от друга на значительном расстоянии, а могут частично или полностью перекрываться — стандарт 802.11 оставляет здесь свободу для проектировщика сети.

В сетях, обладающих инфраструктурой, некоторые станции сети являются базовыми, или, в терминологии 802.11, **точками доступа** (Access Point, AP). Станция, которая выполняет функции AP, является членом какой-нибудь BSS-сети (рис. 14.11). Все базовые станции сети связаны между собой с помощью распределенной системы (Distribution System, DS), в качестве которой может использоваться та же среда (то есть радио- или инфракрасные волны), что и для взаимодействия между станциями, или же отличная от нее, например проводная.

[1] Функции портала стандартом не детализируются, это может быть коммутатор или маршрутизатор.

Точки доступа вместе с распределенной системой поддерживают **службу распределенной системы** (Distribution System Service, DSS). Задачей DSS является передача пакетов между станциями, которые по каким-то причинам не могут или не хотят взаимодействовать между собой непосредственно. Наиболее очевидной причиной использования DSS является принадлежность станций разным BSS-сетям. В этом случае они передают кадр своей точке доступа, которая через DS передает его точке доступа, обслуживающей BSS-сеть со станцией назначения.

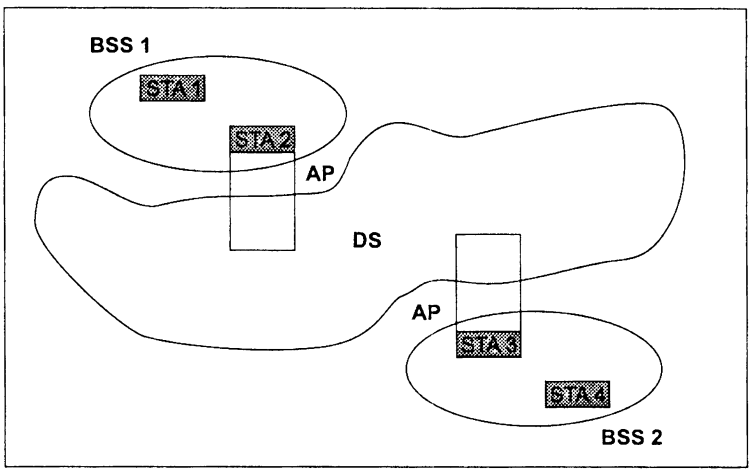


Рис. 14.11. Сеть с расширенным набором услуг

Сеть с расширенным набором услуг (Extended Service Set, ESS) состоит из нескольких BSS-сетей, объединенных распределенной средой.

ESS-сеть обеспечивает станциям мобильность — они могут переходить из одной BSS-сети в другую. Эти перемещения обеспечиваются функциями уровня MAC рабочих и базовых станций, поэтому они совершенно прозрачны для уровня LLC. ESS-сеть может также взаимодействовать с проводной локальной сетью. Для этого в распределенной системе должен присутствовать портал.

Распределенный режим доступа DCF

В сетях 802.11 уровень MAC обеспечивает два режима доступа к разделяемой среде (см. рис. 14.9): **распределенный режим DCF** (Distributed Coordination Function) и **централизованный режим PCF** (Point Coordination Function).

Рассмотрим сначала, как обеспечивается доступ в распределенном режиме DCF. В этом режиме реализуется хорошо известный нам метод CSMA/CA. Вместо неэффективного в беспроводных сетях прямого распознавания коллизий по методу CSMA/CD, здесь используется их косвенное выявление. Для этого каждый переданный кадр должен подтверждаться кадром положительной квитанции, посылаемым станцией назначения. Если же по истечении оговоренного тайм-аута квитанция не поступает, станция-отправитель считает, что произошла коллизия.

Режим доступа DCF требует синхронизации станций. В спецификации 802.11 эта проблема решается достаточно элегантно — временны́е интервалы начинают отсчитываться от момента окончания передачи очередного кадра (рис. 14.12). Это не требует передачи каких-либо специальных синхронизирующих сигналов и не ограничивает размер пакета размером слота, так как слоты принимаются во внимание только при принятии решения о начале передачи кадра.

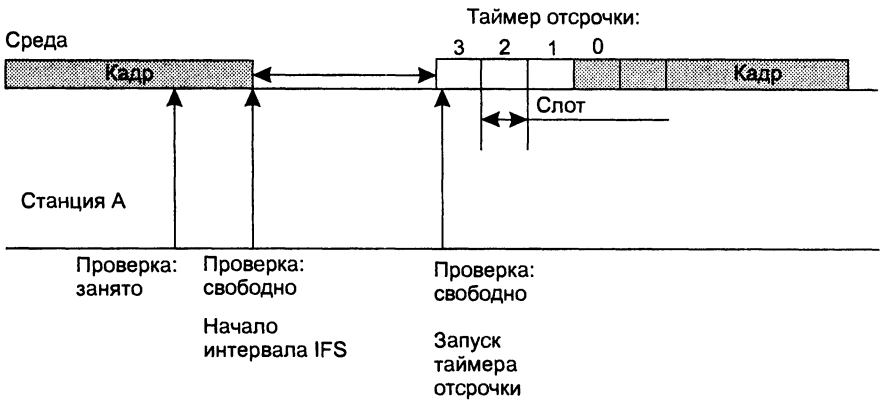


Рис. 14.12. Режим доступа DCF

Станция, которая хочет передать кадр, обязана предварительно прослушать среду. Как только она фиксирует окончание передачи кадра, она обязана отсчитать интервал времени, равный межкадровому интервалу (IFS). Если после истечения IFS среда все еще свободна, то начинается отсчет слотов фиксированной длительности. Кадр можно начать передавать только в начале какого-либо из слотов при условии, что среда свободна. Станция выбирает для передачи слот на основании *усеченного экспоненциального двоичного алгоритма отсрочки*, аналогичного используемому в методе CSMA/CD. Номер слота выбирается как случайное целое число, равномерно распределенное в интервале [0, CW], где CW означает Contention Window (**конкурентное окно**).

О том, как выбирается размер слота и величина конкурентного окна, будет сказано немного позже. А сейчас рассмотрим этот довольно непростой метод доступа на примере (см. рис. 14.12). Пусть станция А выбрала для передачи на основании усеченного экспоненциального двоичного алгоритма отсрочки слот 3. При этом она присваивает **таймеру отсрочки** (назначение которого будет ясно из дальнейшего описания) значение 3 и начинает проверять состояние среды в начале каждого слота. Если среда свободна, то из значения таймера отсрочки вычитается 1, и если результат равен нулю, то начинается передача кадра.

Таким образом, обеспечивается условие незанятости всех слотов, включая выбранный. Это условие является необходимым для начала передачи.

Если же в начале какого-нибудь слота среда оказывается занятой, то вычитания единицы не происходит, и таймер «замораживается». В этом случае станция на-

чинает новый цикл доступа к среде, изменяя только алгоритм выбора слота для передачи. Как и в предыдущем цикле, станция следит за средой и при ее освобождении делает паузу в течение межкадрового интервала. Если среда осталась свободной, то станция *использует значение «замороженного» таймера в качестве номера слота* и выполняет описанную выше процедуру проверки свободных слотов с вычитанием единиц, начиная с замороженного значения таймера отсрочки.

Размер слота зависит от способа кодирования сигнала; так, для метода FHSS размер слота равен 28 мкс, а для метода DSSS — 1 мкс. Размер слота выбирается таким образом, чтобы он превосходил время распространения сигнала между любыми двумя станциями сети плюс время, затрачиваемое станцией на распознавание занятости среды. Если такое условие соблюдается, то каждая станция сети сумеет правильно распознать начало передачи кадра при прослушивании слотов, предшествующих выбранному ею для передачи слоту. Это, в свою очередь, означает следующее.

Коллизия может случиться только в том случае, когда несколько станций выбирают один и тот же слот для передачи.

В этом случае кадры искажаются, и квитанции от станций назначения не приходят. Не получив в течение определенного времени квитанцию, отправители фиксируют факт коллизии и пытаются передать свои кадры снова. При каждой повторной неудачной попытке передачи кадра интервал [0, CW], из которого выбирается номер слота, удваивается. Если, например, начальный размер окна выбран равным 8 (то есть CW = 7), то после первой коллизии размер окна должен быть равен 16 (CW = 15), после второй последовательной коллизии — 32 и т. д. Начальное значение CW в соответствии со стандартом 802.11 должно выбираться в зависимости от типа физического уровня, используемого в беспроводной локальной сети.

Как и в методе CSMA/CD, в данном методе количество неудачных попыток передачи одного кадра ограничено, но стандарт 802.11 не дает точного значения этого верхнего предела. Когда верхний предел в N попыток достигнут, то кадр отбрасывается, а счетчик последовательных коллизий устанавливается в нуль. Этот счетчик также устанавливается в нуль, если кадр после некоторого количества неудачных попыток все же передается успешно.

В режиме доступа DFC применяются меры для *устранения эффекта скрытого терминала*. Для этого станция, которая хочет захватить среду и в соответствии с описанным алгоритмом начинает передачу кадра в определенном слоте, вместо кадра данных сначала посылает станции назначения короткий служебный кадр RTS (Request To Send — запрос на передачу). На этот запрос станция назначения должна ответить служебным кадром CTS (Clear To Send — свободна для передачи), после чего станция-отправитель посылает кадр данных. Кадр CTS должен оповестить о захвате среды те станции, которые находятся вне зоны сигнала станции-отправителя, но в зоне досягаемости станции-получателя, то есть являются скрытыми терминалами для станции-отправителя.

ПРИМЕЧАНИЕ

Максимальная длина кадра данных 802.11 равна 2346 байт, длина RTS-кадра — 20 байт, CTS-кадра — 14 байт. Так как RTS- и CTS-кадры гораздо короче, чем кадр данных, то потери данных в результате коллизии RTS- или CTS-кадров гораздо меньше, чем при коллизии кадров данных. Процедура обмена RTS- и CTS-кадрами не обязательна. От нее можно отказаться при небольшой нагрузке сети, поскольку в такой ситуации коллизии случаются редко, а значит, не стоит тратить дополнительное время на выполнение процедуры обмена RTS- и CTS-кадрами.

Централизованный режим доступа PCF

В том случае, когда в BSS-сети имеется станция, выполняющая функции точки доступа, может применяться также централизованный режим доступа PCF, обеспечивающий приоритетное обслуживание трафика. В этом случае говорят, что точка доступа играет роль арбитра среды.

Режим доступа PCF в сетях 802.11 сосуществует с режимом DCF. Оба режима координируются с помощью трех типов межкадровых интервалов (рис. 14.13).

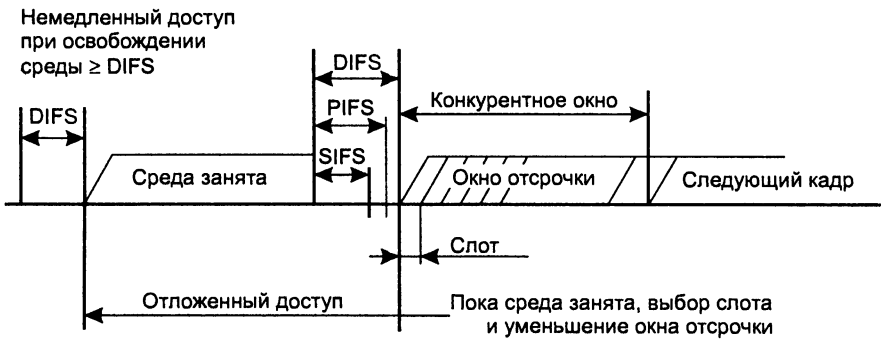


Рис. 14.13. Сосуществование режимов PCF и DCF

После освобождения среды каждая станция отсчитывает время простоя среды, сравнивая его с тремя значениями:

- короткий межкадровый интервал (Short IFS, **SIFS**);
- межкадровый интервал режима PCF (**PIFS**);
- межкадровый интервал режима DCF (**DIFS**).

Захват среды с помощью распределенной процедуры DCF возможен только в том случае, когда среда свободна в течение времени, равного или большего, чем DIFS. То есть в качестве IFS в режиме DCF нужно использовать интервал DIFS — самый длительный период из трех возможных, что дает этому режиму самый низкий приоритет.

Межкадровый интервал SIFS имеет наименьшее значение, он служит для первоочередного захвата среды ответными CTS-кадрами или квитанциями, которые продолжают или завершают уже начавшуюся передачу кадра.

Значение межкадрового интервала PIFS больше, чем SIFS, но меньше, чем DIFS. Промежутком времени между завершением PIFS и DIFS пользуется арбитр среды. В этом промежутке он может передать специальный кадр, который говорит всем станциям, что начинается **контролируемый период**. Получив этот кадр, станции, которые хотели бы воспользоваться алгоритмом DCF для захвата среды, уже не могут этого сделать, они должны дожидаться окончания контролируемого периода. Длительность этого периода объявляется в специальном кадре, но этот период может закончиться и раньше, если у станций нет чувствительного к задержкам трафика. В этом случае арбитр передает служебный кадр, после которого по истечении интервала DIFS начинает работать режим DCF.

На управляемом интервале реализуется *централизованный метод* доступа PCF. Арбитр выполняет процедуру опроса, чтобы по очереди предоставить каждой такой станции право на использование среды, направляя ей специальный кадр. Станция, получив такой кадр, может ответить другим кадром, который подтверждает прием специального кадра и одновременно передает данные (либо по адресу арбитра для транзитной передачи, либо непосредственно станции).

Для того чтобы какая-то доля среды всегда доставалась асинхронному трафику, длительность контролируемого периода ограничена. После его окончания арбитр передает соответствующий кадр и начинается неконтролируемый период.

Каждая станция может работать в режиме PCF, для этого она должна подписаться на эту услугу при присоединении к сети.

Безопасность

Разработчики стандарта IEEE 802.11 поставили перед собой цель — обеспечить безопасность передачи данных по беспроводной локальной сети, эквивалентную безопасности при передаче данных по проводной локальной сети, такой как Ethernet.

Можно заметить, что в технологии проводной локальной сети Ethernet нет каких-то особых мер обеспечения безопасности данных. В стандартах Ethernet не используется аутентификация пользователей или шифрование данных. Тем не менее проводные сети лучше защищены от несанкционированного доступа и нарушения конфиденциальности данных, чем беспроводные — только потому, что они являются проводными. Действительно, для того чтобы получить доступ к проводной сети, злоумышленник должен к ней физически присоединиться. Для этого он должен каким-то образом попасть в помещение, где имеются розетки, и присоединить свой компьютер к одной из них. Такое действие можно заметить и пресечь (хотя возможности для несанкционированного доступа к проводной сети все равно остаются большими).

В беспроводной сети несанкционированный доступ можно осуществить гораздо проще, достаточно оказаться в зоне распространения радиоволн этой сети. Для этого можно даже не входить в здание, где развернута сеть. Физическое подключение к среде в этом случае также не требуется, так что посетитель может принимать данные, не производя подозрительных действий, а просто имея работающий ноутбук в своей сумке.

В стандарте 802.11 предусмотрены средства безопасности, которые повышают защищенность беспроводной локальной сети до уровня обычной проводной локальной сети. Поэтому основной протокол защиты данных в сетях 802.11 так и называется — **WEP** (Wired Equivalent Privacy — секретность, эквивалентная проводной). Он предоставляет возможность шифровать данные, передаваемые через беспроводную среду, и тем самым обеспечивает их конфиденциальность. Технология 802.11 предлагает еще один механизм безопасности — механизм аутентификации — доказательство легальности пользователя, подключающегося к сети. Однако несовершенство средств безопасности 802.11 делают их популярной мишенью для критиков. Например, исследуя зашифрованный трафик 802.11, взломщик может расшифровать информацию в течение 24 часов.

Именно поэтому рабочая группа 802.11i занята в настоящее время разработкой более мощного стандарта защиты информации в сетях 802.11.

Персональные сети и технология Bluetooth

Список ключевых слов: персональная сеть, технология Bluetooth, пикосеть, главное устройство пикосети, подчиненное устройство пикосети, рассредоточенная сеть, синхронный канал, ориентированный на соединение, асинхронный канал, не ориентированный на соединение, профиль, уровень физических радиосигналов, уровень базового диапазона частот, диспетчер каналов, уровень протокола адаптации для управления логическим каналом, аудиоуровень, уровень управления, поле данных, код доступа, заголовок кадра.

Особенности персональных сетей

Персональные сети (Personal Area Networks, PAN) предназначены для взаимодействия устройств, принадлежащих одному владельцу, на небольшом расстоянии, обычно *в радиусе 10 м*. Такими устройствами могут быть ноутбук, мобильный телефон, принтер, карманный компьютер PDA (Personal Digital Assistant), телевизор, а также многочисленные бытовые приборы, например холодильник.

Персональные сети должны обеспечивать как фиксированный доступ, например, в пределах дома, так и мобильный, когда владелец устройств перемещается вместе с ними между помещениями или городами.

Персональные сети во многом похожи на локальные, но у них есть и свои особенности.

- Многие из устройств, которые могут входить в персональную сеть, *гораздо проще*, чем традиционный узел LAN — компьютер. Кроме того, такие устройства обычно имеют небольшие габариты и стоимость. Поэтому в стандартах PAN требуется учитывать, что их реализация должна приводить к недорогим решениям, потребляющим небольшую энергию.

- *Область покрытия PAN меньше области покрытия LAN*, для взаимодействия узлов PAN часто достаточно нескольких метров.
- *Высокие требования к безопасности.* Персональные устройства, путешествуя вместе со своим владельцем, попадают в различное окружение. Иногда они должны взаимодействовать с устройствами других персональных сетей, например, если их владелец встретил на улице своего знакомого и решил переписать из его устройства PDA в свое несколько адресов общих знакомых. В других случаях такое взаимодействие явно нежелательно, так как может привести к утечке конфиденциальной информации. Поэтому протоколы PAN должны обеспечивать разнообразные методы аутентификации устройств и шифрования данных в мобильной обстановке.
- При соединении малогабаритных устройств между собой желание избавиться от кабелей проявляется гораздо сильнее, чем при соединении компьютера с принтером или концентратором. Из-за этого персональные сети в гораздо большей степени *тяготеют к беспроводным решениям*, чем локальные.
- Если человек носит PAN-устройство постоянно с собой и на себе, то оно не должно причинять вред его здоровью. Поэтому такое устройство должно *излучать сигналы небольшой мощности*, желательно не более 100 мВт (обычный сотовый телефон излучает сигналы мощностью от 600 мВт до 3 Вт).

Сегодня самой популярной технологией PAN является **Bluetooth**, которая обеспечивает взаимодействие 8 устройств в разделяемой среде диапазона 2,4 МГц со скоростью передачи данных до 723 Кбит/с.

Архитектура Bluetooth

Стандарт Bluetooth разработан группой Bluetooth SIG (Bluetooth Special Interest Group), которая была организована по инициативе компании Ericsson. Стандарт Bluetooth также адаптирован рабочей группой IEEE 802.15.1 в соответствии с общей структурой стандартов IEEE 802.

В технологии Bluetooth используется концепция **пикосети**. Название подчеркивает небольшую область покрытия, от 10 до 100 м, в зависимости от мощности излучения передатчика устройства. В пикосеть может входить до 255 устройств, но только 8 из них могут в каждый момент времени быть активными и обмениваться данными. Одно из устройств в пикосети является **главным**, остальные — **подчиненными** (рис. 14.14).

Активное подчиненное устройство может обмениваться данными только с главным устройством, прямой обмен между подчиненными устройствами невозможен. Все подчиненные устройства данной пикосети, кроме семи активных, должны находится в режиме пониженного энергопотребления, в котором они только периодически прослушивают команду главного устройства для перехода в активное состояние.

Главное устройство отвечает за доступ к *разделяемой среде пикосети*, которая представляет собой нелицензируемые частоты диапазона 2,4 ГГц. Разделяемая среда передает данные со скоростью 1 Мбит/с, но из-за накладных расходов на заголовки пакетов и смену частот полезная скорость передачи данных

в среде не превышает 777 Кбит/с. Пропускная способность среды делится главным устройством между семью подчиненными устройствами на основе техники TDM.

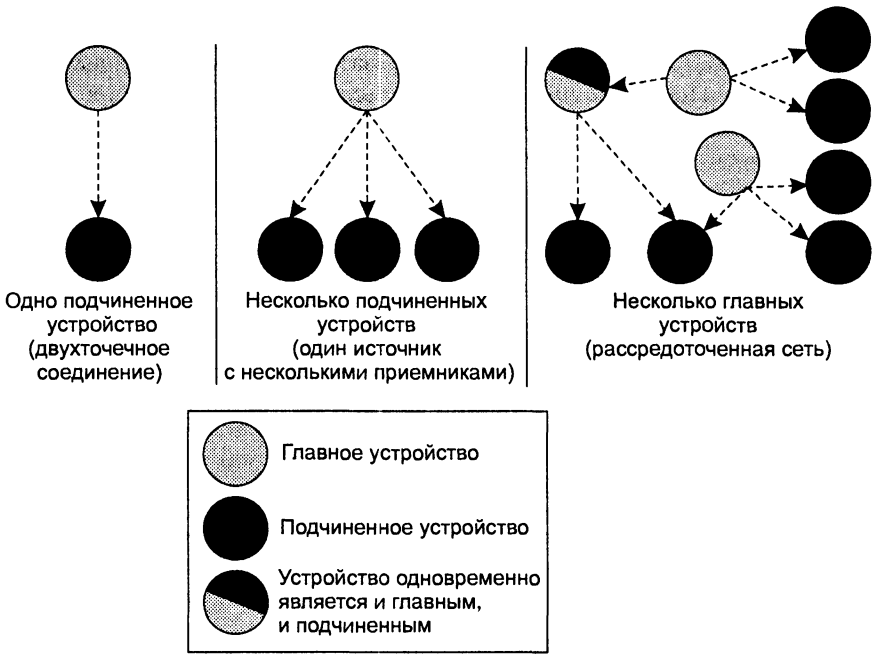


Рис. 14.14. Пикосеть и рассредоточенная сеть

Такая архитектура позволяет применять более простые протоколы в устройствах, выполняющих функции подчиненных (например, в радионаушниках), и отдает более сложные функции управления пикосетью компьютеру, который, скорее всего, будет главным устройством этой сети.

Присоединение к пикосети происходит динамически. Главное устройство пикосети, используя процедуру опроса, собирает информацию об устройствах, которые попадают в зону его пикосети. После обнаружения нового устройства главное устройство проводит с ним переговоры. Если желание подчиненного устройства присоединиться к пикосети совпадает с решением главного устройства (подчиненное устройство прошло проверку аутентичности и оказалось в списке разрешенных устройств), то новое подчиненное устройство присоединяется к сети.

ПРИМЕЧАНИЕ

Безопасность сетей Bluetooth обеспечивается за счет аутентификации устройств и шифрования передаваемого трафика. Протоколы Bluetooth обеспечивают более высокий уровень защиты, чем протокол WEP стандарта IEEE 802.11.

Несколько пикосетей, которые обмениваются между собой данными, образуют **рассредоточенную сеть**. Взаимодействие в пределах рассредоточенной сети осуществляется за счет того, что один узел (называемый *мостом*) одновременно является членом нескольких пикосетей, причем этот узел может исполнять роль главного устройства одной пикосети и подчиненного устройства другой.

Для того чтобы сигналы разных пикосетей не интерферировали, каждое главное устройство использует *собственную* последовательность псевдослучайной перестройки частоты. Использование отличающихся последовательностей псевдослучайной перестройки частоты затрудняет общение пикосетей между собой. Для преодоления этой проблемы устройство, играющее роль моста, должно при подключении к каждой из пикосетей соответствующим образом менять частоту.

Коллизии, хотя и с очень небольшой вероятностью, все же могут происходить, когда два или более устройства из разных пикосетей выберут для работы один и тот же частотный канал.

Как видно из описания, рассредоточенная сеть реализует метод доступа CDMA на основе техники FHSS. Для надежной передачи данных в технологии Bluetooth может выполняться прямая коррекция ошибок FEC, а получение кадра подтверждается с помощью квитанций.

Сети Bluetooth используют разные методы для передачи информации двух типов.

- Для *чувствительного к задержкам трафика* (например, голоса) сеть поддерживает **синхронный канал, ориентированный на соединение** (Synchronous Connection-Oriented link, SCO), работающий со скоростью 64 Кбит/с, Для канала SCO пропускная способность резервируется на все время соединения.
- Для *эластичного трафика* (например, компьютерных данных) используется работающий с переменной скоростью **асинхронный канал, не ориентированный на соединение** (Asynchronous Connection-Less link, ACL). Для канала ACL пропускная способность выделяется по запросу подчиненного устройства или по потребности главного устройства.

Стек протоколов Bluetooth

Bluetooth является законченной оригинальной технологией, рассчитанной на самостоятельное применение в электронных персональных устройствах. Поэтому эта технология поддерживает полный стек протоколов, включая собственные прикладные протоколы. В этом заключается ее отличие от рассмотренных ранее технологий, таких как Ethernet или IEEE 802.11, которые выполняют только функции физического и канального уровней.

Создание для технологии Bluetooth собственных прикладных протоколов объясняется стремлением разработчиков реализовывать ее в разнообразных простых устройствах, которым не под силу, да и не к чему поддерживать стек протоколов TCP/IP. Кстати, технология Bluetooth появилась в результате попыток разработать стандарт для взаимодействия мобильного телефона с беспроводными наушниками. Понятно, что для решения такой простой задачи не нужен ни протокол передачи файлов (FTP), ни протокол передачи гипертекста (HTTP). В резуль-

тате для технологии Bluetooth был разработан оригинальный стек протоколов, в дополнение к которому появилось большое количество профилей.

Профили определяют конкретный набор протоколов для решения той или иной задачи. Например, существует профиль для взаимодействия компьютера или мобильного телефона с беспроводными наушниками. Имеется также профиль для тех устройств, которые могут передавать файлы (наушникам он, скорее всего, не потребуется, хотя будущее предвидеть сложно), профиль эмуляции последовательного порта RS-232 и т. д.

При приведении стандартов Bluetooth в соответствие к архитектуре стандартов IEEE 802 рабочая группа 802.15.1 ограничилась только так называемыми протоколами *ядра Bluetooth*, которые соответствуют функциям физического уровня и уровня MAC (рис. 14.15).

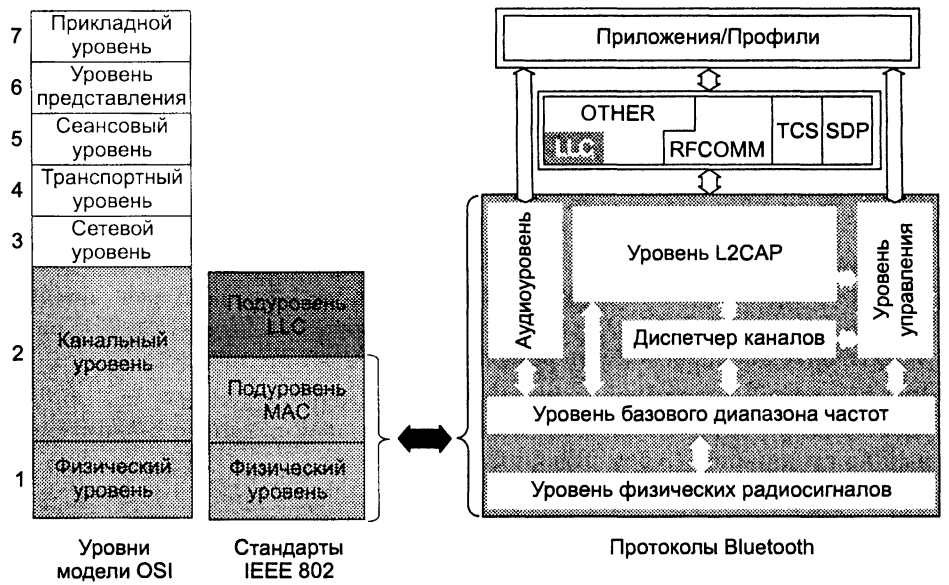


Рис. 14.15. Соответствие протоколов Bluetooth модели OSI и стандартам IEEE 802

- **Уровень физических радиосигналов** описывает частоты и мощности сигналов, используемых для передачи информации.
- **Уровень базового диапазона частот** отвечает за организацию каналов передачи данных в радиосреде. В его обязанности входят выбор последовательности псевдослучайной перестройки частоты, синхронизация устройств в пикосети, формирование и передача кадров по установленным каналам SCO и ACL. Кадр Bluetooth имеет переменную длину, поле данных может содержать от 0 до 2744 бит (343 байт). Для передачи голоса используются кадры фиксированного размера с полем данных 240 бит (30 байт).
- **Диспетчер каналов** отвечает за аутентификацию устройств и шифрование трафика, а также управляет статусом устройств, то есть может сделать подчиненное устройство главным, и наоборот.

- **Уровень протокола адаптации для управления логическим каналом** (Logical Link Control Adaptation Layer, L2CAP) является верхним уровнем протоколов ядра Bluetooth. Этот протокол используется только в тех случаях, когда устройство передает данные, голосовой трафик обходит этот протокол и обращается непосредственно к уровню базового диапазона частот. Уровень L2CAP принимает от протоколов верхнего уровня сегменты данных размером до 64 Кбайт и делит их на небольшие кадры для уровня базового диапазона частот. При приеме уровень L2CAP собирает кадры в исходный сегмент и передает протоколу верхнего уровня.
- **Аудиоуровень** обеспечивает передачу голоса по каналам SCO. На этом уровне применяется импульсно-кодовая модуляция (PCM), что определяет скорость голосового канала в 64 Кбит/с.
- **Уровень управления** передает внешнему блоку информацию о состоянии соединений и принимает от внешнего блока команды, изменяющие конфигурацию и состояние соединений.

Кадры Bluetooth

Разделяемая среда представляет собой последовательность частотных каналов технологии FHSS в диапазоне 2,4 ГГц. Каждый частотный канал имеет ширину 1 МГц, количество каналов равно 79 (в США и большинстве других стран мира) или 23 (в Испании, Франции, Японии).

Чиповая скорость равна 1600 Гц, поэтому период чипа составляет 625 мкс. Главное устройство разделяет общую среду на основе временно́го мультиплексирования (TDM), используя в качестве тайм-слота время пребывания системы на одном частотном канале, то есть 625 мкс. Информация кодируется с тактовой частотой 1 МГц путем двоичной частотной манипуляции (BFSK), в результате битовая скорость составляет 1 Мбит/с. В течение одного тайм-слота пикосеть Bluetooth передает 625 бит, но не все они используются для передачи полезной информации. При смене частоты устройствам сети требуется некоторое время для синхронизации, поэтому из 625 бит только 366 передают кадр данных.

Кадр данных может занимать 1, 3 или 5 слотов. В том случае, когда кадр занимает больше одного слота, частота канала остается неизменной в течение всего времени передачи кадра. В этом случае накладные расходы на синхронизацию меньше, так что размер кадра, состоящего, например, из 5 последовательных слотов, равен 2870 бит (с полем данных до 2744 бит).

ВНИМАНИЕ

Составными могут быть только кадры данных (то есть кадры канала ACL), а кадры, переносящие голос (кадры канала SCO), всегда состоят из одного слота.

Рассмотрим формат кадра, состоящего из одного слота — 366 бит (рис. 14.16):

- **Поле данных** занимает 240 бит.
- **Код доступа** (72 бита) используется для идентификации пикосети. Каждое Bluetooth-устройство имеет глобально уникальный 6-байтовый адрес, поэтому

для идентификации пикосети используется три младших байта уникального адреса главного устройства. Каждое устройство при формировании кадра помещает эти байты в поле кода доступа, дополняя их битами 1/3 для прямой коррекции ошибок (сокращение 1/3 говорит о том, что 1 бит информации преобразуется в 3 бита кода). Если главное или подчиненное устройство получает кадр, содержащий неверный код доступа, то оно отбрасывает этот кадр, считая, что он, скорее всего, получен из другой пикосети.

- **Заголовок кадра** (54 бита) содержит MAC-адрес, однобитовой признак подтверждения приема кадра, тип кадра, а также ряд признаков. MAC-адрес состоит из трех битов, это временный адрес одного их семи подчиненных устройств, при этом адрес 000 является широковещательным. Информация заголовка также передается с помощью битов 1/3 алгоритма FEC.

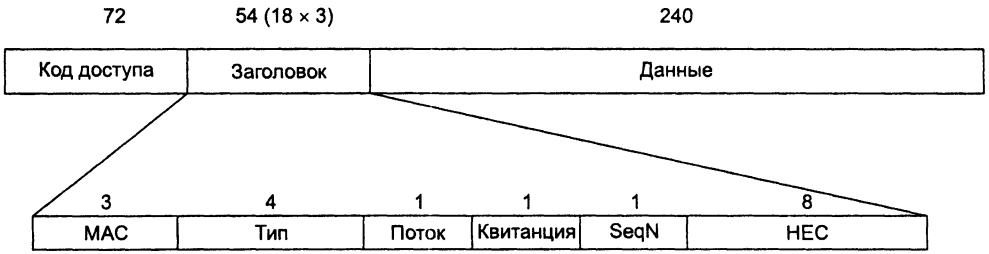


Рис. 14.16. Формат кадра Bluetooth, состоящего из одного слота

Формат кадра, состоящего из трех или пяти слотов, отличается только размером поля данных. Информация, помещаемая в поле данных, может кодироваться с помощью битов 1/3 или 2/3 алгоритма FEC либо передаваться вообще без прямой коррекции ошибок FEC.

Пример работы технологии Bluetooth

Рассмотрим работу пикосети на примере. Пусть пикосеть состоит из главного и трех активных подчиненных устройств. Для упрощения предположим, что все устройства используют кадры, занимающие один слот. На рис. 14.17 показано, каким образом главное устройство распределяет слоты между членами пикосети.

Для дуплексного обмена главное устройство всегда выделяет каждому каналу пару слотов: первый слот используется для передачи данных от главного устройства к подчиненному, а второй — в обратном направлении.

В примере, показанном на рисунке, существует один канал SCO между главным устройством и первым подчиненным устройством. Как мы уже знаем, каналам SCO всегда выделяется фиксированная часть пропускной способности среды, величина которой зависит от того, каким образом будет использоваться метод прямой коррекции ошибок (FEC) голосовой информации.

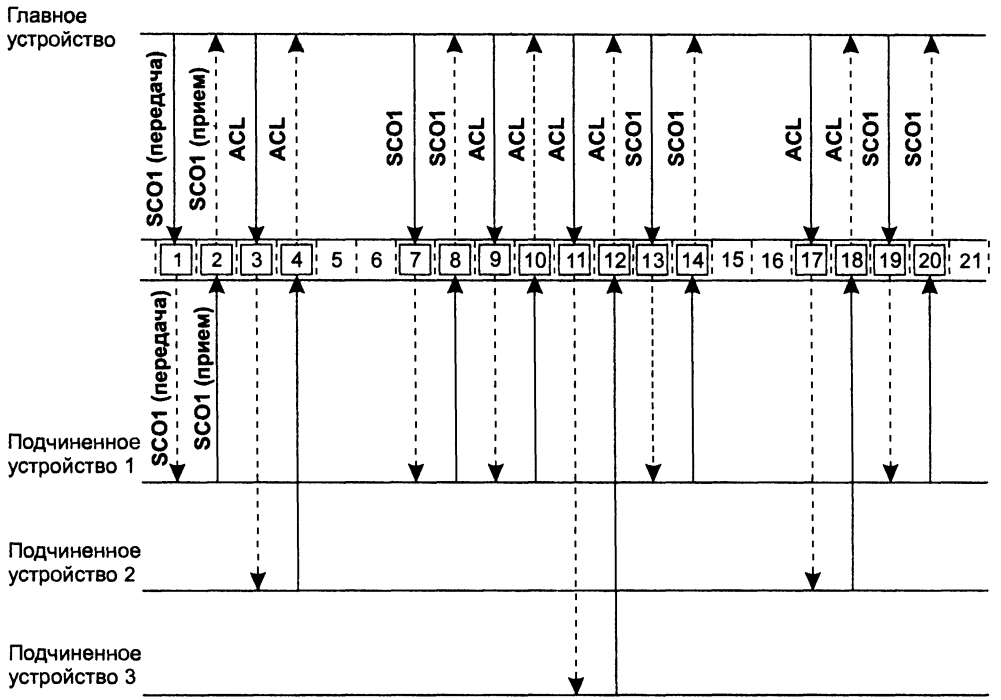


Рис. 14.17. Разделение среды

- Если алгоритм FEC не применяется, то для канала SCO выделяется каждая третья пара слотов, как это и показано на рисунке. Такое распределение слотов обеспечивает передачу 64 Кбит/с потоков в каждом направлении. Убедимся в этом. Кодек PCM оцифровывает голос с частотой 8 кГц (период 125 мкс), представляя каждый замер одним байтом. Каждый кадр переносит 30 байт (240 бит), то есть 30 замеров. Кадры канала SCO в одном направлении повторяются через каждые 6 слотов, поэтому период повторения кадров равен $6 \times 625 = 3750$ мкс. Соответственно, скорость передачи данных в канале SCO (в одном направлении) равна $240/(3750 \times 10^{-6}) = 64$ Кбит/с.
- В том случае, когда используются биты 2/3 алгоритма FEC, то в поле данных кадра размещается не 30, а 20 замеров, поэтому для обеспечения скорости 64 Кбит/с такому каналу SCO нужно выделять каждую вторую пару слотов.
- Наконец, биты 1/3 алгоритма FEC приводят к тому, что кадр переносит только 10 замеров голоса, так что такой канал занимает все слоты разделяемой среды.

Приведенные расчеты показывают, что в пикосети могут одновременно существовать не более трех каналов SCO (возможно, соединяющих с разными подчиненными устройствами), причем только тогда, когда канал не использует алгоритм FEC для снижения доли битовых ошибок. Прямая коррекция ошибок уменьшает число каналов SCO до двух или даже одного.

Оставшаяся от каналов SCO пропускная способность используется для передачи асинхронных данных. Для этого в пикосети имеется канал ACL. Этот канал соединяет один источник (главное устройство) с несколькими приемниками (все подчиненные устройства пикосети). Его не нужно устанавливать, он существует всегда.

Потребности подчиненных устройств в передаче асинхронных данных главное устройство узнает путем их периодического опроса. Для этого оно использует служебный кадр с MAC-адресом устройства. Если у главного устройства есть данные для этого подчиненного устройства, то оно может совместить передачу данных с опросом в одном кадре.

На рис. 14.17 показано, что главное устройство использовало слоты 3 и 4 для обмена кадрами со вторым подчиненным устройством, слоты 9 и 10 — для обмена с первым подчиненным устройством и слоты 11 и 12 — для обмена с третьим подчиненным устройством. Метод опроса исключает коллизии при доступе к каналу ACL, но скорость доступа к этому каналу для каждого отдельного устройства не определена, она зависит от количества устройств, которые хотят передавать асинхронные данные.

Таким образом, в сети Bluetooth совмещаются техники коммутации каналов (для каналов SCO) и коммутации пакетов (для канала ACL).

В том случае, когда каналы SCO в сети не используются, вся пропускная способность среды отводится каналу ACL. При наличии кадров, состоящих из 5 слотов, максимальная скорость передачи данных составляет 432,6 Кбит/с в каждом направлении (без прямой коррекции ошибок). Возможно также несимметричное деление пропускной способности канала ACL, тогда максимальная скорость достигает 723,2 Кбит/с в одном направлении при скорости 57,6 Кбит/с в обратном. Не нужно забывать, что это — суммарные скорости передачи данных в канале ACL, а не скорости потоков данных отдельных устройств. Когда несколько устройств используют канал, скорость делится между всеми устройствами.

Оборудование для локальных сетей с разделяемой средой

Список ключевых слов: сетевой адаптер, или сетевая интерфейсная карта, двухпортовый повторитель, многопортовый повторитель, правило 4-х хабов, автосегментация, поддержка резервных связей, защита от несанкционированного доступа, конфигурационная коммутация.

Концентраторы вместе с сетевыми адаптерами, а также кабельной системой представляют тот минимум оборудования, с помощью которого можно создать локальную сеть с разделяемой средой. Понятно, что такая сеть не может быть слишком большой, так как при большом количестве узлов общая среда передачи данных быстро становится узким местом, снижающим производительность сети. Поэтому концентраторы и сетевые адаптеры позволяют строить небольшие базо-

вые фрагменты сетей, которые затем должны объединяться друг с другом с помощью коммутаторов, мостов и маршрутизаторов.

Основные функции сетевых адаптеров

Сетевой адаптер, или **сетевая интерфейсная карта** (Network Interface Card, NIC), вместе со своим драйвером реализует канальный уровень модели OSI в конечном узле сети — компьютере. Точнее, в сетевой операционной системе пара адаптер-драйвер выполняет только функции физического уровня и уровня MAC, в то время как уровень LLC обычно реализуется модулем операционной системы, единым для всех драйверов и сетевых адаптеров. Например, в ОС Windows XP уровень LLC реализуется в модуле NDIS, общем для всех драйверов сетевых адаптеров независимо от того, какую технологию поддерживает драйвер.

Сетевой адаптер совместно с драйвером выполняют две операции: передачу и прием кадра.

Передача кадра из компьютера в кабель требует выполнения перечисленных ниже этапов.

1. Прием кадра данных уровня LLC через межуровневый интерфейс вместе с адресной информацией уровня MAC. Обычно взаимодействие между протоколами внутри компьютера происходит через буферы, расположенные в оперативной памяти. Данные для передачи в сеть помещаются в эти буферы протоколами верхних уровней, которые извлекают их из дисковой памяти либо из файлового кэша с помощью подсистемы ввода-вывода операционной системы.
2. Оформление кадра данных уровня MAC, в который инкапсулируется кадр уровня LLC. Заполнение адресов приемника и источника, вычисление контрольной суммы.
3. Формирование символов кодов при использовании избыточных кодов типа 4B/5B. Скремблирование кодов для получения более равномерного спектра сигналов. Этот этап выполняется не во всех протоколах, например, технология Ethernet 10 Мбит/с обходится без него.
4. Выдача сигналов в кабель в соответствии с принятым линейным кодом — манчестерским, NRZI, MLT-3 и т. п.

Прием кадра из кабеля в компьютер включает следующие действия.

1. Прием из кабеля сигналов, кодирующих битовый поток.
2. Выделение сигналов на фоне шума. Эту операцию могут выполнять различные специализированные микросхемы или процессоры DSP. В результате в приемнике адаптера образуется некоторая битовая последовательность, с большой степенью вероятности совпадающая с той, которая была послана передатчиком.
3. Если данные перед отправкой в кабель подвергались скремблированию, то они пропускаются через дескремблер, после чего в адаптере восстанавливаются символы кода, посланные передатчиком.

4. Проверка контрольной суммы кадра. Если контрольная сумма неверна, то кадр отбрасывается, а через межуровневый интерфейс наверх, протоколу LLC передается соответствующий код ошибки. Если контрольная сумма верна, то из MAC-кадра извлекается LLC-кадр и передается через межуровневый интерфейс наверх, протоколу LLC.

Распределение обязанностей между сетевым адаптером и его драйвером стандартами не определяется, поэтому каждый производитель решает этот вопрос самостоятельно. Обычно сетевые адаптеры делятся на адаптеры для клиентских компьютеров и адаптеры для серверов.

В адаптерах *для клиентских компьютеров* значительная часть работы перекладывается на драйвер, тем самым адаптер оказывается проще и дешевле. Однако при этом увеличивается степень загрузки центрального процессора компьютера рутинными работами по передаче кадров из оперативной памяти компьютера в сеть.

Адаптеры, предназначенные *для серверов*, обычно снабжаются собственными процессорами, которые самостоятельно выполняют большую часть работы по передаче кадров из оперативной памяти в сеть и обратно.

В зависимости от того, какой протокол реализует адаптер, они делятся на адаптеры Ethernet, Token Ring, FDDI и т. д. Так как протокол Fast Ethernet позволяет за счет процедуры автопереговоров автоматически выбрать скорость работы сетевого адаптера, то многие адаптеры Ethernet сегодня поддерживают две скорости работы и имеют в своем названии приставку 10/100.

В сетевых адаптерах реализована конвейерная схема обработки кадров, то есть процессы приема кадра из оперативной памяти компьютера и передачи его в сеть совмещаются во времени. Таким образом, после приема нескольких первых байтов кадра начинается их передача. Это существенно (на 25–55 %) повышает производительность цепочки *оперативная память — адаптер — физический канал — адаптер — оперативная память*. Такая схема очень чувствительна к порогу начала передачи, то есть к количеству байтов кадра, которое загружается в буфер адаптера перед началом передачи в сеть. Сетевой адаптер осуществляет самонастройку этого параметра путем анализа рабочей среды и расчета, без участия администратора сети. Самонастройка обеспечивает максимально возможную производительность для конкретного сочетания внутренней шины компьютера, его системы прерываний и системы прямого доступа к памяти.

Адаптеры базируются на специализированных интегральных схемах, что повышает производительность и надежность адаптера при одновременном снижении его стоимости.

ВНИМАНИЕ

Повышение производительности канала «память-адаптер» очень важно для повышения производительности сети в целом, так как скорость продвижения кадра по любому маршруту обработки, включающему, к примеру, концентраторы, коммутаторы, маршрутизаторы, глобальные каналы связи и т. п., всегда определяется производительностью самого медленного элемента этого маршрута. Следовательно, если сетевой адаптер сервера или клиентского компьютера работает медленно, никакие другие коммуникационные устройства не смогут повысить скорость работы сети.

Выпускаемые сегодня сетевые адаптеры можно отнести к четвертому поколению. В эти адаптеры обязательно входит интегральная схема ASIC (Application-Specific Integrated Circuit), выполняющая функции уровня MAC, а также большое количество высокоуровневых функций. В набор таких функций может входить поддержка агента удаленного мониторинга, схема приоритезации кадров, функции дистанционного управления компьютером и т. п. В серверных вариантах адаптеров почти обязательно наличие мощного процессора, разгружающего центральный процессор.

Основная функция концентраторов

Практически во всех современных технологиях локальных сетей определено устройство, которое имеет несколько равноправных названий — *концентратор*, *хаб*, *повторитель*. В зависимости от области применения этого устройства в значительной степени изменяется состав его функций и конструктивное исполнение. Неизменной остается только основная функция — *повторение кадра* либо на всех портах (как определено в стандарте Ethernet), либо только на некоторых портах, в соответствии с конкретным алгоритмом, определенным тем или иным стандартом.

Концентратор обычно имеет несколько портов, к которым с помощью отдельных физических сегментов кабеля подключаются конечные узлы сети — компьютеры. Концентратор объединяет отдельные физические сегменты сети в единую разделяемую среду, доступ к которой осуществляется в соответствии с одним из рассмотренных протоколов локальных сетей — Ethernet, Token Ring и т. п. Так как логика доступа к разделяемой среде существенно зависит от технологии, то для каждого типа технологии выпускаются свои концентраторы: Ethernet, Token Ring, FDDI.

Каждый концентратор выполняет некоторую *основную функцию*, определенную в соответствующем стандарте той технологии, которую он поддерживает. Помимо основной функции концентратор может выполнять некоторое количество *дополнительных функций*, которые либо вообще не определены в стандарте, либо являются факультативными. Например, концентратор Token Ring может выполнять функцию отключения некорректно работающих портов и перехода на резервное кольцо, хотя в стандарте такие его возможности не описаны. Концентратор оказался удобным устройством для выполнения дополнительных функций, облегчающих контроль и эксплуатацию сети.

Рассмотрим особенности реализации основной функции концентратора на примере концентраторов Ethernet.

В технологии Ethernet устройства, объединяющие несколько физических сегментов коаксиального кабеля в единую разделяемую среду, использовались давно и получили название «повторителей» по своей основной функции — повторению на всех своих портах сигналов, полученных на входе одного из портов. В сетях на основе коаксиального кабеля обычными являлись **двухпортовые повторители**, соединяющие только два сегмента кабеля, поэтому термин «концентратор» к ним обычно не применялся.

С появлением спецификации 10Base-T для витой пары повторитель стал неотъемлемой частью сети Ethernet, так как без него связь можно было организовать только между двумя узлами сети. **Многопортовые повторители** Ethernet на витой паре стали называть концентраторами, или хабами, так как в одном устройстве действительно концентрировались связи между большим количеством узлов сети. На рис. 14.18 показан типичный концентратор Ethernet, рассчитанный на образование небольших сегментов разделяемой среды. Он имеет 16 портов стандарта 10Base-T с разъемами RJ-45, а также один порт AUI для подключения внешнего трансивера. Обычно к порту AUI подключается трансивер, работающий на коаксиал или оптоволокно. С помощью этого трансивера концентратор подключается к магистральному кабелю, соединяющему несколько концентраторов между собой. Таким же образом обеспечивается подключение станции, удаленной от концентратора более чем на 100 м.

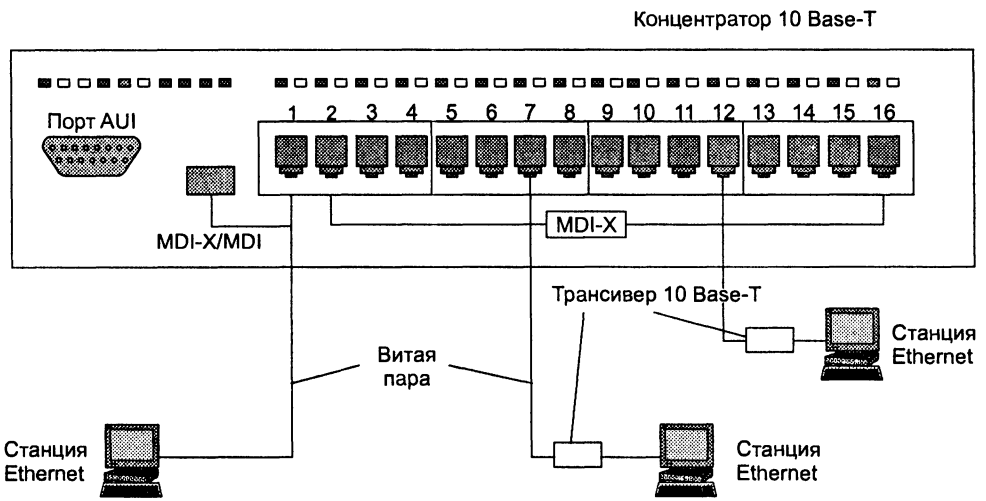


Рис. 14.18. Концентратор Ethernet

ПРИМЕЧАНИЕ

Для соединения концентраторов 10Base-T в иерархическую систему можно применять те же порты, что и для подключения конечных станций, с учетом одного обстоятельства. Дело в том, что обычный порт RJ-45, предназначенный для подключения сетевого адаптера и называемый MDI-X (кроссированный интерфейс MDI), имеет инвертированную разводку контактов разъема, чтобы сетевой адаптер можно было подключить к концентратору с помощью стандартного соединительного кабеля, не кроссирующего контакты (рис. 14.19). В случае соединения концентраторов через стандартный порт MDI-X приходится использовать нестандартный кабель с перекрестным соединением пар. Поэтому некоторые изготовители снабжают концентратор выделенным портом MDI, в котором нет кроссирования пар. Таким образом, два концентратора можно соединить обычным некроссированным кабелем, если это делать через порт MDI-X одного концентратора и порт MDI второго. Чаще один и тот же порт концентратора может работать и как порт MDI-X, и как порт MDI, в зависимости от положения кнопочного переключателя, как это показано в нижней части рис. 14.19.

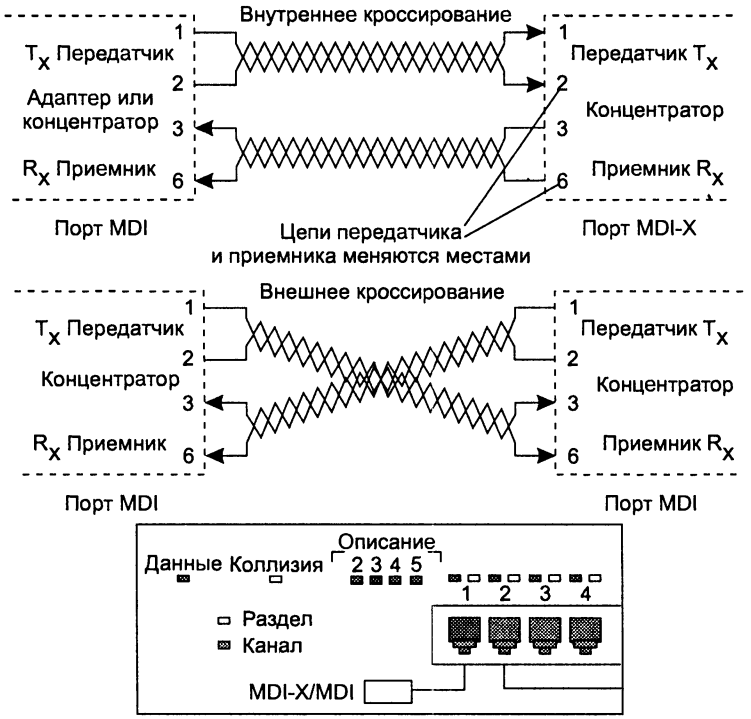


Рис. 14.19. Соединения типа «станция-концентратор» и «концентратор-концентратор» на витой паре

Многопортовый повторитель-концентратор Ethernet может по-разному рассматриваться при использовании **правила 4-х хабов**. В большинстве моделей все порты связаны с единственным блоком повторения, и при прохождении сигнала между двумя портами повторителя блок повторения вносит задержку всего один раз. Поэтому такой концентратор нужно считать одним повторителем с ограничениями, накладываемыми правилом 4-х хабов. Но существуют и другие модели повторителей, в которых на несколько портов имеется свой блок повторения. В таком случае каждый блок повторения нужно считать отдельным повторителем и учитывать его отдельно в правиле 4-х хабов.

Однако если существующие различия при выполнении основной функции концентраторов не столь велики, то их намного превосходит разброс в возможностях реализации концентраторами дополнительных функций.

Дополнительные функции концентраторов

У концентратора есть многочисленные функции, включая **автосегментацию** — способность отключать некорректно работающие порты, изолируя тем самым остальную часть сети от возникших в узле проблем[1]. Основной причиной отключения порта в стандартах Ethernet и Fast Ethernet является отсутствие ответа на

[1] Для концентратора FDDI эта функция является основной, так как определена в протоколе.

последовательность импульсов теста связности, посылаемых во все порты каждые 16 мс. В этом случае неисправный порт отключается, но импульсы теста связности будут продолжать посылаться в порт с тем, чтобы при восстановлении устройства работа с ним была продолжена автоматически.

Рассмотрим ситуации, в которых концентраторы Ethernet и Fast Ethernet выполняют отключение порта.

- *Ошибки на уровне кадра.* Если интенсивность прохождения через порт кадров, имеющих ошибки, превышает заданный порог, то порт отключается, а затем, при отсутствии ошибок в течение заданного времени, включается снова. Такими ошибками могут быть: неверная контрольная сумма, неверная длина кадра (больше 1518 байт или меньше 64 байт), неоформленный заголовок кадра.
- *Множественные коллизии.* Если концентратор фиксирует, что источником коллизии был один и тот же порт 60 раз подряд, то порт отключается. Через некоторое время порт снова будет включен.
- *Затянувшаяся передача.* Как и сетевой адаптер, концентратор контролирует время прохождения одного кадра через порт. Если это время превышает время передачи кадра максимальной длины в 3 раза, то порт отключается.

Поддержка резервных связей. Так как использование резервных связей в концентраторах определено только в стандарте FDDI, то для остальных стандартов разработчики концентраторов поддерживают такую функцию с помощью своих частных решений. Например, концентраторы Ethernet могут образовывать только иерархические связи без петель. Поэтому резервные связи всегда должны соединять отключенные порты, чтобы не нарушать логику работы сети. Обычно при конфигурировании концентратора администратор должен определить, какие порты являются основными, а какие по отношению к ним — резервными (рис. 14.20). Если по какой-либо причине порт отключается (срабатывает механизм автосегментации), концентратор делает активным его резервный порт.

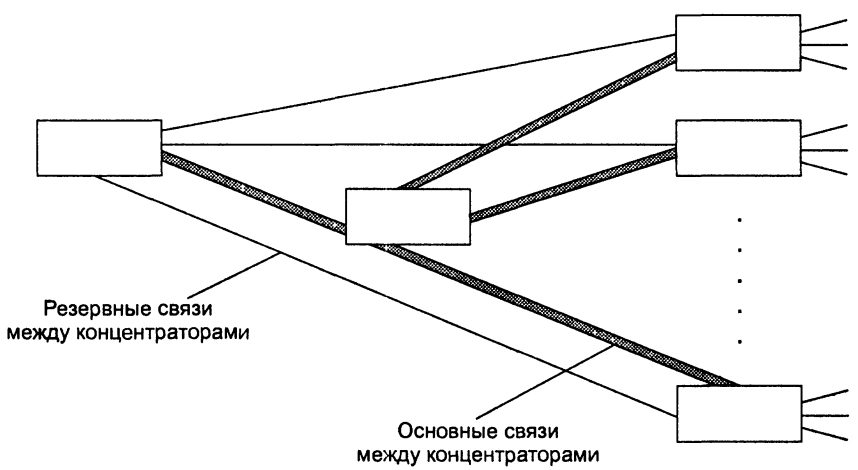


Рис. 14.20. Резервные связи между концентраторами Ethernet

В некоторых моделях концентраторов разрешается использовать механизм назначения резервных портов только для оптоволоконных портов. Это делается из предположения, что нужно резервировать только наиболее важные связи, обычно выполняемые на оптическом кабеле. В других же моделях резервным можно сделать любой порт.

Защита от несанкционированного доступа. Разделяемая среда предоставляет очень удобную возможность для несанкционированного прослушивания сети и получения доступа к передаваемым данным. Для этого достаточно подключить компьютер с программным анализатором протоколов к свободному разъему концентратора, записать на диск весь проходящий по сети трафик, а затем выделить из него нужную информацию.

Разработчики концентраторов предоставляют определенные средства защиты данных в разделяемых средах.

Наиболее простое средство — назначение разрешенных MAC-адресов портам концентратора. В стандартном концентраторе Ethernet порты MAC-адресов не имеют. Защита заключается в том, что администратор вручную связывает с каждым портом концентратора некоторый MAC-адрес. Этот MAC-адрес является адресом станции, которой разрешается подключаться к данному порту. Например, на рис. 14.21 первому порту концентратора назначен MAC-адрес 123 (условная запись), и компьютер с MAC-адресом 123 нормально работает с сетью через данный порт. Если злоумышленник отсоединяет этот компьютер и присоединяет вместо него свой, концентратор заметит, что при старте нового компьютера в сеть начали поступать кадры с адресом источника 789. Так как этот адрес является недопустимым для первого порта, то эти кадры фильтруются, порт отключается, а факт нарушения прав доступа может быть зафиксирован.

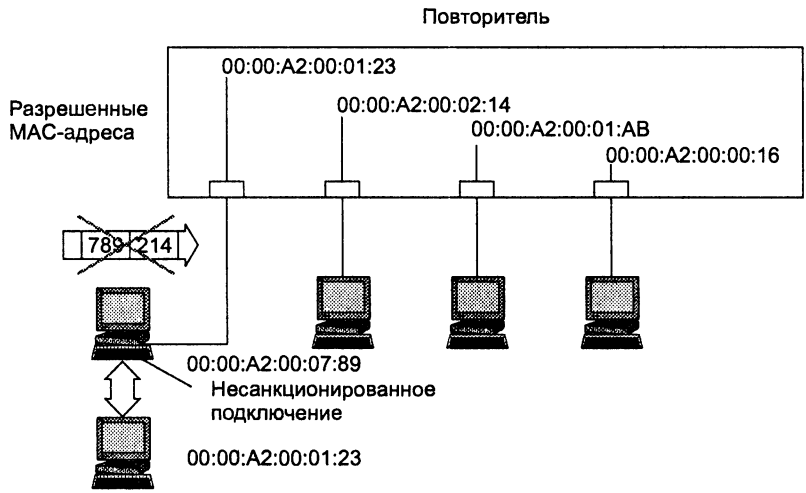


Рис. 14.21. Изоляция портов: передача кадров только от станций с фиксированными адресами

Заметим, что для реализации описанного метода защиты данных концентратор нужно предварительно сконфигурировать. Для этого концентратор должен иметь блок управления. Такие концентраторы обычно называют интеллектуальными. Блок управления представляет собой компактный вычислительный блок со встроенным программным обеспечением. Для взаимодействия администратора с блоком управления концентратор имеет консольный порт (чаще всего RS-232), к которому подключается терминал или персональный компьютер с программой эмуляции терминала. При присоединении терминала блок управления организует на его экране диалог, с помощью которого администратор вводит значения MAC-адресов. Блок управления может поддерживать и другие операции конфигурирования, например ручное отключение или включение портов и т. д. Для этого при подключении терминала блок управления выдает на экран некоторое меню, с помощью которого администратор выбирает нужное действие.

Другим средством защиты данных от несанкционированного доступа является их шифрование. Однако процесс реального шифрования требует большой вычислительной мощности, и для повторителя, не буферизующего кадр, выполнить шифрование «на лету» весьма сложно. Вместо этого в концентраторах применяется метод случайного искажения поля данных в пакетах, передаваемых портам с адресом, отличным от адреса назначения пакета. Этот метод сохраняет логику случайного доступа к среде, так как все станции видят занятость среды кадром данных, но только станция, которой послан этот кадр, может понять содержание поля данных кадра (рис. 14.22). Для реализации этого метода концентратор также нужно снабдить информацией о том, какие MAC-адреса имеют станции, подключенные к его портам. Обычно поле данных в кадрах, направляемых станциям, отличным от адресата, заполняется нулями.

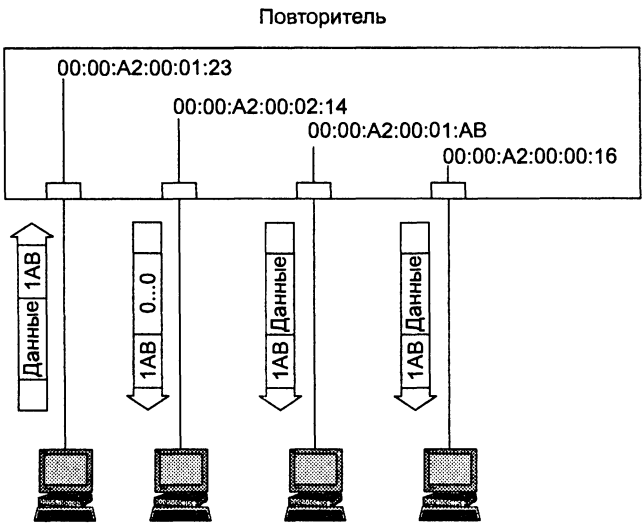


Рис. 14.22. Искажение поля данных в кадрах, не предназначенных для приема станциями

Многосегментные концентраторы

При рассмотрении некоторых моделей концентраторов возникает вопрос — зачем в них имеется такое большое количество портов, например 192 или 240? Имеет ли смысл разделять среду в 10 или 16 Мбит/с между таким большим количеством станций? Возможно, десять — пятнадцать лет назад ответ в некоторых случаях мог бы быть и положительным, например, для тех сетей, в которых компьютеры использовались сетью только для отправки небольших почтовых сообщений или для переписывания небольшого текстового файла. Сегодня таких сетей осталось крайне мало, и даже 5 компьютеров могут полностью загрузить сегмент Ethernet.

Для чего же тогда нужен концентратор с большим количеством портов, если ими практически нельзя воспользоваться из-за ограничений по пропускной способности, приходящейся на одну станцию?

Ответ состоит в том, что в таких концентраторах имеется несколько несвязанных внутренних шин, которые предназначены для создания нескольких разделяемых сред. Так, концентратор, представленный на рис. 14.23, имеет три внутренние шины Ethernet. Если, например, в таком концентраторе 72 порта, то каждый из этих портов может быть связан с любой из трех внутренних шин. На рисунке первые два компьютера связаны с шиной Ethernet 3, а третий и четвертый компьютеры — с шиной Ethernet 1. Первые два компьютера образуют один разделяемый сегмент, а третий и четвертый — другой разделяемый сегмент.

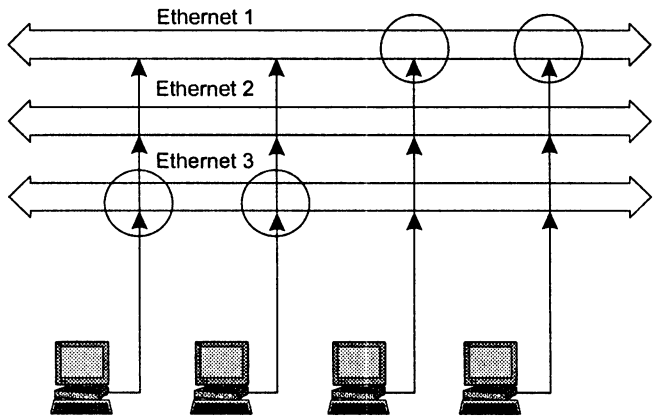


Рис. 14.23. Многосегментный концентратор

Между собой компьютеры, подключенные к разным сегментам, общаться через концентратор не могут, так как шины внутри концентратора никак не связаны.

Многосегментные концентраторы нужны для создания разделяемых сегментов, состав которых может легко изменяться. Большинство многосегментных концентраторов позволяют выполнять операцию соединения порта с одной из внутренних шин чисто программным способом, например путем локального конфи-

гурирования через консольный порт. В результате администратор сети может присоединять компьютеры пользователей к любым портам концентратора, а затем с помощью программы конфигурирования концентратора управлять составом каждого сегмента. Если завтра сегмент 1 окажется перегруженным, то его компьютеры можно распределить между оставшимися сегментами концентратора.

Возможность многосегментного концентратора программно изменять связи портов с внутренними шинами называется **конфигурационной коммутацией**.

ВНИМАНИЕ

Конфигурационная коммутация не имеет ничего общего с коммутацией кадров, которую выполняют мосты и коммутаторы.

Многосегментные концентраторы — это программируемая основа больших сетей. Для соединения сегментов между собой нужны устройства другого типа — мосты/коммутаторы или маршрутизаторы. Такие межсетевые устройства должны подключаться к нескольким портам многосегментного концентратора, подсоединенным к разным внутренним шинам, и выполнять передачу кадров или пакетов между сегментами точно так же, как если бы они были образованы отдельными устройствами-концентраторами.

Для крупных сетей многосегментный концентратор играет роль интеллектуального кроссового шкафа, который выполняет новое соединение не за счет механического перемещения вилки кабеля в новый порт, а за счет программного изменения внутренней конфигурации устройства.

Конструктивное исполнение концентраторов

На конструктивное устройство концентраторов большое влияние оказывает их область применения. Концентраторы рабочих групп чаще всего выпускаются как устройства с фиксированным количеством портов, корпоративные концентраторы — как модульные устройства на основе шасси, а концентраторы отделов могут иметь стековую конструкцию. Такое деление не является жестким, и в качестве корпоративного концентратора может использоваться, например, модульный концентратор.

Концентратор с фиксированным количеством портов — это наиболее простое конструктивное исполнение, когда устройство представляет собой отдельный корпус со всеми необходимыми элементами (портами, органами индикации и управления, блоком питания), и эти элементы заменять нельзя. Обычно все порты такого концентратора поддерживают одну среду передачи, общее количество портов изменяется от 4 до 48. Один порт может быть специально выделен для подключения концентратора к магистрали сети или же для объединения концентраторов (в качестве такого порта часто используется порт AUI, в этом случае применение соответствующего трансивера позволяет подключить концентратор к практически любой физической среде передачи данных).

Модульный концентратор выполняется в виде отдельных модулей с фиксированным количеством портов, устанавливаемых на общее шасси. Шасси имеет

внутреннюю шину для объединения отдельных модулей в единый повторитель. Часто такие концентраторы являются многосегментными, тогда в пределах одного модульного концентратора работает несколько несвязанных между собой повторителей. Для модульного концентратора могут существовать различные типы модулей, отличающиеся количеством портов и типом поддерживаемой физической среды. Модульные концентраторы позволяют более точно подобрать необходимую для конкретного применения конфигурацию концентратора, а также гибко и с минимальными затратами реагировать на изменения конфигурации сети.

Ввиду ответственной работы, которую выполняют корпоративные модульные концентраторы, они снабжаются модулем управления, системой терморегулирования, избыточными источниками питания и возможностью замены модулей «на лету».

Недостатком концентратора на основе шасси является высокая начальная стоимость такого устройства для случая, когда предприятию на первом этапе создания сети нужно установить всего 1–2 модуля. Высокая стоимость шасси вызвана тем, что оно поставляется вместе со всеми общими устройствами, такими как избыточные источники питания и т. п. Поэтому для сетей средних размеров большую популярность завоевали стековые концентраторы.

Стековый концентратор, как и концентратор с фиксированным числом портов, выполнен в виде отдельного корпуса без возможности замены отдельных его модулей. Типичный вид нескольких стековых концентраторов Ethernet показан на рис. 14.24.

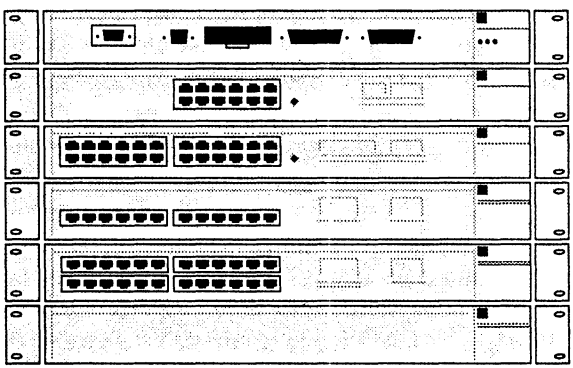


Рис. 14.24. Стековые концентраторы Ethernet

Стековые концентраторы имеют специальные порты и кабели для объединения нескольких таких корпусов в единый повторитель с общим блоком повторения (рис. 14.25), который обеспечивает общую ресинхронизацию сигналов для всех своих портов и поэтому с точки зрения правила 4-х хабов считается одним повторителем. Если стековые концентраторы имеют несколько внутренних шин, то при соединении в стек эти шины объединяются и становятся общими для всех устройств стека. Число объединяемых в стек корпусов может быть достаточно большим (обычно до 8, но бывает и больше). Стековые концентраторы могут поддерживать различные физические среды передачи, что делает их почти такими

же гибкими, как модульные концентраторы, но при этом стоимость этих устройств в расчете на один порт получается обычно ниже, так как сначала предприятие может купить одно устройство без избыточного шасси, а потом нарастить стек еще несколькими аналогичными устройствами.

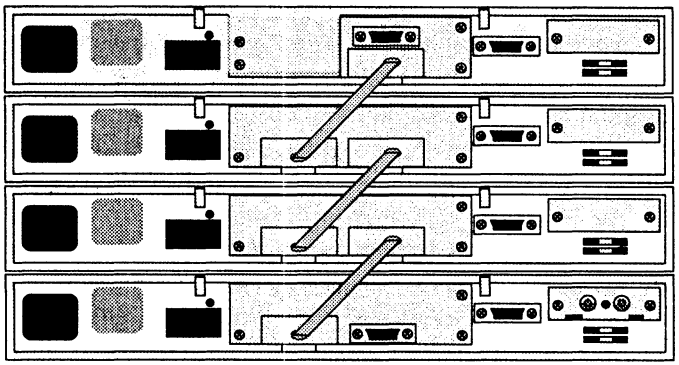


Рис. 14.25. Объединение стековых концентраторов в единое устройство с помощью специальных разъемов на задней панели

Стековые концентраторы, выпускаемые одним производителем, выполняются в едином конструктивном стандарте, что позволяет легко устанавливать их друг на друга, образуя единое настольное устройство, или помещать их в общую стойку. Экономия при организации стека происходит еще и за счет единого для всех устройств стека модуля управления (который вставляется в один из корпусов стека как дополнительный модуль), а также общего избыточного источника питания.

Модульно-стековые концентраторы представляют собой модульные концентраторы, объединенные специальными связями в стек. Как правило, корпуса таких концентраторов рассчитаны на небольшое количество модулей (1–3). Эти концентраторы сочетают достоинства концентраторов обоих типов (модульных и стековых).

Приведенная выше классификация конструктивного исполнения справедлива не только для концентраторов, но и для коммуникационных устройств всех типов — мостов и коммутаторов локальных сетей, коммутаторов глобальных сетей и маршрутизаторов. Нужно только отметить, что не для всех типов устройств стековая конструкция подразумевает столь тесное взаимодействие элементов стека, как у концентраторов. Часто устройства стека объединяют только общие блоки питания и управления, а основные функции устройства стека могут выполнять автономно.

Выводы

В сетях Token Ring используется детерминированный метод доступа с передачей токена. Логической топологией сетей Token Ring является кольцо, физической — звезда. Сети Token Ring работают на двух скоростях (4 и 16 Мбит/с) и могут использовать в качестве физической среды экранированную и неэкранированную витую пару, а также волоконно-оптический кабель.

Максимальное количество станций в кольце — 260, а максимальная длина кольца — 4 км. За счет кольцевой топологии технология Token Ring обеспечивает частичную отказоустойчивость.

Существует значительная преемственность между технологиями Token Ring и FDDI: обе обладают кольцевой топологией и применяют передачу токена в качестве метода доступа. Технология FDDI поддерживает развитые средства отказоустойчивости. При однократных отказах кабельной системы или одной из станций кольца сеть остается работоспособной за счет «свертывания» двойного кольца в одинарное.

В технологии FDDI в качестве физической среды впервые был использован волоконно-оптический кабель и достигнута скорость 100 Мбит/с. Максимальное количество станций двойного подключения в кольце FDDI — 500, максимальный диаметр двойного кольца — 100 км. Это делает технологию FDDI пригодной для работы не только в качестве технологии LAN, но и в качестве технологии MAN.

Беспроводные локальные сети позволяют избавиться от громоздкой кабельной системы и обеспечивают мобильность пользователей, но взамен требуют решения сложного комплекса проблем, связанных с высоким уровнем помех беспроводной среды и неопределенной зоной покрытия сети.

Стандарты IEEE 802.11 являются наиболее перспективными стандартами беспроводных локальных сетей. Существует несколько вариантов спецификаций физического уровня 802.11, отличающихся диапазоном используемых частот (2,4 и 5 ГГц), а также методом кодирования (FHSS, DSSS, OFDM). Физический уровень 802.11b обеспечивает передачу данных со скоростью до 11 Мбит/с.

Метод доступа 802.11 является комбинацией случайного метода доступа с предотвращением коллизий DCF и централизованного детерминированного метода доступа с опросом PCF. Гибкое применение режимов DCF и PCF позволяет обеспечить поддержку показателей QoS для синхронного и асинхронного трафиков.

Персональные сети (PAN) предназначены для взаимодействия устройств, принадлежащих одному владельцу, на небольшом расстоянии, обычно в радиусе от 10 до 100 м. Персональные сети должны обеспечивать как фиксированный доступ, например, в пределах дома, так и мобильный, когда владелец устройств перемещается вместе с ними между помещениями или городами.

Сегодня самой популярной технологией PAN является Bluetooth, использует концепцию пикосети. В пикосеть может входить до 255 устройств, но только 8 из них могут в каждый момент времени быть активными и обмениваться данными. Одно из устройств в пикосети является главным, остальные — подчиненными.

Несколько пикосетей, которые находятся в одной и той же области и обмениваются между собой данными, образуют рассредоточенную сеть. Пикосети, образующие рассредоточенную сеть, взаимодействуют между собой за счет того, что один узел (мост) одновременно является членом нескольких пикосетей.

Для чувствительного к задержкам трафика сеть Bluetooth поддерживает синхронные каналы, ориентированные на соединение (SCO), а для эластичного — асинхронные каналы, не ориентированные на соединение (ACL). Канал SCO обычно используется для передачи голосового трафика со скоростью 64 Кбит/с, а канал ACL — для передачи компьютерного трафика с переменной скоростью до 723 Кбит/с.

Концентраторы локальных сетей помимо основной функции протокола (побитного повторения кадра на всех портах или на следующем порту) всегда выполняют ряд полезных дополнительных функций: автосегментацию, поддержку резервных связей, защиту сети от несанкционированного доступа.

Вопросы и задания

1. Опишите алгоритм доступа к среде для технологии Token Ring.
2. Какие функции выполняет активный монитор?
3. За счет чего сеть Token Ring не теряет связность при отключении одного из компьютеров, входящих в кольцо?
4. Укажите максимально допустимую длину поля данных для технологий Ethernet, Token Ring, FDDI, Bluetooth.
5. Из каких соображений выбирается максимальное время оборота токена в сети Token Ring?
6. Какой элемент сети Token Ring восстанавливает синхронизацию потока битов?
7. В чем заключается преимущество механизма раннего освобождения токена?
8. В чем состоит сходство и различие технологий FDDI и Token Ring?
9. Какие элементы сети FDDI обеспечивают отказоустойчивость?
10. Технология FDDI является отказоустойчивой. Означает ли это, что при любом однократном обрыве кабеля сеть FDDI будет продолжать нормально работать?
11. К каким последствиям может привести двукратный обрыв кабеля в кольце FDDI?
12. Что произойдет, если в сети FDDI будет поврежден кабель SAS?
13. Какие методы кодирования сигналов используются в сетях IEEE 802.11?
14. Какой тип среды используется в рассредоточенной сети для передачи данных между BSS-сетями?
15. На что влияет эффект скрытого терминала?
16. Каким образом обнаруживает коллизии уровень MAC в сетях 802.11?
17. Может ли станция сети 802.11 передать кадр другой станции, входящей в ту же BSS-сеть, не непосредственно, а через точку доступа?
18. С какой целью в режиме DCF разрешенный для передачи кадров период времени делится на слоты? Из каких соображений выбирается длительность слота?
19. За счет чего режим PCF всегда имеет приоритет перед режимом DCF?
20. Как в технологии Bluetooth пикосети объединяются в рассредоточенную сеть?
21. Почему не все 625 бит тайм-слота Bluetooth используются для передачи кадра?
22. В каких случаях один кадр Bluetooth переносит данные одного, двух или трех каналов SCO?
23. Какие методы коммутации используются в технологии Bluetooth?
24. По каким причинам в технологии Bluetooth выбрана архитектура с главным и подчиненными устройствами?
25. Как влияет на производительность сети пропускная способность сетевого адаптера и пропускная способность порта концентратора?

26. Как концентратор поддерживает резервные связи?
27. В соответствии с основной функцией концентратора — повторением сигнала — его относят к устройствам, работающим на физическом уровне модели OSI. Приведите примеры дополнительных функций концентратора, для выполнения которых концентратору требуется информация протоколов более высоких уровней.
28. Чем модульный концентратор отличается от стекового?
29. Почему для соединения концентраторов между собой используются специальные порты?
30. Оцените максимальное время ожидания доступа к среде в сети Token Ring, состоящей из 160 станций и работающей на скорости 16 Мбит/с.
31. Сеть Token Ring состоит из 100 станций, длина кольца равна 2000 м. Скорость передачи данных составляет 16 Мбит/с. Время удержания токена выбрано равным 10 мс. Каждая станция передает кадры фиксированного размера в 4000 байт (с учетом заголовка) и полностью использует время удержания токена для передачи своих кадров. Подсчитайте, какой выигрыш дает механизм раннего освобождения токена для этой сети.
32. Сети IEEE 802.11 и Bluetooth работают в пределах одной и той же территории. Сеть 802.11 использует физическую спецификацию FHSS со скоростью передачи данных 1 Мбит/с. Сеть Bluetooth работает со стандартным для этой технологии значением чиповой скорости в 1600 Гц, а сеть 802.11 поддерживает чиповую скорость 50 Гц. Обе сети задействуют 79 каналов в диапазоне 2,4 ГГц.

    Определите долю кадров в каждой сети, которые были повреждены из-за одновременного использования двумя сетями одного и того же частотного канала. Для определенности считайте, что все данные в сети Bluetooth передаются кадрами размером в один слот, а в сети 802.11 применяются кадры максимальной длины.

ГЛАВА 15 Коммутируемые локальные сети

Разделяемая среда применяется в локальных сетях с момента появления первых сетей этого типа. Такой подход к использованию коммуникационного канала имеет несколько преимуществ, одним из которых является простота коммуникационного оборудования локальной сети. Однако наличие разделяемой среды связано и с очевидным недостатком — плохой масштабируемостью, так как производительность, отводимая каждому узлу сети, снижается пропорционально увеличению их числа.

Естественным решением проблемы масштабирования локальной сети является ее разбиение на сегменты, каждый из которых представляет собой отдельную разделяемую среду. Такая логическая сегментация выполняется с помощью мостов или коммутаторов локальных сетей. Мы уже познакомились с принципами логической структуризации в главе 3, а в данной главе изучим алгоритмы работы мостов и коммутаторов более детально.

Локальные сети, разделенные на логические сегменты, называют коммутируемыми локальными сетями. В том случае, когда сегмент состоит только из одного компьютера, подключенного непосредственно к порту коммутатора, его называют микросегментом. По сути, микросегмент уже не является разделяемой средой, это дуплексный канал, который передатчик порта компьютера или коммутатора использует по своему усмотрению, не разделяя его с другим передатчиком.

И хотя коммутируемая локальная сеть всегда представляет собой более дорогое решение, чем локальная сеть на разделяемой среде, она помимо масштабируемости обладает еще рядом преимуществ, которые мы обсудим в этой главе.

Логическая структуризация сети с помощью мостов и коммутаторов

Список ключевых слов: алгоритм прозрачного моста, адресная таблица, таблица фильтрации, или таблица маршрутизации, продвижение кадра, фильтрация, статическая запись, динамическая запись, затопление сети, широковещательный шторм, пользовательский фильтр, алгоритм покрывающего дерева.

Достоинства и недостатки сети на разделяемой среде

При построении небольших сетей, состоящих из 10–30 узлов, использование стандартных технологий на разделяемой среде приводит к экономичным и эффективным решениям, что проявляется в первую очередь в следующих свойствах:

- простая топология сети допускает легкое наращивание числа узлов (в небольших пределах);
- отсутствуют потери кадров из-за переполнения буферов коммуникационных устройств, так как сам метод доступа к разделяемой среде регулирует поток кадров и приостанавливает станции, слишком часто генерирующие кадры;
- простота протоколов обеспечивает низкую стоимость сетевых адаптеров, повторителей и концентраторов и сети в целом.

Однако справедливым является и другое утверждение — крупные сети, насчитывающие сотни и тысячи узлов, не могут быть построены на основе одной разделяемой среды даже при такой скоростной технологии, как Gigabit Ethernet. И не только потому, что проектировщик сети часто сталкивается с жесткими ограничениями максимальной длины сети, обусловленными особенностями метода доступа Ethernet. И не потому, что практически все технологии ограничивают количество узлов в разделяемой среде: все технологии семейства Ethernet — 1024 узлами, Token Ring — 260 узлами, а FDDI — 500 узлами.

Главная проблема сетей с разделяемой средой — дефицит пропускной способности.

Процессы, происходящие в локальных сетях на разделяемой среде, качественно могут быть описаны моделями массового обслуживания, в частности — моделью M/M/1, рассмотренной в главе 7. Разделяемая среда соответствует обслуживающему устройству этой модели, а кадры, генерируемые каждым компьютером сети, — заявкам на обслуживание. Очередь заявок в действительности распределяется по компьютерам сети, где кадры ожидают своей очереди на использование среды.

Модель M/M/1 не может адекватно отразить многие особенности локальных сетей на разделяемой среде, например коллизии, возникающие в Ethernet. Тем не менее она хорошо показывает качественную картину зависимости задержек доступа к среде от коэффициента использования среды. На рис. 15.1 показаны зави-

симости этого типа, полученные для сетей Ethernet, Token Ring и FDDI путем имитационного моделирования.

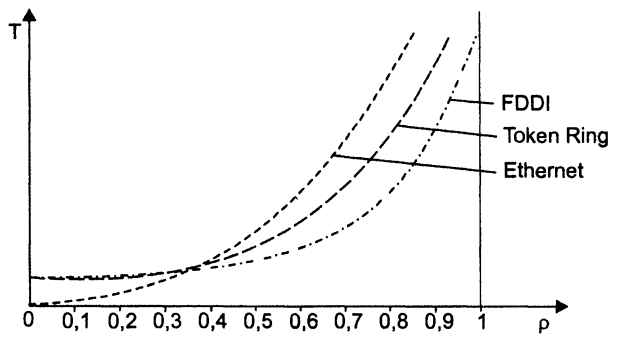


Рис. 15.1. Задержки доступа к среде передачи данных для технологий Ethernet, Token Ring и FDDI

Как видно из рисунка, всем технологиям присуща качественно одинаковая картина экспоненциального роста величины задержек доступа при увеличении коэффициента использования сети. Однако их отличает порог, при котором наступает резкий перелом в поведении сети, когда почти прямолинейная зависимость переходит в крутую экспоненциальную. Для всего семейства технологий Ethernet — это 30–50 % (сказывается эффект коллизий), для технологии Token Ring — 60 %, а технологии FDDI — 70–80 %.

Количество узлов, при которых коэффициент использования сети начинает приближаться к опасной границе, зависит от типа функционирующих в узлах приложений. Если раньше для сетей Ethernet считалось, что 30 узлов — это вполне приемлемое число для одного разделяемого сегмента, то сегодня, в условиях, когда мультимедийные приложения передают по сети большие файлы данных, предельное число узлов может составлять 5–10.

Преимущества логической структуризации сети

Ограничения, возникающие из-за использования одной разделяемой среды, можно преодолеть, выполнив *логическую структуризацию сети*, то есть сегментировать единую разделяемую среду на несколько и соединить полученные сегменты сети такими устройствами, как мосты, коммутаторы или маршрутизаторы (рис. 15.2).

Перечисленные устройства передают кадры с одного своего порта на другой, анализируя адрес назначения, помещенный в этих кадрах. Мосты и коммутаторы выполняют операцию передачи кадров на основе плоских адресов канального уровня, то есть MAC-адресов, а маршрутизаторы используют для этой цели иерархические адреса сетевого уровня. Изучение маршрутизаторов мы отложим до части IV, а сейчас сконцентрируем свое внимание на мостах и коммутаторах.

Мы уже коротко обсуждали необходимость логической структуризации в главе 3, а в этом разделе мы рассмотрим эту проблему более детально. Логическая струк-

туризация позволяет решить несколько задач, основные из них: повышение производительности, гибкости, безопасности и управляемости сети.

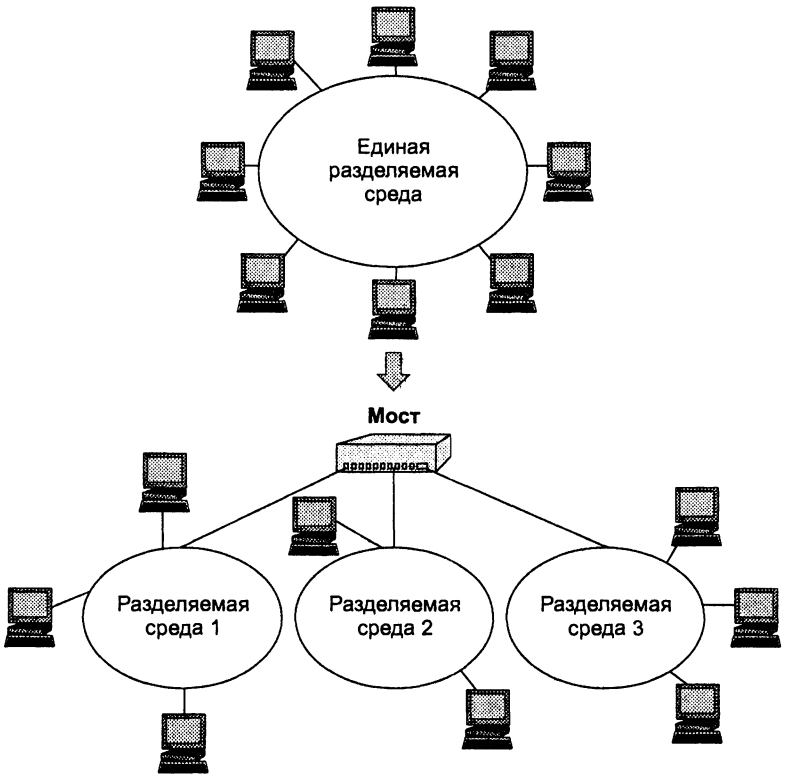


Рис. 15.2. Логическая структуризация сети

Повышение производительности. Для иллюстрации эффекта повышения производительности, который является главной целью логической структуризации, рассмотрим рис. 15.3. На нем показаны два сегмента Ethernet, соединенные мостом. Внутри сегментов имеются повторители. До деления сети на сегменты весь трафик, генерируемый узлами сети, был общим (представим, что вместо моста был повторитель) и учитывался при определении коэффициента использования сети. Если обозначить среднюю интенсивность трафика, идущего от узла i к узлу j, через C_{ij}, то суммарный трафик, который должна была передавать сеть до деления на сегменты, равен $C_{\Sigma} = \Sigma C_{ij}$ (считаем, что суммирование проводится по всем узлам).

После разделения сети подсчитаем нагрузку отдельно для каждого сегмента. Например, нагрузка сегмента S1 стала равна $C_{S1} + C_{S1\text{-}S2}$, где C_{S1} — внутренний трафик сегмента S1, а $C_{S1\text{-}S2}$ — межсегментный трафик. Чтобы показать, что нагрузка сегмента S1 стала меньше, чем нагрузка исходной сети, заметим, что общую нагрузку сети до разделения на сегменты можно представить в таком виде:

$$C_{\Sigma} = C_{S1} + C_{S1\text{-}S2} + C_{S2}.$$

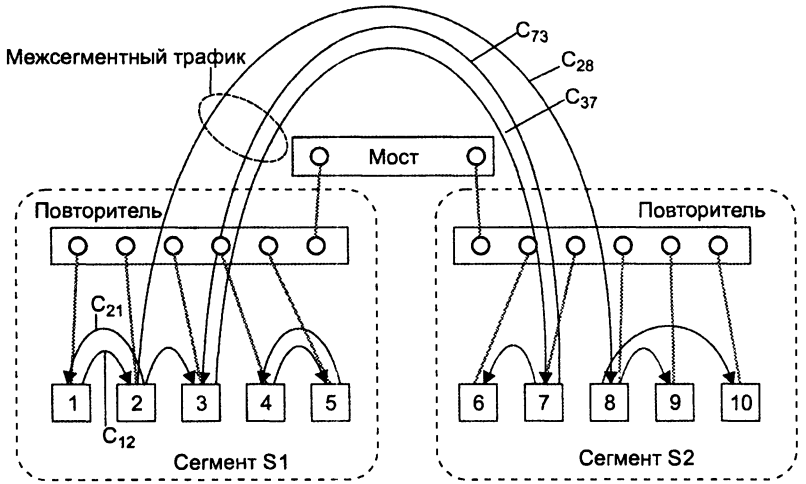


Рис. 15.3. Изменение нагрузки при делении сети на сегменты

Значит, нагрузка сегмента S1 после разделения стала равной $C\Sigma - C_{S2}$, то есть стала меньше на величину внутреннего трафика сегмента S2. Аналогичные рассуждения можно повторить относительно сегмента S2. Следовательно, в соответствии с графиками, приведенными на рис. 15.1, задержки в сегментах уменьшились, а полезная пропускная способность, приходящаяся на один узел, увеличилась.

Ранее было отмечено, что деление сети на логические сегменты *почти* всегда снижает нагрузку в новых сегментах. Слово «почти» учитывает очень редкий случай, когда сеть разбита на сегменты так, что внутренний трафик каждого сегмента равен нулю, то есть весь трафик является межсегментным. Для примера, представленного на рис. 15.3, это означало бы, что все компьютеры сегмента S1 обмениваются данными только с компьютерами сегмента S2, и наоборот.

На практике в сети всегда можно выделить группу компьютеров, которые принадлежат сотрудникам, решающим общую задачу. Это могут быть сотрудники одной рабочей группы, отдела, другого структурного подразделения предприятия. В большинстве случаев им нужен доступ к ресурсам сети их отдела и только изредка — доступ к удаленным ресурсам.

В 80-е годы существовало эмпирическое правило, говорящее о том, что можно разделить сеть на сегменты так, что 80 % трафика составят обращения к локальным ресурсам и только 20 % — к удаленным. Сегодня такая закономерность не всегда соответствует действительности, она может трансформироваться в правило 50 на 50 % и даже 20 на 80 % (например, большая часть обращений направлена к ресурсам Интернета или к централизованным серверам предприятия). Тем не менее в любом случае внутрисегментный трафик существует. Если его нет, значит, сеть разбита на логические сегменты неверно.

Повышение гибкости сети. При построении сети как совокупности сегментов каждый из них может быть адаптирован к специфическим потребностям рабочей группы или отдела. Например, в одном сегменте может использоваться техноло-

гия Ethernet и ОС Unix, в другом — Token Ring и OS-400. Вместе с тем, у пользователей обоих сегментов есть возможность обмениваться данными через мосты/коммутаторы. Процесс разбиения сети на логические сегменты можно рассматривать и в обратном направлении, как процесс создания большой сети из уже имеющихся небольших сетей.

Повышение безопасности данных. Устанавливая различные логические фильтры на мостах/коммутаторах, можно контролировать доступ пользователей к ресурсам других сегментов, чего не позволяют делать повторители.

Повышение управляемости сети. Побочным эффектом снижения трафика и повышения безопасности данных является упрощение управления сетью. Проблемы очень часто локализуются внутри сегмента. Сегменты образуют логические домены управления сетью.

Как уже не раз отмечалось, сеть можно разделить на логические сегменты с помощью устройств двух типов — мостов и коммутаторов. Сразу после появления коммутаторов в начале 90-х годов отделы маркетинга компаний, которые первые начали производить эти новые устройства, пытались создать у производителей впечатление, что мост и коммутатор — это принципиально разные устройства, что совершенно неверно.

Мост и коммутатор — это *функциональные* близнецы. Основное отличие коммутатора от моста заключается в том, что мост обрабатывает кадры последовательно, а коммутатор — параллельно.

Оба эти устройства продвигают кадры на основании одного и того же алгоритма, а именно **алгоритма прозрачного моста**, описанного в стандарте IEEE 802.1D.

Этот стандарт, разработанный задолго до появления первого коммутатора, описывал работу *моста*, поэтому совершенно естественно, что в его названии и содержании используется термин «мост». Некоторая путаница возникла, когда на свет появились первые модели коммутаторов — они выполняли тот же алгоритм продвижения кадров, описанный в стандарте IEEE 802.1D, который был с десяток лет отработан мостами. И хотя мосты, для которых алгоритм был разработан, сегодня уже относятся к практически «вымершему» виду коммуникационных устройств, в стандартах, описывающих работу коммутатора, следуя традиции, используют термин «мост». Мы же не будем столь консервативными, и при описании алгоритмов 802.1D в следующем разделе будем позволять себе иногда использовать термин «коммутатор», кроме тех случаев, когда речь пойдет об официальном названии стандарта или когда необходимо будет подчеркнуть разницу между двумя типами устройств.

Алгоритм прозрачного моста IEEE 802.1D

Слово «прозрачный» в названии *алгоритм прозрачного моста* отражает тот факт, что мосты и коммутаторы в своей работе не учитывают существование в сети сетевых адаптеров конечных узлов, концентраторов, повторителей. С другой стороны, и перечисленные выше сетевые устройства функционируют, «не замечая» присутствия в сети мостов и коммутаторов.

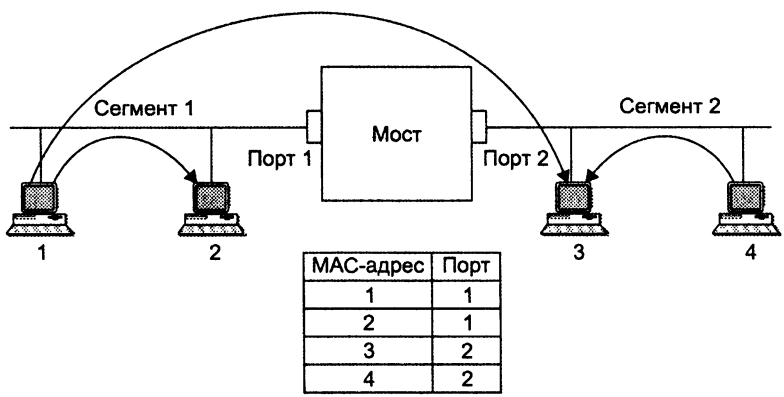
Алгоритм прозрачного моста не зависит от технологии локальной сети, в которой устанавливается мост/коммутатор, поэтому прозрачные мосты/коммутаторы Ethernet работают точно так же, как прозрачные мосты/коммутаторы FDDI или Token Ring.

Коммутатор строит свою адресную таблицу на основании пассивного наблюдения за трафиком, циркулирующим в подключенных к его портам сегментах. При этом коммутатор учитывает адреса источников кадров данных, поступающих на порты коммутатора. По адресу источника кадра коммутатор делает вывод о принадлежности узла-источника тому или иному сегменту сети.

ВНИМАНИЕ

Каждый порт коммутатора работает, как конечный узел своего сегмента, за одним исключением — порт коммутатора не имеет собственного MAC-адреса. Порты коммутатора не нуждаются в адресах, так как они работают в так называемом неразборчивом режиме захвата кадров, когда все поступающие на порт кадры, независимо от их адреса назначения, запоминаются на время в буферной памяти. Работая в неразборчивом режиме, коммутатор «слушает» весь трафик, передаваемый в присоединенных к нему сегментах, и использует проходящие через него кадры для изучения структуры сети.

Рассмотрим процесс автоматического создания адресной таблицы коммутатора и ее использования на примере простой сети, представленной на рис. 15.4.



MAC-адрес	Порт
1	1
2	1
3	2
4	2

Рис. 15.4. Принцип работы прозрачного моста/коммутатора

Коммутатор соединяет два сетевых сегмента. Сегмент 1 составляют компьютеры, подключенные с помощью одного отрезка коаксиального кабеля к порту 1 коммутатора, а сегмент 2 — компьютеры, подключенные с помощью другого отрезка коаксиального кабеля к порту 2 коммутатора. В исходном состоянии коммутатор не знает о том, компьютеры с какими MAC-адресами подключены к каждому из его портов. В этой ситуации коммутатор просто передает любой захваченный и буферизованный кадр на *все* свои порты за исключением того порта, от которого этот кадр получен. В нашем примере у коммутатора только два порта, поэтому он передает кадры с порта 1 на порт 2, и наоборот. Отличие работы коммутатора в этом режиме от повторителя заключается в том, что он передает кадр, предва-

рительно буферизуя его, а не бит за битом, как это делает повторитель. Буферизация разрывает логику работы всех сегментов как единой разделяемой среды. Когда коммутатор собирается передать кадр с сегмента на сегмент, например с сегмента 1 на сегмент 2, он, как обычный конечный узел, пытается получить доступ к разделяемой среде сегмента 2 по правилам алгоритма доступа, в данном примере — по правилам алгоритма CSMA/CD.

Одновременно с передачей кадра на все порты коммутатор изучает адрес источника кадра и делает запись о его принадлежности к тому или иному сегменту в своей **адресной таблице**. Эту таблицу также называют **таблицей фильтрации**, или **таблицей маршрутизации**. Например, получив на порт 1 кадр от компьютера 1, коммутатор делает первую запись в своей адресной таблице.

MAC-адрес 1 — порт 1.

Эта запись означает, что компьютер, имеющий MAC-адрес 1, принадлежит сегменту, подключенному к порту 1 коммутатора. Если все четыре компьютера данной сети проявляют активность и посылают друг другу кадры, то скоро коммутатор построит полную адресную таблицу сети, состоящую из 4 записей — по одной записи на узел (см. рис. 15.4).

При каждом поступлении кадра на порт коммутатора он, прежде всего, пытается найти адрес назначения кадра в адресной таблице. Продолжим рассмотрение действий коммутатора на примере (см. рис. 15.4).

1. При получении кадра, направленного от компьютера 1 компьютеру 3, коммутатор просматривает адресную таблицу на предмет совпадения адреса в какой-либо из ее записей с адресом назначения — MAC-адресом 3. Запись с искомым адресом имеется в адресной таблице.
2. Коммутатор выполняет второй этап анализа таблицы — проверяет, находятся ли компьютеры с адресами источника и назначения в одном сегменте. В примере компьютер 1 (MAC-адрес 1) и компьютер 3 (MAC-адрес 3) находятся в разных сегментах. Следовательно, коммутатор выполняет операцию **продвижения** (forwarding) кадра — передает кадр на порт 2, ведущий в сегмент получателя, получает доступ к сегменту и передает туда кадр.
3. Если бы оказалось, что компьютеры принадлежали одному сегменту, то кадр просто был бы удален из буфера. Такая операция называется **фильтрацией** (filtering).
4. Если бы запись MAC-адрес 3 отсутствовала в адресной таблице, то есть, другими словами, *адрес назначения был неизвестен* коммутатору, то он передал бы кадр на все свои порты, кроме порта — источника кадра, как и на начальной стадии процесса обучения.

Процесс обучения коммутатора никогда не заканчивается и происходит одновременно с продвижением и фильтрацией кадров. Коммутатор постоянно следит за адресами источника буферизуемых кадров, чтобы автоматически приспосабливаться к изменениям, происходящим в сети, — перемещениям компьютеров из одного сегмента сети в другой, отключению и появлению новых компьютеров.

Входы адресной таблицы могут быть динамическими, создаваемыми в процессе самообучения коммутатора, и статическими, создаваемыми вручную админист-

ратором сети. **Статические записи** не имеют срока жизни, что дает администратору возможность влиять на работу коммутатора, например, ограничивая передачу кадров с определенными адресами из одного сегмента в другой.

Динамические записи имеют срок жизни — при создании или обновлении записи в адресной таблице с ней связывается отметка времени. По истечении определенного тайм-аута запись помечается как недействительная, если за это время коммутатор не принял ни одного кадра с данным адресом в поле адреса источника. Это дает возможность коммутатору автоматически реагировать на перемещения компьютера из сегмента в сегмент — при его отключении от старого сегмента запись о его принадлежности к нему со временем вычеркивается из адресной таблицы. После подключения этого компьютера к другому сегменту его кадры начнут попадать в буфер коммутатора через другой порт, и в адресной таблице появится новая запись, соответствующая текущему состоянию сети.

Кадры с широковещательными MAC-адресами, как и кадры с неизвестными адресами назначения, передаются коммутатором на все его порты. Такой режим распространения кадров называется **затоплением сети** (flooding). Наличие коммутаторов в сети не препятствует распространению широковещательных кадров по всем сегментам сети. Однако это является достоинством только тогда, когда широковещательный адрес выработан корректно работающим узлом.

Нередко в результате каких-либо программных или аппаратных сбоев протокол верхнего уровня или сетевой адаптер начинают работать некорректно, а именно постоянно с высокой интенсивностью генерировать кадры с широковещательным адресом. Коммутатор в соответствии со своим алгоритмом передает ошибочный трафик во все сегменты. Такая ситуация называется **широковещательным штормом** (broadcast storm).

К сожалению, коммутаторы не защищают сети от широковещательного шторма, во всяком случае, по умолчанию, как это делают маршрутизаторы (вы познакомитесь с этим свойством маршрутизаторов в части IV). Максимум, что может сделать администратор с помощью коммутатора для борьбы с широковещательным штормом, — установить для каждого узла предельно допустимую интенсивность генерации кадров с широковещательным адресом. Но при этом нужно точно знать, какая интенсивность является нормальной, а какая — ошибочной. При смене протоколов ситуация в сети может измениться, и то, что вчера считалось ошибочным, сегодня может оказаться нормой.

На рис. 15.5 показана типичная структура коммутатора. Функции доступа к среде при приеме и передаче кадров выполняют микросхемы MAC, которые идентичны микросхемам сетевого адаптера.

Протокол, реализующий алгоритм коммутатора, располагается между уровнями MAC и LLC (рис. 15.6).

На рис. 15.7 показана копия экрана терминала с адресной таблицей модуля локального коммутатора. Терминал подключен к консольному порту, и информация на его экране высвечена модулем управления коммутатора.

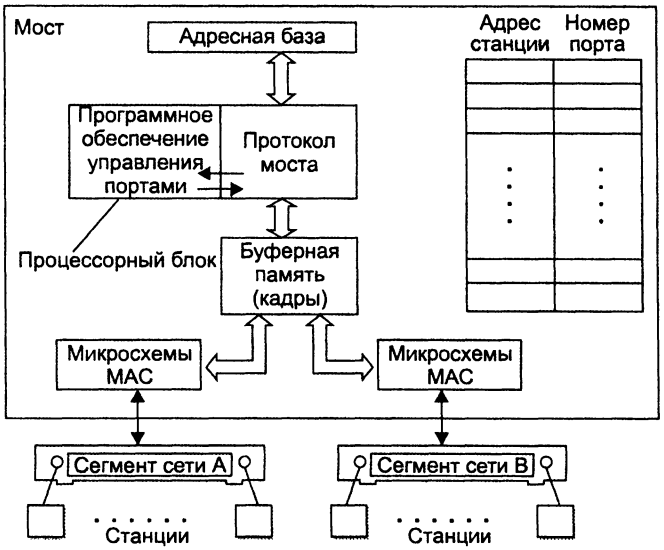


Рис. 15.5. Структура моста/коммутатора

Рис. 15.6. Место протокола коммутатора в стеке протоколов

```
                          Forwarding Table                    Page 1 of 1

Address          Dispn    Address          Dispn    Address          Dispn
 00608CB17E58    LAN B     0000810298D6    LAN A     02070188ACA     LAN A
 00008101C4DF    LAN B   + 000081016A52    LAN A   * 010081000100    Flood
* 010081000101   Discard   * 0180C2000000  Discard   * 000081FFD166    Flood

                                  Статус адреса:
                                  срок жизни записи истек

Exit      Next Page      Prev Page      Edit Table      Search Item      Go Page
          + Unlearned    * Static    Total Entries = 9    Static Entries = 4
          Use cursor keys to choose option. Press <RETURN> to select.
                    Press <CTRL> <P> to return to Main Menu
```

Рис. 15.7. Адресная таблица коммутатора

Из выводимой на экран адресной таблицы видно, что сеть состоит из двух сегментов — LAN A и LAN B. В сегменте LAN A имеются, по крайней мере, 3 станции, а в сегменте LAN B — 2 станции. Четыре адреса, помеченные звездочками, являются статическими, то есть назначенными администратором вручную. Адрес, помеченный плюсом, является динамическим адресом с истекшим сроком жизни.

Таблица имеет поле *Dispn* — «disposition» (это «распоряжение» коммутатору о том, какую операцию нужно проделать с кадром, имеющим данный адрес назначения). Обычно при автоматическом составлении таблицы в этом поле ставится условное обозначение порта назначения, но при ручном задании адреса в это поле можно внести нестандартную операцию обработки кадра. Например, операция *Flood* (затопление) заставляет коммутатор распространять кадр в широковещательном режиме, несмотря на то, что его адрес назначения не является широковещательным. Операция *Discard* (отбросить) говорит коммутатору, что кадр с таким адресом не нужно передавать на порт назначения. Вообще говоря, операции, задаваемые в поле *Dispn*, определяют особые условия фильтрации кадров, дополняющие стандартные условия их распространения. Такие условия обычно называют **пользовательскими фильтрами**, мы их рассмотрим немного позже, в подразделе «Фильтрация трафика».

Топологические ограничения коммутаторов в локальных сетях

Серьезным ограничением функциональных возможностей мостов и коммутаторов является отсутствие поддержки петлеобразных конфигураций сети.

Рассмотрим это ограничение на примере сети, показанной на рис. 15.8.

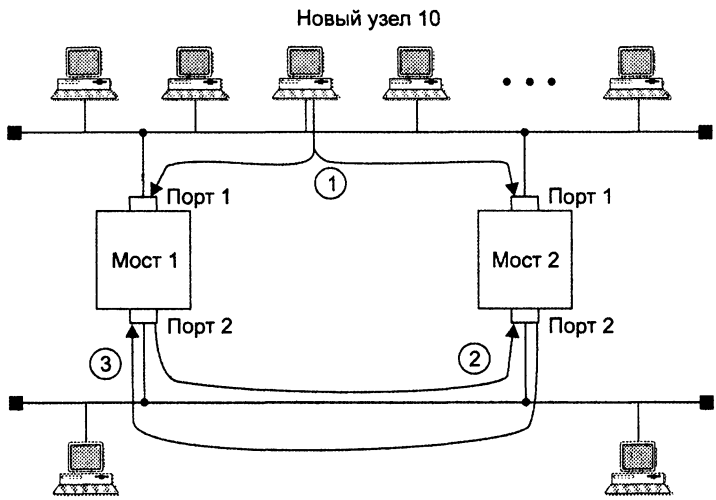


Рис. 15.8. Влияние замкнутых маршрутов на работу коммутаторов

Два сегмента Ethernet параллельно соединены двумя коммутаторами, так что образовалась петля. Пусть новая станция с MAC-адресом 123 впервые начинает работу в данной сети. Обычно начало работы любой операционной системы сопровождается рассылкой широковещательных кадров, в которых станция заявляет о своем существовании и одновременно ищет серверы сети.

На этапе 1 станция посылает первый кадр с широковещательным адресом назначения и адресом источника 123 в свой сегмент. Кадр попадает как в коммутатор 1, так и в коммутатор 2. В обоих коммутаторах новый адрес источника 123 заносится в адресную таблицу с пометкой о его принадлежности сегменту 1, то есть создается новая запись вида:

MAC-адрес 123 — Порт 1.

Так как адрес назначения широковещательный, то каждый коммутатор должен передать кадр на сегмент 2. Эта передача происходит поочередно в соответствии с методом случайного доступа технологии Ethernet. Пусть первым доступ к сегменту 2 получает коммутатор 1 (этап 2 на рис. 15.8). При появлении кадра на сегменте 2 коммутатор 2 принимает его в свой буфер и обрабатывает. Он видит, что адрес 123 уже есть в его адресной таблице, но пришедший кадр является более свежим, и он решает, что адрес 123 принадлежит сегменту 2, а не 1. Поэтому коммутатор 2 корректирует содержимое базы и делает запись о том, что адрес 123 принадлежит сегменту 2:

MAC-адрес 123 — Порт 2.

Аналогично поступает коммутатор 1, когда коммутатор 2 передает свою копию кадра на сегмент 2.

Ниже перечислены последствия наличия петли в сети.

- «Размножение» кадра, то есть появление нескольких его копий (в данном случае — двух, но если бы сегменты были соединены тремя коммутаторами — то трех и т. д.).
- Бесконечная циркуляция обеих копий кадра по петле в противоположных направлениях, а значит, засорение сети ненужным трафиком.
- Постоянная перестройка коммутаторами своих адресных таблиц, так как кадр с адресом источника 123 будет появляться то на одном порту, то на другом.

В целях исключения всех этих нежелательных эффектов коммутаторы нужно применять так, чтобы между логическими сегментами не было петель, то есть строить с помощью коммутаторов только древовидные структуры, гарантирующие наличие единственного пути между любыми двумя сегментами. Тогда кадры от каждой станции будут поступать в коммутатор всегда с одного и того же порта, и коммутатор сможет правильно решать задачу выбора рационального маршрута в сети.

В небольших сетях сравнительно легко гарантировать существование одного и только одного пути между двумя сегментами. Но когда количество соединений возрастает, то вероятность непреднамеренного образования петли оказывается высокой.

Возможна и другая причина возникновения петель. Так, для повышения надежности желательно иметь между коммутаторами резервные связи, которые не участвуют в нормальной работе основных связей по передаче информационных кадров станций, но при отказе какой-либо основной связи образуют новую связную рабочую конфигурацию без петель.

Избыточные связи необходимо блокировать, то есть переводить их в неактивное состояние. В сетях с простой топологией эта задача решается вручную, путем блокирования соответствующих портов коммутаторов. В больших сетях со сложными связями используются алгоритмы, которые позволяют решать задачу обнаружения петель автоматически. Наиболее известным из них является стандартный **алгоритм покрывающего дерева** (Spanning Tree Algorithm, STA), который будет детально рассмотрен в главе 16.

Коммутаторы

Список ключевых слов: процессор пакетов Ethernet, коммутационная матрица, коммутации «на лету», или «напролет», неблокирующий коммутатор, метод обратного давления, агрессивный захват среды, пользовательский фильтр, коммутатор с общей шиной, разделяемая многовходовая память, комбинированный коммутатор, скорость фильтрации, скорость продвижения, задержка передачи кадра, производительность коммутатора, максимальная емкость адресной таблицы.

Особенности коммутаторов

При появлении в конце 80-х начале 90-х годов быстрых протоколов, производительных персональных компьютеров, мультимедийной информации, разделении сети на большое количество сегментов классические *мосты* перестали справляться с работой. Обслуживание потоков кадров между теперь уже несколькими портами с помощью одного процессорного блока требовало значительного повышения быстродействия процессора, а это довольно дорогостоящее решение.

Более эффективным оказалось решение, которое и «породило» коммутаторы: для обслуживания потока, поступающего на каждый порт, в устройство ставился отдельный специализированный процессор, который реализовывал алгоритм моста. По сути, коммутатор — это мультипроцессорный мост, способный параллельно продвигать кадры сразу между всеми парами своих портов. Но если при добавлении процессорных блоков компьютер не перестали называть компьютером, а добавили только прилагательное «мультипроцессорный», то с мультипроцессорными мостами произошла метаморфоза — они превратились в коммутаторы. Этому способствовал способ связи между отдельными процессорами коммутатора — они связывались коммутационной матрицей, похожей на матрицы мультипроцессорных компьютеров, связывающие процессоры с блоками памяти.

Со временем коммутаторы вытеснили из локальных сетей классические однопроцессорные мосты. Основная причина этого — существенно более высокая

производительность, с которой коммутаторы передают кадры между сегментами сети. Если мосты могли даже замедлять работу сети, то коммутаторы всегда выпускаются с процессорами портов, которые могут передавать кадры с той максимальной скоростью, на которую рассчитан протокол. Добавление к этому возможности параллельной передачи кадров между портами предопределило судьбу мостов и коммутаторов.

Производительность коммутаторов на несколько порядков выше, чем мостов — коммутаторы могут передавать до нескольких миллионов кадров в секунду, в то время как мосты обычно обрабатывают 3–5 тысяч кадров в секунду.

За время своего существования уже без конкурентов-мостов коммутаторы вобрали в себя многие дополнительные функции, которые появлялись в результате естественного развития сетевых технологий. К этим функциям относятся, например, поддержка виртуальных сетей (VLAN), приоритезация трафика, использование магистрального порта по умолчанию и т. п.

Технология коммутации сегментов Ethernet была предложена небольшой компанией Kalpana в 1990 году в ответ на растущие потребности в повышении пропускной способности связей высокопроизводительных серверов с сегментами рабочих станций. У коммутатора компании Kalpana при свободном в момент приема кадра состоянии выходного порта задержка между получением первого байта кадра и появлением этого же байта на выходе порта адреса назначения составляла всего 40 мкс, что было гораздо меньше задержки кадра при его передаче мостом.

Структурная схема коммутатора EtherSwitch, предложенного фирмой Kalpana, представлена на рис. 15.9.

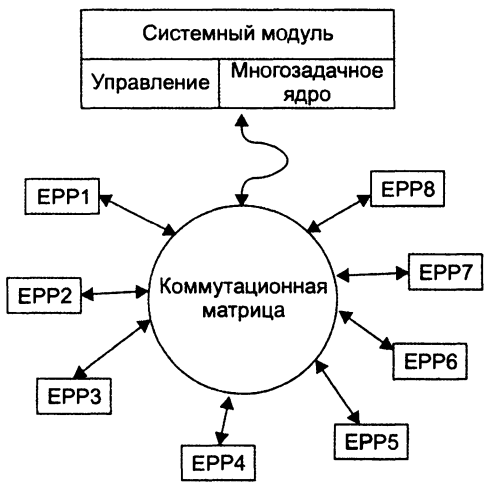


Рис. 15.9. Структура коммутатора EtherSwitch компании Kalpana

Каждый из 8 портов 10Base-T обслуживается одним **процессором пакетов Ethernet** (Ethernet Packet Processor, EPP). Кроме того, коммутатор имеет системный мо-

дуль, который координирует работу всех процессоров EPP, в частности ведет общую адресную таблицу коммутатора. Для передачи кадров между портами используется **коммутационная матрица**. Она функционирует по принципу коммутации каналов, соединяя порты коммутатора. Для 8 портов матрица может одновременно обеспечить 8 внутренних каналов при полудуплексном режиме работы портов и 16 — при дуплексном, когда передатчик и приемник каждого порта работают независимо друг от друга.

При поступлении кадра в какой-либо порт соответствующий процессор EPP буферизует несколько первых байтов кадра, чтобы прочитать адрес назначения. После получения адреса назначения процессор сразу же приступает к обработке кадра, не дожидаясь прихода остальных его байтов.

1. Процессор EPP просматривает свой кэш адресной таблицы, и если не находит там нужного адреса, обращается к системному модулю, который работает в многозадачном режиме, параллельно обслуживая запросы всех процессоров EPP. Системный модуль производит просмотр общей адресной таблицы и возвращает процессору найденную строку, которую тот буферизует в своем кэше для последующего использования.
2. Если адрес назначения найден в адресной таблице и кадр нужно отфильтровать, процессор просто прекращает записывать в буфер байты кадра, очищает буфер и ждет поступления нового кадра.
3. Если же адрес найден и кадр нужно передать на другой порт, процессор, продолжая прием кадра в буфер, обращается к коммутационной матрице, пытаясь установить в ней путь, связывающий его порт с портом, через который идет маршрут к адресу назначения.
4. Коммутационная матрица может это сделать только в том случае, когда порт адреса назначения в этот момент свободен, то есть не соединен с другим портом данного коммутатора.
5. Если же порт занят, то, как и в любом устройстве с коммутацией каналов, матрица в соединении отказывает. В этом случае кадр полностью буферизуется процессором входного порта, после чего процессор ожидает освобождения выходного порта и образования коммутационной матрицей нужного пути.
6. После того как нужный путь установлен, в него направляются буферизованные байты кадра, которые принимаются процессором выходного порта. Как только процессор выходного порта получает доступ к подключенному к нему сегменту Ethernet по алгоритму CSMA/CD, байты кадра сразу же начинают передаваться в сеть. Процессор входного порта постоянно хранит несколько байтов принимаемого кадра в своем буфере, что позволяет ему независимо и асинхронно принимать и передавать байты кадра (рис. 15.10).

Описанный выше способ передачи кадра без его полной буферизации получил название **коммутации «на лету»** (on-the-fly), или **«напролет»** (cut-through). Этот способ представляет собой, по сути, *конвейерную обработку* кадра, когда частично совмещаются во времени несколько этапов его передачи.

1. Прием первых байтов кадра процессором входного порта, включая прием байтов адреса назначения.

2. Поиск адреса назначения в адресной таблице коммутатора (в кэше процессора или в общей таблице системного модуля).
3. Коммутация матрицы.
4. Прием остальных байтов кадра процессором входного порта.
5. Прием байтов кадра (включая первые) процессором выходного порта через коммутационную матрицу.
6. Получение доступа к среде процессором выходного порта.
7. Передача байтов кадра процессором выходного порта в сеть.

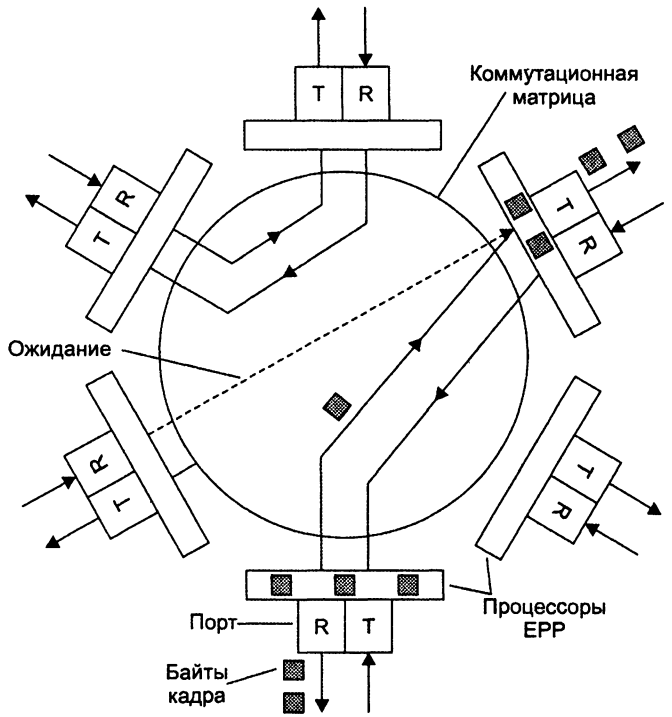


Рис. 15.10. Передача кадра через коммутационную матрицу

На рис. 15.11 подставлены два режима обработки кадра: режим коммутации «на лету» с частичным совмещением во времени нескольких этапов, и режим полной буферизации кадра с последовательным выполнением всех этапов. (Заметим, что этапы 2 и 3 совместить во времени нельзя, так как без знания номера выходного порта операция коммутации матрицы не имеет смысла.)

Как показывает схема, экономия от конвейеризации получается ощутимой.

Однако главной причиной повышения производительности сети при использовании коммутатора является *параллельная* обработка нескольких кадров.

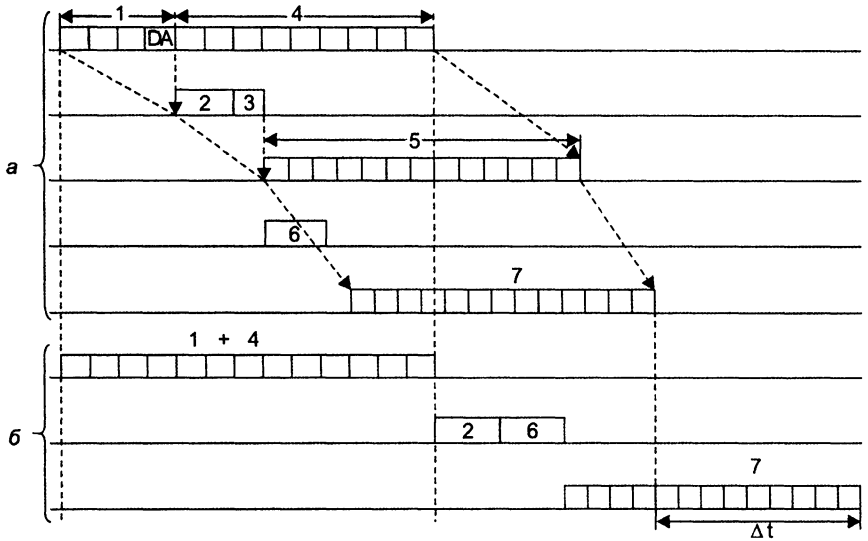


Рис. 15.11. Экономия времени при конвейерной обработке кадра: *а* — конвейерная обработка, *б* — обычная обработка с полной буферизацией

Этот эффект иллюстрирует рис. 15.12, на котором показана идеальная в отношении производительности ситуация, когда четыре порта из восьми передают данные с максимальной для протокола Ethernet скоростью 10 Мбит/с. Причем они передают эти данные на остальные четыре порта коммутатора не конфликтуя — потоки данных между узлами сети распределились так, что для каждого принимающего кадры порта есть свой выходной порт.

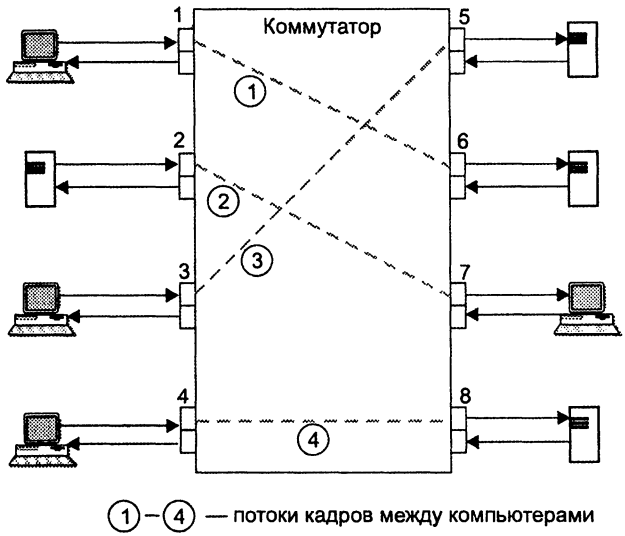


Рис. 15.12. Параллельная передача кадров коммутатором

Если коммутатор успевает обрабатывать входной трафик при максимальной интенсивности поступления кадров на входные порты, то общая производительность коммутатора в приведенном примере составит 4 × 10 = 40 Мбит/с, а при обобщении примера для N портов — (N/2) × 10 Мбит/с. В таком случае говорят, что *коммутатор предоставляет каждой станции или сегменту, подключенным к его портам, выделенную пропускную способность протокола.*

Естественно, что в сети не всегда складывается описанная ситуация. Если двум станциям, например станциям, подключенным к портам 3 и 4, одновременно нужно записывать данные на один и тот же сервер, подключенный к порту 8, то коммутатор не сможет выделить каждой станции по 10 Мбит/с, так как порт 8 не может передавать данные со скоростью 20 Мбит/с. Кадры станций будут ожидать во внутренних очередях входных портов 3 и 4, когда освободится порт 8 для передачи очередного кадра. Очевидно, хорошим решением для такого распределения потоков данных было бы подключение сервера к более высокоскоростному порту, например Fast Ethernet.

Неблокирующие коммутаторы

Коммутатор называют **неблокирующим**, если он может передавать кадры через свои порты с той же скоростью, с которой они на них поступают.

Когда говорят, что коммутатор может поддерживать *устойчивый неблокирующий режим* работы коммутатора, то имеют в виду, что коммутатор передает кадры со скоростью их поступления в течение произвольного промежутка времени. Для обеспечения подобного режима нужно таким образом распределить потоки кадров по выходным портам, чтобы: во-первых, порты справлялись с нагрузкой, во-вторых, коммутатор мог всегда в среднем передать на выходы столько кадров, сколько их поступило на входы. Если же входной поток кадров (просуммированный по всем портам) в среднем будет превышать выходной поток кадров (также просуммированный по всем портам), то кадры будут накапливаться в буферной памяти коммутатора, и при переполнении — просто отбрасываться.

Для поддержания устойчивого неблокирующего режима работы коммутатора необходимо, чтобы его производительность удовлетворяла условию $C_к = (\Sigma C_{pi})/2$, где $C_к$ — производительность коммутатора, C_{pi} — максимальная производительность протокола, поддерживаемого i-м портом коммутатора.

Суммарная производительность портов учитывает каждый проходящий кадр дважды — как входящий кадр и как выходящий, а так как в устойчивом режиме входной трафик равен выходному, то минимально достаточная производительность коммутатора для поддержки неблокирующего режима равна половине суммарной производительности портов. Если порт, например Ethernet 10 Мбит/с, работает в полудуплексном режиме, то производительность порта C_{pi} равна 10 Мбит/с, а если в дуплексном, — 20 Мбит/с.

Иногда говорят, что коммутатор поддерживает *мгновенный неблокирующий режим.* Это означает, что он может принимать и обрабатывать кадры от всех своих

портов на максимальной скорости протокола, независимо от того, обеспечиваются ли условия устойчивого равновесия между входным и выходным трафиком. Правда, обработка некоторых кадров при этом может быть неполной — при занятости выходного порта кадр помещается в буфер коммутатора.

Для поддержки мгновенного неблокирующего режима коммутатор должен обладать большей собственной производительностью, а именно она должна быть равна суммарной производительности его портов: $C_к = \Sigma C_{pi}$.

Первый коммутатор для локальных сетей не случайно появился для технологии Ethernet. Помимо очевидной причины, связанной с наибольшей популярностью сетей Ethernet, существовала и другая, не менее важная причина — эта технология больше других страдает от увеличения времени ожидания доступа к среде при повышении загрузки сегмента. Поэтому сегменты Ethernet в крупных сетях в первую очередь нуждались в средстве разгрузки узких мест сети, и этим средством стали коммутаторы фирмы Kalpana, а затем и других компаний.

Некоторые компании стали развивать технологию коммутации для повышения производительности других технологий локальных сетей, таких как Token Ring и FDDI. Внутренняя организация коммутаторов различных производителей иногда очень отличалась от структуры первого коммутатора EtherSwitch, однако принцип параллельной обработки кадров по всем портам оставался неизменным.

Широкому применению коммутаторов, безусловно, способствовало то обстоятельство, что внедрение технологии коммутации не требовало замены установленного в сетях оборудования — сетевых адаптеров, концентраторов, кабельной системы. Порты коммутаторов работали в обычном полудуплексном режиме, поэтому к ним прозрачно можно было подключить как конечный узел, так и концентратор, организующий целый логический сегмент. Так как коммутаторы и мосты прозрачны для протоколов сетевого уровня, то их появление в сети не оказало никакого влияния на маршрутизаторы сети, если они там имелись.

Борьба с перегрузками

В классическом полудуплексном режиме у коммутатора имеется возможность воздействовать на конечный узел с помощью алгоритма доступа к среде, который соседний узел обязан отрабатывать. Применяются два основных способа управления потоком кадров — обратное давление на конечный узел и агрессивный захват среды.

Метод обратного давления (backpressure) состоит в создании искусственных коллизий в сегменте, который чересчур интенсивно посылает кадры в коммутатор. Для этого коммутатор обычно использует jam-последовательность, отправляемую на выход порта, к которому подключен сегмент (или узел), чтобы приостановить его активность.

Другой метод «торможения» обычно применяется в том случае, когда соседом является конечный узел. Метод основан на так называемом **агрессивном захвате среды** либо после окончания передачи очередного кадра, либо после коллизии. Эти два случая иллюстрирует рис. 15.13.

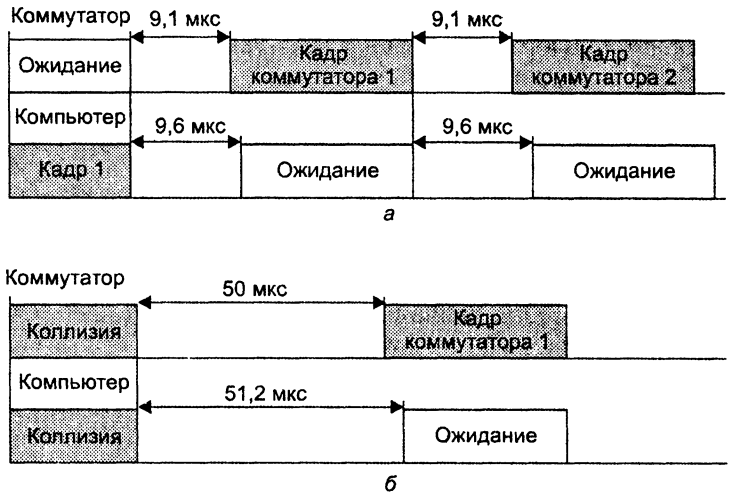


Рис. 15.13. Агрессивное поведение коммутатора при перегрузках буферов

В первом случае (рис. 15.13, *а*) коммутатор окончил передачу очередного кадра и вместо технологической паузы в 9,6 мкс сделал паузу в *9,1 мкс*, после чего начал передачу нового кадра. Компьютер не смог захватить среду, так как он выдержал стандартную паузу в 9,6 мкс и обнаружил после этого, что среда уже занята.

Во втором случае (рис. 15.13, *б*) кадры коммутатора и компьютера столкнулись, то есть была зафиксирована коллизия. Так как компьютер сделал паузу после коллизии в 51,2 мкс, как это положено по стандарту (интервал отсрочки равен 512 битовых интервалов), а коммутатор — *50 мкс*, то и в этом случае компьютеру не удалось передать свой кадр.

Коммутатор может пользоваться этим механизмом адаптивно, увеличивая степень своей агрессивности по мере необходимости.

Многие производители путем сочетания этих двух методов реализуют достаточно тонкие механизмы управления потоком кадров при перегрузках. Эти методы используют алгоритмы чередования передаваемых и принимаемых кадров (frame interleave). Алгоритм чередования должен быть гибким и позволять коммутатору в критических ситуациях на каждый принимаемый кадр передавать несколько своих, разгружая внутренний буфер кадров, причем не обязательно снижая при этом интенсивность приема кадров до нуля, а просто уменьшая ее до необходимого уровня.

Трансляция протоколов канального уровня

Коммутаторы в соответствии со спецификациями IEEE 802.1H и RFC 1042 могут выполнять трансляцию одного протокола канального уровня в другой, например Ethernet в FDDI, Fast Ethernet в Token Ring и т. п. Трансляцию протоколов локальных сетей облегчает тот факт, что наиболее сложную работу, а именно

работу по трансляции адресов, которую при объединении гетерогенных сетей выполняют маршрутизаторы и шлюзы, в данном случае выполнять не нужно.

Все конечные узлы локальных сетей имеют уникальные адреса одного и того же формата (MAC-адреса) независимо от поддерживаемого протокола.

Поэтому адрес сетевого адаптера Ethernet понятен сетевому адаптеру FDDI, и они оба могут использовать эти адреса в полях своих кадров, не задумываясь о том, что узел, с которым они взаимодействуют, принадлежит сети, работающей по другой технологии. Поэтому при согласовании протоколов локальных сетей коммутаторы просто переносят адреса приемника и источника из кадра одного протокола в кадр другого.

Помимо изменения порядка следования битов при передаче байтов адреса трансляция протокола Ethernet (и Fast Ethernet, который имеет формат кадров Ethernet) в протоколы FDDI и Token Ring включает выполнение перечисленных ниже (возможно, не всех) операций.

1. Вычисление длины поля данных кадра и помещение этого значения в поле длины при передаче кадра из сети FDDI или Token Ring в сеть Ethernet 802.3 (в кадрах FDDI и Token Ring поле длины отсутствует).
2. Заполнение полей статуса кадра при передаче кадров из сети FDDI или Token Ring в сеть Ethernet. Кадры FDDI и Token Ring имеют два бита, устанавливаемые станцией, которой предназначался кадр, — бит распознавания адреса A и бит копирования кадра C. При передаче коммутатором кадра в другую сеть нет стандартных правил для установки битов A и C в кадре, который возвращается по кольцу к станции-источнику. Поэтому производители коммутаторов решают эту проблему по своему усмотрению.
3. Отбрасывание кадров, передаваемых из сетей FDDI или Token Ring в сеть Ethernet с размером поля данных большим, чем 1500 байт, так как это максимально возможное значение поля данных для сетей Ethernet. В дальнейшем, не дождавшись ответа от станции назначения из сети Ethernet, протокол верхнего уровня станции из сети FDDI/Token Ring, возможно, уменьшит размер передаваемых в одном кадре данных, и тогда коммутатор сможет передавать кадры между этими станциями. Другим вариантом решения проблемы является поддержка коммутатором IP-фрагментации, но это требует, во-первых, реализации в коммутаторе протокола сетевого уровня, а во-вторых, поддержки протокола IP взаимодействующими узлами транслируемых сетей.
4. Заполнение поля *типа протокола* кадра Ethernet II при приходе кадров из сетей, поддерживающих кадры FDDI или Token Ring, в которых это поле отсутствует, зато имеются поля DSAP и SSAP того же назначения, но с другими кодами для обозначения протоколов. Для упрощения трансляции спецификация RFC 1042 предлагает всегда использовать в сетях FDDI и Token Ring кадры с заголовками LLC/SNAP, которые имеют то же поле типа и с теми же значениями, что и кадры Ethernet II. При преобразовании кадров значение из поля типа протокола заголовка LLC/SNAP переносится в поле типа протокола кадра Ethernet II, и наоборот. Если в сети Ethernet имеются форматы

кадров, отличные от Ethernet II, то они также должны иметь заголовок LLC/SNAP.

5. Пересчет контрольной суммы кадра в соответствии со сформированными значениями служебных полей кадра.

Фильтрация трафика

Многие модели коммутаторов позволяют администраторам задавать дополнительные условия фильтрации кадров наряду со стандартными условиями их фильтрации в соответствии с информацией адресной таблицы.

Пользовательские фильтры предназначены для создания дополнительных барьеров на пути кадров, позволяющих ограничивать доступ определенных групп пользователей к отдельным службам сети.

Наиболее простыми являются пользовательские фильтры на основе MAC-адресов станций. Так как MAC-адреса — это та информация, с которой работает коммутатор, то он позволяет создавать такие фильтры в удобной для администратора форме, возможно, проставляя некоторые условия в дополнительном поле адресной таблицы, подобно тем, которые были указаны в адресной таблице коммутатора на рис. 15.7 (например, условие отбрасывать кадры с определенным адресом). Таким способом пользователю, работающему на компьютере с данным MAC-адресом, полностью запрещается доступ к ресурсам другого сегмента сети.

Часто администратору требуется задать более тонкие условия фильтрации, например запретить некоторому пользователю печатать свои документы на сервере печати Windows, находящемся в чужом сегменте, а остальные ресурсы этого сегмента сделать доступными. Для реализации подобного фильтра нужно запретить передачу кадров, которые удовлетворяют следующим условиям: во-первых, имеют определенный MAC-адрес, во-вторых, содержат в поле данных SMB-пакеты, в-третьих, в соответствующем поле этих пакетов указан тип сервиса «печать». Коммутаторы не анализируют протоколы верхних уровней, такие как SMB, поэтому администратору приходится для задания условий фильтрации «вручную» определять поле, по значению которого нужно осуществлять фильтрацию. В качестве признака фильтрации администратор указывает пару «смещение-размер» относительно начала поля данных кадра канального уровня, а затем еще приводит шестнадцатеричное значение этого поля.

Сложные условия фильтрации обычно записываются в виде булевых выражений, формируемых с помощью логических операторов AND и OR.

Архитектура и конструктивное исполнение коммутаторов

Для ускорения операций коммутации сегодня все коммутаторы используют заказные специализированные БИС — ASIC, которые оптимизированы для выполнения основных операций коммутации. Часто в одном коммутаторе имеется несколько специализированных БИС, каждая из которых выполняет функцио-

нально законченную часть операций. Помимо процессорных микросхем для успешной неблокирующей работы коммутатору нужно также иметь быстродействующий *узел обмена* для передачи кадров между процессорными микросхемами портов.

В настоящее время в коммутаторах узел обмена строится на основе одной из трех схем:

- коммутационная матрица;
- общая шина;
- разделяемая многовходовая память.

Часто эти три схемы комбинируются в одном коммутаторе.

Коммутационная матрица обеспечивает наиболее простой способ взаимодействия процессоров портов, и именно этот способ был реализован в первом промышленном коммутаторе локальных сетей. Однако реализация матрицы возможна только для определенного числа портов, причем сложность схемы возрастает пропорционально квадрату количества портов коммутатора (рис. 15.14).

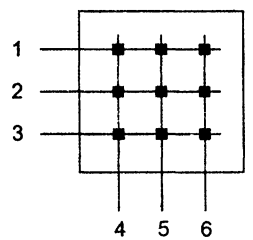


Рис. 15.14. Коммутационная матрица

Более детальное представление одного из возможных вариантов реализации коммутационной матрицы для восьми портов дано на рис. 15.15. Входные блоки процессоров портов на основании просмотра адресной таблицы коммутатора определяют по адресу назначения номер выходного порта. Эту информацию они добавляют к байтам исходного кадра в виде специального ярлыка — тега. Для данного примера тег представляет собой просто 3-разрядное двоичное число, соответствующее номеру выходного порта.

Матрица состоит из трех уровней двоичных переключателей, которые соединяют свой вход с одним из двух выходов в зависимости от значения бита тега. Переключатели первого уровня управляются первым битом тега, второго — вторым, а третьего — третьим.

Матрица может быть реализована и иначе, на основании комбинационных схем другого типа, но ее особенностью все равно остается технология коммутации физических каналов. Известным недостатком этой технологии является отсутствие буферизации данных внутри коммутационной матрицы — если составной канал невозможно построить из-за занятости выходного порта или промежуточного коммутационного элемента, то данные должны накапливаться в их источнике, в данном случае — во входном блоке порта, принявшего кадр. Основные досто-

инства таких матриц — высокая скорость коммутации и регулярная структура, которую удобно реализовывать в интегральных микросхемах. Зато после реализации матрицы N × N в составе БИС проявляется еще один ее недостаток — сложность наращивания числа коммутируемых портов.

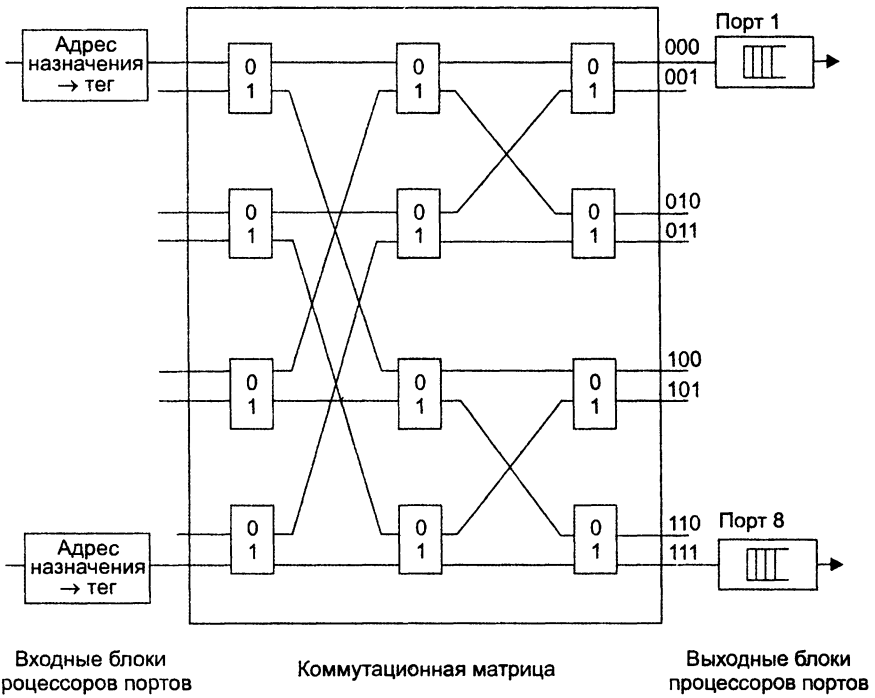


Рис. 15.15. Реализация коммутационной матрицы 8 × 8 с помощью двоичных переключателей

В **коммутаторах с общей шиной** процессоры портов связывают высокоскоростной шиной, используемой в режиме разделения времени.

Пример такой архитектуры приведен на рис. 15.16. Чтобы шина не блокировала работу коммутатора, ее производительность должна равняться, по крайней мере, сумме производительностей всех портов коммутатора. Для модульных коммутаторов характерно, что путем удачного подбора модулей с низкоскоростными портами можно обеспечить неблокирующий режим работы, и в то же время некоторые сочетания модулей с высокоскоростными портами могут приводить к структурам, у которых узким местом является общая шина.

Кадр должен передаваться по шине небольшими частями, по несколько байтов, чтобы передача кадров между портами происходила в псевдопараллельном режиме, не внося задержек в передачу кадра в целом. Размер такой ячейки данных определяется производителем коммутатора. Некоторые производители выбирают в качестве порции данных, переносимых за одну операцию по шине, ячейку ATM с ее полем данных в 48 байт. Такой подход облегчает трансляцию протоколов локальных сетей в протокол ATM, если коммутатор поддерживает эти технологии.

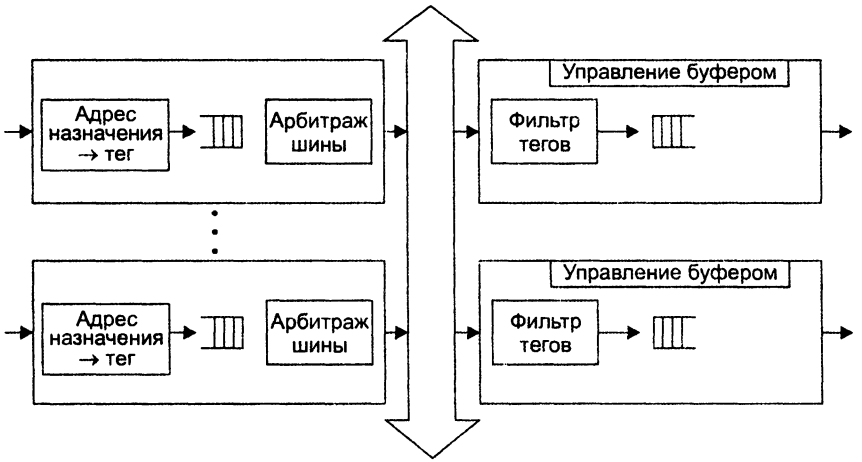


Рис. 15.16. Архитектура коммутатора с общей шиной

Входной блок процессора помещает в ячейку, переносимую по шине, тег, в котором указывает номер порта назначения. Каждый выходной блок процессора порта содержит фильтр тегов, который выбирает теги, предназначенные данному порту.

Шина, так же как и коммутационная матрица, не может осуществлять промежуточную буферизацию, но поскольку данные кадра разбиваются на небольшие ячейки, задержек с начальным ожиданием доступности выходного порта в такой схеме нет — здесь работает принцип коммутации пакетов, а не каналов.

Разделяемая многовходовая память представляет собой третью базовую архитектуру взаимодействия портов. Пример такой архитектуры приведен на рис. 15.17.

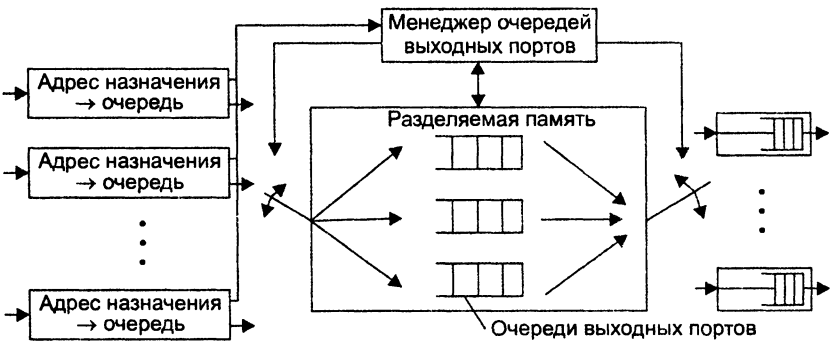


Рис. 15.17. Архитектура коммутаторов с разделяемой памятью

Входные блоки процессоров портов соединяются с переключаемым входом разделяемой памяти, а выходные блоки этих же процессоров — с ее переключаемым выходом. Переключением входа и выхода разделяемой памяти управляет *менеджер очередей выходных портов*. В разделяемой памяти менеджер организует несколько очередей данных, по одной для каждого выходного порта. Входные блоки процессоров передают менеджеру портов запросы на запись данных в очередь

того порта, который соответствует адресу назначения кадра. Менеджер по очереди подключает вход памяти к одному из входных блоков процессоров, и тот переписывает часть данных кадра в очередь определенного выходного порта. По мере заполнения очередей менеджер производит также поочередное подключение выхода разделяемой памяти к выходным блокам процессоров портов, и данные из очереди переписываются в выходной буфер процессора.

Применение общей буферной памяти, гибко распределяемой менеджером между отдельными портами, снижает требования к размеру буферной памяти процессора порта. Однако она должна быть достаточно быстродействующей для поддержания необходимой скорости обмена данными между N портами коммутатора.

Комбинированные коммутаторы. У каждой из описанных архитектур есть свои достоинства и недостатки, поэтому часто в сложных коммутаторах эти архитектуры применяются в комбинации друг с другом. Пример такого комбинирования приведен на рис. 15.18.

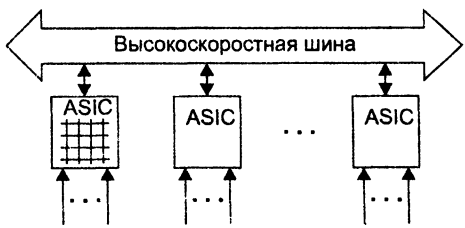


Рис. 15.18. Комбинирование архитектур коммутационной матрицы и общей шины

Коммутатор состоит из модулей с фиксированным количеством портов (2–12), выполненных на основе специализированной БИС, реализующей архитектуру коммутационной матрицы. Если порты, между которыми нужно передать кадр данных, принадлежат одному модулю, то передача кадра осуществляется процессорами модуля на основе имеющейся в модуле коммутационной матрицы. Если же порты принадлежат разным модулям, то процессоры общаются по общей шине. При такой архитектуре передача кадров внутри модуля будет происходить быстрее, чем при межмодульной передаче, так как коммутационная матрица — наиболее быстрый, хотя и наименее масштабируемый способ взаимодействия портов. Скорость внутренней шины коммутаторов может достигать нескольких гигабит в секунду, а у наиболее мощных моделей — до нескольких десятков гигабит в секунду.

Характеристики производительности коммутаторов

Скорости фильтрации и продвижения кадров — две основные характеристики производительности коммутатора. Эти характеристики являются интегральными, они не зависят от того, каким образом технически реализован коммутатор.

Скорость фильтрации — это скорость, с которой коммутатор выполняет перечисленные ниже этапы обработки кадров.

1. Прием кадра в свой буфер.
2. Просмотр адресной таблицы с целью нахождения порта для адреса назначения кадра.
3. Уничтожение кадра, так как его порт назначения и порт источника принадлежат одному логическому сегменту.

Скорость фильтрации практически у всех коммутаторов является неблокирующей — коммутатор успевает отбрасывать кадры в темпе их поступления.

Скорость продвижения — это скорость, с которой коммутатор выполняет следующие этапы обработки кадров.

1. Прием кадра в свой буфер.
2. Просмотр адресной таблицы с целью нахождения порта для адреса назначения кадра.
3. Передача кадра в сеть через найденный по адресной таблице порт назначения.

Как скорость фильтрации, так и скорость продвижения измеряются обычно в кадрах в секунду. Если в характеристиках коммутатора не уточняется, для какого протокола и для какого размера кадра приведены значения скоростей фильтрации и продвижения, то по умолчанию считается, что эти показатели даются для протокола Ethernet и кадров минимального размера, то есть кадров длиной 64 байт. Как мы уже обсуждали, режим передачи кадров минимальной длины используется как наиболее сложный тест, который должен подтвердить способность коммутатора работать при наихудшем сочетании параметров трафика.

Задержка передачи кадра измеряется как время, прошедшее с момента прихода первого байта кадра на входной порт коммутатора до момента появления этого байта на его выходном порту. Задержка складывается из времени, затрачиваемого на буферизацию байтов кадра, а также времени, затрачиваемого на обработку кадра коммутатором, — просмотр адресной таблицы, принятие решения о фильтрации или продвижении, получение доступа к среде выходного порта. Величина вносимой коммутатором задержки зависит от режима его работы. Если коммутация осуществляется «на лету», то задержки обычно невелики и составляют от 5 до 40 мкс, а при полной буферизации кадров — от 50 до 200 мкс (для кадров минимальной длины).

Производительность коммутатора определяется количеством пользовательских данных, переданных в единицу времени через его порты (измеряется в мегабитах в секунду). Так как коммутатор работает на канальном уровне, для него пользовательскими данными являются те данные, которые переносятся в поле данных кадров протоколов канального уровня — Ethernet, Token Ring, FDDI и т. п. Максимальное значение производительности коммутатора всегда достигается на кадрах максимальной длины, так как при этом минимальна доля накладных расходов на служебную информацию кадра. Коммутатор — это многопортовое устройство, поэтому для него в качестве характеристики принято давать максимальную суммарную производительность при одновременной передаче трафика по всем его портам.

На производительности коммутатора сказывается способ передачи кадров — «на лету» или с буферизацией. Коммутаторы, передающие кадры «на лету», вносят меньшие задержки передачи кадров на каждом промежуточном коммутаторе, поэтому общее уменьшение задержки доставки данных может быть значительным, что важно для мультимедийного трафика. Кроме того, выбранный способ коммутации оказывает влияние на возможности реализации некоторых полезных дополнительных функций, например трансляцию протоколов канального уровня. В табл. 15.1 сравниваются два способа коммутации.

Таблица 15.1. Возможности коммутаторов при коммутации «на лету» и с полной буферизацией

Функция	На лету	С буферизацией
Защита от плохих кадров	Нет	Да
Поддержка разнородных сетей (Ethernet, Token Ring, FDDI, ATM)	Нет	Да
Задержка передачи кадров	Низкая (5–40 мкс) при низкой нагрузке, средняя при высокой нагрузке	Средняя при любой нагрузке
Поддержка резервных связей	Нет	Да
Функция анализа трафика	Нет	Да

Средняя величина задержки коммутаторов, работающих «на лету», при высокой нагрузке объясняется тем, что в этом случае выходной порт часто бывает занят приемом другого кадра, поэтому вновь поступивший кадр для данного порта все равно приходится буферизовать.

Коммутатор, работающий «на лету», может выполнять проверку некорректности передаваемых кадров, но не может изъять плохой кадр из сети, так как часть его байтов (и, как правило, большая часть) уже переданы в сеть.

ПРИМЕЧАНИЕ

Так как каждый способ имеет свои достоинства и недостатки, в тех моделях коммутаторов, которым не нужно транслировать протоколы, иногда применяется механизм адаптивной смены режима работы коммутатора. Основной режим такого коммутатора — коммутация «на лету», но коммутатор постоянно контролирует трафик и при превышении интенсивности появления плохих кадров некоторого порога переходит на режим полной буферизации. Затем коммутатор может вернуться к коммутации «на лету».

Еще одной важной конструктивной характеристикой коммутатора является **максимальная емкость адресной таблицы**. Она определяет предельное количество MAC-адресов, с которыми может одновременно оперировать коммутатор.

Коммутаторы чаще всего используют для выполнения операций каждого порта выделенный процессорный блок со своей памятью для хранения собственного

экземпляра адресной таблицы. Каждый порт хранит только те наборы адресов, с которыми он работал в последнее время, поэтому экземпляры адресной таблицы разных процессорных модулей, как правило, не совпадают.

Значение максимального числа MAC-адресов, которое может запомнить процессор порта, зависит от области применения коммутатора. Коммутаторы рабочих групп обычно поддерживают всего несколько адресов на порт, так как они предназначены для образования микросегментов. Коммутаторы отделов должны поддерживать несколько сотен адресов, а коммутаторы магистралей сетей — до нескольких тысяч, обычно 4000–8000 адресов.

Недостаточная емкость адресной таблицы может служить причиной замедления работы коммутатора и засорения сети избыточным трафиком. Если адресная таблица процессора порта полностью заполнена, а он встречает новый адрес источника в поступившем кадре, процессор должен вытеснить из таблицы какой-либо старый адрес и поместить на его место новый. Эта операция сама по себе отнимает у процессора часть времени, но главные потери производительности наблюдаются при поступлении кадра с адресом назначения, который пришлось удалить из адресной таблицы. Так как адрес назначения кадра неизвестен, то коммутатор должен передать этот кадр на все остальные порты. Некоторые производители коммутаторов решают эту проблему за счет изменения алгоритма обработки кадров с неизвестным адресом назначения. Один из портов коммутатора конфигурируется как магистральный порт, на который по умолчанию передаются все кадры с неизвестным адресом[1]. Передача кадра на магистральный порт производится в расчете на то, что этот порт подключен к вышестоящему коммутатору (при иерархическом соединении коммутаторов в крупной сети), который имеет достаточную емкость адресной таблицы и знает, куда нужно передать любой кадр.

Дуплексные протоколы локальных сетей

Список ключевых слов: микросегментация, полудуплексный и дуплексный режимы работы коммутатора, механизм обратной связи, подуровень управления уровня MAC, технология 10G Ethernet, стандарт IEEE 802.3ae, интерфейс XGMII.

Изменения в работе MAC-уровня в дуплексном режиме

Технология коммутации сама по себе не имеет непосредственного отношения к методу доступа к среде, который используется портами коммутатора. При подключении к порту коммутатора сегмента, представляющего собой разделяемую

[1] В маршрутизаторах такой прием применяется давно, позволяя сократить размеры адресных таблиц в сетях, организованных по иерархическому принципу.

среду, данный порт, как и все остальные узлы такого сегмента, должен поддерживать полудуплексный режим.

Однако когда к каждому порту коммутатора подключен не сегмент, а только *один* компьютер, причем по двум физически раздельным каналам, как это происходит почти во всех стандартах Ethernet, кроме коаксиальных версий Ethernet, ситуация становится не такой однозначной. Порт может работать как в обычном полудуплексном режиме, так и в дуплексном.

Подключение к портам коммутатора не сегментов, а отдельных компьютеров называется микросегментацией.

В обычном для Ethernet **полудуплексном режиме** работы порт коммутатора попрежнему распознает коллизии. Доменом коллизий в этом случае является участок сети, включающий передатчик коммутатора, приемник коммутатора, передатчик сетевого адаптера компьютера, приемник сетевого адаптера компьютера и две витые пары, соединяющие передатчики с приемниками (рис. 15.19).

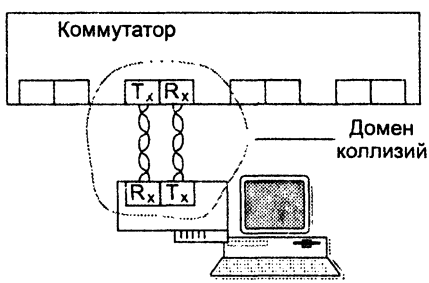


Рис. 15.19. Домен коллизий, образуемый компьютером и портом коммутатора

Коллизия возникает, когда передатчики порта коммутатора и сетевого адаптера одновременно или почти одновременно начинают передачу своих кадров, считая, что сегмент свободен (см. рисунок). Правда, вероятность коллизии в таком сегменте гораздо меньше, чем в сегменте, состоящем из 20–30 узлов, но она не нулевая. При этом максимальная производительность сегмента Ethernet в 14 880 кадров в секунду при минимальной длине кадра делится между передатчиком порта коммутатора и передатчиком сетевого адаптера. Если считать, что она делится пополам, то каждому предоставляется возможность передавать примерно по 7440 кадров в секунду.

В **дуплексном режиме** одновременная передача данных передатчиком порта коммутатора и сетевого адаптера коллизией не считается. В принципе, это достаточно естественный режим работы для отдельных дуплексных каналов передачи данных, и он часто используется в протоколах глобальных сетей. При дуплексной связи порты Ethernet 10 Мбит/с могут передавать данные со скоростью 20 Мбит/с — по 10 Мбит/с в каждом направлении.

Естественно, необходимо, чтобы MAC-узлы взаимодействующих устройств поддерживали дуплексный режим. В случае когда только один узел поддерживает дуплексный режим, второй узел будет постоянно фиксировать коллизии и при-

останавливать свою работу, в то время как другой узел продолжит передавать данные, которые никто в этот момент не принимает. Изменения, которые нужно внести в логику работы MAC-узла, чтобы он мог работать в дуплексном режиме, минимальны — нужно просто отменить фиксацию и обработку коллизий в сетях Ethernet. Если же микросегмент образован компьютером, поддерживающим протокол Token Ring или FDDI, то сетевой адаптер и порт коммутатора должны посылать свои кадры, не дожидаясь прихода токена доступа, а тогда, когда в этом возникнет необходимость. Фактически, при работе в дуплексном режиме MAC-узел игнорирует метод доступа к среде, разработанный для данной технологии.

При разработке технологий Fast Ethernet и Gigabit Ethernet дуплексный режим стал одним из двух полноправных стандартных режимов работы узлов сети. Сетевые адаптеры сегодня могут поддерживать оба режима работы, отрабатывая логику алгоритма доступа CSMA/CD при подключении к порту концентратора и работая в дуплексном режиме при подключении к порту коммутатора.

Перегрузки при дуплексной работе

Простой отказ от поддержки алгоритма доступа к разделяемой среде без какой-либо модификации протокола ведет к повышению вероятности потерь кадров коммутаторами, так как при этом теряется контроль за потоками кадров, направляемых конечными узлами в сеть. В полудуплексном режиме, свойственном технологиям с разделяемой средой, поток кадров регулировался самим методом доступа к разделяемой среде. При переходе на полнодуплексный режим узлу разрешается отправлять кадры в коммутатор всегда, когда это ему нужно, поэтому коммутаторы сети могут в этом режиме сталкиваться с перегрузками, не имея при этом никаких средств «притормаживания» потока кадров.

Причина перегрузок обычно кроется не в том, что коммутатор является блокирующим, то есть ему не хватает производительности процессоров для обслуживания потоков кадров, а в ограниченной пропускной способности отдельного выходного порта, которая определяется параметрами протокола.

Поэтому, если входной трафик неравномерно распределяется между выходными портами, легко представить ситуацию, когда в какой-либо выходной порт коммутатора будет направляться трафик с суммарной средней интенсивностью большей, чем протокольный максимум. На рис. 15.20 показана как раз такая ситуация, когда в порт 3 коммутатора Ethernet направляется от портов 1, 2, 4 и 6 поток кадров размером в 64 байт с суммарной интенсивностью в 22 100 кадров в секунду. Вспомним, что максимальная скорость в кадрах в секунду для сегмента Ethernet составляет 14 880. Естественно, что когда кадры поступают в буфер порта со скоростью 22 100 кадров в секунду, а уходят со скоростью 14 880 кадров в секунду, то внутренний буфер выходного порта начинает неуклонно заполняться необработанными кадрами.

Нетрудно подсчитать, что при размере буфера в 100 Кбайт в приведенном примере полное заполнение буфера произойдет через 0,22 с после начала работы в таком интенсивном режиме. Увеличение буфера до 1 Мбайт даст увеличение времени заполнения буфера до 2,2 с, что также неприемлемо. Проблему можно

решить с помощью *средств контроля перегрузки*, которые были рассмотрены в главе 7.

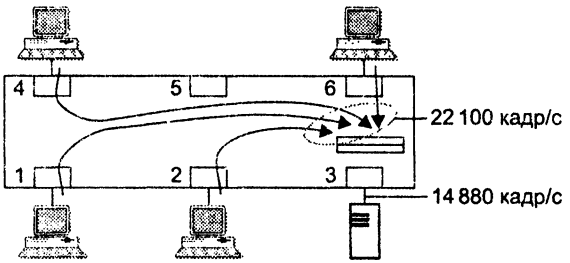


Рис. 15.20. Переполнение буфера порта из-за несбалансированности трафика

Как мы знаем, существуют различные типы средств контроля перегрузки: управление очередями в коммутаторах, обратная связь, резервирование пропускной способности. На основе этих средств можно создать эффективную систему поддержки показателей QoS для трафика разных классов.

В этом разделе мы рассмотрим **механизм обратной связи**, который был стандартизован для сетей Ethernet в марте 1997 года как спецификация IEEE 802.3x. Механизм обратной связи 802.3x используется только в дуплексном режиме работы портов коммутатора. Этот механизм очень важен для коммутаторов локальных сетей, так как он позволяет уменьшить потери кадров из-за переполнения буферов независимо от того, обеспечивает сеть дифференцированную поддержку показателей QoS для разных типов трафика или же предоставляет базовый сервис по доставке с максимальными усилиями («по возможности»). Другие механизмы поддержания показателей QoS будут рассмотрены в следующей главе.

Спецификация 802.3x вводит новый подуровень в стеке протоколов Ethernet — **подуровень управления уровня MAC**. Он располагается над уровнем MAC и является необязательным (рис. 15.21).

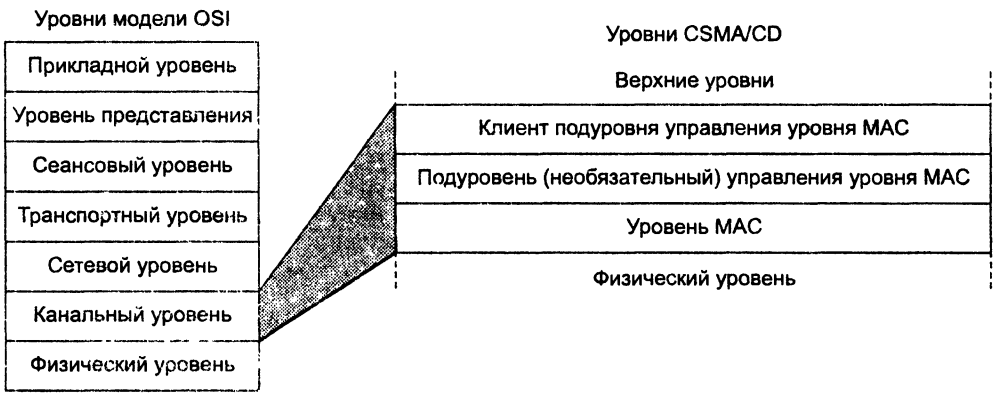


Рис. 15.21. Подуровень управления уровня MAC

Кадры этого подуровня могут использоваться в различных целях, но пока в стандартах Ethernet для них определена только одна задача — приостановка передачи кадров другими узлами на определенное время.

Кадр подуровня управления отличается от кадров пользовательских данных тем, что в поле длины/типа всегда содержится шестнадцатеричное значение 88-08. Формат кадра подуровня управления рассчитан на универсальное применение, поэтому он достаточно сложен (рис. 15.22).

Рис. 15.22. Формат кадра подуровня управления

Коммутатор использует кадр подуровня управления в том случае, когда ему нужно на время приостановить поступление кадров от соседнего узла, чтобы разгрузить свои внутренние очереди.

В качестве адреса назначения можно использовать зарезервированное для этой цели значение группового адреса 01-80-C2-00-00-01. Это удобно в том случае, когда соседний узел также является коммутатором (так как порты коммутатора не имеют уникальных MAC-адресов). Если сосед — конечный узел, можно также использовать уникальный MAC-адрес.

В поле кода операции подуровня управления указывается шестнадцатеричный код 00-01, поскольку, как уже было отмечено, пока определена только одна операция подуровня управления, она называется *PAUSE* (пауза) и имеет шестнадцатеричный код 00-01.

В поле параметров подуровня управления указывается время, на которое узел, получивший такой код, должен прекратить передачу кадров узлу, отправившему кадр с операцией *PAUSE*. Время измеряется в 512 битовых интервалах конкретной реализации Ethernet, диапазон возможных вариантов приостановки равен 0–65535.

Как видно из описания, этот механизм обратной связи относится к типу 2 в соответствии с классификацией, приведенной в главе 7. Специфика его состоит в том, что в нем предусмотрена только одна операция — приостановка на определенное время. Обычно же в механизмах этого типа используются две операции — приостановка и возобновление передачи кадров. Именно так этот механизм реализован в одном из наиболее старых протоколов сетей с коммутацией пакетов — протоколе сети X.25 под названием LAP-B.

Технология 10G Ethernet

Стандарт 10G Ethernet определяет только дуплексный режим работы, поэтому он используется исключительно в *коммутируемых* локальных сетях.

Формально этот стандарт имеет обозначение **IEEE 802.3ae** и является поправкой к основному тексту стандарта 802.3. Этот документ описывает семь новых спецификаций физического уровня, которые взаимодействуют с уровнем MAC с помощью нового варианта подуровня согласования (рис. 15.23). Этот подуровень обеспечивает для всех вариантов физического уровня 10G Ethernet единый интерфейс **XGMII** (eXtended Gigabit Medium Independent Interface — расширенный интерфейс независимого доступа к гигабитной среде), который предусматривает параллельный обмен четырьмя байтами, образующими четыре потока данных.

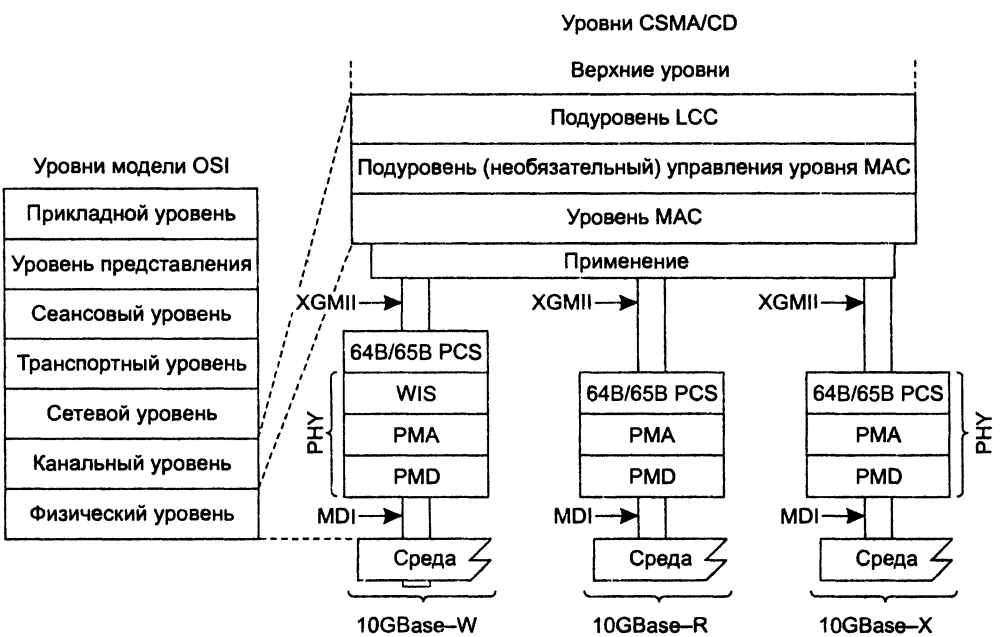


Рис. 15.23. Три группы физических интерфейсов 10G

Как видно из рисунка, существуют три группы физических интерфейсов стандарта 10G Ethernet: 10GBase-X, 10GBase-R и 10GBase-W. Они отличаются

способом кодирования данных: в варианте 10Base-X используется код 8B/10B, а в остальных двух — код 64B/66B. Все они задействуют оптическую среду для передачи данных.

Группа 10GBase-X в настоящее время состоит из одного интерфейса подуровня PMD — 10GBase-LX4. Буква L говорит о том, что информация передается с помощью волн второго диапазона прозрачности, то есть 1310 нм. Информация в каждом направлении передается одновременно с помощью четырех волн (что отражает цифра 4 в названии интерфейса), которые мультиплексируются на основе техники WDM (рис. 15.24). Каждый из четырех потоков интерфейса XGMII передается в оптическом волокне со скоростью 2,5 Гбит/с.

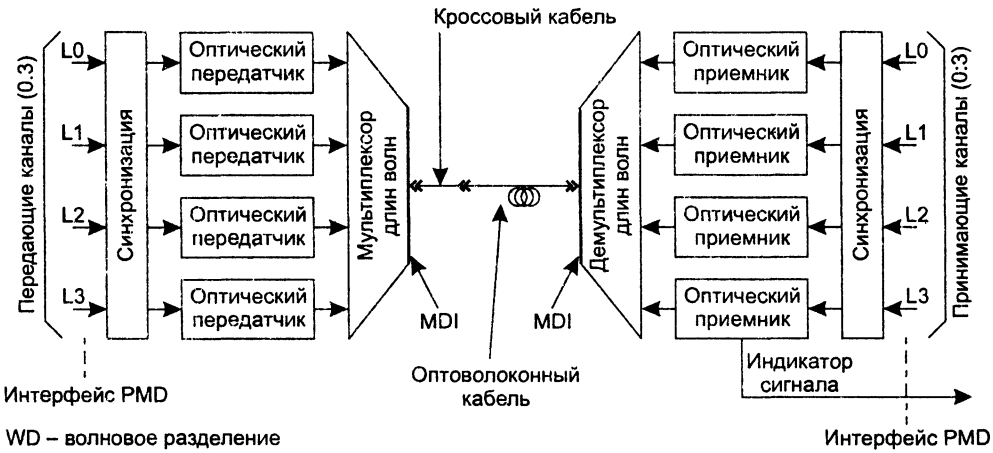


Рис. 15.24. Интерфейс 10GBase-LX4 использует технику WDM

Максимальное расстояние между передатчиком и приемником стандарта 10GBase-LX4 на многомодовом волокне равно 200–300 м (в зависимости от полосы пропускания волокна), а на одномодовом — 10 км.

В каждой из групп 10GBase-W и 10GBase-R может быть три варианта подуровня PMD: S, L и E в зависимости от используемого для передачи информации диапазона волн — 850, 1310 или 1550 нм соответственно. Таким образом, существуют интерфейсы 10GBase-WS, 10GBase-WL, 10GBase-WE и 10GBase-RS, 10GBase-RL и 10GBase-RE. Каждый из них передает информацию с помощью одной волны соответствующего диапазона.

В отличие от 10GBase-R физические интерфейсы группы 10GBase-W обеспечивают скорость передачи и формат данных, совместимые с интерфейсом SONET STS-192/SDH STM-64. Пропускная способность интерфейсов группы W равна 9,95328 Гбит/с, а эффективная скорость передачи данных — 9,58464 Гбит/с (часть пропускной способности тратится на заголовки кадров STS/STM). Из-за того что скорость передачи информации у этой группы интерфейсов ниже, чем 10 Гбит/с, они могут взаимодействовать только между собой, то есть соединение, например, интерфейсов 10GBase-RL и 10Base-WL невозможно.

Интерфейсы группы W не являются полностью совместимыми по электрическим характеристикам с интерфейсами SONET STS-192/SDH STM-64. Поэтому для соединения сетей 10G Ethernet через первичную сеть SONET/SDH у мультиплексоров первичной сети должны быть специальные интерфейсы 10G, совместимые со спецификациями 10GBase-W. Поддержка оборудованием 10GBase-W скорости 9,95328 Гбит/с обеспечивает принципиальную возможность передачи трафика 10G Ethernet через сети SONET/SDH в кадрах STS-192/STM-64.

Физические интерфейсы, работающие в окне прозрачности E, обеспечивают передачу данных на расстояния до 40 км. Это позволяет строить не только локальные сети, но и сети мегаполисов, что нашло отражение в поправках к исходному тексту стандарта 802.3.

Выводы

Логическая структуризация сети необходима при построении сетей средних и крупных размеров. Использование общей разделяемой среды приемлемо только для сети, состоящей из 5--10 компьютеров.

Деление сети на логические сегменты повышает производительность, надежность, гибкость построения и управляемость сети.

Для логической структуризации сети применяются мосты и их современные преемники — коммутаторы. Первые два типа устройств позволяют разделить сеть на логические сегменты с помощью минимума средств — только на основе протоколов канального уровня. Кроме того, эти устройства не требуют конфигурирования.

Пассивный способ построения адресной таблицы коммутаторами — путем слежения за проходящим трафиком — приводит к невозможности работы в сетях с петлевидными связями. Другим недостатком сетей, построенных на коммутаторах, является отсутствие защиты от широковещательного шторма, который эти устройства обязаны передавать в соответствии с алгоритмом своей работы.

Применение коммутаторов позволяет сетевым адаптерам использовать дуплексный режим работы протоколов локальных сетей (Ethernet, Fast Ethernet, Gigabit Ethernet, Token Ring, FDDI). В этом режиме отсутствует этап доступа к разделяемой среде, а общая скорость передачи данных удваивается.

В дуплексном режиме для борьбы с перегрузками коммутаторов используется метод обратной связи, описанный в стандарте 802.3x. Он позволяет приостановить на некоторое время поступление кадров от непосредственных соседей перегруженного коммутатора.

При полудуплексном режиме работы коммутаторы используют для управления потоком при перегрузках два метода: агрессивный захват среды и обратное давление на конечный узел. Применение этих методов позволяет достаточно гибко управлять потоком, чередуя несколько передаваемых кадров с одним принимаемым.

Основными характеристиками производительности коммутатора являются: скорость фильтрации кадров, скорость продвижения кадров, общая пропускная способность по всем портам в мегабитах в секунду, задержка передачи кадра.

На характеристики производительности коммутатора влияют: тип коммутации — «на лету» или с полной буферизацией, размер адресной таблицы, размер буфера кадров.

Коммутаторы умеют фильтровать передаваемый трафик по различным критериям, учитывающим адреса получателя и отправителя, а также значения произвольных полей. Однако способ

задания фильтров на канальном уровне достаточно сложен, он требует от администратора хорошего знания протоколов и кропотливой работы по определению местоположения нужного признака внутри кадра.

Вопросы и задания

1. Назовите главные недостатки сетей на разделяемой среде.
2. Чем обусловлен тот факт, что в сетях Ethernet резкий рост задержек начинается при меньших значениях коэффициента использования среды, чем в сетях Token Ring и FDDI?
3. Что можно отнести к преимуществам коммутируемых локальных сетей?
4. На основе каких адресов строится таблица продвижения коммутатора?
5. Можно ли сказать, что при разделении среды на два сегмента нагрузка каждого сегмента уменьшается в два раза?
6. К каким негативным последствиям приводит наличие петель в сети, построенной на коммутаторах, работающих в соответствии с алгоритмом прозрачного моста?
7. Для какой цели записи таблицы продвижения имеют ограниченный срок жизни?
8. Почему мост в алгоритме прозрачного моста называется «прозрачным»?
9. Для каких целей создаются пользовательские фильтры в коммутаторах?
10. Какие параметры можно задействовать при создании пользовательского фильтра коммутатора?
11. Может ли скорость продвижения превосходить скорость фильтрации?
12. Что такое «неблокирующий коммутатор»?
13. Может ли неблокирующий коммутатор терять пакеты из-за переполнения очереди?
14. Какие механизмы коммутаторы используют при переполнении внутренних очередей?
15. Может ли в технологии 10G Ethernet использоваться разделяемая среда?
16. Какой особенности физического интерфейса соответствует цифра 4 в спецификации 10GBase-LX4?
17. Можно ли коммутатор локальной сети с интерфейсом 10GBase-WL непосредственно присоединить к порту STM-64 мультиплексора SDH?
18. Какое свойство технологий локальных сетей облегчает трансляцию протоколов Ethernet, Token Ring и FDDI?
19. В каких случаях кадр FDDI невозможно транслировать в кадр Ethernet?
20. Назовите основные типы коммутирующих блоков.
21. В чем основной недостаток коммутационной матрицы?

22. Какое дополнительное преимущество связано с применением разделяемой памяти в качестве коммутирующего блока?
23. По какой причине коммутация «на лету» нашла ограниченное применение в коммутаторах?
24. Что происходит, если количество адресов локальной сети превосходит размер адресной таблицы коммутатора?
25. Пользовательский фильтр включает логическое условие и действие, которое нужно выполнить над кадром, если это условие соблюдается. Напишите условие фильтра, который отбрасывает кадры (это действие), идущие от компьютера A с MAC-адресом 06-DB-00-34-5E-27 и от компьютера B с MAC-адресом CC-33-00-D5-43-4D к серверу S с MAC-адресом CC-33-00-65-44-AA.
26. Вам необходимо улучшить производительность сети на разделяемой среде, показанной на рис. 15.25. В вашем распоряжении имеется только один коммутатор, имеющий два порта 1000 Мбит/с и 8 портов 100 Мбит/с. Каким образом вы преобразуете сеть, если имеющиеся в начальном варианте хабы можно продолжать использовать?

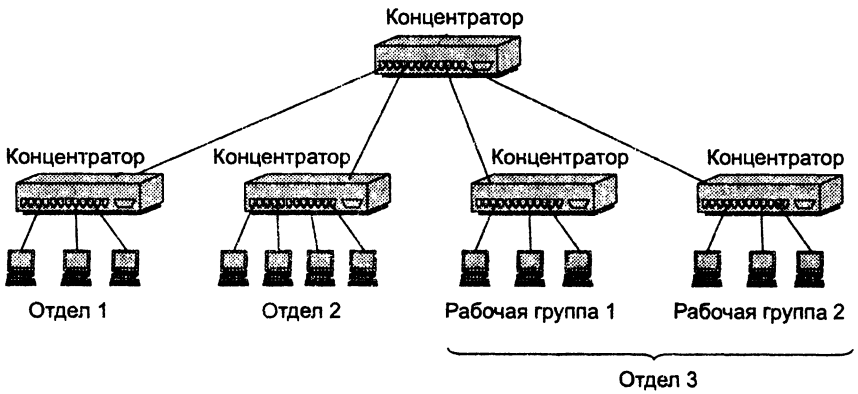


Рис. 15.25. Модернизируемая сеть

ГЛАВА 16 Интеллектуальные функции коммутаторов

Коммутаторы позволяют значительно увеличивать размеры сетей за счет разбиения разделяемой среды на части или полного отказа от принципа разделения среды в пользу коммутируемых локальных сетей Однако при росте сетей возникают новые проблемы, с которыми не в состоянии справиться коммутатор, работа которого строится лишь на основе алгоритма прозрачного моста. Прежде всего, остается нерешенной проблема надежности сети, так как древовидная топология коммутируемых локальных сетей очень уязвима — отказ любой линии связи или коммутатора приводит к потере связности сети, сеть фактически распадается на два или более сегмента.

Ограничения древовидной топологии преодолеваются благодаря интеллектуальным функциям коммутаторов, которые наделяют локальные сети дополнительными возможностями. Так, в коммутируемых локальных сетях широко применяется алгоритм покрывающего дерева (STA), который позволяет автоматически находить новый вариант древовидной топологии при отказах коммутаторов или линий связи и тем самым обеспечить отказоустойчивость сети. Алгоритм покрывающего дерева был разработан одновременно с алгоритмом прозрачного моста (то есть в начале 80-х) и с тех пор успешно применяется в локальных сетях.

Сравнительно недавно, с началом широкого применения коммутируемых локальных сетей был разработан механизм агрегирования линий связи. Поддержка коммутаторами агрегирования позволяет объединять несколько линий связи (физических каналов) в один логический канал. Это повышает как производительность, так и надежность сети.

Новые развитые возможности коммутаторов локальных сетей обеспечивают поддержку методов QoS для различных типов трафика, включая приоритетные и взвешенные очереди, обратную связь, резервирование ресурсов.

Несмотря на прогресс, обеспечиваемый алгоритмом покрывающего дерева и механизмом агрегирования линий связи, в локальных сетях, построенных *только* на коммутаторах и не использующих маршрутизаторы, остается немало проблем. Часть из этих проблем позволяет снять техника виртуальных локальных сетей (VLAN). Функция VLAN дает возможность за счет конфигурирования коммутаторов (то есть программным, а не физическим способом, соединяя и разъединяя разъемы кабелей) разделить локальную сеть на несколько обособленных логических сегментов. Такие обособленные сегменты затем могут быть соединены в составную сеть уже с помощью протокола сетевого уровня, при этом благодаря программному делению сети на сегменты можно очень удобно и быстро менять структуру сети.

Алгоритм покрывающего дерева

Список ключевых слов: алгоритм покрывающего дерева, протокол покрывающего дерева, сегмент сети, корневой коммутатор, метрика, идентификатор коммутатора, корневой порт, идентификатор порта, назначенный порт, назначенный коммутатор, протокольная единица данных моста.

В тех локальных сетях, где технологии и оборудование реализуют функции только первого и второго уровней модели ISO/OSI, проблема использования альтернативных путей имеет свою специфику: базовые протоколы поддерживают только *древовидные*, то есть не содержащие замкнутых контуров, топологии связей.

Для автоматического перевода в резервное состояние всех альтернативных связей, не вписывающихся в топологию дерева, в локальных сетях используются **алгоритм покрывающего дерева** (Spanning Tree Algorithm, STA) и реализующий его **протокол покрывающего дерева** (Spanning Tree Protocol, STP).

Алгоритм покрывающего дерева, разработанный достаточно давно, в 1983 году, был признан IEEE удачным и включен в ту же спецификацию 802.1D, в которой описывается и сам алгоритм работы прозрачного моста. Хотя мосты, для которых был разработан алгоритм, сегодня уже относятся к практически «вымершему» виду коммуникационных устройств, STA широко применяется в наиболее массовых устройствах современных локальных сетей — коммутаторах. Алгоритм STA позволил без применения маршрутизаторов только на основе коммутаторов строить крупные локальные сети, за счет избыточных связей обладающие высокой надежностью.

Производители оборудования, как правило, реализуют алгоритм STA в коммутаторах, предлагаемых для тех участков сети, к которым предъявляются повышенные требования к надежности, — в магистральных коммутаторах, а также в коммутаторах отделов и крупных рабочих групп.

Необходимые определения

Алгоритм STA формализует сеть (рис. 16.1, *а*) в виде графа (рис. 16.1, *б*), вершинами которого являются коммутаторы и сегменты сети.

Сегмент — связная часть сети, не содержащая коммутаторов (и маршрутизаторов). Сегмент может быть разделяемым (во время создания алгоритма STA это был единственный тип сегмента) и включать устройства физического уровня: повторители/концентраторы, существование которых коммутатор, будучи устройством канального уровня, «не замечает». Сегодня сегмент часто представляет собой дуплексный двухточечный канал между смежными портами двух коммутаторов.

Алгоритм покрывающего дерева обеспечивает построение древовидной топологии связей с единственным путем минимальной длины от каждого коммутатора и от каждого сегмента до некоторого выделенного **корневого коммутатора** — корня дерева. *Единственность* пути гарантирует отсутствие петель, а *минималь-*

ность расстояния — рациональность маршрутов следования трафика от периферии сети к ее магистрали, роль которой исполняет корневой коммутатор.

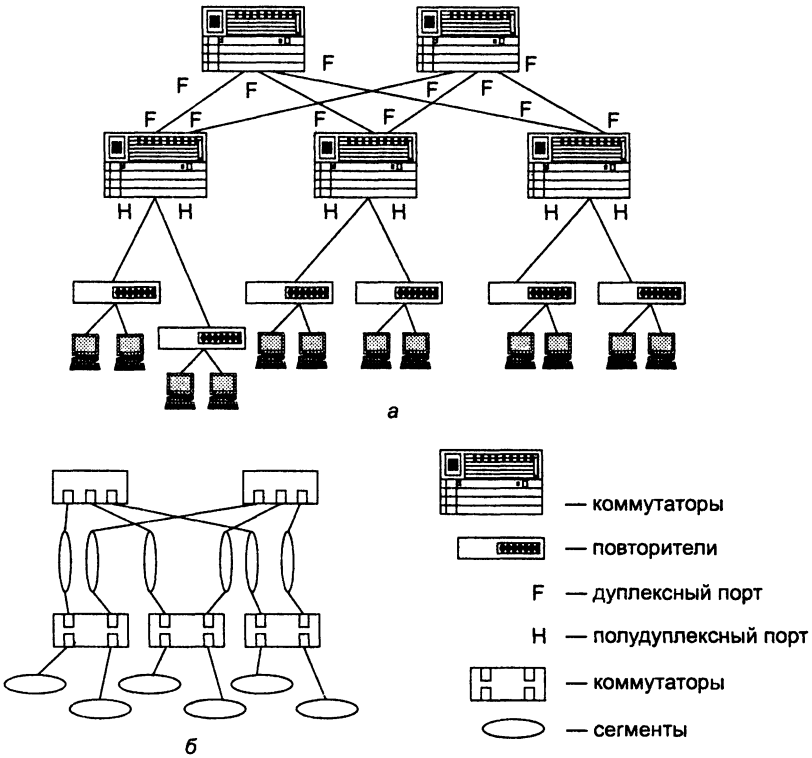


Рис. 16.1. Формализованное представление сети в соответствии с алгоритмом STA

В качестве расстояния в STA используется **метрика**, традиционная для протоколов маршрутизации, — величина, обратно пропорциональная пропускной способности сегмента. В STA метрика также определяется как *условное время передачи бита сегментом*. Это время измеряется в 10-наносекундных единицах. Так, для сегмента Ethernet 10 Мбит/с метрика равна 10 условных единиц, для сегмента Ethernet 100 Мбит/с — 1, а для сегмента Token Ring 16 Мбит/с — 6,25. Учитывая, что сети становятся все более и более скоростными, существует так называемая пересмотренная версия шкалы условных единиц: 10 Мбит/с — 100, 100 Мбит/с — 19, 1 Гбит/с — 4, 10 Гбит/с — 2.

Идентификатор коммутатора — это 8-байтовое число, шесть младших байтов которого составляет MAC-адрес его блока управления, отрабатывающего алгоритм STA (напомним, что портам коммутаторов и мостов для выполнения своей основной функции MAC-адреса не требуются), а два старших байта конфигурируются вручную, что, как мы дальше увидим, позволяет администратору сети влиять на процесс выбора корневого коммутатора.

Корневой порт коммутатора — порт, который имеет кратчайшее расстояние до корневого коммутатора (точнее, до любого из портов корневого коммутатора).

Идентификатором порта служит 2-байтовое число. Младший байт содержит порядковый номер данного порта в коммутаторе, а значение старшего байта задается администратором.

Назначенный порт — порт, который среди всех портов всех коммутаторов данного сегмента сети имеет минимальное расстояние до корневого коммутатора.

Назначенным коммутатором сегмента объявляется коммутатор, которому принадлежит назначенный порт данного сегмента.

Протокольными единицами данных моста (Bridge Protocol Data Unit, BPDU) называются специальные пакеты, которыми периодически обмениваются коммутаторы для автоматического определения конфигурации дерева. Пакеты BPDU переносят данные об идентификаторах коммутаторов и портов, а также о расстоянии до корневого коммутатора. Интервал генерации пакетов BPDU, называемый в алгоритме *интервалом hello*, настраивается администратором и обычно составляет от 1 до 4 секунд.

Три этапа построения дерева

На рис. 16.2 приведен пример сети, которую мы будем использовать для иллюстрации процедуры построения покрывающего дерева.

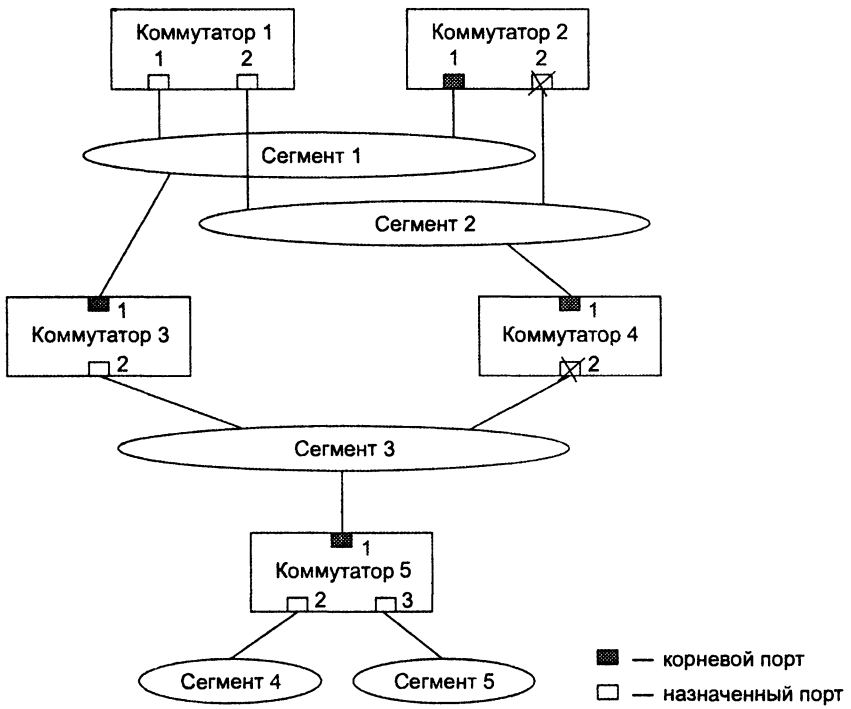


Рис. 16.2. Пример покрывающего дерева STA

Алгоритм STA определяет активную конфигурацию сети за три этапа.

Первый этап — определение корневого коммутатора, от которого строится дерево.

В соответствии с алгоритмом STA в качестве корневого коммутатора выбирается коммутатор с *наименьшим значением идентификатора*. Если администратор не вмешается в этот процесс, корневой коммутатор будет выбран достаточно случайным образом — им станет устройство с минимальным MAC-адресом блока управления. Очевидно, что такой выбор может оказаться далеко не рациональным. Например, при выборе коммутатора 5 в качестве корневого значительная часть трафика проходила бы через большое количество транзитных сегментов и коммутаторов. Поэтому не стоит администратору пускать данный процесс «на самотек» — лучше в него вмешаться и назначить корневой коммутатор осознанно (за счет соответствующего конфигурирования старших байтов идентификаторов коммутатора), чтобы выбранный коммутатор действительно занимал центральное место в соединениях сегментов. Предположим, что идентификаторы коммутаторов совпадают с их номерами, приведенными на рисунке. Тогда корневым коммутатором является коммутатор 1.

Второй этап — выбор корневого порта для каждого коммутатора.

Расстояние определяется по пакетам BPDU, поступающим от корневого коммутатора. На основании этих пакетов каждый коммутатор может определить минимальные расстояния от всех своих портов до корневого коммутатора. Каждый коммутатор анализирует и ретранслирует BPDU, увеличивая расстояние до корня, указанное в полученном пакете BPDU, на условное время того сегмента, из которого принят данный пакет. Тем самым в пакете BPDU по мере прохождения через коммутаторы наращивается расстояние до корневого коммутатора. Например, если считать, что все сегменты в рассматриваемом примере являются сегментами Ethernet 10 Мбит/с, то коммутатор 2, приняв из сегмента 1 пакет BPDU с расстоянием, равным 0, увеличивает его на 10 условных единиц (измерения метрики).

Ретранслируя пакеты, каждый коммутатор для каждого своего порта запоминает минимальное расстояние до корня, встретившееся во всех принятых этим портом пакетах BPDU. По завершении процедуры определения конфигурации покрывающего дерева каждый коммутатор находит свой корневой порт (с минимальным расстоянием до корня).

При равных метриках для разрешения неоднозначности к процедуре выбора минимального расстояния привлекаются значения идентификаторов коммутаторов и портов. Предпочтение отдается портам и коммутаторам с наименьшими идентификаторами. Например, для сегмента 3 существует два равноценных в отношении метрики пути к корневому коммутатору 1 — через коммутатор 3 и через коммутатор 4. Выбранный путь проходит через коммутатор, с *меньшим значением идентификатора*, а именно 3 (номера портов внутри коммутатора в данном случае совпадают, но при сравнении сначала принимается во внимание *идентификатор коммутатора*, а потом уже номер порта).

В нашем примере коммутатор 3 выбирает порт 1 в качестве корневого, так как для него минимальное расстояние до корня равно 10 условных единиц (пакет

BPDU с таким расстоянием принят от корневого коммутатора через сегмент 1). Порт 2 коммутатора 3 устанавливает на основании принятых пакетов, что минимальное расстояние равно 20 условных единиц — это соответствует прохождению пакета от порта 2 корневого коммутатора через сегмент 2, затем через коммутатор 4 и сегмент 3. Коммутатор 2 при выборе корневого порта «сталкивается» с ситуацией, когда у его портов 1 и 2 равное расстояние до корня — по 10 условных единиц (порт 1 принимает пакеты от порта 1 корневого коммутатора через один промежуточный сегмент — сегмент 1, так же как порт 2 получает пакеты от порта 2 корневого коммутатора через сегмент 2). Поскольку числовое значение идентификатора порта 1 меньше, чем порта 2, то корневым и выбирается порт 1.

Третий этап — выбор назначенных порта и коммутатора.

Из всех портов всех коммутаторов в пределах каждого сегмента сети выбирается назначенный порт и соответствующий данному порту назначенный коммутатор сегмента. Аналогично выбору корневого порта здесь используется распределенная процедура. Каждый коммутатор сегмента прежде всего исключает из рассмотрения свой корневой порт (для сегмента, к которому он подключен, всегда существует другой коммутатор, который расположен ближе к корню). Для каждого из оставшихся портов выполняется сравнение принятых по ним минимальных расстояний до корня (еще до наращивания на условное время сегмента) с расстоянием до корня корневого порта данного коммутатора. Если все принятые на этом порту расстояния оказываются больше, чем расстояние от собственного корневого порта, значит, для сегмента, к которому подключен порт, кратчайший путь к корневому коммутатору проходит через него, и он становится назначенным. Коммутатор делает все свои порты, для которых такое условие выполняется, назначенными. Когда имеется несколько портов с одинаковым кратчайшим расстоянием до корневого коммутатора, выбирается порт с наименьшим идентификатором.

В рассматриваемом примере коммутатор 2 при проверке порта 2 обнаруживает, что через этот порт принимаются пакеты с минимальным расстоянием 0 (это пакеты от порта 2 корневого коммутатора 1). Так как собственный корневой порт у коммутатора 2 имеет расстояние до корня 10, то порт 2 этого коммутатора не является назначенным для сегмента 2.

На выполнение всех трех этапов коммутаторам сети отводится по умолчанию 15 секунд. Предполагается, что за это время каждый коммутатор получит столько пакетов BPDU, сколько будет достаточно для определения состояния своих портов.

Все остальные порты, кроме корневых и назначенных, блокируются (на рисунке они перечеркнуты), и в результате завершается построение покрывающего дерева. Математически доказано, что при таком выборе активных портов в сети исключаются петли, и оставшиеся связи образуют *покрывающее дерево* (если оно вообще может быть построено при существующих связях в сети).

После построения покрывающего дерева коммутатор начинает принимать (но не продвигать) пакеты данных и на основе их адресов источника строить таблицу продвижения. Это обычный режим обучения прозрачного моста, который ранее

нельзя было активизировать, так как порт не был уверен в том, что он останется корневым или назначенным и будет передавать пакеты данных. Состояние обучения по умолчанию также выдерживается в течение интервала 15 с. При этом порт продолжает участвовать в работе алгоритма STA, так что поступление пакетов BPDU с лучшими параметрами переводит его в заблокированное состояние.

ПРИМЕЧАНИЕ

Отметим, что выбранная по алгоритму STA древовидная топология в общем случае не оптимальна для всех путей передачи трафика. Так, в описываемом примере при передаче пакетов из сегмента 3 в сегмент 2 трафик проходит путь: коммутатор 3 — сегмент 1 — коммутатор 1 — сегмент 2. Метрика этого пути — 30. Если бы порт 2 коммутатора 4 не был заблокирован, то путь мог бы быть короче — через коммутатор 4. При этом метрика пути была бы равна 20 — лучше, чем в предыдущем случае. Такой вариант возможен при выборе кратчайшего пути к корневому коммутатору для сегмента 4 через коммутатор 4, а не 3, например, за счет соответствующего назначения старших частей идентификаторов коммутаторов. Однако при таком варианте путь из сегмента 4 в сегмент 1 уже не окажется оптимальным.

И только после двукратной выдержки по таймеру порт переходит в состояние продвижения и обрабатывает пакеты данных в соответствии с построенной таблицей (которая продолжает модифицироваться, отражая изменения в структуре сети).

В процессе нормальной работы корневой коммутатор продолжает генерировать конфигурационные пакеты BPDU с интервалом hello, а остальные коммутаторы получают их через свои корневые порты и ретранслируют через назначенные порты. У коммутатора могут отсутствовать назначенные порты, как у коммутаторов 2 и 4, но он все равно участвует в работе протокола STA, так как корневой порт принимает служебные пакеты BPDU.

Если по истечении максимального времени жизни сообщения (по умолчанию — 20 с) корневой порт любого коммутатора сети не получает служебный пакет BPDU, то он инициализирует новую процедуру построения покрывающего дерева. При этом на все порты генерируется и передается пакет BPDU, в котором коммутатор указывает себя в качестве корневого. Аналогичным образом ведут себя и другие коммутаторы сети, у которых истек таймер максимального времени жизни сообщения, в результате чего выбирается новая активная конфигурация.

Недостатки и достоинства STA

Одним из основных достоинств алгоритма покрывающего дерева является то, что, в отличие от многих упрощенных алгоритмов, где переход в резервное соединение осуществляется исключительно при отказе соседнего устройства, он принимает решение о реконфигурировании с учетом не только связей с соседями, но и связей в отдаленных сегментах сети.

К недостаткам алгоритма можно отнести то, что в сетях с большим количеством коммутаторов время определения новой активной конфигурации может оказаться слишком большим. Если в сети используются заданные по умолчанию значе-

ния тайм-аутов, переход на новую конфигурацию может занять свыше 50 секунд: 20 секунд понадобится на констатацию факта потери связи с корневым коммутатором (истечение таймера — единственный способ узнать об этом событии в стандартном варианте STA) и еще 2 × 15 секунд потребуется для перехода портов в состояние продвижения.

Имеющиеся многочисленные нестандартные версии STA позволяют сократить время реконфигурирования за счет усложнения алгоритма, например добавления новых типов служебных сообщений. В 2001 году разработана стандартная ускоренная версия STA (спецификация IEEE 802.1w).

Агрегирование линий связи в локальных сетях

Список ключевых слов: агрегирование линий связи, транк, динамический способ распределения кадров, статический способ распределения кадров, протокол управления агрегированием линий связи.

Транки и логические каналы

Агрегирование линий связи (физических каналов) между двумя коммуникационными устройствами в один логический канал является еще одной формой использования избыточных альтернативных связей в локальных сетях.

Отличие техники агрегирования линий связи от алгоритма покрывающего дерева достаточно принципиально.

- Алгоритм STA переводит избыточные связи в горячий резерв, оставляя в рабочем состоянии только минимальный набор линий, необходимых для обеспечения связности сегментов сети. В этом случае *повышается надежность* сети, но не ее производительность.
- При агрегировании физических каналов все избыточные связи остаются в рабочем состоянии. В результате повышается как *надежность* сети, так и ее *производительность*.

При отказе одной из составляющих агрегированного логического канала, который часто называют **транком**, трафик распределяется между оставшимися линиями (рис. 16.3). На рисунке примером такой ситуации является транк 2, в котором один из физических каналов (центральный) отказал, так что все кадры передаются по оставшимся двум каналам. Этот пример демонстрирует повышение *надежности* при агрегировании.

Покажем теперь, как агрегирование линий связи повышает *производительность* сети. Так, на рисунке коммутаторы 1 и 3 соединены тремя параллельными линиями связи, что в три раза повышает производительность этого участка сети по сравнению со стандартным вариантом топологии дерева, которая не допускает таких параллельных связей. Повышение производительности связи между коммутаторами путем агрегирования линий связи в некоторых случаях является более эффективным, чем замена единственной линии связи более скоростной.

Например, несмотря на то что семейство Ethernet предлагает широкий выбор скоростей физического канала, от 10 Мбит/с до 10 Гбит/с, десятикратное повышение скорости при переходе от одного стандарта Ethernet к другому не всегда нужно и экономически оправдано. Например, если в установленных в сети коммутаторах отсутствует возможность добавления модуля с портом Gigabit Ethernet, то повышение скорости на некоторых каналах до 1000 Мбит/с потребует полной замены коммутаторов. В то же время вполне возможно, что у таких коммутаторов имеются свободные порты Fast Ethernet, поэтому скорость передачи данных можно было бы повысить, например, до 600 Мбит/с, объединив в агрегированный канал шесть портов Fast Ethernet.

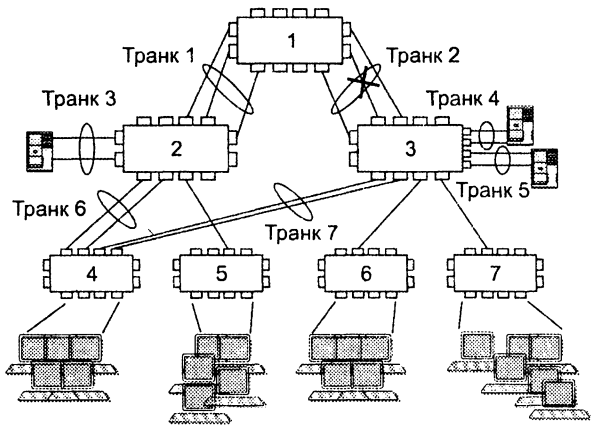


Рис. 16.3. Агрегирование физических каналов

Агрегирование линий связи является обобщением одного из способов применения альтернативных маршрутов (см. раздел «Альтернативные маршруты» в главе 6): «Сеть заранее находит два маршрута, однако использует только один». При агрегировании отыскивается N маршрутов (где $N \geq 2$), каждый из которых используется для одного потока, а при отказе какого-либо маршрута «пострадавший» поток переводится на любой из оставшихся (N – 1) работающих маршрутов.

Агрегирование линий связи используется как для связей между портами коммутаторов локальной сети, так и для связей между компьютером и коммутатором. Чаще всего этот вариант выбирают для высокоскоростных и ответственных серверов. В этом случае все сетевые адаптеры, входящие в транк, принадлежат одному компьютеру и разделяют один и тот же сетевой адрес. Поэтому для протокола IP или другого протокола сетевого уровня порты транка неразличимы, что соответствует концепции единого логического канала, лежащей в основе агрегирования.

Почти все методы агрегирования, применяемые в настоящее время, обладают существенным ограничением — в них учитываются только связи между двумя соседними коммутаторами сети и полностью игнорируется все, что происходит вне этого участка сети. Например, работа транка 1 никак не координируется с работой транка 2, и наличие обычной связи между коммутаторами 2 и 3, которая создает

вместе с транками 1 и 2 петлю, не учитывается. Поэтому технику агрегирования линий связи необходимо применять *одновременно* с алгоритмом покрывающего дерева — если администратор сети хочет использовать все топологические возможности объединения узлов сети. Для STA транк должен выглядеть как одна линия связи, тогда логика работы алгоритма останется в силе.

Существует большое количество фирменных реализаций механизма агрегирования линий связи. Наиболее популярные принадлежат, естественно, лидерам в секторе оборудования для локальных сетей. Это такие реализации, как Fast EtherChannel и Gigabit EtherChannel компании Cisco, MultiLink Trunking компании Nortel, Adaptive Load Balancing компании Intel и ряд других. Стандарт IEEE 802.3ad Link Aggregation обобщает эти подходы.

Борьба с «размножением» пакетов

Рассмотрим теперь подробней, в чем состоят особенности работы коммутатора в случае, когда его порты образуют транк. Во фрагменте сети, приведенном на рис. 16.4, два коммутатора — 1 и 2 — связаны четырьмя физическими каналами.

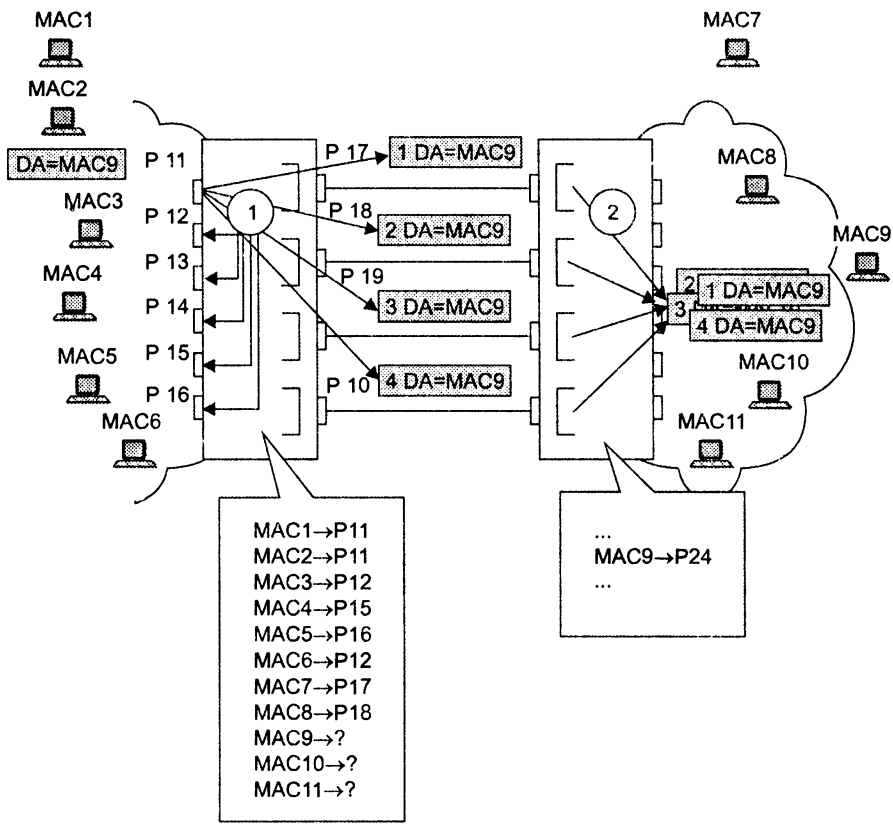


Рис. 16.4. Размножение пакетов с неизученным адресом при наличии параллельных каналов между коммутаторами

Необходимо отметить, что транк может быть односторонним или двусторонним. Каждый коммутатор контролирует только отправку кадра, принимая решение, на какой из выходных портов его нужно передать. Поэтому если оба коммутатора считают связывающие их каналы транком, то он будет двусторонним, в противном случае — односторонним.

Рисунок иллюстрирует поведение коммутатора 1 по отношению к параллельным каналам. В том случае, когда они не рассматриваются данным коммутатором как агрегированный канал, возникают проблемы с кадрами двух типов:

- кадрами с *еще не изученными* коммутатором уникальными адресами;
- кадрами, в которых указан *широковещательный* или *групповой адрес.*

Алгоритм прозрачного моста требует от коммутатора передавать кадр с неизученным (отсутствующим в таблице продвижения) адресом на все порты, кроме того, с которого кадр был принят. При наличии параллельных каналов такой кадр будет «размножен» в количестве, равном количеству каналов, — в приведенном примере коммутатор 2 примет четыре копии оригинального кадра.

При этом происходит также зацикливание кадров — они будут постоянно циркулировать между двумя коммутаторами, причем удалить их из сети окажется невозможно, так как в кадрах канального уровня отсутствует поле срока жизни, часто используемое в протоколах верхних уровней, например в IP и IPX.

В любом случае кадр с неизученным адресом повысит нагрузку на сеть за счет увеличения числа кадров, что чревато возникновением заторов, задержек и потерь данных. Помимо роста нагрузки дублирование кадров может привести также к неэффективной работе многих протоколов верхнего уровня. Примером может служить узел, работающий по протоколу TCP, для которого дублирование положительных квитанций, подтверждающих факт доставки данных адресату, служит косвенным признаком перегрузки сети.

Еще больше проблем создают кадры с широковещательным адресом — они всегда должны передаваться на все порты, кроме исходного, так что в любом случае «засорение» сети посторонним трафиком окажется значительным, и кадры будут зацикливаться.

С кадрами, у которых адрес назначения изучен, проблем у коммутаторов, связанных параллельными каналами, не возникает — коммутатор передает такой кадр на тот единственный порт, по которому этот кадр впервые пришел от источника.

Разработчики механизмов агрегирования учли проблемы, возникающие при обработке кадров с неизученными, широковещательными и групповыми адресами. Решение достаточно простое — все порты, связанные с параллельными каналами, считаются одним *логическим портом*, который и фигурирует в таблице продвижения вместо нескольких *физических портов.*

В примере, представленном на рис. 16.4, в таблице продвижения вместо портов P17, P18, P19 и P10 фигурирует логический порт AL11. С этим портом связаны адреса всех узлов, путь к которым лежит через коммутатор 2. При этом изучение нового адреса по кадру, поступившему от любого из физических портов, входящих в транк, приводит к появлению в таблице продвижения коммутатора новой записи с идентификатором логического порта. Поступающий в коммутатор кадр,

адрес назначения которого изучен и связан с идентификатором логического порта, передается на один (и только один!) выходной физический порт, входящий в состав транка. Точно так же коммутатор поступает с неизученными, широковещательными и групповыми адресами — для передачи кадра используется только одна из связей. На порты коммутатора, не входящие в транк, это изменение в логике обработки кадров не распространяется. Так, коммутатор 1 всегда передает кадр с неизученным или широковещательным адресом на порты P11–P16. Благодаря такому решению кадры не дублируются и описанные проблемы не возникают.

ВНИМАНИЕ

Вышесказанное справедливо только тогда, когда агрегированная линия связи сконфигурирована в качестве транка с обеих сторон.

Выбор порта

Остается открытым вопрос — какой из портов коммутатора нужно использовать для продвижения кадра через транк?

Можно предложить несколько вариантов ответов. Учитывая, что одной из целей агрегирования линий связи является повышение суммарной производительности участка сети между двумя коммутаторами (или коммутатором и сервером), следует распределять кадры по портам транка динамически, учитывая текущую загрузку каждого порта и направляя кадры в наименее загруженные (с меньшей длиной очереди) порты. **Динамический способ распределения кадров**, учитывающий текущую загрузку портов и обеспечивающий баланс нагрузки между всеми связями транка, должен приводить, казалось бы, к максимальной пропускной способности транка.

Однако такое утверждение справедливо не всегда, так как в нем не учитывается поведение протоколов верхнего уровня. Существует ряд таких протоколов, производительность которых может существенно снизиться, если пакеты сеанса связи между двумя конечными узлами будут приходить не в том порядке, в котором они отправлялись узлом-источником. А такая ситуация может возникнуть, если два или более последовательных кадра одного сеанса будут передаваться через разные порты транка — по причине того, что очереди в буферах этих портов имеют разную длину. Следовательно, и задержка передачи кадра может быть разной, так что более поздний кадр обгонит более ранний.

Поэтому в большинстве реализаций механизмов агрегирования используются методы статического, а не динамического распределения кадров по портам. **Статический способ распределения кадров** подразумевает закрепление за определенным портом транка потока кадров определенного сеанса между двумя узлами, так что все кадры будут проходить через одну и ту же очередь и их упорядоченность не изменится.

Обычно при статическом распределении выбор порта для некоторого сеанса выполняется на основании определенных признаков, имеющихся в поступающих

пакетах. Чаще всего такими признаками являются MAC-адреса источника или приемника, или оба вместе. В популярной реализации механизма Fast EtherChannel компании Cisco для коммутаторов семейства Catalyst 5000/6000 при выборе номера порта транка используется операция исключающего ИЛИ (XOR) над двумя последними битами MAC-адресов источника и приемника. Результат этой операции имеет четыре значения: 00, 01, 10 и 11, которые и являются условными номерами портов транка.

На рис. 16.5 приведен пример сети, в которой работает механизм Fast EtherChannel. Распределение потоков для сеансов между конечными узлами получается при этом достаточно случайным. Так как распределение не учитывает реальной нагрузки, которую создает каждый сеанс, общая пропускная способность транка может использоваться нерационально, особенно если интенсивности сеансов намного отличаются друг от друга. Кроме того, алгоритм распределения не гарантирует даже равномерного в количественном отношении распределения сеансов по портам. Случайный набор MAC-адресов в сети может привести к тому, что через один порт будут проходить несколько десятков сеансов, а через другой — только два-три. Выравнивание нагрузки портов можно при данном алгоритме достигнуть только при большом количестве компьютеров и сеансов связи между ними.

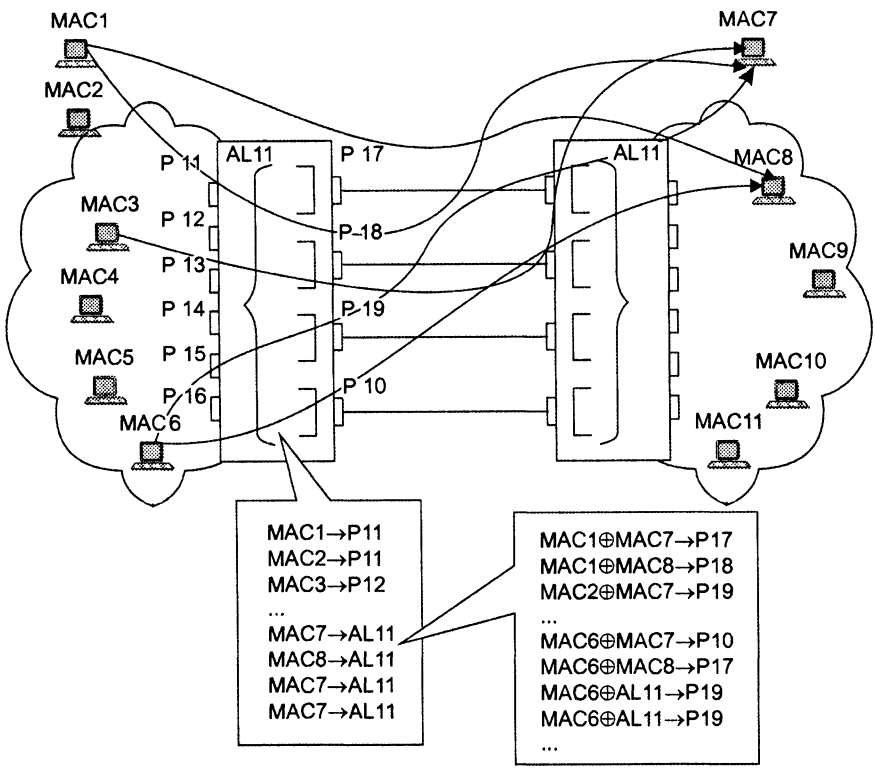


Рис. 16.5. Пример сети с механизмом Fast EtherChannel

Можно предложить и другие способы распределения сеансов по портам. Например, в соответствии с IP-адресами пакетов, которые инкапсулированы в кадры канального уровня, типами прикладных протоколов (почта по одному порту, веб-трафик по другому и т. д.). Полезным оказывается назначение порту сеансов с MAC-адресами, которые были изучены именно через этот порт — чтобы трафик сеанса проходил через один и тот же порт в обоих направлениях.

Стандартный способ создания агрегированных каналов, описанный в спецификации 802.3ad, предполагает возможность создания логического порта путем объединения нескольких физических портов, принадлежащих разным коммутаторам. Для того чтобы коммутаторы могли автоматически обеспечиваться информацией о принадлежности какого-либо физического порта определенному логическому порту, в спецификации предложен служебный **протокол управления агрегированием линий связи** (Link Control Aggregation Protocol, LCAP). Поэтому возможны такие конфигурации агрегированных каналов, которые увеличивают отказоустойчивость сети не только на участках между двумя коммутаторами, но и в более сложных топологиях (рис. 16.6).

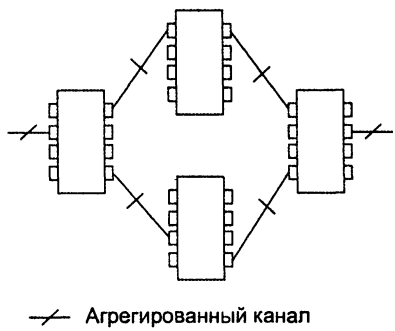


Рис. 16.6. Распределенное агрегирование каналов

При отказе какого-либо канала транка все пакеты сеансов, назначенные для соответствующего порта, будут направляться на один из оставшихся портов. Обычно восстановление связности при таком отказе занимает от единиц до десятков миллисекунд. Это объясняется тем, что во многих реализациях транка после отказа физического канала все MAC-адреса, которые были с ним связаны, принудительно помечаются как неизученные. Затем коммутатор повторяет процедуру изучения этих адресов. После этого процедура назначения сеанса портам выполняется заново, естественно, учитываются только работающие порты. Так как тайм-ауты в сеансах протоколов локальных сетей обычно небольшие, коротким оказывается и время восстановления соединения.

Виртуальные локальные сети

Список ключевых слов: виртуальная локальная сеть, коммутатор 3-го уровня, стандарт IEEE 802.1Q тег виртуальной сети.

Важным свойством коммутатора локальной сети является способность контролировать передачу кадров между сегментами сети. По различным причинам (соблюдение прав доступа, политика безопасности и т. д.) некоторые кадры не следует передавать по адресу назначения.

Как мы видели в главе 15, такого типа ограничения можно реализовать с помощью *пользовательских фильтров*. Однако пользовательский фильтр может запретить коммутатору передачу кадров только по конкретным адресам, а широковещательный трафик он *обязан* передать всем сегментам сети. Так требует алгоритм его работы. Поэтому сети, созданные на основе коммутаторов, иногда называют *плоскими* — из-за отсутствия барьеров на пути широковещательного трафика. Технология виртуальных локальных сетей позволяет преодолеть указанное ограничение.

Виртуальной локальной сетью (Virtual LAN, VLAN) называется группа узлов сети, трафик которой, в том числе широковещательный, на канальном уровне полностью изолирован от трафика других узлов сети.

Это означает, что передача кадров между разными виртуальными сетями на основании адреса канального уровня невозможна независимо от типа адреса — уникального, группового или широковещательного. В то же время внутри виртуальной сети кадры передаются по технологии коммутации, то есть только на тот порт, который связан с адресом назначения кадра.

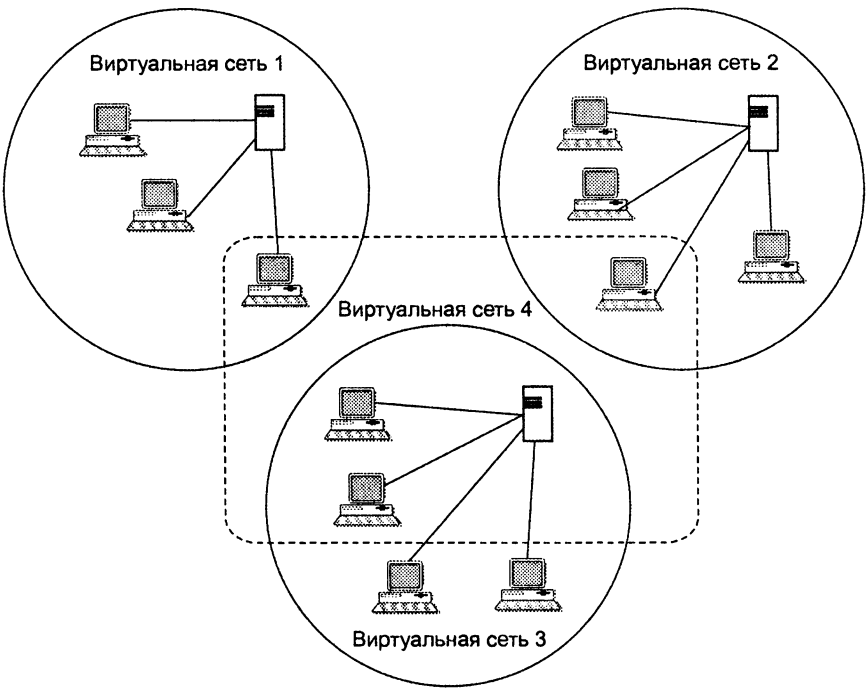


Рис. 16.7. Виртуальные локальные сети

Виртуальные локальные сети могут *перекрываться*, если один или несколько компьютеров входят в состав более чем одной виртуальной сети. На рис. 16.7 сервер электронной почты входит в состав виртуальных сетей 3 и 4. Это означает, что его кадры передаются коммутаторами всем компьютерам, входящим в эти сети. Если же какой-то компьютер входит в состав только виртуальной сети 3, то его кадры до сети 4 доходить не будут, но он может взаимодействовать с компьютерами сети 4 через общий почтовый сервер. Такая схема не полностью защищает виртуальные сети друг от друга — так, широковещательный шторм, возникший на сервере электронной почты, затопит и сеть 3, и сеть 4.

Говорят, что виртуальная сеть образует *домен широковещательного трафика*, по аналогии с доменом коллизий, который образуется повторителями сетей Ethernet.

Назначение виртуальных сетей

Основное назначение технологии VLAN состоит в облегчении процесса создания изолированных сетей, которые затем обычно связываются между собой с помощью маршрутизаторов. Такое построение сети создает мощные барьеры на пути нежелательного трафика из одной сети в другую. Сегодня считается очевидным, что любая крупная сеть должна включать маршрутизаторы, иначе потоки ошибочных кадров, например широковещательных, будут периодически «затапливать» всю сеть через прозрачные для них коммутаторы, приводя ее в неработоспособное состояние.

Достоинством технологии виртуальных сетей является то, что она позволяет создавать полностью изолированные сегменты сети путем логического конфигурирования коммутаторов, не прибегая к изменению физической структуры.

До появления технологии VLAN для создания отдельной сети использовались либо физически изолированные сегменты коаксиального кабеля, либо несвязанные между собой сегменты, построенные на повторителях и мостах. Затем эти сети связывались маршрутизаторами в единую составную сеть (рис. 16.8).

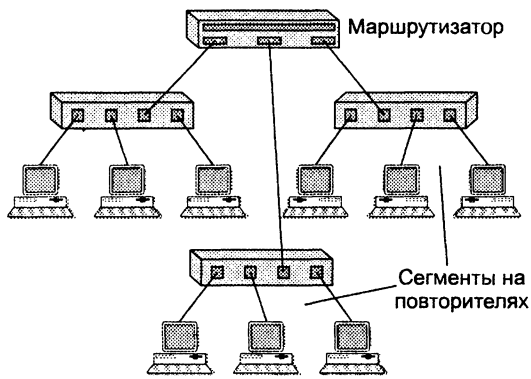


Рис. 16.8. Составная сеть, состоящая из сетей, построенных на основе повторителей

Изменение состава сегментов (переход пользователя в другую сеть, дробление крупных сегментов) при таком подходе подразумевает физическую перекоммутацию разъемов на передних панелях повторителей или в кроссовых панелях, что не очень удобно в больших сетях — много физической работы, к тому же высока вероятность ошибки.

Частично эта проблема была решена с помощью многосегментных концентраторов, которые дали возможность программировать состав разделяемого сегмента без физической перекоммутации (см. главу 14). Однако решение действительно является только «частичным», так как накладывает большие ограничения на структуру сети. Во-первых, количество сегментов такого концентратора обычно невелико, поэтому выделить каждому узлу свой сегмент, как это можно сделать с помощью коммутатора, нереально. Во-вторых, при таком подходе вся работа по передаче данных между сегментами ложится на маршрутизаторы, а коммутаторы со своей высокой производительностью остаются «не у дел». Поэтому сети, построенные на основе концентраторов с конфигурационной коммутацией, попрежнему основаны на разделении среды передачи данных между большим количеством узлов и, следовательно, обладают гораздо меньшей производительностью по сравнению с сетями, построенными на основе коммутаторов.

Для связи виртуальных сетей в общую сеть требуется привлечение сетевого уровня. Он может быть реализован в отдельном маршрутизаторе, а может работать и в составе программного обеспечения коммутатора, который тогда становится комбинированным устройством — так называемым **коммутатором 3-го уровня**. Коммутаторы 3-го уровня рассматриваются в главе 20.

Технология виртуальных сетей долгое время не стандартизировалась, хотя и была реализована в очень широком спектре моделей коммутаторов разных производителей. Положение изменилось после принятия в 1998 году стандарта IEEE 802.1Q, который определяет базовые правила построения виртуальных локальных сетей, не зависящие от протокола канального уровня, поддерживаемого коммутатором.

Создание виртуальных сетей на базе одного коммутатора

При создании виртуальных сетей на основе одного коммутатора обычно используется механизм *группирования портов* коммутатора (рис. 16.9). При этом каждый порт приписывается той или иной виртуальной сети. Кадр, пришедший от порта, принадлежащего, например, виртуальной сети 1, никогда не будет передан порту, который не принадлежит этой виртуальной сети. Порт можно приписать нескольким виртуальным сетям, хотя на практике так делают редко — пропадает эффект полной изоляции сетей.

ПРИМЕЧАНИЕ

Если к порту коммутатора подключен сегмент, построенный на основе повторителя, то узлы такого сегмента не имеет смысла включать в разные виртуальные сети — все равно трафик этих узлов будет общим.

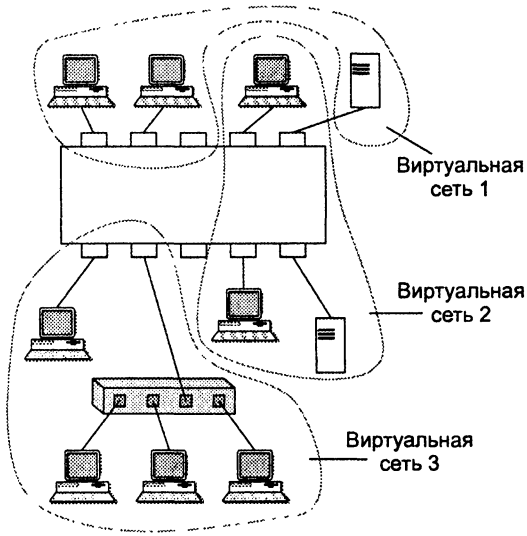


Рис. 16.9. Виртуальные сети, построенные на одном коммутаторе

Создание виртуальных сетей путем группирования портов не требует от администратора большого объема ручной работы — достаточно каждый порт приписать к одной из нескольких заранее поименованных виртуальных сетей. Обычно такая операция выполняется с помощью специальной программы, прилагаемой к коммутатору.

Второй способ образования виртуальных сетей основан на *группировании MAC-адресов*. Каждый MAC-адрес, который изучен коммутатором, приписывается той или иной виртуальной сети. При существовании в сети множества узлов этот способ требует от администратора выполнения большого количества ручных операций. Однако при построении виртуальных сетей на основе нескольких коммутаторов он оказывается более гибким, чем группирование портов.

Создание виртуальных сетей на базе нескольких коммутаторов

Рисунок 16.10 иллюстрирует проблему, возникающую при создании виртуальных сетей на основе нескольких коммутаторов, поддерживающих технику *группирования портов*.

Если узлы какой-либо виртуальной сети подключены к разным коммутаторам, то для подключения каждой такой сети на коммутаторах должна быть выделена специальная пара портов. В противном случае, если коммутаторы будут связаны только одной парой портов, информация о принадлежности кадра той или иной виртуальной сети при передаче из коммутатора в коммутатор будет утеряна. Таким образом, коммутаторы с группированием портов требуют для своего соединения столько портов, сколько виртуальных сетей они поддерживают. Порты и кабели используются при таком способе очень расточительно. Кроме того, при

соединении виртуальных сетей через маршрутизатор для каждой виртуальной сети выделяется отдельный кабель и отдельный порт маршрутизатора, что также приводит к большим накладным расходам.

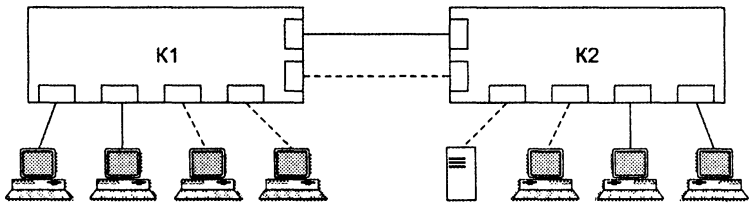


Рис. 16.10. Построение виртуальных сетей на нескольких коммутаторах с группированием портов

Группирование MAC-адресов в виртуальную сеть на каждом коммутаторе избавляет от необходимости связывать их по нескольким портам, поскольку в этом случае MAC-адрес является меткой виртуальной сети. Однако этот способ требует выполнения большого количества ручных операций по маркировке MAC-адресов на каждом коммутаторе сети.

Описанные два подхода основаны только на добавлении дополнительной информации к адресным таблицам коммутатора, и в них отсутствует возможность встраивания в передаваемый кадр информации о принадлежности кадра виртуальной сети. В остальных подходах используются имеющиеся или дополнительные поля кадра для сохранения информации о принадлежности кадра той или иной виртуальной локальной сети при его перемещениях между коммутаторами сети. При этом нет необходимости запоминать в каждом коммутаторе принадлежность всех MAC-адресов составной сети виртуальным сетям.

Дополнительное поле с пометкой о номере виртуальной сети используется только тогда, когда кадр передается от коммутатора к коммутатору, а при передаче кадра конечному узлу оно обычно удаляется. При этом модифицируется протокол взаимодействия «коммутатор-коммутатор», а программное и аппаратное обеспечение конечных узлов остается неизменным. До принятия стандарта IEEE 802.1Q существовало много фирменных протоколов этого типа, но все они имели один недостаток — оборудование различных производителей оказывалось несовместимым при образовании VLAN.

В стандарте **IEEE 802.1Q** для хранения номера виртуальной сети предусмотрен дополнительный заголовок в 2 байта, который этот протокол делит с протоколом 802.1p (он рассматривается в следующем разделе этой главы). Помимо 3 бит для хранения приоритета кадра, описанных стандартом 802.1p, в этом заголовке 12 бит используются для хранения номера виртуальной сети, к которой принадлежит кадр. Эта дополнительная информация, которая называется **тегом виртуальной сети**, позволяет коммутаторам разных производителей создавать до 4096 общих виртуальных сетей. Кадр с такой информацией называют «помеченным». Длина помеченного кадра Ethernet увеличивается на 4 байта, так как помимо 2 байт собственно тега добавляются еще 2 байта. Структура помеченного

кадра Ethernet показана на рис. 16.11. При добавлении заголовка 802.1p/Q поле данных уменьшается на 2 байта.

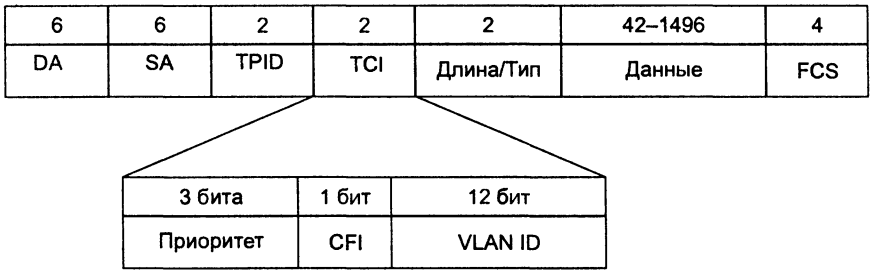


Рис. 16.11. Структура помеченного кадра Ethernet

Введение стандарта 802.1Q позволило производителям оборудования преодолеть различия в фирменных реализациях VLAN и добиться совместимости при построении виртуальных локальных сетей. Поддерживают технику VLAN как производители коммутаторов, так и сетевых адаптеров. В последнем случае сетевой адаптер может генерировать и принимать помеченные кадры Ethernet, содержащие поле тега виртуальной сети. Если сетевой адаптер генерирует помеченные кадры, то тем самым он определяет их принадлежность к той или иной виртуальной локальной сети, поэтому коммутатор должен обрабатывать их соответствующим образом, то есть передавать или не передавать на выходной порт в зависимости от принадлежности порта. Драйвер сетевого адаптера может получить номер своей (или своих) виртуальной локальной сети путем ручного конфигурирования либо от некоторого приложения, работающего на данном узле или на одном из серверов сети.

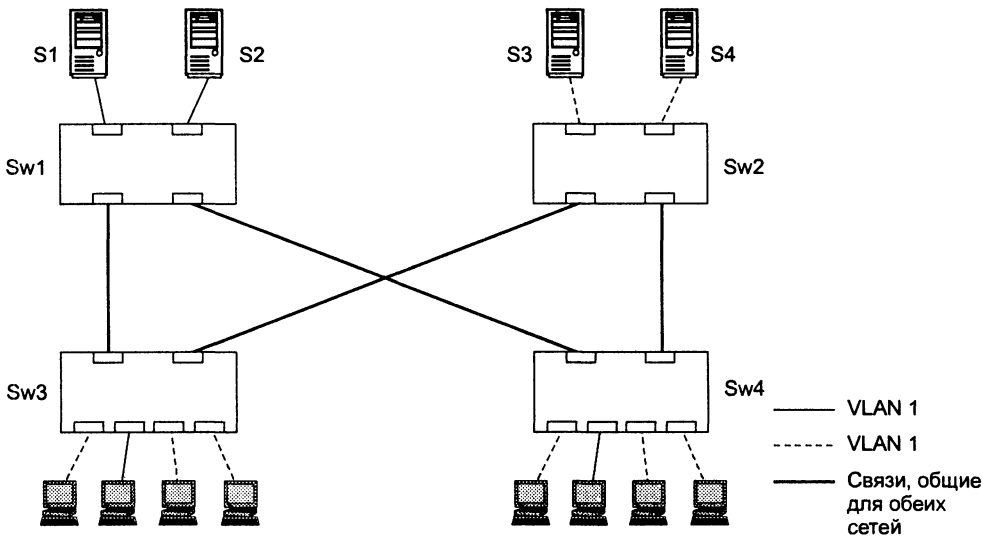


Рис. 16.12. Сети с VLAN и избыточные связи

Существование виртуальных локальных сетей в сети влияет на выбор активной топологии покрывающего дерева. Рассмотрим пример на рис. 16.12.

В этой сети образовано две виртуальные локальные сети: VLAN 1 и VLAN 2. Эти виртуальные сети построены на основе техники группирования портов. На рисунке связи между портами, принадлежащими VLAN 1, показаны тонкими сплошными линиями, а связи между портами VLAN 2 — пунктирными.

Если при построении активной топологии STA мы не будем принимать во внимание наличие в сети виртуальных сетей, то получим покрывающее дерево, показанное на рис. 16.13 (в качестве корневого коммутатора выбран коммутатор Sw1). Очевидно, что эта топология не эффективна для VLAN 2, так как, например, путь от компьютера W1 до сервера S3 проходит через четыре коммутатора. Для сравнения, путь от компьютеров сети VLAN 1 до сервера S1 проходит через два коммутатора. Если бы в качестве корневого коммутатора был выбран коммутатор Sw2, то такая топология оказалась бы не эффективной для сети VLAN 1.

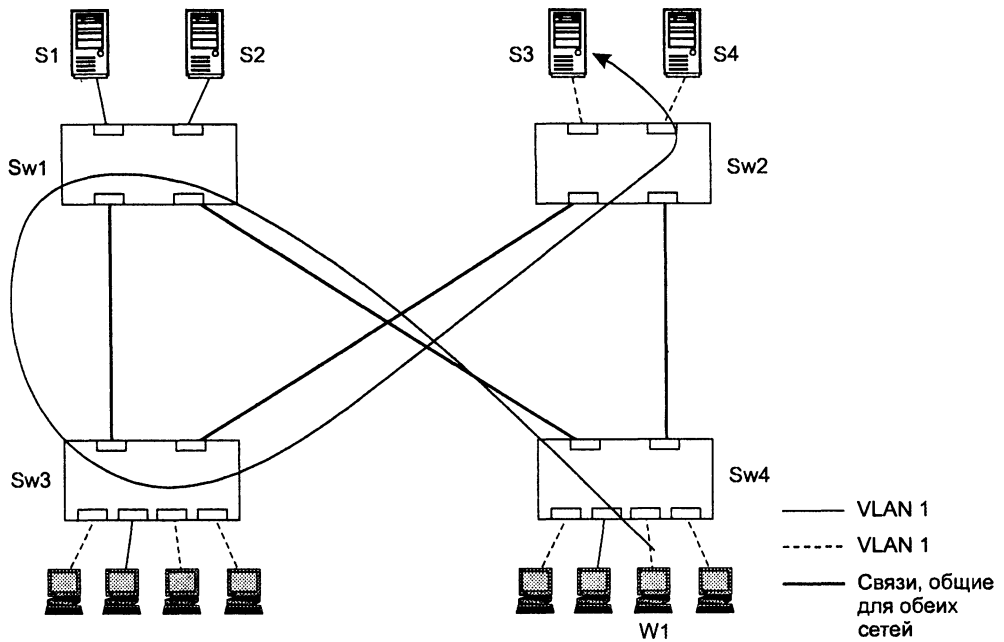


Рис. 16.13. Покрывающее дерево, построенное без учета существования виртуальных локальных сетей

Другое решение состоит в том, чтобы активная топология строилась для каждой виртуальной сети отдельно. В рассматриваемом примере такой подход приводит к двум деревьям с корневым коммутатором Sw1 для VLAN 1 и корневым коммутатором Sw2 для VLAN 2 (рис. 16.14).

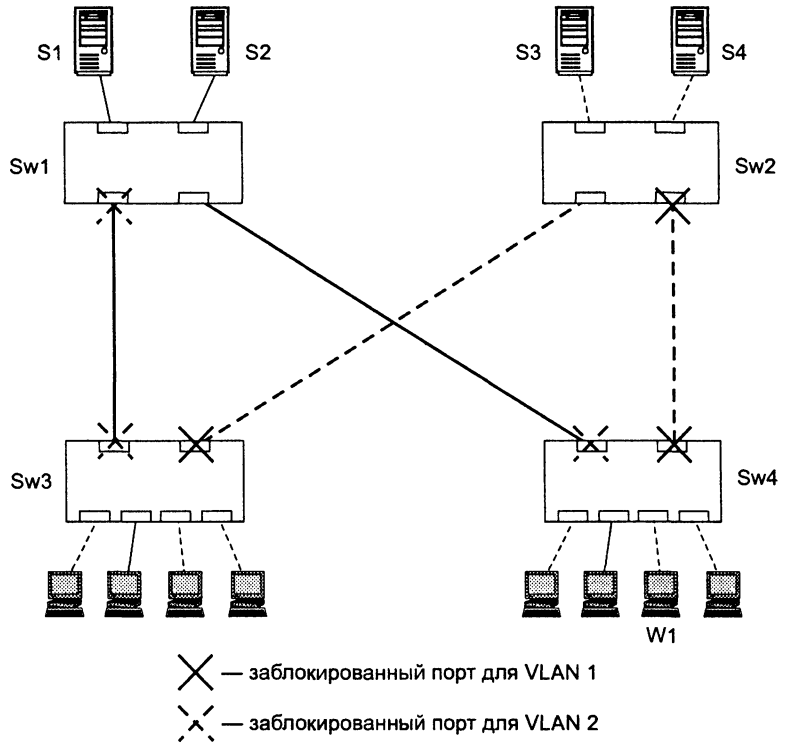


Рис. 16.14. Покрывающие деревья, построенные с учетом существования локальных сетей

Качество обслуживания в виртуальных сетях

Коммутаторы локальных сетей поддерживают практически все механизмы QoS, которые мы обсуждали в главе 7. Это утверждение относится к коммутаторам локальных сетей как к классу коммуникационных устройств, каждая же конкретная модель коммутатора может быть наделена только определенным набором механизмов поддержания параметров QoS или же не иметь их вовсе. Как правило, коммутаторы рабочих групп средств QoS не поддерживают, в то время как для магистральных коммутаторов эта поддержка является обязательной.

Классификация трафика. Коммутаторы локальных сетей являются устройствами второго уровня, которые анализируют заголовки только протоколов канального уровня. Поэтому коммутаторы обычно используют для классификации трафика только MAC-адреса источника и приемника, а также номер порта, через который поступил кадр. Возможно также использование для классификации значения произвольного подполя внутри поля данных, заданного путем указания смещения в байтах. Эти способы не очень удобны для администратора, которому необходимо, например, отделить голосовой трафик от трафика передачи файлов. Поэтому некоторые коммутаторы, не поддерживая протоколы верхних уровней в полном объеме (например, не применяя протокол IP для продвиже-

ния пакетов), выполняют классификацию на основе признаков, содержащихся в заголовках пакетов этих протоколов — IP-адресах и портах TCP/UDP.

Маркирование трафика обычно выполняется на границе сети, а затем его результаты используются во всех промежуточных устройствах сети. В кадре Ethernet 802.3 отсутствует поле, в которое можно было бы поместить результат маркировки трафика. Однако этот недостаток исправляет спецификация 802.1p, в которой имеются три бита дополнительного заголовка 802.1Q/p для хранения приоритета кадра.

Фактически эти три бита используются для хранения признака одного из восьми классов трафика. Именно так трактует это поле стандарт 802.1D-1998, куда вошла спецификация 802.1p. В приложении H стандарта 802.1D-1998 даются рекомендации по разделение всего трафика локальных сетей на семь классов, которые представлены в табл. 16.1.

Таблица 16.1. Типы трафика в локальной сети

Приоритет	Обозначение	Тип трафика
1	BK	Фоновый трафик
2	—	Экономный трафик
0 (вариант по умолчанию)	BE	Трафик, передаваемый с максимальными усилиями («по возможности»)
3	EE	Приоритетный трафик
4	CL	Контролируемый трафик
5	VI	Видео (задержка и джиттер менее 100 мс)
6	VO	Голос (задержка и джиттер менее 10 мс)
7	NC	Управление сетью

Фоновый трафик — это наименее чувствительный к задержкам трафик, например трафик резервного копирования, источник которого может передавать большие объемы данных, поэтому его целесообразно выделить в особый класс, чтобы он не замедлял обработку других типов трафика.

Классы трафика BE (Best Effort), EE (Excellent Effort) и CE (Controlled Effort) не являются классами реального времени, то есть не предъявляют жестких требований к предельным уровням задержек. Однако для этих классов желательно обеспечение некоторого минимального уровня пропускной способности. Обслуживание этих классов целесообразно проводить с помощью механизма взвешенных очередей.

Трафики видео, голоса и управления сетью чувствительны к задержкам, в таблице приведены рекомендуемые значения пороговых величин задержек. Их обслуживание целесообразно проводить с помощью приоритетных очередей. Трафику управления сетью дается высший приоритет при обслуживании, так как от свое-

временного принятия решения и доставки управляющей информации сетевым устройствам зависят любые характеристики сети.

Управление очередями. Коммутатор, поддерживающий параметры QoS, позволяет использовать несколько очередей для дифференцированной обработки классов трафика. Очереди могут обслуживаться в соответствии с алгоритмом приоритетной обработки, взвешенного обслуживания или на основе комбинации этих алгоритмов.

Коммутатор обычно поддерживает некоторое максимальное количество очередей, которое может оказаться меньше, чем требуемое число классов трафика. В этой ситуации несколько классов будут обслуживаться одной очередью, то есть фактически сольются в один класс. Стандарт 802.1D-1998 дает рекомендации в отношении того, какие классы трафика нужно реализовывать в сети в условиях ограниченного количества очередей в коммутаторах (табл. 16.2).

Таблица 16.2. Классы трафика и количество очередей

Число очередей	Класс трафика							
1	BE							
2	BE				VO			
3	BE				CL		VO	
4	BK		BE		CL		VO	
5	BK		BE		CL	VI	VO	
6	BK		BE	EE	CL	VI	VO	
7	BK		BE	EE	CL	VI	VO	NC
8	BK	–	BE	EE	CL	VI	VO	NC

При существовании только одной очереди в сети существует только один класс трафика — BE (Best Effort). Качество его обслуживания невозможно улучшить за счет управления очередями, хотя остаются такие возможности, как обратная связь и резервирование полосы пропускания.

Две очереди дают возможность разделить трафик на два класса — BE и VO (Voice). При таком делении к классу VO следует относить любой чувствительный к задержкам трафик, то есть не только голос, но и видео, а также трафик управления сетью.

Дальнейшее увеличение количества очередей позволяет более дифференцированно обслуживать трафик, вплоть до рекомендуемых семи классов. Предложенная схема является только рекомендацией, администратор сети может делить трафик на классы по своему усмотрению.

Кроме того, допускается обслуживание индивидуальных потоков трафика, но при этом каждый коммутатор должен самостоятельно выделять поток из общего трафика, так как в кадре Ethernet нет поля для переноса через сеть метки потока.

В качестве признака класса трафика можно использовать номер виртуальной сети. Этот признак можно также комбинировать со значениями поля приоритета кадра, получая большое число различных классов.

Резервирование и профилирование. Коммутаторы локальных сетей поддерживают методы резервирования пропускной способности интерфейсов для классов трафика или индивидуальных потоков. Обычно коммутатор разрешает назначить классу или потоку минимальную скорость передачи данных, которая гарантируется в периоды перегрузок, а также максимальную скорость передачи данных, которая контролируется механизмом профилирования.

Для коммутаторов локальных сетей не существует стандартного протокола резервирования ресурсов. Поэтому для выполнения резервирования администратор сети должен сконфигурировать каждый коммутатор сети отдельно.

Ограничения мостов и коммутаторов

Список ключевых слов: отсутствие петель, широковещательный шторм, фильтрация трафика, MAC-адрес, трансляция протоколов, коммутируемая сеть, виртуальные локальные сети, транки.

Применение коммутаторов позволяет преодолеть ограничения, свойственные сетям с разделяемой средой. Коммутируемые локальные сети могут покрывать значительные территории, плавно переходя в сети мегаполисов; они могут состоять из сегментов различной пропускной способности, образуя сети с очень высокой производительностью; они могут использовать альтернативные маршруты для повышения надежности и производительности. Однако построение сложных сетей только на основе повторителей, мостов и коммутаторов, то есть без применения устройств сетевого уровня, имеет существенные ограничения.

- Серьезные ограничения по-прежнему накладываются на топологию коммутируемой локальной сети. Требование *отсутствия петель* преодолевается с помощью техники STA и агрегирования каналов только частично. Действительно, STA не позволяет использовать все альтернативные маршруты для передачи пользовательского трафика, а агрегирование каналов разрешает так делать только на участке сети между двумя соседними коммутаторами. Подобные ограничения не позволяют применять многие эффективные топологии, которые могли бы использоваться для передачи трафика.
- Логические сегменты сети, расположенные между коммутаторами, *слабо изолированы* друг от друга, а именно не защищены от так называемых широковещательных штормов. Использование же механизма виртуальных сетей, реализованного во многих коммутаторах, хотя и позволяет достаточно гибко создавать изолированные по трафику группы станций, но при этом изолирует их полностью, то есть так, что узлы одной виртуальной сети не могут взаимодействовать с узлами другой виртуальной сети.
- В сетях, построенных на основе мостов и коммутаторов, достаточно *сложно решается задача фильтрации трафика* на основе данных, содержащихся в пакете.

В таких сетях фильтрации выполняется только с помощью пользовательских фильтров, для создания которых администратору приходится иметь дело с двоичным представлением содержимого пакетов.

- Реализация транспортной подсистемы только средствами физического и канального уровней приводит к *недостаточно гибкой, одноуровневой системе адресации*: в качестве адреса назначения используется MAC-адрес, жестко связанный с сетевым адаптером.
- У коммутаторов *ограничены возможности по трансляции протоколов* при создании гетерогенной сети. Они не могут транслировать WAN-протоколы в LAN-протоколы из-за различий в системе адресации этих сетей, а также различных значений максимального размера поля данных.

Наличие серьезных ограничений у протоколов канального уровня показывает, что построение на основе средств этого уровня больших неоднородных сетей является весьма проблематичным. Естественное решение в этих случаях — привлечение средств более высокого, сетевого уровня.

Коммутируемая сеть завода «Трансмаш»

В главе 12 мы рассмотрели структуру локальной сети завода «Трансмаш», построенной на повторителях Ethernet 10 Мбит/с. Тот пример соответствовал типичной для начала 90-х годов ситуации, когда единая разделяемая среда с пропускной способностью 10 Мбит/с полностью удовлетворяла потребности предприятия в обмене трафиком между немногочисленными компьютерами подразделений. Здесь мы опишем модернизированный вариант локальной сети этого завода, который стал характерен для многих крупных локальных сетей второй половины 90-х годов.

Основной особенностью этой локальной сети является то, что она *полностью построена на коммутаторах* (рис. 16.15). Переход на коммутируемую сеть был продиктован резко возросшими в середине 90-х годов требованиями, предъявляемыми к производительности и надежности локальной сети. К этому времени компьютеризованная обработка данных на заводе «Трансмаш» стала одним из основных средств производства, при этом увеличилось как число компьютеров, так и качественно изменились приложения, которые стали передавать мультимедийную информацию больших объемов.

Основу локальной сети каждого из пяти зданий завода составляет мощный центральный коммутатор на основе шасси, оснащенный портами Fast Ethernet и Gigabit Ethernet (коммутаторы BS1–BS5). Коммутатор здания объединяет коммутаторы этажей, которые подключены к нему транками, состоящими из двух-трех портов Fast Ethernet. Каждый коммутатор этажа применяется для подключения пользовательского оборудования двух типов: персональных компьютеров и технологического оборудования (более детально эти соединения показаны на примере сети здания 2). Пользователи персональных компьютеров работают с приложениями автоматизированной системы управления предприятием (АСУП), а технологическое оборудование образует автоматизированную систему управления технологическими процессами (АСУТП).

Центральные коммутаторы зданий 2, 3 и 4 образуют магистраль локальной сети завода. Они объединены двухпортовыми транками Gigabit Ethernet, что обеспечивает большой запас магистрали по производительности. Здания 5 и 4 подключены к магистрали с помощью обычных (без транков) соединений Gigabit Ethernet. Для связи коммутаторов этих зданий задействуется многомодовое оптическое волокно, которое было

проложено еще для локальной сети на повторителях и качества которого оказалось достаточно для устойчивой работы портов 1000Base-SX.

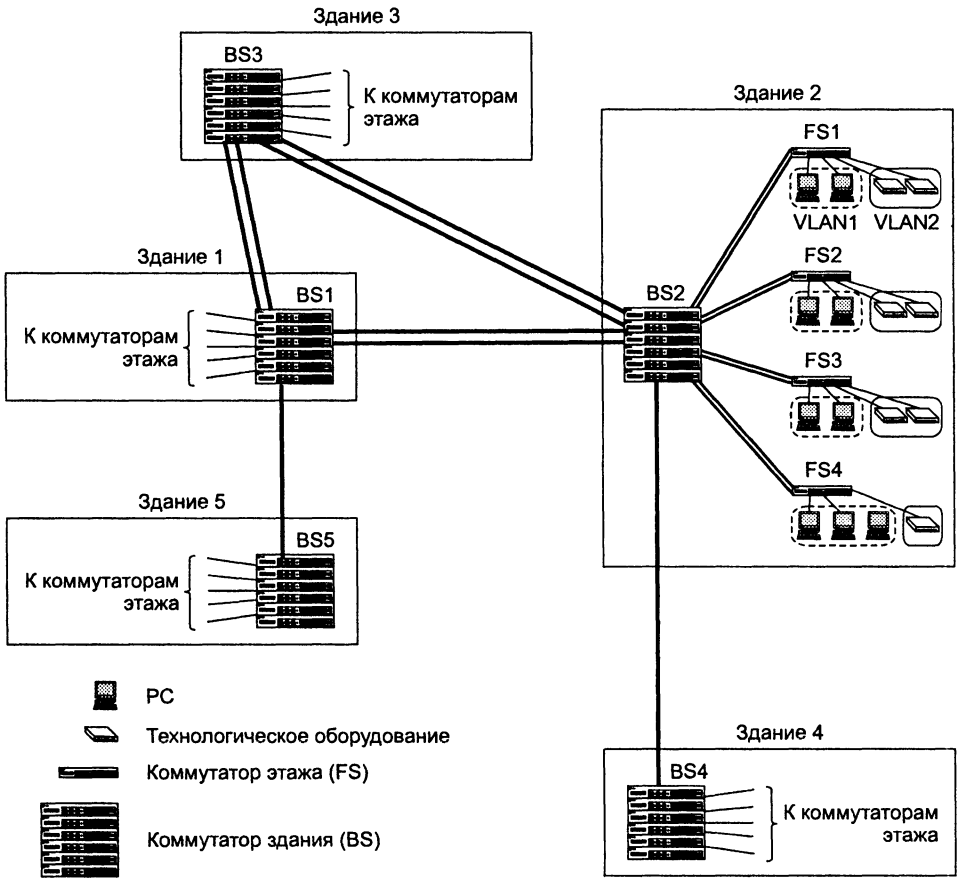


Рис. 16.15. Коммутируемая сеть завода «Трансмаш»

В сети «Трансмаш» передается трафик приложений двух типов: АСУП и АСУТП. Эти классы трафика отличаются требованиями к качеству обслуживания, например, трафик АСУТП является трафиком реального времени, а АСУП — нет. Поэтому в сети «Трансмаш» организованы две виртуальные локальные сети — VLAN 1 для трафика АСУП и VLAN 2 для трафика АСУТП. Это позволяет надежно изолировать каждый тип трафика и, кроме того, упрощает поддержку параметров QoS коммутаторами, так как признаком необходимости обработки трафика в приоритетной очереди является номер VLAN, в данном случае — номер 2.

Так как магистраль локальной сети обладает избыточными связями, то коммутаторы используют алгоритм STA, причем отдельно для каждой виртуальной локальной сети. Для VLAN 1 резервной является связь между коммутаторами BS4 и BS2, а для VLAN 2 — между коммутаторами BS4 и BS3. Обмен данными между АСУП и АСУТП происходит за счет того, что несколько серверов являются членами обеих сетей.

Выводы

Для автоматического поддержания в сложных сетях резервных связей в коммутаторах реализуется алгоритм покрывающего дерева. Этот алгоритм описан в документе IEEE 802.1D и основан на периодическом обмене коммутаторов специальными кадрами, с помощью которых выявляются и блокируются петлевидные связи в сети.

Протокол STA находит конфигурацию покрывающего дерева за три этапа. На первом этапе определяется корневой коммутатор, на втором — корневые порты, на третьем — назначенные порты сегментов.

Недостатком протокола STA 802.1D является сравнительно большое время установления новой активной конфигурации — около 50 с. Новый стандарт 802.1w устраняет этот недостаток.

Агрегирование нескольких физических каналов в один логический является одной из форм использования нескольких активных альтернативных маршрутов в локальных сетях на коммутаторах. Агрегирование каналов повышает как производительность, так и надежность сети.

Агрегированный канал может быть образован не только между двумя соседними коммутаторами, но и распределяться между портами нескольких коммутаторов. Для автоматического уведомления о принадлежности физического порта определенному агрегированному порту разработан протокол LCAP.

Технология виртуальных локальных сетей (VLAN) позволяет в сети, построенной на коммутаторах, программным путем создать изолированные группы конечных узлов, между которыми отсутствует любой трафик, в том числе широковещательный.

Конфигурирование VLAN обычно ведется путем группирования портов или MAC-адресов.

Для построение виртуальной локальной сети на основе нескольких коммутаторов желательно помечать передаваемые кадры специальной меткой — тегом, идентифицирующем номер сети, которой принадлежит отправитель кадра. Стандартный формат тега VLAN определен в спецификации 802.1Q.

Коммутаторы LAN поддерживают многие механизмы QoS: классификацию и профилирование трафика, приоритетные и взвешенные очереди, резервирование пропускной способности.

Вопросы и задания

1. Для какой цели используется алгоритм покрывающего дерева?
2. Дайте определение покрывающего дерева.
3. Какой порт коммутатора называется корневым?
4. Какой порт называется назначенным?
5. Каким образом измеряется расстояние между коммутаторами в STA?
6. Назовите три этапа построения активной топологии покрывающего дерева.
7. Каким образом выбирается корневой порт из нескольких претендентов, если расстояния до корневого коммутатора у них равны?
8. Может ли администратор влиять на выбор корневого коммутатора?
9. Каким образом коммутаторы решают, что выбор активной топологии завершен?

10. Что побуждает коммутатор начать процедуру поиска новой активной топологии?
11. В чем основной недостаток STA?
12. Чего позволяет добиться агрегирование каналов?
13. Как взаимодействуют алгоритм покрывающего дерева и агрегирование каналов?
14. В чем ограничения агрегирования каналов?
15. В чем отличие между односторонним и двусторонним транком?
16. По каким соображениям выбирает порт транка при передаче кадра?
17. Зачем учитывать принадлежность кадров к одному сеансу при использовании агрегированного канала?
18. Почему VLAN можно назвать доменом широковещательного трафика?
19. Каким образом можно объединить несколько виртуальных локальных сетей?
20. Назовите основные способы образования VLAN.
21. Почему группирование портов плохо работает в сети, построенной на нескольких коммутаторах?
22. Можно ли одновременно использовать группирование портов и стандарт IEEE 802.1Q?
23. Какие механизмы обеспечения показателей QoS поддерживают коммутаторы локальных сетей?
24. На сколько классов рекомендует разделять трафик стандарт IEEE 802.1D-1998?
25. Перечислите ограничения сетей, построенных на основе коммутаторов.
26. В каких случаях целесообразнее выполнить агрегирование каналов, чем выбрать более скоростную версию технологии Ethernet?
27. Каким образом стандарт IEEE 802.1Q решает проблему построения VLAN на нескольких коммутаторах?
28. Должен ли алгоритм покрывающего дерева учитывать наличие в сети VLAN?
29. Что делать, если коммутаторы сети поддерживают меньшее количество очередей, чем существует классов трафика?

Часть IV

Сети TCP/IP

Прежде чем перейти к последним двум частям книги, давайте вспомним, что мы уже изучили в первых трех частях, и обсудим, с чем нам еще предстоит познакомиться. В части I на концептуальном уровне были рассмотрены большинство проблем, которым посвящен этот учебник. Возможно это самая сложная и важная часть книги — ведь от того, насколько хорошо заложен фундамент, зависит прочность основанных на нем знаний. Мы не раз обращались и будем обращаться к материалам части I в дальнейшем.

Части II и III были посвящены конкретным технологиям передачи данных соответственно физического и канального уровней. В них существенно реже использовались абстрактные модели сети в виде графа или «облака», в котором плавают компьютеры. Вместо этого на первый план вышли конкретные протоколы, форматы кадров и реальное оборудование.

Что же ждет читателя в следующей части — части IV? Следуя логике, диктуемой моделью OSI, вслед за частями, в которых были изучены технологии физического и канального уровней, мы рассмотрим в части IV средства сетевого уровня, то есть средства, которые обеспечивают возможность объединения множества сетей в единую сеть. Учитывая, что бесспорным лидером среди протоколов сетевого уровня является протокол IP, мы будем рассматривать вопросы построения объединенных сетей на его примере. При этом мы дадим по возможности широкую картину взаимодействия всех протоколов этого стека.

Заметим, что в предыдущих частях не раз затрагивались, а иногда и достаточно серьезно обсуждались вопросы межсетевого взаимодействия TCP/IP. Так, в главе 2 мы уже рассмотрели, хотя и в самом общем виде, понятие маршрутизации. В главе 4 в разделе «Модель OSI», изучая сетевой уровень, мы познакомились с понятием «составная сеть», которую можно представить как совокупность нескольких сетей (подсетей). Подсети в составной сети, которые могут быть как локальными, так и глобальными, соединяются между собой маршрутизаторами. В пределах каждой подсети все узлы взаимодействуют по единой для них технологии, например Ethernet, Token Ring, FDDI, frame relay, X.25. Однако ни одна из этих технологий не способна построить информационную связь между произвольно выбранными узлами, принадлежащими разным сетям. Именно эту задачу — организацию взаимодействия между любой произвольной парой узлов в «большой» составной сети эффективно решают протоколы стека TCP/IP. В главе 5 было дано описание структуры Интернета — самой известной и масштабной сети, построенной на основе технологии TCP/IP. Читателю настоятельно рекомендуется еще раз внимательно просмотреть этот материал.

Забегая вперед, мы хотим предупредить читателя, что в последней части книги, посвященной технологиям WAN, мы еще не раз вернемся к протоколам TCP/IP. Мы рассмотрим особенности работы протокола IP «поверх» сетей ATM/FR, тесно связанную с IP технологию MPLS, основной протокол систем управления SNMP, а также защищенную версию протокола IP — протокол IPSec.

- Глава 17. Адресация в сетях TCP/IP
- Глава 18. Протокол межсетевого взаимодействия
- Глава 19. Базовые протоколы TCP/IP
- Глава 20. Дополнительные функции маршрутизаторов IP-сетей

ГЛАВА 17 Адресация в сетях TCP/IP

Значительная часть технологии TCP/IP направлена на решение следующих задач адресации:

- *Задача согласованного использования адресов различного типа* включает отображение адресов разных типов, например, преобразование сетевого IP-адреса в локальный, доменного имени — в IP-адрес.
- *Обеспечение уникальности адресов*. В зависимости от типа адреса требуется обеспечивать однозначность адресации в пределах компьютера, подсети, корпоративной сети или Интернета.
- *Конфигурирование сетевых интерфейсов и сетевых приложений*.

Каждая из перечисленных задач имеет достаточно простое решение для сети, число узлов которой не превосходит нескольких десятков. Например, для отображения символьного доменного имени в IP-адрес достаточно поддерживать на каждом хосте таблицу всех символьных имен, используемых в сети, и соответствующих им IP-адресов. Так же просто «вручную» присвоить всем интерфейсам в небольшой сети уникальные адреса. Однако в крупных сетях эти же задачи усложняются настолько, что требуют принципиально других решений.

Ключевым словом, которое характеризует подход к решению этих проблем, принятый в TCP/IP, является **масштабируемость**.

Процедуры, предлагаемые TCP/IP для назначения, отображения и конфигурирования адресов, одинаково хорошо работают в сетях разного масштаба. В этой главе наряду с собственно схемой образования IP-адресов мы познакомимся с наиболее популярными масштабируемыми средствами поддержки адресации в сетях TCP/IP: технологией бесклассовой междоменной маршрутизации, системой доменных имен, протоколом динамического конфигурирования хостов.

Типы адресов стека TCP/IP

Список ключевых слов: MAC-адрес, локальный (аппаратный) адрес, номер сети, номер узла в сети, сетевой адрес, IP-адрес, символьное имя, доменное имя, система доменных имен, DNS-имя.

Итак, для идентификации сетевых интерфейсов используются три типа адресов:

- локальные (аппаратные) адреса;
- сетевые адреса (IP-адреса);
- символьные (доменные) имена.

Локальные адреса

В большинстве технологий LAN (Ethernet, FDDI, Token Ring) для однозначной адресации интерфейсов используются **MAC-адреса**. Существует немало технологий (X.25, ATM, frame relay), в которых применяются другие схемы адресации. Будучи автономными, такие сети используют свою схему адресации исключительно для обеспечения связи собственных узлов. Однако как только некоторая сеть объединяется с другими сетями, функциональность этих адресов расширяется, они становятся необходимым элементом вышележащей объединяющей технологии — в данном случае технологии TCP/IP. Роль, которую играют эти адреса в TCP/IP, не зависит от того, какая именно технология используется в подсети, поэтому они имеют общее название — **локальные (аппаратные) адреса**.

Слово «локальный» в контексте TCP/IP означает «действующий не во всей составной сети, а лишь в пределах подсети». Именно в таком смысле понимаются здесь термины: «локальная технология» (технология, на основе которой построена подсеть), «локальный адрес» (адрес, который используется некоторой локальной технологией для адресации узлов в пределах подсети). Напомним, что в качестве подсети («локальной сети») может выступать сеть, построенная как на основе локальной технологии, например Ethernet, FDDI, так и на основе глобальной технологии, например X.25, Frame Relay. Следовательно, говоря о подсети, мы используем слово «локальная» не как характеристику технологии, на которой построена эта подсеть, а как указание на роль, которую играет эта подсеть в архитектуре составной сети.

Сложности могут возникнуть и при интерпретации определения «аппаратный». В данном случае термин «аппаратный» подчеркивает концептуальное представление разработчиков стека TCP/IP о подсети как о некотором вспомогательном *аппаратном* средстве, единственной функцией которого является перемещение IP-пакета через подсеть до ближайшего шлюза (маршрутизатора). И не важно, что реально нижележащая локальная технология может быть достаточно сложной, все ее сложности игнорируются технологией TCP/IP.

Рассмотрим, например, случай, когда в составную сеть TCP/IP входит сеть IPX/SPX. Последняя сама может быть разделена на подсети, и так же как IP-сеть, она идентифицирует свои узлы аппаратными и сетевыми IPX-адресами. Но технология TCP/IP игнорирует многоуровневое строение сети IPX/SPX и рассматрива-

ет в качестве локальных адресов узлов подсети IPX/SPX адреса сетевого уровня данной технологии (IPX-адреса). Аналогично, если в составную сеть включена сеть X.25, то локальными адресами узлов этой сети для протокола IP будут соответственно адреса X.25.

Сетевые IP-адреса

Чтобы технология TCP/IP могла решать свою задачу объединения сетей, ей необходима собственная глобальная система адресации, *не зависящая от способов адресации узлов в отдельных сетях*. Эта система адресации должна позволять универсальным и однозначным способом идентифицировать любой интерфейс составной сети. Очевидным решением является уникальная нумерация всех сетей составной сети, а затем нумерация всех узлов в пределах каждой из этих сетей. Пара, состоящая из **номера сети** и **номера узла**, отвечает поставленным условиям и может служить в качестве **сетевого адреса**.

В качестве номера узла может выступать либо локальный адрес этого узла (такая схема принята в стеке IPX/SPX), либо некоторое число, никак не связанное с локальной технологией и однозначно идентифицирующее узел в пределах данной подсети. В первом случае сетевой адрес становится зависимым от локальных технологий, что ограничивает его применение. Например, сетевые адреса IPX/SPX рассчитаны на работу в составных сетях, объединяющих сети, в которых используются только MAC-адреса или адреса аналогичного формата. Второй подход более универсален, он характерен для стека TCP/IP[1].

В технологии TCP/IP сетевой адрес называют **IP-адрес**.

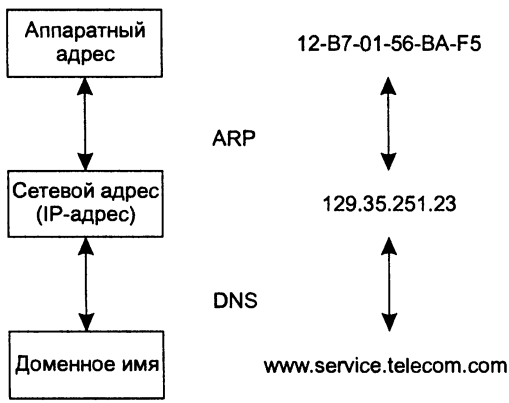


Рис. 17.1. Преобразование адресов

Каждый раз, когда пакет направляется адресату через составную сеть, в его заголовке указывается IP-адрес узла назначения. По номеру сети назначения каждый очередной маршрутизатор находит IP-адрес следующего маршрутизатора.

[1] Заметим, что использование локального адреса в качестве номера узла имеет ряд преимуществ. Как будет показано далее, именно такая схема принята в протоколе IPv6.

Перед тем как отправить пакет в следующую сеть, маршрутизатор должен определить на основании найденного IP-адреса следующего маршрутизатора его локальный адрес. Для этой цели протокол IP, как показано на рис. 17.1, обращается к протоколу разрешения адресов (ARP).

ВНИМАНИЕ

Рассмотрим IP-сеть. Маршрутизатор по определению входит сразу в несколько сетей, следовательно, каждый его интерфейс имеет собственный IP-адрес. Конечный узел также может входить в несколько IP-сетей. В этом случае компьютер должен иметь несколько IP-адресов, по числу сетевых связей. Таким образом, IP-адрес идентифицирует не отдельный компьютер или маршрутизатор, а одно сетевое соединение.

Доменные имена

Для идентификации компьютеров аппаратное и программное обеспечение в сетях TCP/IP полагается на IP-адреса. Например, команда ftp://192.45.66.17 будет устанавливать сеанс связи с нужным ftp-сервером, а команда http://203.23.106.33 откроет начальную страницу на корпоративном веб-сервере. Однако пользователи обычно предпочитают работать с более удобными **символьными именами** компьютеров.

Символьные идентификаторы сетевых интерфейсов в пределах составной сети строятся по иерархическому признаку. Составляющие полного символьного (или доменного) имени в IP-сетях разделяются точкой и перечисляются в следующем порядке: сначала простое имя хоста, затем имя группы хостов (например, имя организации), затем имя более крупной группы (домена) и так до имени домена самого высокого уровня (например, домена, объединяющего организации по географическому принципу: RU — Россия, UK — Великобритания, US — США). Примером доменного имени может служить имя base2.sales.zil.ru.

Между **доменным именем** и IP-адресом узла нет никакой функциональной зависимости, поэтому единственный способ установления соответствия — это таблица. В сетях TCP/IP используется специальная **система доменных имен** (Domain Name System, DNS), которая устанавливает это соответствие на основании создаваемых администраторами сети таблиц соответствия. Поэтому доменные имена называют также **DNS-именами**.

В общем случае сетевой интерфейс может иметь несколько локальных адресов, сетевых адресов, доменных имен.

Формат IP-адреса

Список ключевых слов: номер сети, номер узла в сети, фиксированная граница поля адреса, маска, классы адресов A, B, C, D и E, групповой адрес, индивидуальный адрес, неопределенный адрес, ограниченный широковещательный адрес, широковещательный адрес, адрес обратной петли, групповой адрес.

В заголовке IP-пакета для хранения IP-адресов отправителя и получателя отводятся два поля, каждое имеет фиксированную длину 4 байт (32 бит). IP-адрес состоит из двух логических частей — **номера сети** и **номера узла в сети**.

Наиболее распространенной формой представления IP-адреса является запись в виде четырех чисел, представляющих значения каждого байта в десятичной форме и разделенных точками, например:

128.10.2.30

Этот же адрес может быть представлен в двоичном формате:

10000000 00001010 00000010 00011110

и в шестнадцатеричном формате:

80.0A.02.1D

Заметим, что запись адреса не предусматривает *специального разграничительного знака* между номером сети и номером узла. Вместе с тем при передаче пакета по сети часто возникает необходимость разделить адрес на эти две части. Например, маршрутизация, как правило, осуществляется на основании номера сети, поэтому каждый маршрутизатор, получая пакет, должен прочитать из соответствующего поля заголовка адрес назначения и выделить из него номер сети. Каким образом маршрутизаторы определяют, какая часть из 32 бит, отведенных под IP-адрес, относится к номеру сети, а какая — к номеру узла?

Можно предложить несколько вариантов решения этой проблемы.

- Простейший из них состоит в использовании **фиксированной границы**. При этом все 32-битовое поле адреса заранее делится на две части не обязательно равной, но фиксированной длины, в одной из которых всегда будет размещаться номер сети, а в другой — номер узла. Решение очень простое, но хорошее ли? Поскольку поле, которое отводится для хранения номера узла, имеет фиксированную длину, все сети будут иметь одинаковое максимальное число узлов. Если, например, под номер сети отвести один первый байт, то все адресное пространство распадется на сравнительно небольшое (2^8) число сетей огромного размера (2^{24} узлов). Если границу передвинуть дальше вправо, то сетей станет больше, но все равно все они будут одинакового размера. Очевидно, что такой жесткий подход не позволяет дифференцированно удовлетворять потребности отдельных предприятий и организаций. Именно поэтому он не нашел применения, хотя и использовался на начальном этапе существования технологии TCP/IP (RFC 760).
- Второй подход (RFC 950, RFC 1518) основан на использовании *маски*, которая позволяет максимально гибко устанавливать границу между номером сети и номером узла. При таком подходе адресное пространство можно использовать для создания множества сетей разного размера.

  Маска — это число, применяемое в паре с IP-адресом, причем двоичная запись маски содержит непрерывную последовательность единиц в тех разрядах, которые должны в IP-адресе интерпретироваться как номер сети. Граница между последовательностями единиц и нулей в маске соответствует границе между номером сети и номером узла в IP-адресе.

- И, наконец, наиболее распространенный до недавнего времени способ решения данной проблемы заключается в использовании **классов адресов** (RFC 791). Этот способ представляет собой компромисс по отношению к двум предыдущим: размеры сетей хотя и не могут быть произвольными, как при использовании масок, но и не должны быть одинаковыми, как при установлении фиксированных границ. Вводится пять классов адресов: A, B, C, D, E. Три из них — A, B и C — используются для адресации сетей, а два — D и E — имеют специальное назначение. Для каждого класса сетевых адресов определено собственное положение границы между номером сети и номером узла.

Классы IP-адресов

Признаком, на основании которого IP-адрес относится к тому или иному классу, являются значения нескольких первых битов адреса. Таблица 17.1 иллюстрирует структуру IP-адресов разных классов.

Таблица 17.1. Классы IP-адресов

Класс	Первые биты	Наименьший номер сети	Наибольший номер сети	Максимальное число узлов в сети
A	0	1.0.0.0 (0 — не используется)	126.0.0.0 (127 — зарезервирован)	2^{24}, поле 3 байта
B	10	128.0.0.0	191.255.0.0	2^{16}, поле 2 байта
C	110	192.0.0.0	223.255.255.0	2^{8}, поле 1 байт
D	1110	224.0.0.0	239.255.255.255	Групповые адреса
E	11110	240.0.0.0	247.255.255.255	Зарезервировано

- К **классу A** относится адрес, в котором старший бит имеет значение 0. В адресах класса A под идентификатор сети отводится 1 байт, а остальные 3 байта интерпретируются как номер узла в сети. Сети, все IP-адреса которых имеют значение первого байта в диапазоне от 1 (00000001) до 126 (01111110), называются сетями класса A. Значение 0 (00000000) первого байта не используется, а значение 127 (01111111) зарезервировано для специальных целей, о чем будет рассказано далее. Сетей класса A сравнительно немного, зато количество узлов в них может достигать 2^{24}, то есть 16 777 216 узлов.
- К **классу B** относятся все адреса, старшие два бита которых имеют значение 10. В адресах класса B под номер сети и под номер узла отводится по два байта. Сети, значения первых двух байтов адресов которых находятся в диапазоне от 128.0. (10000000 00000000) до 191.255 (10111111 11111111), называются сетями класса B. Ясно, что сетей класса B больше, чем сетей класса A, а размеры их меньше. Максимальное количество узлов в сетях класса B составляет 2^{16} (65 536).

- К **классу C** относятся все адреса, старшие три бита которых имеют значение 110. В адресах класса C под номер сети отводится 3 байта, а под номер узла — 1 байт. Сети, старшие три байта которых находятся в диапазоне от 192.0.0 (11000000 00000000 00000000) до 223.255 (11011111 11111111 11111111), называются сетями класса C. Сети класса C наиболее распространены и имеют наименьшее максимальное число узлов — 2^8 (256).
- Если адрес начинается с последовательности 1110, то он является адресом **класса D** и обозначает особый, **групповой адрес** (multicast address). В то время как адреса классов A, B и C используются для идентификации отдельных сетевых интерфейсов, то есть являются **индивидуальными адресами** (unicast address), групповой адрес идентифицирует группу сетевых интерфейсов, которые в общем случае могут принадлежать разным сетям. Интерфейс, входящий в группу, получает наряду с обычным индивидуальным IP-адресом еще один групповой адрес. Если при отправке пакета в качестве адреса назначения указан адрес класса D, то такой пакет должен быть доставлен всем узлам, которые входят в группу.
- Если адрес начинается с последовательности 11110, то это значит, что данный адрес относится к **классу E**. Адреса этого класса зарезервированы для будущих применений.

Чтобы получить из IP-адреса номер сети и номер узла, требуется не только разделить адрес на две соответствующие части, но и дополнить каждую из них нулями до полных 4 байт. Возьмем, например, адрес класса B 129.64.134.5. Первые два байта идентифицируют сеть, а последующие два — узел. Таким образом, номером сети является адрес 129.64.0.0, а номером узла — адрес 0.0.134.5.

Особые IP-адреса

В TCP/IP существуют ограничения при назначении IP-адресов, а именно номера сетей и номера узлов *не могут состоять из одних двоичных нулей или единиц.* Отсюда следует, что максимальное количество узлов, приведенное в табл. 17.1 для сетей каждого класса, должно быть уменьшено на 2. Например, в адресах класса C под номер узла отводится 8 бит, которые позволяют задавать 256 номеров: от 0 до 255. Однако в действительности максимальное число узлов в сети класса C не может превышать 254, так как адреса 0 и 255 запрещены для адресации сетевых интерфейсов. Из этих же соображений следует, что конечный узел не может иметь адрес типа 98.255.255.255, поскольку номер узла в этом адресе класса A состоит из одних двоичных единиц.

Итак, некоторые IP-адреса интерпретируются особым образом.

- Если IP-адрес состоит только из двоичных нулей, то он называется **неопределенным адресом** и обозначает адрес того узла, который сгенерировал этот пакет. Адрес такого вида в особых случаях помещается в заголовок IP-пакета в поле адреса отправителя.
- Если в поле номера сети стоят только нули, то по умолчанию считается, что узел назначения принадлежит той же самой сети, что и узел, который отпра-

вил пакет. Такой адрес также может быть использован только в качестве адреса отправителя.

- Если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета. Такой адрес называется **ограниченным широковещательным** (limited broadcast). Ограниченность в данном случае означает, что пакет не выйдет за границы данной сети ни при каких условиях.
- Если в поле адреса назначения в разрядах, соответствующих номеру узла, стоят только единицы, то пакет, имеющий такой адрес, рассылается *всем* узлам сети, номер которой указан в адресе назначения. Например, пакет с адресом 192.190.21.255 будет направлен всем узлам сети 192.190.21.0. Такой тип адреса называется **широковещательным** (broadcast).

ВНИМАНИЕ

В протоколе IP нет понятия широковещания в том смысле, в котором оно используется в протоколах канального уровня локальных сетей, когда данные должны быть доставлены абсолютно всем узлам сети. Как ограниченный, так и обычный варианты широковещательной рассылки имеют пределы распространения в составной сети — они ограничены либо сетью, которой принадлежит источник пакета, либо сетью, номер которой указан в адресе назначения. Поэтому деление сети с помощью маршрутизаторов на части локализует широковещательный шторм пределами одной из подсетей просто потому, что нет способа адресовать пакет одновременно всем узлам всех сетей составной сети.

Особый смысл имеет IP-адрес, первый октет которого равен 127. Этот адрес является *внутренним адресом стека протоколов* компьютера (или маршрутизатора). Он используется для тестирования программ, а также для организации работы клиентской и серверной частей приложения, установленных на одном компьютере. Обе программные части данного приложения спроектированы в расчете на то, что они будут обмениваться сообщениями по сети. Но какой же IP-адрес они должны использовать для этого? Адрес сетевого интерфейса компьютера, на котором они установлены? Но это приводит к избыточным передачам пакетов в сеть. Экономичным решением является применение внутреннего адреса 127.0.0.0. В IP-сети запрещается присваивать сетевым интерфейсам IP-адреса, начинающиеся со 127. Когда программа посылает данные по IP-адресу 127.x.x.x, то данные не передаются в сеть, а возвращаются модулям верхнего уровня того же компьютера как только что принятые. Маршрут перемещения данных образует «петлю», поэтому этот адрес называется **адресом обратной петли** (loopback).

Групповые адреса (multicast), относящиеся к классу D, предназначены для экономичного распространения в Интернете или большой корпоративной сети аудио- или видеопрограмм, адресованных сразу большой аудитории слушателей или зрителей. Если групповой адрес помещен в поле адреса назначения IP-пакета, то данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу с номером, указанным в поле адреса. Один и тот же узел может входить в несколько групп. В общем случае члены группы могут распределяться по различным сетям, находящимся друг от друга на произвольно большом расстоянии.

Групповой адрес не делится на номера сети и узла и обрабатывается маршрутизатором особым образом. Основное назначение групповых адресов — распространение информации по схеме «один ко многим». От того, найдут групповые адреса широкое применение (сейчас их используют в основном небольшие экспериментальные «островки» в Интернете), зависит, сможет ли Интернет создать серьезную конкуренцию радио и телевидению.

Использование масок при IP-адресации

Снабжая каждый IP-адрес маской, можно отказаться от понятий классов адресов и сделать более гибкой систему адресации.

Пусть, например, для IP-адреса 129.64.134.5 указана маска 255.255.128.0, то есть в двоичном виде IP-адрес 129.64.134.5 — это:

10000001.01000000.10000110.00000101,

а маска 255.255.128.0 — это:

11111111.11111111.10000000.00000000.

Если игнорировать маску и интерпретировать адрес 129.64.134.5 на основе классов, то номером сети является 129.64.0.0, а номером узла — 0.0.134.5 (поскольку адрес относится к классу B).

Если же использовать маску, то 17 последовательных двоичных единиц в маске 255.255.128.0, «наложенные» на IP-адрес 129.64.134.5, делят его на две части:

- номер сети: 10000001.01000000.1;
- номер узла: 0000110.00000101.

В десятичной форме записи номера сети и узла, дополненные нулями до 32 бит, выглядят, соответственно, как 129.64.128.0 и 0.0.6.5.

Наложение маски можно интерпретировать как выполнение логической операции «И» (AND). Так, в предыдущем примере номер сети из адреса 129.64.134.5 является результатом выполнения логической операции AND с маской 255.255.128.0:

10000001 01000000 10000110 00000101
AND
11111111.11111111.10000000.00000000

Для стандартных классов сетей маски имеют следующие значения:

- класс A — 11111111. 00000000. 00000000. 00000000 (255.0.0.0);
- класс B — 11111111. 11111111. 00000000. 00000000 (255.255.0.0);
- класс C — 11111111. 11111111. 11111111. 00000000 (255.255.255.0).

ПРИМЕЧАНИЕ

Для записи масок используются и другие форматы. Например, удобно интерпретировать значение маски, записанной в шестнадцатеричном коде: FF.FF.00.00 — маска для адресов класса B. Еще более часто встречается и такое обозначение 185.23.44.206/16 — эта запись говорит о том, что маска для этого адреса содержит 16 единиц или что в указанном IP-адресе под номер сети отведено 16 двоичных разрядов.

Механизм масок широко распространен в IP-маршрутизации, причем маски могут использоваться для самых разных целей. С их помощью администратор может разбивать одну, выделенную ему поставщиком услуг сеть определенного класса, на несколько других, не требуя дополнительных номеров сетей — эта операция называется *разделением на подсети* (subnetting). На основе этого же механизма поставщики услуг могут объединять адресные пространства нескольких сетей путем введения так называемых «префиксов» для уменьшения объема таблиц маршрутизации и повышения за счет этого производительности маршрутизаторов — такая операция называется *объединением подсетей* (supernetting). Подробнее об этом мы поговорим при изучении технологии бесклассовой междоменной маршрутизации.

Порядок назначения IP-адресов

Список ключевых слов: частный адрес, организация ICANN, префикс адреса, RFC 2050, уникальность номеров сетей и узлов, централизованная процедура, дефицит адресов, IPv6, NAT, технология бесклассовой междоменной маршрутизации, пул адресов.

По определению схема IP-адресации должна обеспечивать уникальность нумерации сетей, а также уникальность нумерации узлов в пределах каждой из сетей. Следовательно, процедуры назначения номеров как сетям, так и узлам сетей должны быть *централизованными*. Рекомендуемый порядок назначения IP-адресов дается в RFC 2050.

Назначение адресов автономной сети

Когда дело касается сети, являющейся частью Интернета, уникальность нумерации может быть обеспечена только усилиями специально созданных для этого центральных органов. В небольшой же автономной IP-сети условие уникальности номеров сетей и узлов может быть выполнено силами сетевого администратора.

В этом случае в распоряжении администратора имеются все адресное пространство, так как совпадение IP-адресов в не связанных между собой сетях не вызовет никаких отрицательных последствий. Администратор может выбирать адреса произвольным образом, соблюдая лишь синтаксические правила и учитывая ограничения на особые адреса. (Напомним, что номер узла в технологии TCP/IP назначается независимо от его локального адреса.)

Однако при таком подходе исключена возможность в будущем подсоединить данную сеть к Интернету. Действительно, произвольно выбранные адреса данной сети могут совпасть с централизовано назначенными адресами Интернета. Для того чтобы избежать коллизий, связанных с такого рода совпадениями, в стандартах Интернета определено несколько так называемых **частных адресов**, рекомендуемых для автономного использования:

- в классе A — сеть 10.0.0.0;
- в классе B — диапазон из 16 номеров сетей 172.16.0.0–172.31.0.0;
- в классе C — диапазон из 255 сетей — 192.168.0.0–192.168.255.0.

Эти адреса, исключенные из множества централизованно распределяемых, составляют огромное адресное пространство, достаточное для нумерации узлов автономных сетей практически любых размеров. Заметим также, что частные адреса, как и при произвольном выборе адресов, в разных автономных сетях могут совпадать. В то же время использование частных адресов для адресации автономных сетей делает возможным корректное подключение их к Интернету. Применяемые при этом специальные технологии подключения[1] исключают коллизии адресов.

Централизованное распределение адресов

В больших сетях, подобных Интернету, уникальность сетевых адресов гарантируется централизованной, иерархически организованной системой их распределения. Номер сети может быть назначен только по рекомендации специального подразделения Интернета. Главным органом регистрации глобальных адресов в Интернете с 1998 года является неправительственная некоммерческая организация **ICANN** (Internet Corporation for Assigned Names and Numbers). Эта организация координирует работу региональных отделов, деятельность которых охватывает большие географические площади: ARIN — Америка, RIPE (Европа), APNIC (Азия и Тихоокеанский регион). Региональные отделы выделяют блоки адресов сетей крупным поставщикам услуг, а те, в свою очередь, распределяют их между своими клиентами, среди которых могут быть и более мелкие поставщики.

Проблемой централизованного распределения адресов является их дефицит. Уже сравнительно давно очень трудно получить адрес класса B и практически невозможно стать обладателем адреса класса A. При этом надо отметить, что дефицит обусловлен не только ростом сетей, но и тем, что имеющееся адресное пространство используется нерационально. Очень часто владельцы сетей класса C расходуют лишь небольшую часть из имеющихся у них 254 адресов. Рассмотрим пример, когда две сети необходимо соединить глобальной связью. В таких случаях в качестве линии связи используют два маршрутизатора, соединенных по двухточечной схеме (рис. 17.2). Для вырожденной сети, образованной линией связи, связывающей порты двух смежных маршрутизаторов, приходится выделять отдельный номер сети, хотя в этой сети имеется всего два узла.

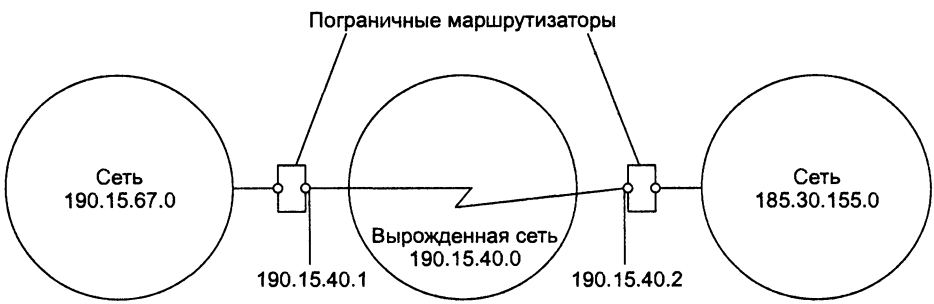


Рис. 17.2. Нерациональное использование пространства IP-адресов

[1] Например, такой технологией является NAT, которая будет рассмотрена в главе 20.

Для смягчения проблемы дефицита адресов разработчики стека TCP/IP предлагают разные подходы. Принципиальным решением является переход на новую версию протокола IP — протокол IPv6, в котором резко расширяется адресное пространство. Однако и текущая версия протокола IP (IPv4) поддерживает технологии, направленные на более экономное расходование IP-адресов, такие например, как NAT и CIDR.

Адресация и технология CIDR

Технология бесклассовой междоменной маршрутизации (Classless Inter-Domain Routing, CIDR), которая описана в документах RFC 1517, RFC 1518, RFC 1519, RFC 1520 и о которой впервые было официально объявлено в 1993 году, позволяет центрам распределения адресов избежать выдачи абонентам излишних адресов.

Деление IP-адреса на номер сети и номер узла в технологии CIDR происходит не на основе нескольких старших битов, определяющих класс сети (A, B или C), а на основе маски переменной длины, назначаемой поставщиком услуг. Непременным условием применимости CIDR является наличие у организации, распоряжающейся адресами, непрерывных диапазонов адресов. Такие адреса имеют одинаковый **префикс**, то есть одинаковые цифры в нескольких старших разрядах. Пусть в распоряжении некоторого поставщика услуг имеется непрерывное пространство IP-адресов в количестве 2^n (рис. 17.3). Отсюда следует, что префикс имеет длину (32 – n) разрядов. Оставшиеся n разрядов играют роль счетчика последовательных номеров.

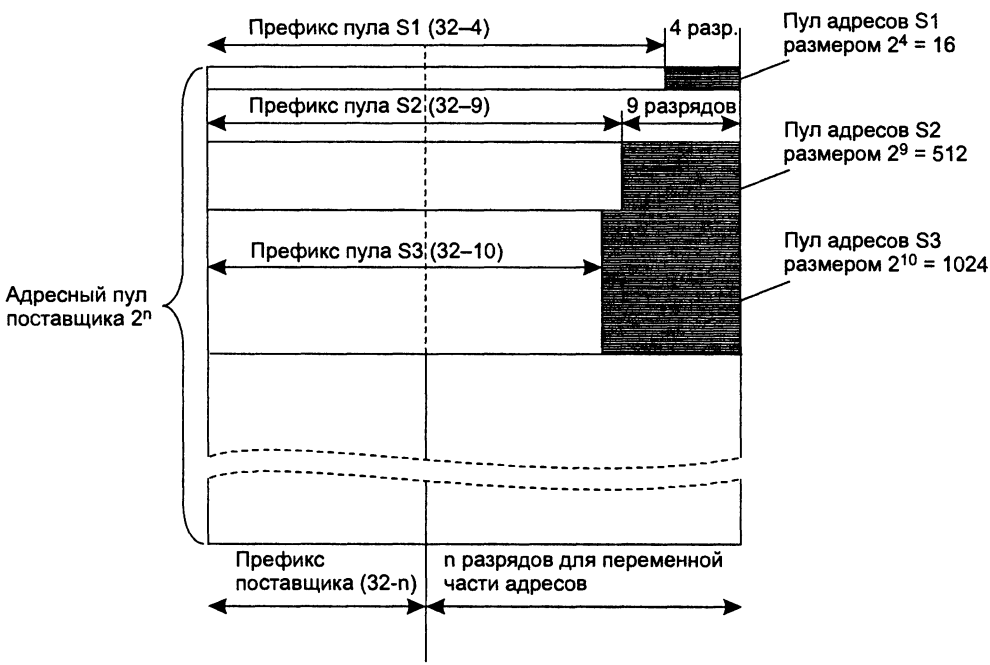


Рис. 17.3. Распределение адресов на основе технологии CIDR

Когда потребитель обращается к поставщику услуг с просьбой о выделении ему некоторого числа адресов, то в имеющемся пуле адресов «вырезается» непрерывная область S1, S2 или S3, в зависимости от требуемого количества адресов. При этом должны быть выполнены следующие условия:

- количество адресов в выделяемой области должно быть равно степени двойки,
- начальная граница выделяемого пула адресов должна быть кратна требуемому количеству узлов.

Очевидно, что префикс каждой из показанных на рисунке областей имеет собственную длину — чем меньше количество адресов в данной области, тем длиннее ее префикс.

Пример

Пусть поставщик услуг Интернета располагает пулом адресов в диапазоне 193.20.0.0–193.23.255.255 (1100 0001.0001 0100.0000 0000.0000 0000–1100 0001.0001 0111.1111 1111.1111 1111), то есть количество адресов равно 2^{18}. Соответственно префикс поставщика услуг имеет длину 14 разрядов — 1100 0001.0001 01, или в другом виде — 193.20/14.

Если абоненту этого поставщика услуг требуется совсем немного адресов, например 13, то поставщик мог бы предложить ему различные варианты: сеть 193.20.30.0/28, сеть 193.20.30.16/28 или сеть 193.21.204.48/28. Во всех случаях в распоряжении абонента для нумерации узлов имеются 4 младших бита. Таким образом, наименьшее число, удовлетворяющее потребностям абонента (13), которое можно представить степенью двойки (2^4), является 16. Префикс для каждого из выделяемых пулов во всех этих случаях играет роль номера сети, он имеет длину 32 – 4 = 28 разрядов.

Рассмотрим другой вариант, когда к поставщику услуг обратился крупный заказчик, сам, возможно, собирающийся оказывать услуги по доступу в Интернет. Ему требуется блок адресов в 4000 узлов. На нумерацию такого количества узлов пойдет 12 двоичных разрядов, следовательно, размер выделенного пула адресов будет несколько больше требуемого — 4096. Граница, с которой должен начинаться выделяемый участок, должна быть кратна размеру участка, то есть это могут быть любые адреса из следующих: 193.20.0.0, 193.20.16.0, 193.20.32.0, 193.20.48.0, и другие числа оканчивающиеся на 12 нулей. Пусть поставщик услуг предложил потребителю диапазон адресов 193.20.16.0–193.20.31.255. Для этого диапазона агрегированный номер сети (префикс) имеет длину 20 двоичных разрядов и равен 193.20.16.0/20.

Благодаря CIDR поставщик услуг получает возможность «нарезать» блоки из выделенного ему адресного пространства в соответствии с действительными требованиями каждого клиента.

Мы еще вернемся к технологии CIDR в главе 18, чтобы обсудить, каким образом эта технология помогает не только экономно расходовать адреса, но и более эффективно осуществлять маршрутизацию.

Отображение IP-адресов на локальные адреса

Список ключевых слов: протокол разрешения адресов, ARP-запрос, статическая запись, динамическая запись, ARP-кэш, ARP-сервер, протокол Proxy-ARP.

Одной из главных задач, которая ставилась при создании протокола IP, являлось обеспечение совместной согласованной работы в сети, состоящей из подсетей, в общем случае использующих разные сетевые технологии. Взаимодействие технологии TCP/IP с локальными технологиями подсетей происходит многократно при перемещении IP-пакета по составной сети. На каждом маршрутизаторе протокол IP определяет, какому следующему маршрутизатору в этой сети надо направить пакет. В результате решения этой задачи протоколу IP становится известен *IP-адрес* интерфейса следующего маршрутизатора (или конечного узла, если эта сеть является сетью назначения). Чтобы локальная технология сети смогла доставить пакет на следующий маршрутизатор, необходимо:

- упаковать пакет в кадр соответствующего для данной сети формата (например, Ethernet);
- снабдить данный кадр *локальным адресом* следующего маршрутизатора.

Решением этих задач, как уже отмечалось[1], занимается уровень сетевых интерфейсов стека TCP/IP.

Протокол разрешения адресов

Как уже было сказано, никакой зависимости между локальным адресом и его IP-адресом не существует, следовательно, единственный способ установления соответствия — ведение таблиц. В результате конфигурирования сети каждый интерфейс знает свои IP-адрес и локальный адрес, что можно рассматривать как таблицу, состоящую из одной строки. Проблема состоит в том, как организовать обмен имеющейся информацией между узлами сети.

Для определения локального адреса по IP-адресу используется **протокол разрешения адресов** (Address Resolution Protocol, ARP). Протокол разрешения адресов реализуется различным образом в зависимости от того, работает ли в данной сети протокол локальной сети (Ethernet, Token Ring, FDDI) с возможностью широковещания или же какой-либо из протоколов глобальной сети (X.25, Frame Relay), которые, как правило, не поддерживают широковещательный доступ.

Рассмотрим работу протокола ARP в локальных сетях с *широковещанием*.

На рис. 17.4 показан фрагмент IP-сети, включающий две сети — Ethernet1 (из трех конечных узлов A, B и C) и Ethernet2 (из двух конечных узлов D и E). Сети подключены соответственно к интерфейсам 1 и 2 маршрутизатора. Каждый сетевой интерфейс имеет IP-адрес и MAC-адрес. Пусть в какой-то момент IP-модуль узла C направляет пакет узлу D. Протокол IP узла C определил IP-адрес интерфейса следующего маршрутизатора — это IP1. Теперь, прежде чем упаковать пакет в кадр Ethernet и направить его маршрутизатору, необходимо определить соответствующий MAC-адрес. Для решения этой задачи протокол IP обращается к протоколу ARP. Протокол ARP поддерживает на каждом интерфейсе сетевого адаптера или маршрутизатора отдельную ARP-таблицу, в которой в ходе функционирования сети накапливается информация о соответствии между IP-адресами

[1] См. раздел «Стек TCP/IP» в главе 4.

и MAC-адресами других интерфейсов данной сети. Первоначально, при включении компьютера или маршрутизатора в сеть все его ARP-таблицы пусты.

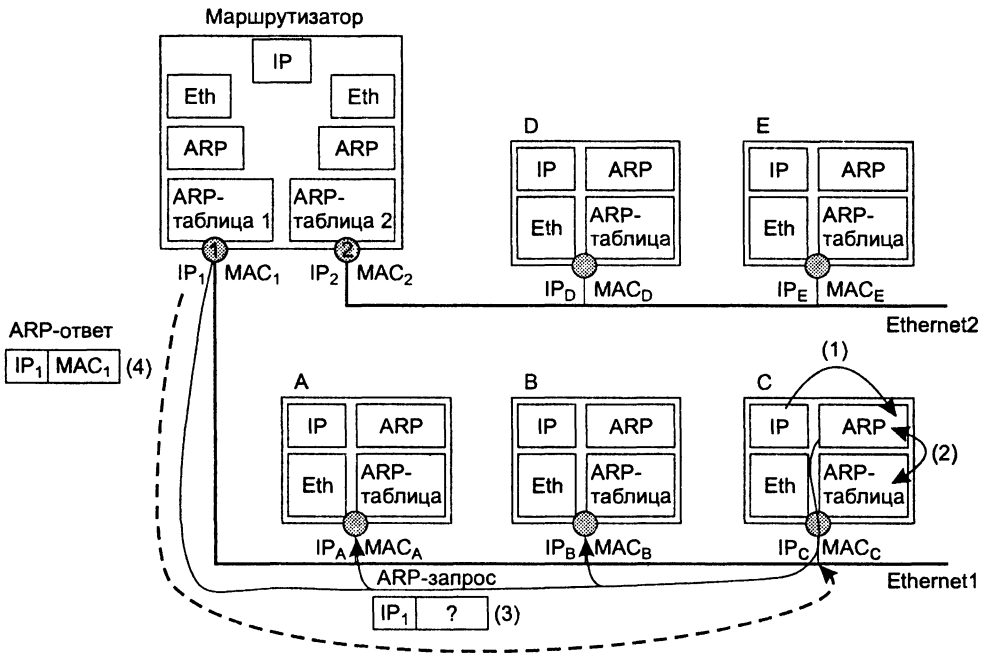


Рис. 17.4. Схема работы протокола ARP

1. На первом шаге происходит передача от протокола IP протоколу ARP примерно такого сообщения: «Какой MAC-адрес имеет интерфейс с адресом IP_1?»
2. Работа протокола ARP начинается с просмотра ARP-таблицы соответствующего интерфейса. Предположим, что среди содержащихся в ней записей отсутствует запрашиваемый IP-адрес.
3. В этом случае исходящий IP-пакет, для которого оказалось невозможным определить локальный адрес из ARP-таблицы, запоминается в буфере, а протокол ARP формирует **ARP-запрос**, вкладывает его в кадр протокола Ethernet и широковещательно рассылает.
4. Все интерфейсы сети Ethernet1 получают ARP-запрос и направляют его «своему» протоколу ARP. ARP сравнивает указанный в запросе адрес IP_1 с IP-адресом интерфейса, на который поступил этот запрос. Протокол ARP, который констатировал совпадение (в данном случае это ARP маршрутизатора 1), формирует ARP-ответ.

В ARP-ответе маршрутизатор указывает локальный адрес MAC_1 своего интерфейса и отправляет его запрашивающему узлу (в данном примере узлу C), используя его локальный адрес. Широковещательный ответ в этом случае не требуется, так как формат ARP-запроса предусматривает поля локального и сетевого адресов отправителя. Заметим, что зона распространения ARP-запросов

ограничивается сетью Ethernet1, так как на пути широковещательных кадров барьером стоит маршрутизатор.

На рис. 17.5 показан кадр Ethernet с вложенным в него ARP-сообщением. ARP-запросы и ARP-ответы имеют один и тот же формат. В табл. 17.2 в качестве примера приведены значения полей реального ARP-запроса, переданного по сети Ethernet[1].

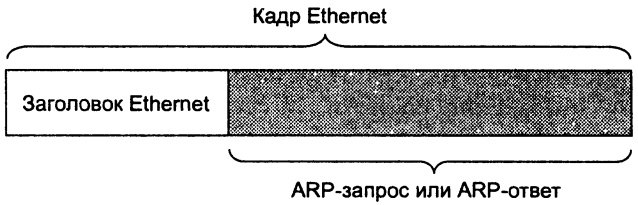


Рис. 17.5. Инкапсуляция ARP-сообщений в кадр Ethernet

Таблица 17.2. Пример ARP-запроса

Поле	Значение
Тип сети	1 (0x1)
Тип протокола	2048 (0x800)
Длина локального адреса	6 (0x6)
Длина сетевого адреса	4 (0x4)
Операция	1 (0x1)
Локальный адрес отправителя	008048EB7E60
Сетевой адрес отправителя	194.85.135.75
Локальный (искомый) адрес получателя	000000000000
Сетевой адрес получателя	194.85.135.65

В поле типа сети для сетей Ethernet указывается значение 1. Поле типа протокола позволяет использовать протокол ARP не только с протоколом IP, но и с другими сетевыми протоколами. Для IP значение этого поля равно 0x0800. Длина локального адреса для протокола Ethernet равна 6 байт, а длина IP-адреса — 4 байт. В поле операции для ARP-запросов указывается значение 1, для ARP-ответов — значение 2.

Из этого запроса видно, что в сети Ethernet узел с IP-адресом 194.85.135.75 пытается определить, какой MAC-адрес имеет другой узел той же сети, сетевой адрес которого 194.85.135.65. *Поле искомого локального адреса заполнено нулями.*

[1] Символы 0x означают, что за ними следует число, записанное в шестнадцатеричном формате.

Ответ присылает узел, опознавший свой IP-адрес. Если в сети нет машины с искомым IP-адресом, то ARP-ответа не будет. Протокол IP уничтожает IP-пакеты, направляемые по этому адресу. В табл. 17.3 показаны значения полей ARP-ответа, который мог бы поступить на приведенный в табл. 17.2 ARP-запрос.

Таблица 17.3. Пример ARP-ответа

Поле	Значение
Тип сети	1 (0x1)
Тип протокола	2048 (0x800)
Длина локального адреса	6 (0x6)
Длина сетевого адреса	4 (0x4)
Операция	1 (0x1)
Локальный адрес отправителя	00E0F77F1920
Сетевой адрес отправителя	194.85.135.65
Локальный (искомый) адрес получателя	008048EB7E60
Сетевой адрес получателя	194.85.135.75

В результате обмена ARP-сообщениями модуль IP, пославший запрос с интерфейса, имеющего адрес 194.85.135.75, определил, что IP-адресу 194.85.135.65 соответствует MAC-адрес 00E0F77F1920. Этот адрес будет затем помещен в заголовок кадра Ethernet, ожидавшего отправления IP-пакета.

Чтобы уменьшить число ARP-обращений в сети, найденное соответствие между IP-адресом и MAC-адресом сохраняется в ARP-таблице соответствующего интерфейса, в данном случае — это запись:

194.85.135.65 — 00E0F77F1920

Данная запись в ARP-таблице появляется автоматически, спустя несколько миллисекунд после того, как модуль ARP проанализирует ARP-ответ. Теперь, если вдруг вновь возникнет необходимость послать пакет по адресу 194.85.135.65, то протокол IP, прежде чем посылать широковещательный запрос, проверит, нет ли уже такого адреса в ARP-таблице.

ARP-таблица пополняется *не только за счет поступающих на данный интерфейс ARP-ответов*, но и в результате извлечения полезной информации из широковещательных ARP-запросов. Действительно, в каждом запросе, как это видно из табл. 17.2 и 17.3, содержатся IP-адрес и MAC-адрес отправителя. Все интерфейсы, получившие этот запрос, могут поместить информацию о соответствии локального и сетевого адресов отправителя в собственную ARP-таблицу. В частности, все узлы, получившие ARP-запрос (см. табл. 17.2), могут пополнить свою ARP-таблицу записью:

194.85.135.75 — 008048EB7E60

Таким образом, вид ARP-таблицы, в которую в ходе работы сети были добавлены две упомянутые нами записи, иллюстрирует табл. 17.4.

Таблица 17.4. Пример ARP-таблицы

IP-адрес	MAC-адрес	Тип записи
194.85.135.65	00E0F77F1920	Динамический
194.85.135.75	008048EB7E60	Динамический
194.85.60.21	008048EB7567	Статический

В ARP-таблицах существует два типа записей: динамические и статические. **Статические записи** создаются вручную с помощью утилиты arp и не имеют срока устаревания, точнее, они существуют до тех пор, пока компьютер или маршрутизатор остается включенным. **Динамические записи** должны периодически обновляться. Если запись не обновлялась в течение определенного времени (порядка нескольких минут), то она исключается из таблицы. Таким образом, в ARP-таблице содержатся записи не обо всех узлах сети, а только о тех, которые активно участвуют в сетевых операциях. Поскольку такой способ хранения информации называют кэшированием, ARP-таблицы иногда называют **ARP-кэшем**.

ПРИМЕЧАНИЕ

Некоторые реализации протоколов IP и ARP не ставят IP-пакеты в очередь на время ожидания ARP-ответов. Вместо этого IP-пакет просто уничтожается, а его восстановление возлагается на модуль TCP или прикладной процесс, работающий через протокол UDP. Такое восстановление выполняется за счет тайм-аутов и повторных передач. Повторная передача сообщения проходит успешно, так как первая попытка уже вызвала заполнение ARP-таблицы.

Совсем другой способ разрешения адресов используется *в глобальных сетях*, в которых не поддерживается широковещательная рассылка. Здесь администратору сети чаще всего приходится вручную формировать и помещать на какой-либо сервер ARP-таблицы, в которых он задает, например, соответствие IP-адресов адресам X.25, имеющих для протокола IP смысл локальных адресов. В то же время сегодня наметилась тенденция автоматизации работы протокола ARP и в глобальных сетях. Для этой цели среди всех маршрутизаторов, подключенных к какой-либо глобальной сети, выделяется специальный маршрутизатор, который ведет ARP-таблицу для всех остальных узлов и маршрутизаторов этой сети.

При таком централизованном подходе для всех узлов и маршрутизаторов вручную нужно задать только IP-адрес и локальный адрес выделенного для этих целей маршрутизатора. При включении каждый узел и маршрутизатор регистрирует свои адреса в выделенном маршрутизаторе. Всякий раз, когда возникает необходимость определения по IP-адресу локального адреса, модуль ARP обращается к выделенному маршрутизатору с запросом и автоматически получает ответ без участия администратора. Работающий таким образом маршрутизатор называют **ARP-сервером**.

В некоторых случаях возникает обратная задача — нахождение IP-адреса по известному локальному адресу. Тогда в действие вступает *реверсивный протокол* ARP (Reverse Address Resolution Protocol, RARP). Этот протокол используется, например, при старте бездисковых станций, не знающих в начальный момент времени своего IP-адреса, но знающих MAC-адрес своего сетевого адаптера.

Протокол Proxy-ARP

Протокол Proxy-ARP — это одна из разновидностей протокола ARP, позволяющая отображать IP-адреса на аппаратные адреса в сетях, поддерживающих широковещание, даже в тех случаях, когда искомый узел находится за пределами данного домена коллизий.

На рис. 17.6 показана сеть, один из конечных узлов которой (компьютер D) работает в режиме удаленного узла. Подробнее об этом режиме вы прочитаете в главе 23 части V, а сейчас достаточно знать, что конечный узел в таком режиме обладает всеми возможностями компьютеров, работающих в основной части сети Ethernet, в частности, он имеет IP-адрес (IP_D), относящийся к той же сети. Для всех конечных узлов сети Ethernet особенности подключения удаленного узла (наличие модемов, коммутируемая связь, протокол PPP) абсолютно прозрачны — они взаимодействуют с ним обычным образом. Чтобы такой режим взаимодействия стал возможным, среди прочего, необходим протокол Proxy-ARP. Поскольку удаленный узел подключен к сети по протоколу PPP, то он, очевидно, *не имеет MAC-адреса.*

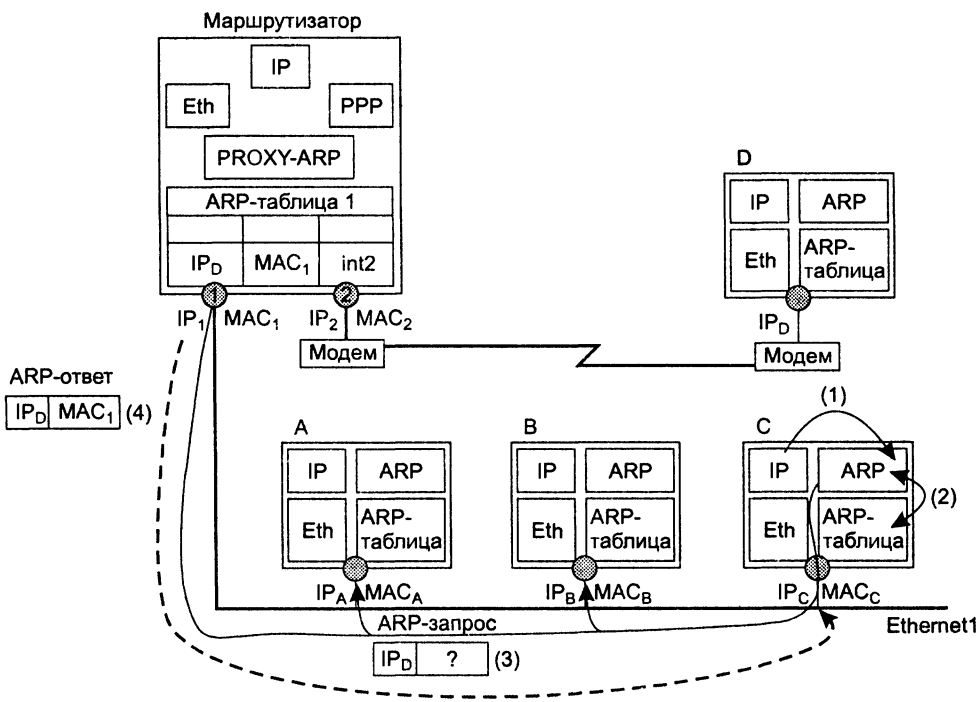


Рис. 17.6. Схема работы протокола Proxy-ARP

Пусть приложение, работающее, например, на компьютере C, решает послать пакет компьютеру D. Ему известен IP-адрес узла назначения (IP_D), однако как мы уже не раз отмечали, для передачи пакета по сети Ethernet его необходимо упаковать в кадр Ethernet и снабдить MAC-адресом. Для определения MAC-адреса IP-протокол узла C обращается к протоколу ARP, который посылает широковещательное сообщение с ARP-запросом. Если бы в этой сети на маршрутизаторе не был установлен протокол Proxy-ARP, на этот запрос не откликнулся бы ни один узел.

Однако протокол Proxy-ARP установлен на маршрутизаторе и работает следующим образом. При подключении к сети удаленного узла D в таблицу ARP-маршрутизатора заносится запись

$$IP_D - MAC_1 - int2,$$

которая означает, что:

- при поступлении ARP-запроса на маршрутизатор относительно адреса IP_D в ARP-ответ будет помещен аппаратный адрес MAC_1, соответствующий аппаратному адресу интерфейса 1 маршрутизатора;
- узел, имеющий адрес IP_D, подключен к интерфейсу 2 маршрутизатора.

В ответ на посланный узлом C широковещательный ARP-запрос откликается маршрутизатор с установленным протоколом Proxy-ARP. Он посылает «ложный» ARP-ответ, в котором на место аппаратного адреса компьютера D помещает собственный адрес MAC_1. Узел C, не подозревая «подвоха», посылает кадр с IP-пакетом по адресу MAC_1. Получив кадр, маршрутизатор с установленным протоколом Proxy-ARP «понимает», что он направлен не ему (в пакете указан чужой IP-адрес) и, следовательно, надо искать адресата в ARP-таблице. Из таблицы видно, что кадр надо направить узлу, подключенному ко второму интерфейсу.

Мы рассмотрели простейшую схему применения протокола Proxy-ARP, которая тем не менее достаточно полно отражает логику его работы.

Система DNS

Список ключевых слов: плоское имя, домен имен, поддомен, краткое доменное имя, относительное доменное имя, полное доменное имя, централизованная служба DNS, нерекурсивная и рекурсивная процедуры разрешения имени, обратная зона.

Плоские символьные имена

В операционных системах, которые первоначально разрабатывались для локальных сетей, таких как Novell NetWare, Microsoft Windows или IBM OS/2, пользователи всегда работали с символьными именами компьютеров. Так как локальные сети состояли из небольшого числа компьютеров, применялись так называемые **плоские имена**, состоящие из последовательности символов, не раз-

деленных на части. Примерами таких имен являются: NW1_1, mail2, MOSCOW_SALES_2. Для установления соответствия между символьными именами и MAC-адресами в этих операционных системах применялся механизм широковещательных запросов, подобный механизму запросов протокола ARP. Так, широковещательный способ разрешения имен реализован в протоколе NetBIOS, на котором были построены многие локальные ОС. Так называемые NetBIOS-имена стали на долгие годы одним из основных типов плоских имен в локальных сетях.

Для стека TCP/IP, рассчитанного в общем случае на работу в больших территориально распределенных сетях, подобный подход оказывается неэффективным.

Иерархические символьные имена

В стеке TCP/IP применяется доменная система имен, которая имеет иерархическую древовидную структуру, допускающую наличие в имени произвольного количества составных частей (рис. 17.7).

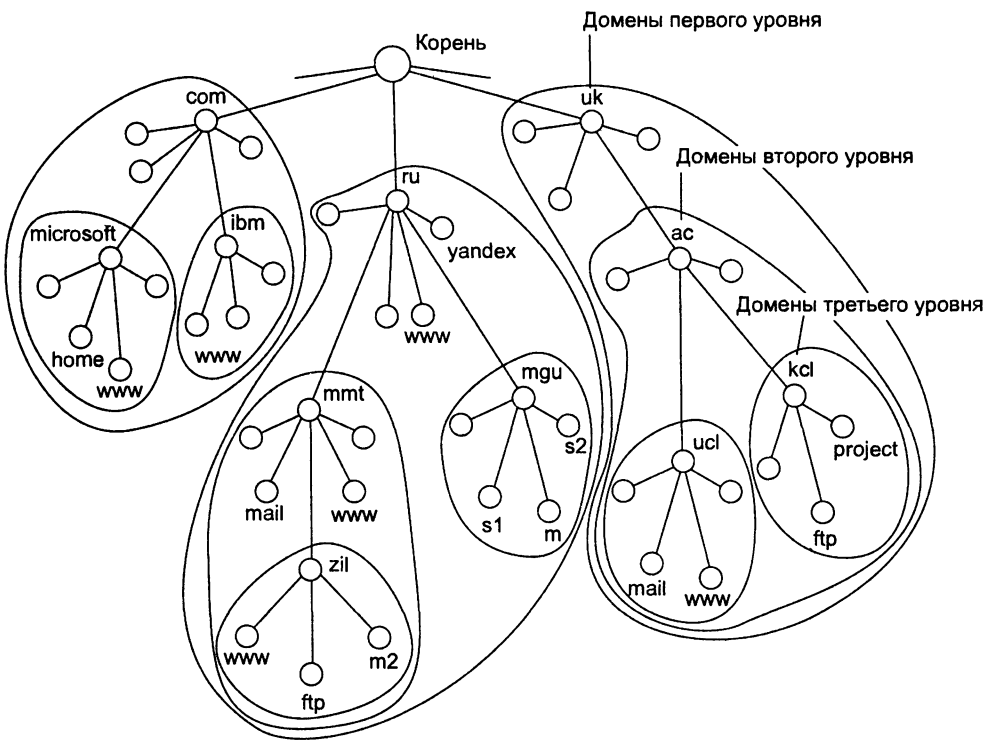


Рис. 17.7. Пространство доменных имен

Иерархия доменных имен аналогична иерархии имен файлов, принятой во многих популярных файловых системах. Дерево имен начинается с корня, обозначаемого здесь точкой (.). Затем следует старшая символьная часть имени, вторая по старшинству символьная часть имени и т. д. Младшая часть имени соответст-

вует конечному узлу сети. В отличие от имен файлов, при записи которых сначала указывается самая старшая составляющая, затем составляющая более низкого уровня и т. д., запись доменного имени начинается с самой младшей составляющей, а заканчивается самой старшей. Составные части доменного имени отделяются друг от друга точкой. Например, в имени partnering.microsoft.com составляющая partnering является именем одного из компьютеров в домене microsoft.com.

Разделение имени на части позволяет *разделить административную ответственность* за назначение уникальных имен между различными людьми или организациями в пределах своего уровня иерархии. Так, для примера, приведенного на рис. 17.7, один человек может нести ответственность за то, чтобы все имена, которые имеют окончание «ru», имели уникальную следующую вниз по иерархии часть. Если этот человек справляется со своими обязанностями, то все имена типа www.ru, mail.mmt.ru или m2,zil.mmt.ru будут отличаться второй по старшинству частью.

Разделение административной обязанности позволяет решить проблему образования уникальных имен без взаимных консультаций между организациями, отвечающими за имена одного уровня иерархии. Очевидно, что должна существовать одна организация, отвечающая за назначение имен верхнего уровня иерархии.

Совокупность имен, у которых несколько старших составных частей совпадают, образуют **домен имен** (domain). Например, имена www1.zil.mmt.ru, ftp.zil.mmt.ru, yandex.ru и s1.mgu.ru входят в домен ru, так как все они имеют одну общую старшую часть — имя ru. Другим примером является домен mgu.ru. Из представленных на рис. 17.7 имен в него входят имена s1.mgu.ru, s2.mgu.ru и m.mgu.ru. Этот домен образуют имена, у которых две старшие части всегда равны mgu.ru. Администратор домена mgu.ru несет ответственность за уникальность имен следующего уровня, входяших в домен, то есть имен s1, s2 и m. Образованные домены s1.mgu.ru, s2.mgu.ru и m.mgu.ru являются **поддоменами** домена mgu.ru, так как имеют общую старшую часть имени. Часто поддомены для краткости называют только младшей частью имени, то есть поддомены s1, s2 и m.

ПРИМЕЧАНИЕ

Термин «домен» очень многозначен, поэтому его нужно трактовать в рамках определенного контекста. Помимо доменов имен стека TCP/IP в компьютерной литературе также часто упоминаются домены Windows NT, домены коллизий и некоторые другие. Общим у всех этих терминов является то, что они описывают некоторое множество компьютеров, обладающее каким-либо определенным свойством.

Если в каждом домене и поддомене обеспечивается уникальность имен следующего уровня иерархии, то и вся система имен будет состоять из уникальных имен.

По аналогии с файловой системой в доменной системе имен различают краткие имена, относительные имена и полные доменные имена. **Краткое имя** — это имя конечного узла сети: хоста или порта маршрутизатора. Краткое имя — это лист дерева имен. **Относительное имя** — это составное имя, начинающееся с некото-

рого уровня иерархии, но не самого верхнего. Например, www1.zil — это относительное имя. **Полное доменное имя** (Fully Qualified Domain Name, FQDN) включает составляющие всех уровней иерархии, начиная от краткого имени и кончая корневой точкой: www1.zil.mmt.ru.

Корневой домен управляется центральными органами Интернета IANA и InterNIC. Домены верхнего уровня назначаются для каждой страны, а также для различных типов организаций. Имена этих доменов должны следовать международному стандарту ISO 3166. Для обозначения стран используются трехбуквенные и двухбуквенные аббревиатуры, например ru (Россия), uk (Великобритания), fi (Финляндия), us (Соединенные Штаты), а для различных типов организаций — например, следующие обозначения:

- com — коммерческие организации (например, microsoft.com);
- edu — образовательные организации (например, mit.edu);
- gov — правительственные организации (например, nsf.gov);
- org — некоммерческие организации (например, fidonet.org);
- net — сетевые организации (например, nsf.net).

Каждый домен администрирует отдельная организация, которая обычно разбивает свой домен на поддомены и передает функции администрирования этих поддоменов другим организациям. Чтобы получить доменное имя, необходимо зарегистрироваться в какой-либо организации, которой орган InterNIC делегировал свои полномочия по распределению имен доменов.

ВНИМАНИЕ

Компьютеры входят в домен в соответствии со своими составными именами, при этом они могут иметь абсолютно независимые друг от друга IP-адреса, принадлежащие различным сетям и подсетям. Например, в домен mgu.ru могут входить хосты с адресами 132.13.34.15, 201.22.100.33 и 14.0.0.6.

Доменная система имен реализована в Интернете, но она может работать и как автономная система имен в любой крупной корпоративной сети, которая хотя и использует стек TCP/IP никак не связана с Интернетом.

Схема работы DNS

Широковещательный способ установления соответствия между символьными именами и локальными адресами, подобный протоколу ARP, хорошо работает только в небольшой локальной сети, не разделенной на подсети. В крупных сетях, где возможность всеобщей широковещательной рассылки не поддерживается, нужен другой способ разрешения символьных имен. Хорошей альтернативой широковещательной рассылке является применение централизованной службы, поддерживающей соответствие между различными типами адресов всех компьютеров сети. Например, компания Microsoft для своей корпоративной операционной системы Windows NT разработала централизованную службу WINS, которая поддерживала базу данных NetBIOS-имен и соответствующих им IP-адресов.

В сетях TCP/IP соответствие между доменными именами и IP-адресами может устанавливаться средствами как локального хоста, так и централизованной службы.

На раннем этапе развития Интернета на каждом хосте вручную создавался текстовый файл с известным именем hosts.txt. Этот файл состоял из некоторого количества строк, каждая из которых содержала одну пару «доменное имя — IP-адрес», например:

rhino.acme.com — 102.54.94.97.

По мере роста Интернета файлы hosts.txt также увеличивались в объеме, и создание *масштабируемого* решения для разрешения имен стало необходимостью.

Таким решением стала *централизованная служба* **DNS** (Domain Name System — система доменных имен), основанная на распределенной базе отображений «доменное имя — IP-адрес». Служба DNS использует в своей работе DNS-серверы и DNS-клиенты. DNS-серверы поддерживают распределенную базу отображений, а DNS-клиенты обращаются к серверам с запросами о разрешении доменного имени в IP-адрес.

Служба DNS использует текстовые файлы почти такого формата, как и файл hosts, и эти файлы администратор также подготавливает вручную. Однако служба DNS опирается на иерархию доменов, и каждый сервер службы DNS хранит только часть имен сети, а не все имена, как это происходит при использовании файлов hosts. При росте количества узлов в сети проблема масштабирования решается созданием новых доменов и поддоменов имен и добавлением в службу DNS новых серверов.

Для каждого домена имен создается свой DNS-сервер. Имеется два распределения имен на серверах. В первом случае сервер может хранить отображения «доменное имя — IP-адрес» для всего домена, включая все его поддомены. Однако такое решение оказывается плохо масштабируемым, так как при добавлении новых поддоменов нагрузка на этот сервер может превысить его возможности. Чаще используется другой подход, когда сервер домена хранит только имена, которые заканчиваются на следующем ниже уровне иерархии по сравнению с именем домена. (Аналогично каталогу файловой системы, который содержит записи о файлах и подкаталогах, непосредственно в него «входящих».) Именно при такой организации службы DNS нагрузка по разрешению имен распределяется более-менее равномерно между всеми DNS-серверами сети. Например, в первом случае DNS-сервер домена mmt.ru будет хранить отображения для всех имен, заканчивающихся на mmt.ru (www1.zil.mmt.ru, ftp.zil.mmt.ru, mail.mmt.ru и т. д.). Во втором случае этот сервер хранит отображения только имен типа mail.mmt.ru, www.mmt.ru, а все остальные отображения должны храниться на DNS-сервере поддомена zil.

Каждый DNS-сервер помимо таблицы отображений имен содержит ссылки на DNS-серверы своих поддоменов. Эти ссылки связывают отдельные DNS-серверы в единую службу DNS. Ссылки представляют собой IP-адреса соответствующих серверов. Для обслуживания корневого домена выделено несколько дублирующих друг друга DNS-серверов, IP-адреса которых являются широко известными (их можно узнать, например, в InterNIC).

Процедура разрешения DNS-имени во многом аналогична процедуре поиска файловой системой адреса файла по его символьному имени. Действительно, в обоих случаях составное имя отражает иерархическую структуру организации соответствующих справочников — каталогов файлов или DNS-таблиц. Здесь домен и доменный DNS-сервер являются аналогом каталога файловой системы. Для доменных имен, так же как и для символьных имен файлов, характерна независимость именования от физического местоположения.

Процедура поиска адреса файла по символьному имени заключается в последовательном просмотре каталогов, начиная с корневого. При этом предварительно проверяются кэш и текущий каталог. Для определения IP-адреса по доменному имени также необходимо просмотреть все DNS-серверы, обслуживающие цепочку поддоменов, входящих в имя хоста, начиная с корневого домена.

Существенным отличием файловой системы от службы DNS является то, что первая расположена на одном компьютере, а вторая по своей природе является *распределенной*.

Существует две основные схемы разрешения DNS-имен. В первом варианте работу по поиску IP-адреса координирует DNS-клиент.

1. DNS-клиент обращается к корневому DNS-серверу с указанием полного доменного имени.
2. DNS-сервер отвечает клиенту, указывая адрес следующего DNS-сервера, обслуживающего домен верхнего уровня, заданный в следующей старшей части запрошенного имени.
3. DNS-клиент делает запрос следующего DNS-сервера, который отсылает его к DNS-серверу нужного поддомена и т. д., пока не будет найден DNS-сервер, в котором хранится соответствие запрошенного имени IP-адресу. Этот сервер дает окончательный ответ клиенту.

Такая процедура разрешения имени называется **нерекурсивной**, когда клиент сам итеративно выполняет последовательность запросов к разным серверам имен. Эта схема загружает клиента достаточно сложной работой, и она применяется редко.

Во втором варианте реализуется **рекурсивная** процедура.

1. DNS-клиент запрашивает локальный DNS-сервер, то есть тот сервер, обслуживающий поддомен, которому принадлежит имя клиента.
2. Далее возможны два варианта действий.
   - если локальный DNS-сервер знает ответ, то он сразу же возвращает его клиенту (это может произойти, когда запрошенное имя входит в тот же поддомен, что и имя клиента, или когда сервер уже узнавал данное соответствие для другого клиента и сохранил его в своем кэше);
   - если локальный сервер не знает ответ, то он выполняет итеративные запросы к корневому серверу и т. д. точно так же, как это делал клиент в предыдущем варианте, а получив ответ, передает его клиенту, который все это время просто ждет его от своего локального DNS-сервера.

В этой схеме клиент перепоручает работу своему серверу, поэтому схема называется косвенной, или рекурсивной. Практически все DNS-клиенты используют рекурсивную процедуру.

Для ускорения поиска IP-адресов DNS-серверы широко применяют *кэширование* проходящих через них ответов. Чтобы служба DNS могла оперативно отрабатывать изменения, происходящие в сети, ответы кэшируются на относительно короткое время — обычно от нескольких часов до нескольких дней.

Обратная зона

Служба DNS предназначена не только для нахождения IP-адреса по имени хоста, но и для решения *обратной задачи* — нахождению DNS-имени по известному IP-адресу.

Многие программы и утилиты, пользующиеся службой DNS, пытаются найти имя узла по его адресу в том случае, когда пользователем задан только адрес (или этот адрес программа узнала из пришедшего пакета). Обратная запись не всегда существует даже для тех адресов, для которых есть прямые записи. Ее могли просто забыть создать или же ее создание требует дополнительной оплаты. Обратная задача решается в Интернете путем организации так называемых обратных зон.

Обратная зона — это система таблиц, которая хранит соответствие между IP-адресами и DNS-имена хостов некоторой сети. Для организации распределенной службы и использования для поиска имен того же программного обеспечения, что и для поиска адресов, применяется оригинальный подход, связанный с представлением IP-адреса в виде DNS-имени.

Первый этап преобразования заключается в том, что составляющие IP-адреса интерпретируются как составляющие DNS-имени. Например, адрес 192.31.106.0 рассматривается как состоящий из старшей части, соответствующей домену 192, затем идет домен 31, в который входит домен 106.

Далее, учитывая, что при записи IP-адреса старшая часть является самой *левой* частью адреса, а при записи DNS-имени — самой *правой*, то составляющие в преобразованном адресе указываются в обратном порядке, то есть для данного примера — 106.31.192.

Для хранения соответствия всех адресов, начинающихся, например, с числа 192, заводится зона 192 со своими серверами имен. Для записей о серверах, поддерживающих старшие в иерархии обратные зоны, создана специальная зона in-addr.arpa, поэтому полная запись для использованного в примере адреса выглядит так:

106.31.192.in-addr.arpa

Серверы для обратных зон используют файлы баз данных, не зависящие от файлов основных зон, в которых имеются записи о прямом соответствии тех же имен и адресов. Такая организация данных может приводить к несогласованности, так как одно и то же соответствие вводится в файлы дважды.

Протокол DHCP

Список ключевых слов: конфигурирование, конфигурационные параметры, протокол динамического конфигурирования хостов, ручной режим конфигурирования, режим автоматического назначения статических адресов, динамическое распределение адресов, срок аренды, DHCP-агент.

Для нормальной работы сети каждому сетевому интерфейсу компьютера и маршрутизатора должен быть назначен IP-адрес.

Процедура присвоения адресов происходит в ходе **конфигурирования** компьютеров и маршрутизаторов. Назначение IP-адресов может происходить вручную в результате выполнения процедуры конфигурирования интерфейса, для компьютера сводящейся, например, к заполнению системы экранных форм. При этом администратор должен помнить, какие адреса из имеющегося множества он уже использовал для других интерфейсов, а какие еще свободны. При конфигурировании помимо IP-адресов сетевых интерфейсов (и соответствующих масок) устройству сообщается ряд других **конфигурационных параметров**. При конфигурировании администратор должен назначить клиенту не только IP-адрес, но и другие параметры стека TCP/IP, необходимые для его эффективной работы, например маску и IP-адрес маршрутизатора по умолчанию, IP-адрес сервера DNS, доменное имя компьютера и т. п. Даже при не очень большом размере сети эта работа представляет для администратора утомительную процедуру.

Протокол динамического конфигурирования хостов (Dynamic Host Configuration Protocol, DHCP) автоматизирует процесс конфигурирования сетевых интерфейсов, гарантируя от дублирования адресов за счет централизованного управления их распределением. Работа DHCP описана в RFC 2131 и 2132.

Режимы DHCP

Протокол DHCP работает в соответствии с моделью *клиент-сервер*. Во время старта системы компьютер, являющийся DHCP-клиентом, посылает в сеть широковещательный запрос на получение IP-адреса. DHCP-сервер откликается и посылает сообщение-ответ, содержащее IP-адрес и некоторые другие конфигурационные параметры.

При этом сервер DHCP может работать в разных режимах, включая:

- ручное назначение статических адресов;
- автоматическое назначение статических адресов;
- автоматическое распределение динамических адресов.

Во всех режимах работы администратор при конфигурировании DHCP-сервера сообщает ему один или несколько диапазонов IP-адресов, причем все эти адреса относятся к одной сети, то есть имеют одно и то же значение в поле номера сети.

В **ручном** режиме администратор, помимо пула доступных адресов, снабжает DHCP-сервер информацией о жестком соответствии IP-адресов физическим адресам или другим идентификаторам клиентских узлов. DHCP-сервер, пользуясь

этой информацией, *всегда* выдает определенному DHCP-клиенту *один и тот же* назначенный ему администратором IP-адрес (а также набор других конфигурационных параметров[1]).

В режиме **автоматического** назначения статических адресов DHCP-сервер самостоятельно без вмешательства администратора произвольным образом выбирает клиенту IP-адрес из пула наличных IP-адресов. Адрес дается клиенту из пула в постоянное пользование, то есть между идентифицирующей информацией клиента и его IP-адресом по-прежнему, как и при ручном назначении, существует постоянное соответствие. Оно устанавливается в момент первого назначения DHCP-сервером IP-адреса клиенту. При всех последующих запросах сервер возвращает клиенту тот же самый IP-адрес.

При **динамическом** распределении адресов DHCP-сервер выдает адрес клиенту на ограниченное время, называемое **сроком аренды**. Когда компьютер, являющийся DHCP-клиентом, удаляется из подсети, назначенный ему IP-адрес автоматически освобождается. Когда компьютер подключается к другой подсети, то ему автоматически назначается новый адрес. Ни пользователь, ни сетевой администратор не вмешиваются в этот процесс.

Это дает возможность впоследствии повторно использовать этот IP-адрес для назначения другому компьютеру. Таким образом, помимо основного преимущества DHCP — автоматизации рутинной работы администратора по конфигурированию стека TCP/IP на каждом компьютере, динамическое разделение адресов в принципе позволяет строить IP-сеть, количество узлов в которой превышает количество имеющихся в распоряжении администратора IP-адресов.

Пример

Рассмотрим преимущества, которые дает динамическое разделение пула адресов на примере организации, в которой сотрудники значительную часть рабочего времени проводят вне офиса — дома или в командировках. Каждый из них имеет портативный компьютер, который во время пребывания в офисе подключается к корпоративной IP-сети. Возникает вопрос, сколько IP-адресов необходимо этой организации?

Первый ответ — столько, *скольким сотрудникам необходим доступ в сеть*. Если их 500 человек, то каждому из них должен быть назначен IP-адрес и выделено рабочее место. То есть администрация должна получить у поставщика услуг адреса двух сетей класса C и оборудовать соответствующим образом помещение. Однако вспомним, что сотрудники в этой организации редко появляются в офисе, значит, большая часть ресурсов при таком решении будет простаивать.

Второй ответ — столько, *сколько сотрудников обычно присутствует в офисе* (с некоторым запасом). Если обычно в офисе работает не более 50 сотрудников, то достаточно получить у поставщика услуг пул из 64 адресов и установить в рабочем помещении сеть с 64 коннекторами для подключения компьютеров. Но возникает другая проблема — кто и как будет конфигурировать компьютеры, состав которых постоянно меняется?

Существует два пути. Во-первых, администратор (или сам мобильный пользователь) может конфигурировать компьютер вручную каждый раз, когда возникает необходимость подключения к офисной сети. Такой подход требует от администратора (или пользователей) выполнения большого объема рутинной работы, следовательно — это плохое ре-

[1] Иногда для краткости мы будем опускать это уточнение.

шение. Гораздо привлекательнее выглядят возможности автоматического динамического назначения адресов DHCP. Действительно, администратору достаточно один раз при настройке DHCP-сервера указать диапазон из 64 адресов, а каждый вновь прибывающий мобильный пользователь будет просто физически подключать в сеть свой компьютер, на котором запускается DHCP-клиент. Он запросит конфигурационные параметры и автоматически получит их от DHCP-сервера. Таким образом, для работы 500 мобильных сотрудников достаточно иметь в офисной сети 64 IP-адреса и 64 рабочих места.

Алгоритм динамического назначения адресов

Администратор управляет процессом конфигурирования сети, определяя два основных параметра конфигурации DHCP-сервера: *пул адресов, доступных распределению*, и *срок аренды*. Срок аренды диктует, как долго компьютер может использовать назначенный IP-адрес, перед тем как снова запросить его от DHCP-сервера. Срок аренды зависит от режима работы пользователей сети. Если это небольшая сеть учебного заведения, куда со своими компьютерами приходят многочисленные студенты для выполнения лабораторных работ, то срок аренды может быть равен длительности лабораторной работы. Если же это корпоративная сеть, в которой сотрудники предприятия работают на регулярной основе, то срок аренды может быть достаточно длительным — несколько дней или даже недель.

DHCP-сервер должен находиться в одной подсети с клиентами, учитывая, что клиенты посылают ему широковещательные запросы. Для снижения риска выхода сети из строя из-за отказа DHCP-сервера в сети иногда ставят резервный DHCP-сервер (такой вариант соответствует сети 1 на рис. 17.8).

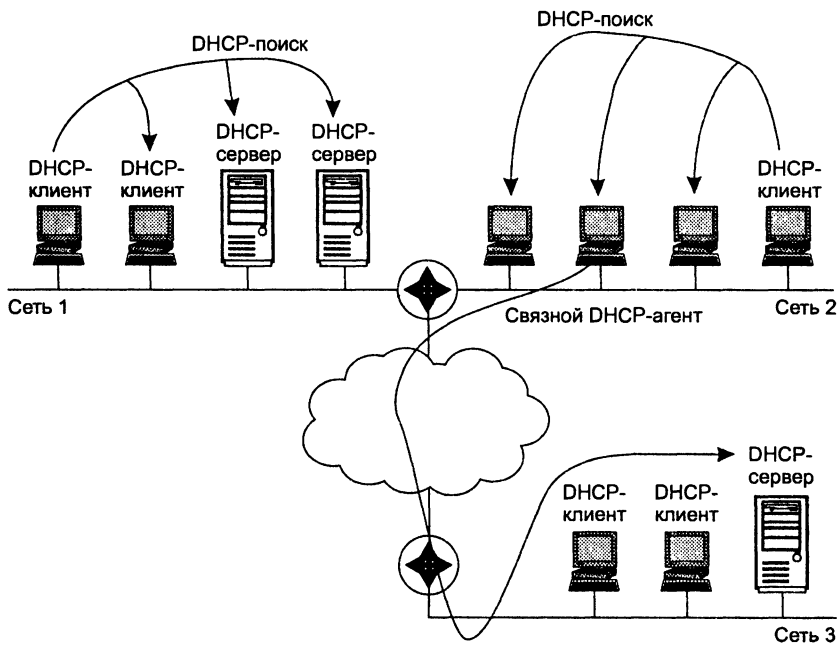


Рис. 17.8. Схемы взаимного расположения DHCP-серверов и DHCP-клиентов

Иногда наблюдается и обратная картина: в сети нет ни одного DHCP-сервера, его подменяет связной **DHCP-агент** — программное обеспечение, играющее роль посредника между DHCP-клиентами и DHCP-серверами (пример такого варианта — сеть 2 на рисунке). Связной агент переправляет запросы клиентов из сети 2 DHCP-серверу сети 3. Таким образом, один DHCP-сервер может обслуживать DHCP-клиентов нескольких разных сетей.

Ниже дана упрощенная схема обмена сообщениями между клиентскими и серверными частями DHCP.

1. Когда компьютер включают, установленный на нем DHCP-клиент посылает ограниченное широковещательное сообщение DHCP-поиска (IP-пакет с адресом назначения, состоящим из одних единиц, который должен быть доставлен всем узлам данной IP-сети).
2. Находящиеся в сети DHCP-серверы получают это сообщение. Если в сети DHCP-серверы отсутствуют, то сообщение DHCP-поиска получает связной DHCP-агент. Он пересылает это сообщение в другую, возможно, значительно отстоящую от него сеть DHCP-серверу, IP-адрес которого ему заранее известен.
3. Все DHCP-серверы, получившие сообщение DHCP-поиска, посылают DHCP-клиенту, обратившемуся с запросом, свои DHCP-предложения. Каждое предложение содержит IP-адрес и другую конфигурационную информацию. (DHCP-сервер, находящийся в другой сети, посылает ответ через агента.)
4. DHCP-клиент собирает конфигурационные DHCP-предложения ото всех DHCP-серверов. Как правило, он выбирает первое из поступивших предложений и отправляет в сеть широковещательный DHCP-запрос. В этом запросе содержатся идентификационная информация о DHCP-сервере, предложение которого принято, а также значения принятых конфигурационных параметров.
5. Все DHCP-серверы получают DHCP-запрос, и только один выбранный DHCP-сервер посылает положительную DHCP-квитанцию (подтверждение IP-адреса и параметров аренды), а остальные серверы аннулируют свои предложения, в частности возвращают в свои пулы предложенные адреса.
6. DHCP-клиент получает положительную DHCP-квитанцию и переходит в рабочее состояние.

Время от времени компьютер пытается обновить параметры аренды у DHCP-сервера. Первую попытку он делает задолго до истечения срока аренды, обращаясь к тому серверу, от которого он получил текущие параметры. Если ответа нет или ответ отрицательный, он через некоторое время снова посылает запрос. Так повторяется несколько раз, и, если все попытки получить параметры у того же сервера оказываются безуспешными, клиент обращается к другому серверу. Если и другой сервер отвечает отказом, то клиент теряет свои конфигурационные параметры и переходит в режим автономной работы.

DHCP-клиент может и по своей инициативе досрочно отказаться от выделенных ему параметров.

В сети, где адреса назначаются динамически, нельзя быть уверенным в адресе, который в данный момент имеет тот или иной узел. И такое непостоянство IP-адресов влечет за собой некоторые проблемы.

Во-первых, *возникают сложности при преобразовании символьного доменного имени в IP-адрес.* Действительно, представьте себе функционирование системы DNS, которая должна поддерживать таблицы соответствия символьных имен IP-адресам в условиях, когда последние меняются каждые два часа! Учитывая это обстоятельство, для серверов, к которым пользователи часто обращаются по символьному имени, назначают статические IP-адреса, оставляя динамические только для клиентских компьютеров. Однако в некоторых сетях количество серверов настолько велико, что их ручное конфигурирование становится слишком обременительным. Это привело к разработке усовершенствованной версии DNS (так называемой динамической системы DNS), в основе которой лежит согласование информационной адресной базы в службах DHCP и DNS.

Во-вторых, *трудно осуществлять удаленное управление и автоматический мониторинг интерфейса* (например, сбор статистики), если в качестве его идентификатора выступает динамически изменяемый IP-адрес.

Наконец, для обеспечения безопасности сети многие сетевые устройства могут блокировать (фильтровать) пакеты, определенные поля которых имеют некоторые заранее заданные значения. Другими словами, при динамическом назначении адресов *усложняется фильтрация пакетов по IP-адресам.*

Последние две проблемы проще всего решаются отказом от динамического назначения адресов для интерфейсов, фигурирующих в системах мониторинга и безопасности.

Выводы

В стеке TCP/IP используются три типа адресов: локальные (называемые также аппаратными), IP-адреса и символьные доменные имена. Все эти типы адресов присваиваются узлам составной сети независимо друг от друга.

IP-адрес имеет длину 4 байта и состоит из номера сети и номера узла. Для определения границы, отделяющей номер сети от номера узла, сегодня используется два подхода. Первый основан на классах адресов, второй — масок.

Класс адреса определяется значениями нескольких первых битов адреса. В адресах класса A под номер сети отводится один байт, а остальные три байта — под номер узла, поэтому они используются в самых больших сетях. Для небольших сетей больше подходят адреса класса C, в которых номер сети занимает три байта, а для нумерации узлов может быть использован только один байт. Промежуточное положение занимают адреса класса B.

Для разделения IP-адреса на номер сети и номер узла используется связанная с этим адресом маска. Двоичная запись маски содержит единицы в тех разрядах, которые в данном IP-адресе должны интерпретироваться как номер сети.

IP-адреса уникально идентифицируют узел в пределах составной сети, поэтому они должны назначаться централизовано. Если сеть автономная, то уникальность IP-адресов в пределах этой сети может быть обеспечена администратором сети. При этом он может выбирать для нумерации сетей и узлов любые синтаксически правильные IP-адреса. Однако предпочтитель-

нее в этом случае применять адреса, специально выделенные для автономных сетей (так называемые частные адреса).

Если сеть очень велика, как, например, Интернет, то процесс назначения IP-адресов усложняется, разбиваясь на два этапа. Первый — распределение номеров сетей — регулируется специальным административным органом, обеспечивающим однозначность нумерации сетей. После того как сеть получила номер, наступает второй этап — назначение номеров узлам сети.

Назначение IP-адресов узлам сети может происходить либо вручную (администратор сам ведет списки свободных и занятых адресов и конфигурирует сетевой интерфейс), либо автоматически (с использованием протокола DHCP). В последнем случае администратор заранее назначает DHCP-серверу диапазон свободных для распределения адресов, из которого последний автоматически выделяет адреса узлам в ответ на поступившие от них запросы.

Установление соответствия между IP-адресом и аппаратным адресом сетевого интерфейса осуществляется протоколом разрешения адресов (ARP).

Протокол ARP, работающий в сетях Ethernet, Token Ring, FDDI, для трансляции IP-адреса в MAC-адрес выполняет широковещательный ARP-запрос. Поступающие ARP-ответы запоминаются в таблицах, создаваемых на каждом сетевом интерфейсе. В сетях, в которых не поддерживается широковещательная рассылка, ARP-таблицы хранятся централизовано на выделенном ARP-сервере.

В стеке TCP/IP применяется система доменных символьных имен, которая имеет иерархическую древовидную структуру, допускающую использование в имени произвольного количества составных частей. Совокупность имен, у которых несколько старших составных частей совпадают, образуют домен имен. Доменные имена назначаются централизованно, если сеть является частью Интернета, в противном случае — локально.

Соответствие между доменными именами и IP-адресами может устанавливаться как средствами локального хоста с использованием файла hosts, так и с помощью централизованной службы DNS, основанной на распределенной базе отображений «доменное имя — IP-адрес».

Вопросы и задания

1. В чем состоит отличие процедур назначения аппаратных и сетевых адресов?
2. Какие из адресов могли бы в составной IP-сети являться локальными, а какие нет? Варианты ответов:
   - 6-байтовый MAC-адрес, например, 12-B3-3B-51-A2-10;
   - адрес X.25, например, 25012112654987;
   - 12-байтовый IPX-адрес, например, 13.34.B4.0A.C5.10.11.32.54.C5.3B.01;
   - адрес VPI/VCI сети ATM.
3. Какие из следующих утверждений верны всегда?
   - Каждый интерфейс моста/коммутатора имеет MAC-адрес.
   - Каждый мост/коммутатор имеет сетевой адрес.
   - Каждый интерфейс моста/коммутатора имеет сетевой адрес.
   - Каждый маршрутизатор имеет сетевой адрес.
   - Каждый интерфейс маршрутизатора имеет MAC-адрес.
   - Каждый интерфейс маршрутизатора имеет сетевой адрес.

4. Какие из приведенных адресов не могут быть использованы в качестве IP-адресов сетевого интерфейса для узлов Интернета? Для синтаксически правильных адресов определите их класс: A, B, C, D или E. Варианты адресов:
   - 127.0.0.1;
   - 201.13.123.245;
   - 226.4.37.105;
   - 103.24.254.0;
   - 10.234.17.25;
   - 154.12.255.255;
   - 13.13.13.13;
   - 204.0.3.1;
   - 193.256.1.16;
   - 194.87.45.0;
   - 195.34.116.255;
   - 161.23.45.305.
5. Пусть IP-адрес некоторого узла подсети равен 198.65.12.67, а значение маски для этой подсети — 255.255.255.240. Определите номер подсети. Какое максимальное число узлов может быть в этой подсети?
6. Пусть вам известно соответствие между IP-адресами и доменными именами для всех компьютеров в сети, кроме одного. Для этого компьютера вы знаете только доменное имя. Можете ли вы, обладая всей этой информацией, с уверенностью определить его IP-адрес?
7. Сколько ARP-таблиц имеет компьютер? Маршрутизатор? Коммутатор?
8. Протокол ARP функционально можно разделить на клиентскую и серверную части. Опишите, какие функции вы отнесли бы к клиентской части, а какие — к серверной?
9. Какие адреса и с какой целью заносит администратор в ARP-таблицу?
10. В каких случаях полезно использовать протокол Proxy-ARP?
11. Можно ли определить по доменным именам компьютеров, насколько близко (территориально) они находятся?
12. Известно, что компьютер с адресом 204.35.101.24 имеет доменное имя new.firm.net. Определите, если это возможно, какое из доменных имен имеет компьютер с адресом 204.35.101.25. Варианты ответов:
    - new1.firm.net;
    - new.firm1.net;
    - new.1firm.net.
13. Что общего между системой DNS и файловой системой?
14. Сколько DHCP-серверов достаточно, чтобы обслужить сеть, разделенную двумя маршрутизаторами?

15. Если в сети для надежности установлено два DHCP-сервера, то каким образом следует администратору назначать для каждого из них пул распределяемых адресов: выделить каждому из них неперекрывающиеся части общего пула или назначить каждому из них один и тот же общий пул?
16. Пусть поставщик услуг Интернета имеет в своем распоряжении адрес сети класса B. Для адресации узлов собственной сети он использует 254 адреса. Определите максимально возможное число абонентов этого поставщика услуг, если размеры требуемых для них сетей соответствуют классу C? Какая маска должна быть установлена на маршрутизаторе поставщика услуг, соединяющем его сеть с сетями абонентов?
17. Какое максимальное количество подсетей теоретически можно организовать, если в вашем распоряжении имеется сеть класса C? Какое значение должна при этом иметь маска?
18. Почему для решения обратной задачи — поиска IP-адресов по известным именам — не используют тот же подход, что и для решения прямой задачи, то есть те же файлы зон и доменов, организованных в дерево, соответствующее иерархии имен.

ГЛАВА 18 Протокол межсетевого взаимодействия

Эта глава посвящена протоколу **IP** (Intrenet Protocol — межсетевой протокол), описанному в документе RFC 751. В каждой очередной сети, лежащей на пути перемещения пакета, протокол IP обращается к средствам транспортировки этой сети, чтобы с их помощью передать пакет на маршрутизатор, ведущий к следующей сети, или непосредственно на узел-получатель. Таким образом, одной из важнейших функций IP является *поддержание интерфейса с нижележащими технологиями* сетей, образующих составную сеть. Кроме того, в функции протокола IP входит *поддержание интерфейса с протоколами вышележащего транспортного уровня*, в частности с протоколом TCP, который решает все вопросы обеспечения надежной доставки данных по составной сети в стеке TCP/IP.

Протокол IP относится к протоколам *без установления соединений*, он поддерживает обработку каждого IP-пакета как независимой единицы обмена, не связанной с другими IP-пакетами. В протоколе IP нет механизмов, обычно применяемых для обеспечения достоверности конечных данных. Если во время продвижения пакета происходит какая-либо ошибка, то протокол IP по своей инициативе ничего не предпринимает для исправления этой ошибки. Например, если на промежуточном маршрутизаторе пакет был отброшен из-за ошибки по контрольной сумме, то модуль IP не пытается заново послать потерянный пакет. Другими словами, протокол IP реализует политику доставки «по возможности» (с максимальными усилиями).

В этой главе мы подробно рассмотрим основную функцию протокола IP — *маршрутизацию*. Основательно изучим структуру таблиц маршрутизации как без использования, так и с использованием масок. Приведем примеры применения масок одинаковой и переменной длины, перекрывающихся адресных пространств, разделения на подсети и объединения подсетей. Также мы исследуем возможности протокола IP, связанные с фрагментацией пакетов.

При обсуждении особенностей новой версии IPv6 мы наиболее подробно остановимся на модернизации схемы адресации, сделавшей ее более масштабируемой, а также на изменении формата IP-заголовка, что позволило повысить пропускную способность сети за счет сокращения объема работ, выполняемых маршрутизаторами.

Формат IP-пакета

Список ключевых слов: номер версии протокола IP, длина заголовка IP-пакета, тип сервиса, байт дифференцированного обслуживания, или DS-байт, приоритет IP-пакета, критерий выбора маршрута, общая длина IP-пакета, идентификатор IP-пакета, флаги IP-пакета, смещение фрагмента, время жизни IP-пакета, протокол верхнего уровня, контрольная сумма заголовка, IP-адреса источника и приемника, поле параметров IP-пакета, выравнивание заголовка.

Имеется прямая связь между количеством полей заголовка пакета и функциональной сложностью протокола, который работает с этим заголовком. Чем проще заголовок — тем проще соответствующий протокол. Большая часть действий протокола связана с обработкой той служебной информации, которая переносится в полях заголовка пакета. Изучая назначение каждого поля заголовка IP-пакета, мы получаем не только формальные знания о структуре пакета, но и знакомимся с основными функциями протокола IP.

IP-пакет состоит из заголовка и поля данных. Ниже перечислены поля заголовка (рис. 18.1).

Рис. 18.1. Структура заголовка IP-пакета

Поле **номера версии** занимает 4 бита и идентифицирует версию протокола IP. Сейчас повсеместно используется версия 4 (IPv4), хотя все чаще встречается и новая версия (IPv6).

Значение **длины заголовка** IP-пакета также занимает 4 бита и измеряется в 32-битовых словах. Обычно заголовок имеет длину в 20 байт (пять 32-битовых слов), но при добавлении некоторой служебной информации это значение может быть увеличено за счет дополнительных байтов в поле параметров. Наибольшая длина заголовка составляет 60 байт.

Поле **типа сервиса** (Type of Service, ToS) имеет и другое, более современное название — **байт дифференцированного обслуживания**, или **DS-байт**. Этим двум

названиям соответствуют два варианта интерпретации этого поля. В обоих случаях данное поле служит одной цели — хранению признаков, которые отражают требования к качеству обслуживания пакета. В прежнем варианте первые три бита содержат значение **приоритета** пакета: от самого низкого — 0 до самого высокого — 7. Маршрутизаторы и компьютеры могут принимать во внимание приоритет пакета и обрабатывать более важные пакеты в первую очередь. Следующие три бита поля ToS определяют **критерий выбора маршрута**. Если бит D (Delay — задержка) установлен в 1, то маршрут должен выбираться для минимизации задержки доставки данного пакета, установленный бит T (Throughput — пропускная способность) — для максимизации пропускной способности, а бит R (Reliability — надежность) — для максимизации надежности доставки. Оставшиеся два бита имеют нулевое значение.

Стандарты дифференцированного обслуживания, принятые в конце 90-х годов, дали новое название этому полю и переопределили назначение его битов. В DS-байте также используются только старшие 6 бит, а два младших бита остаются в качестве резерва. Назначение битов DS-байта будет рассмотрено в подразделе «Дифференцированное обслуживание» раздела «Стандарты QoS в IP-сетях» главы 20.

Поле **общей длины** занимает 2 байта и характеризует общую длину пакета с учетом заголовка и поля данных. Максимальная длина пакета ограничена разрядностью поля, определяющего эту величину, и составляет 65 535 байт, однако в большинстве компьютеров и сетей столь большие пакеты не используются. При передаче по сетям различного типа длина пакета выбирается с учетом максимальной длины пакета протокола нижнего уровня, несущего IP-пакеты. Если это кадры Ethernet, то выбираются пакеты с максимальной длиной 1500 байт, умещающиеся в поле данных кадра Ethernet. В стандартах TCP/IP предусматривается, что все хосты должны быть готовы принимать пакеты вплоть до 576 байт длиной (независимо от того, приходят ли они целиком или фрагментами).

Идентификатор пакета занимает 2 байта и используется для распознавания пакетов, образовавшихся путем деления на части (фрагментации) исходного пакета. Все части (фрагменты) одного пакета должны иметь одинаковое значение этого поля.

Флаги занимают 3 бита и содержат признаки, связанные с фрагментацией. Установленный в 1 бит DF (Do not Fragment — не фрагментировать) запрещает маршрутизатору фрагментировать данный пакет, а установленный в 1 бит MF (More Fragments — больше фрагментов) говорит о том, что данный пакет является промежуточным (не последним) фрагментом. Оставшийся бит зарезервирован.

Поле **смещения фрагмента** занимает 13 бит и задает смещение в байтах поля данных этого фрагмента относительно начала поля данных исходного нефрагментированного пакета. Используется при сборке/разборке фрагментов пакетов. Смещение должно быть кратно 8 байт.

Поле **времени жизни** (Time To Live, TTL) занимает один байт и используется для задания предельного срока, в течение которого пакет может перемещаться по сети. Время жизни пакета измеряется в секундах и задается источником. По истечении каждой секунды пребывания на каждом из маршрутизаторов, через

которые проходит пакет во время своего «путешествия» по сети, из его текущего времени жизни вычитается единица; единица вычитается и в том случае, если время пребывания было меньше секунды. Поскольку современные маршрутизаторы редко обрабатывают пакет дольше, чем за одну секунду, то время жизни можно интерпретировать как максимальное число транзитных узлов, которые разрешено пройти пакету. Если значение поля времени жизни становится нулевым до того, как пакет достигает получателя, пакет уничтожается. Таким образом, время жизни является своего рода часовым механизмом самоуничтожения пакета.

Поле **протокола верхнего уровня** занимает один байт и содержит идентификатор, указывающий, какому протоколу верхнего уровня принадлежит информация, размещенная в поле данных пакета. Значения идентификаторов для разных протоколов приводятся в документе RFC 1700, доступном по адресу http://www.iana.org. Например, 6 означает, что в пакете находится сообщение TCP, 17 — сообщение UDP, 1 — сообщение ICMP.

Контрольная сумма заголовка занимает 2 байта (16 бит) и рассчитывается только по заголовку. Поскольку некоторые поля заголовка меняют свое значение в процессе передачи пакета по сети (например, поле времени жизни), контрольная сумма проверяется и повторно рассчитывается на каждом маршрутизаторе и конечном узле как дополнение к сумме всех 16-битовых слов заголовка. При вычислении контрольной суммы значение самого поля контрольной суммы устанавливается в нуль. Если контрольная сумма неверна, то пакет отбрасывается, как только обнаруживается ошибка.

Поля **IP-адресов источника** и **приемника** имеют одинаковую длину — 32 бита.

Поле **параметров** является необязательным и используется обычно только при отладке сети. Это поле состоит из нескольких подполей одного из восьми предопределенных типов. В этих подполях можно указывать точный маршрут, регистрировать проходимые пакетом маршрутизаторы, помещать данные системы безопасности или временные отметки.

Так как число подполей в поле параметров может быть произвольным, то в конце заголовка должно быть добавлено несколько нулевых байтов для **выравнивания** заголовка пакета по 32-битной границе.

Ниже приведена распечатка значений полей заголовка одного из реальных IP-пакетов, захваченных в сети Ethernet средствами анализатора протоколов сетевого монитора (Network Monitor, NM) от Microsoft. В данной распечатке NM в скобках дает шестнадцатеричные значения полей, кроме того, программа иногда представляет числовые коды полей в виде, более удобном для чтения. Например, дружественный программный интерфейс NM интерпретирует код 6 в поле протокола, помещая туда название соответствующего протокола — TCP (см. строку, выделенную полужирным шрифтом).

```
IP: Version = 4 (0x4)
IP: Header Length = 20 (0x14)
IP: Service Type = 0 (0x0)
IP: Precedence = Routine
IP: ...0.... = Normal Delay
```

```
IP: . ..0... = Normal Throughput
IP:    ..0.  = Normal Reliability
IP: Total Length = 54 (0x36)
IP: Identification = 31746 (0x7C02)
IP: Flags Summary = 2 (0x2)
IP: ..  ...0 = Last fragment in datagram
IP: ... ..1. = Cannot fragment datagram
IP: Fragment Offset = 0 (0x0) bytes
IP: Time to Live = 128 (0x80)
IP: Protocol = TCP - Transmission Control
IP: Checksum = 0xEB86
IP: Source Address = 194.85 135.75
IP: Destination Address = 194.85.135.66
IP: Data: Number of data bytes remaining = 34 (0x0022)
```

Схема IP-маршрутизации

Список ключевых слов: таблица маршрутизации, адрес назначения пакетов, сетевой адрес следующего маршрутизатора, сетевой адрес выходного интерфейса, специфический маршрут, маршрут по умолчанию, маршрутизатор по умолчанию, признак непосредственно подключенной сети, время жизни записи, поле источника, программное обеспечение стека TCP/IP, администратор, протокол маршрутизации.

Рассмотрим механизм IP-маршрутизации на примере составной сети, представленной на рис. 18.2. В этой сети 20 маршрутизаторов (изображенных в виде пронумерованных квадрантных блоков) объединяют 18 сетей в общую сеть; N1, N2, ..., N18 — это номера сетей. На каждом маршрутизаторе и конечных узлах A и B установлены протоколы IP.

Маршрутизаторы имеют по несколько интерфейсов (портов), к которым присоединяются сети. Каждый интерфейс маршрутизатора можно рассматривать как отдельный узел сети: он имеет сетевой адрес и локальный адрес в той подсети, которая к нему подключена. Например, маршрутизатор под номером 1 имеет три интерфейса, к которым подключены сети N1, N2, N3. На рисунке сетевые адреса этих портов обозначены IP_{11}, IP_{12} и IP_{13}. Интерфейс IP_{11} является узлом сети N1, и следовательно в поле номера сети порта IP_{11} содержится номер N1. Аналогично интерфейс IP_{12} — узел в сети N2, а порт IP_{13} — узел в сети N3. Таким образом, маршрутизатор можно рассматривать как совокупность нескольких узлов, каждый из которых входит в свою сеть. Как единое устройство маршрутизатор не имеет ни отдельного сетевого, ни локального адреса.

ПРИМЕЧАНИЕ

Если маршрутизатор имеет блок управления (например, SNMP-управления), то этот блок имеет собственные локальный и сетевой адреса, по которым к нему обращается центральная станция управления.

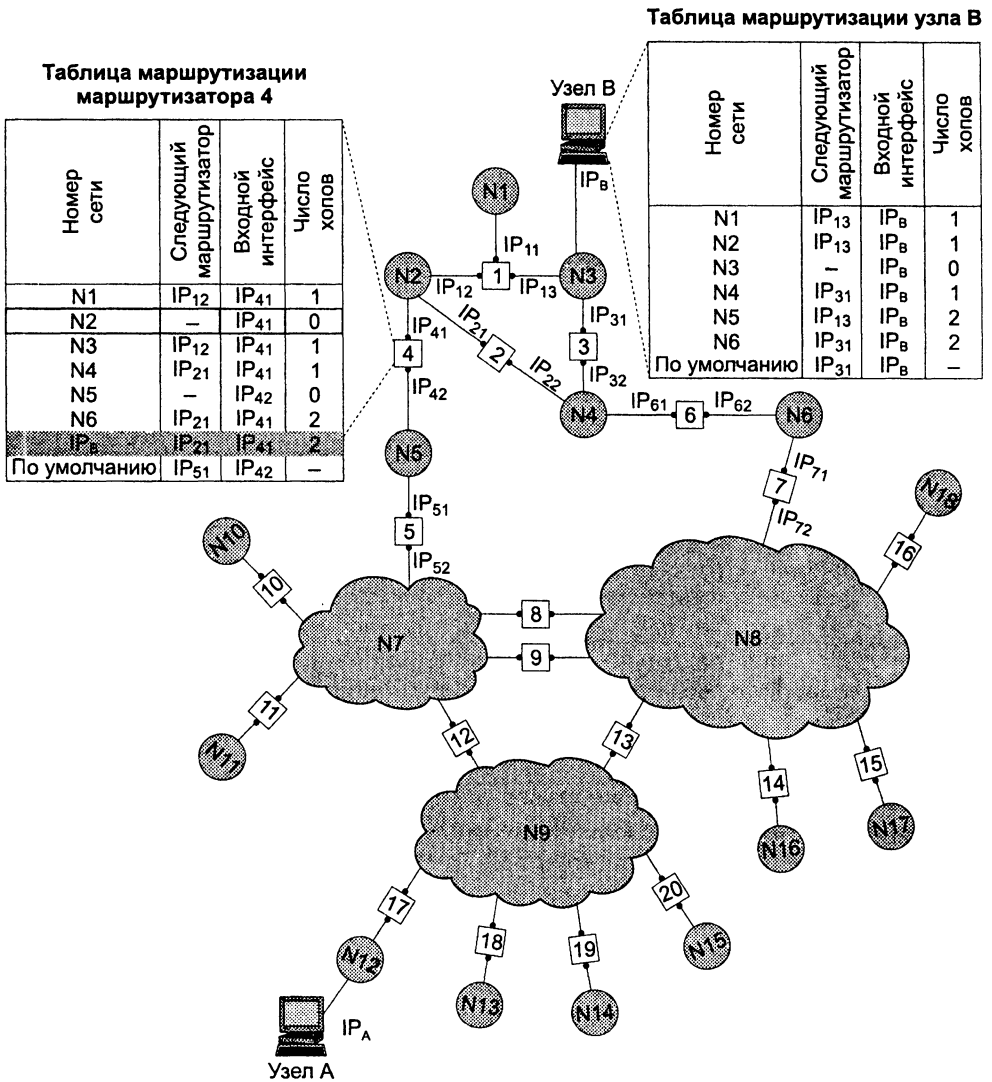


Рис. 18.2. Принципы маршрутизации в составной сети

В сложных составных сетях почти всегда существуют несколько альтернативных маршрутов для передачи пакетов между двумя конечными узлами. Так, пакет, отправленный из узла А в узел В, может пройти через маршрутизаторы 17, 12, 5, 4 и 1 или маршрутизаторы 17, 13, 7, 6 и 3. Нетрудно найти еще несколько маршрутов между узлами А и В.

Задачу выбора маршрута из нескольких возможных решают маршрутизаторы, а также конечные узлы. Маршрут выбирается на основании имеющейся у этих устройств информации о текущей конфигурации сети, а также на основании критерия выбора маршрута. В качестве критерия часто выступает задержка про-

хождения маршрута отдельным пакетом, средняя пропускная способность маршрута для последовательности пакетов или наиболее простой критерий, учитывающий только количество пройденных в маршруте промежуточных маршрутизаторов (*ретрансляционных участков*, или *хопов*). Полученная в результате анализа информация о маршрутах дальнейшего следования пакетов помещается в **таблицу маршрутизации**.

Упрощенная таблица маршрутизации

Используя условные обозначения для сетевых адресов маршрутизаторов и номеров сетей в том виде, как они приведены на рис. 18.2, посмотрим, как могла бы выглядеть таблица маршрутизации, например, в маршрутизаторе 4 (табл. 18.1).

Таблица 18.1. Таблица маршрутизации маршрутизатора 4

Адрес назначения	Сетевой адрес следующего маршрутизатора	Сетевой адрес выходного порта	Расстояние до сети назначения
N1	IP_{12} (R1)	IP41	1
N2	—	IP41	0 (подсоединена)
N3	IP_{12} (R1)	IP41	1
N4	IP_{21} (R2)	IP41	1
N5	—	IP42	0 (подсоединена)
N6	IP_{21} (R2)	IP21	2
IP_B	IP_{21} (R2)	IP41	2
Маршрут по умолчанию	IP_{51} (R5)	IP42	—

ПРИМЕЧАНИЕ

Таблица 18.1 значительно упрощена по сравнению с реальными таблицами, например, здесь отсутствуют столбцы с масками, признаками состояния маршрута, временем, в течение которого действительны записи данной таблицы (их применение будет рассмотрено позже). Вместо номера сети назначения может быть указан полный сетевой адрес отдельного узла назначения. Кроме того, как уже отмечалось, здесь указаны адреса сетей условного формата, не соответствующие какому-либо определенному сетевому протоколу. Тем не менее эта таблица содержит основные поля, имеющиеся в реальных таблицах.

Первый столбец таблицы содержит **адреса назначения пакетов**.

В каждой строке таблицы следом за адресом назначения указывается **сетевой адрес следующего маршрутизатора** (точнее, сетевой адрес интерфейса следующего маршрутизатора), на который надо направить пакет, чтобы тот передвигался по направлению к заданному адресу по рациональному маршруту.

Перед тем как передать пакет следующему маршрутизатору, текущий маршрутизатор должен определить, на какой из нескольких собственных портов (IP_{41} или IP_{42}) он должен поместить данный пакет. Для этого служит третий столбец таблицы маршрутизации, содержащий **сетевые адреса выходных интерфейсов**.

Некоторые реализации сетевых протоколов допускают наличие в таблице маршрутизации сразу *нескольких строк*, соответствующих одному и тому же адресу назначения. В этом случае при выборе маршрута принимается во внимание столбец «расстояние до сети назначения». При этом расстояние измеряется в любой метрике, используемой в соответствии с заданным в сетевом пакете критерием. Расстояние может измеряться временем прохождения пакета по линиям связи, различными характеристиками надежности линий связи на данном маршруте, пропускной способностью или другой величиной, отражающей качество данного маршрута по отношению к заданному критерию. В табл. 18.1 расстояние между сетями измеряется хопами. Расстояние для сетей, непосредственно подключенных к портам маршрутизатора, здесь принимается равным 0, однако в некоторых реализациях отсчет расстояний начинается с 1.

Когда пакет поступает на маршрутизатор, модуль IP извлекает из поступившего заголовка кадра номер сети назначения и последовательно сравнивает его с номерами сетей из каждой строки таблицы. Строка с совпавшим номером сети указывает ближайший маршрутизатор, на который следует направить пакет. Например, если на какой-либо порт маршрутизатора 4 поступает пакет, адресованный в сеть N6, то из таблицы маршрутизации следует, что адрес следующего маршрутизатора — IP_{21}, то есть очередным этапом движения данного пакета будет движение к порту 1 маршрутизатора 2.

Чаще всего в качестве адреса назначения в таблице указывается не весь IP-адрес, а только номер сети назначения. Таким образом, для всех пакетов, направляемых в одну и ту же сеть, протокол IP будет предлагать один и тот же маршрут (мы пока не принимаем во внимание возможные изменения в состоянии сети — отказы маршрутизаторов или обрывы кабелей). Однако в некоторых случаях возникает необходимость для одного из узлов сети определить **специфический маршрут**, отличающийся от маршрута, заданного для всех остальных узлов сети. Для этого в таблицу маршрутизации помещают для данного узла отдельную строку, содержащую его полный IP-адрес и соответствующую маршрутную информацию. Такого рода запись имеется в табл. 18.1 для узла B. Пусть, например, администратор маршрутизатора 4, руководствуясь соображениями безопасности, решил, что пакеты, следующие в узел B (полный адрес IP_B), должны идти через маршрутизатор 2 (интерфейс IP_{21}), а не маршрутизатор 1 (интерфейс IP_{12}), через который передаются пакеты всем остальным узлам сети N3. Если в таблице имеются записи о маршрутах как к сети в целом, так и к ее отдельному узлу, то при поступлении пакета, адресованного данному узлу, маршрутизатор отдаст предпочтение специфическому маршруту.

Поскольку пакет может быть адресован *в любую сеть* составной сети, может показаться, что каждая таблица маршрутизации должна иметь записи обо *всех* сетях, входящих в составную сеть. Но при таком подходе в случае крупной сети объем таблиц маршрутизации может оказаться очень большим, что повлияет на

время ее просмотра, потребует много места для хранения и т. п. Поэтому на практике широко известен прием уменьшения количества записей в таблице маршрутизации, основанный на введении **маршрута по умолчанию** (default route). В этом приеме используются особенности топологии сети. Рассмотрим, например, маршрутизаторы, находящиеся на периферии составной сети. В их таблицах достаточно записать номера только тех сетей, которые непосредственно подсоединены к данному маршрутизатору или расположены поблизости, на тупиковых маршрутах. Обо всех же остальных сетях можно сделать в таблице единственную запись, указывающую на маршрутизатор, через который пролегает путь ко всем этим сетям. Такой маршрутизатор называется **маршрутизатором по умолчанию** (default router). В нашем примере маршрутизатор 4 указывает специфические маршруты только для пакетов, следующих в сети N1–N6. Для всех остальных пакетов, адресованных в сети N7–N18, маршрутизатор предлагает продолжить путь через один и тот же порт IP_{51} маршрутизатора 5, который в данном случае и является маршрутизатором по умолчанию.

Таблицы маршрутизации конечных узлов

Задачу маршрутизации решают не только промежуточные (маршрутизаторы), но и конечные узлы — компьютеры. Решение этой задачи начинается с того, что протокол IP, установленный на конечном узле, определяет, направляется ли пакет в другую сеть или адресован какому-нибудь узлу данной сети. Если номер сети назначения совпадает с номером данной сети, это означает, что пакет маршрутизировать не требуется. В противном случае маршрутизация нужна.

Структуры таблиц маршрутизации конечных узлов и транзитных маршрутизаторов аналогичны. Обратимся снова к сети, изображенной на рис. 18.2. Таблица маршрутизации конечного узла B, принадлежащего сети N3, могла бы выглядеть следующим образом (табл. 18.2). Здесь IP_B — сетевой адрес интерфейса компьютера B. На основании этой таблицы конечный узел B выбирает, на какой из двух имеющихся в локальной сети N3 маршрутизаторов (R1 или R3) следует посылать тот или иной пакет.

Таблица 18.2. Таблица маршрутизации конечного узла B

Номер сети назначения	Сетевой адрес следующего маршрутизатора	Сетевой адрес выходного порта	Расстояние до сети назначения
N1	IP_{13} (R1)	IP_B	1
N2	IP_{13} (R1)	IP_B	1
N3	–	IP_B	0
N4	IP_{31} (R3)	IP_B	1
N5	IP_{13} (R1)	IP_B	2
N6	IP_{31} (R3)	IP_B	2
Маршрут по умолчанию	IP_{31} (R3)	IP_B	–

Конечные узлы в еще большей степени, чем маршрутизаторы, пользуются приемом маршрутизации по умолчанию. Хотя они также в общем случае имеют в своем распоряжении таблицу маршрутизации, ее объем обычно незначителен, что объясняется периферийным расположением всех конечных узлов. Конечный узел часто вообще работает без таблицы маршрутизации, имея только сведения об адресе маршрутизатора по умолчанию. При наличии одного маршрутизатора в локальной сети этот вариант — единственно возможный для всех конечных узлов. Но даже при наличии нескольких маршрутизаторов в локальной сети, когда перед конечным узлом стоит проблема их выбора, часто в компьютерах для повышения производительности прибегают к заданию маршрута по умолчанию.

Рассмотрим таблицу маршрутизации другого конечного узла составной сети — узла А (табл. 18.3). Компактный вид таблицы маршрутизации узла А отражает тот факт, что все пакеты, направляемые из узла А, либо не выходят за пределы сети N12, либо проходят через порт 1 маршрутизатора 17. Этот маршрутизатор и определен в таблице маршрутизации в качестве маршрутизатора по умолчанию.

Таблица 18.3. Таблица маршрутизации конечного узла А

Номер сети назначения	Сетевой адрес следующего маршрутизатора	Сетевой адрес выходного порта	Расстояние до сети назначения
N12	—	IPA	0
Маршрут по умолчанию	$IP_{17,1}$ (R17)	IPA	—

Еще одним отличием работы маршрутизатора и конечного узла является способ построения таблицы маршрутизации. Если маршрутизаторы, как правило, автоматически создают таблицы маршрутизации, обмениваясь служебной информацией, то для конечных узлов таблицы маршрутизации часто создаются вручную администраторами и хранятся в виде постоянных файлов на дисках.

Просмотр таблиц маршрутизации без масок

Ниже приведен алгоритм просмотра таблицы маршрутизации протоколом IP, установленным на маршрутизаторе. При его описании мы будем использовать табл. 18.1 и рис. 18.2.

1. Пусть на один из интерфейсов маршрутизатора поступает пакет. Протокол IP извлекает из пакета IP-адрес назначения (предположим, адрес назначения IP_B).
2. Выполняется *первая фаза* просмотра таблицы — *поиск конкретного маршрута к узлу*. IP-адрес (целиком) последовательно строка за строкой сравнивается с содержимым поля адреса назначения таблицы маршрутизации. Если произошло совпадение (как в табл. 18.1), то из соответствующей строки извлекаются адрес следующего маршрутизатора (IP_{21}) и идентификатор выходного интерфейса (IP_{41}). На этом просмотр таблицы заканчивается.
3. Предположим теперь, что в таблице нет строки с адресом назначения IP_B, а значит, совпадения не произошло. В таком случае протокол IP переходит ко *второй фазе* просмотра — *поиску маршрута к сети назначения*. Из IP-адреса выделя-

ется номер сети (в нашем примере из адреса IP_B выделяется номер сети N3), и таблица снова просматривается на предмет совпадения номера сети в какой-либо строке с номером сети из пакета. При совпадении (а в нашем примере оно произошло) из соответствующей строки таблицы извлекаются адрес следующего маршрутизатора (IP_{12}) и идентификатор выходного интерфейса (IP_{41}). Просмотр таблицы на этом завершается.

4. Наконец, предположим, что адрес назначения в пакете был таков, что совпадения не произошло ни в первой, ни во второй фазах просмотра. В таком случае протокол IP либо выбирает маршрут по умолчанию (и пакет направляется по адресу IP_{51}), либо, если маршрут по умолчанию отсутствует, отбрасывает пакет[1]. Просмотр таблицы на этом заканчивается.

ВНИМАНИЕ

Последовательность фаз в данном алгоритме строго определена, в то время как последовательность просмотра или, что одно и то же, порядок расположения строк в таблице, включая запись о маршруте по умолчанию, никак не сказывается на результате.

Примеры таблиц маршрутизации разных форматов

Структура реальных таблиц маршрутизации стека TCP/IP в целом соответствует упрощенной структуре таблиц, рассмотренных выше. Отметим, однако, что вид таблицы IP-маршрутизации зависит от конкретной реализации стека TCP/IP. Приведем пример нескольких вариантов таблицы маршрутизации, с которыми мог бы работать маршрутизатор R1 в сети, представленной на рис. 18.3.

Начнем с «придуманного», предельно упрощенного варианта таблицы маршрутизации (табл. 18.4). Здесь имеются три маршрута к сетям (записи 56.0.0.0, 116.0.0.0 и 129.13.0.0), две записи о непосредственно подсоединенных сетях (129.13.0.0 и 213.34.12.0), а также запись о маршруте по умолчанию.

Таблица 18.4. Упрощенная таблица маршрутизации маршрутизатора R1

Адрес сети назначения	Адрес следующего маршрутизатора	Адрес выходного интерфейса	Расстояние до сети назначения
56.0.0.0	213.34.12.4	213.34.12.3	15
116.0.0.0	213.34.12.4	213.34.12.3	13
129.13.0.0	198.21.17.6	198.21.17.5	2
129.13.0.0	198.21.17.5	198.21.17.5	1
213.34.12.0	213.34.12.3	213.34.12.3	1
Маршрут по умолчанию	198.21.17.7	198.21.17.5	–

[1] Стандарты технологии TCP/IP не требуют, чтобы в таблице маршрутизации непременно содержались маршруты для всех пакетов, которые могут прийти на интерфейсы маршрутизатора, более того, в таблице может отсутствовать маршрут по умолчанию.

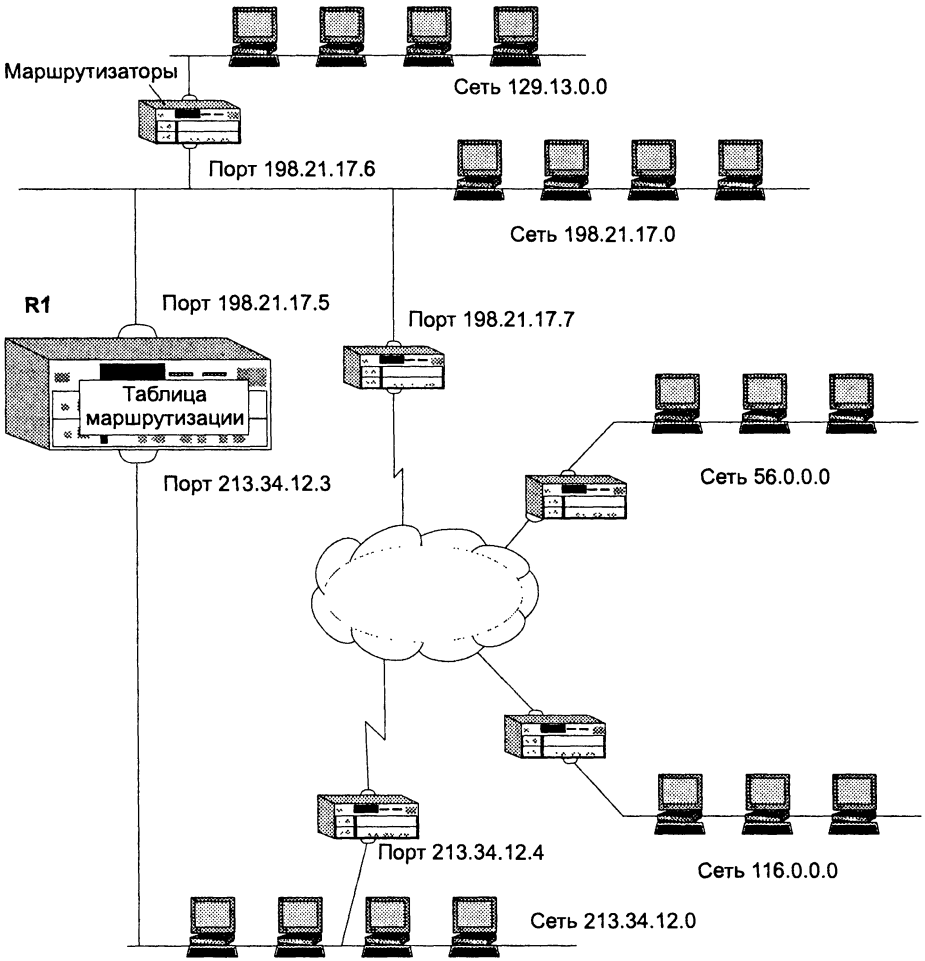


Рис. 18.3. Пример маршрутизируемой сети

Более сложный вид имеют таблицы, которые генерируются в промышленно выпускаемом сетевом оборудовании.

Если представить, что в качестве маршрутизатора R1 в данной сети работает штатный программный маршрутизатор операционной системы Microsoft Windows 2000, то его таблица маршрутизации могла бы выглядеть так, как табл. 18.5.

Если на месте маршрутизатора R1 установить один из популярных *аппаратных* маршрутизаторов, то его таблица маршрутизации для этой же сети может выглядеть совсем иначе (табл. 18.6).

И, наконец, табл. 18.7 представляет собой таблицу маршрутизации для того же маршрутизатора R1, реализованного в виде программного маршрутизатора одной из версий операционной системы Unix.

Таблица 18.5. Таблица программного маршрутизатора ОС Windows 2000

Сетевой адрес	Маска	Адрес шлюза	Интерфейс	Метрика
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
0.0.0.0	0.0.0.0	198.21.17.7	198.21.17.5	1
56.0.0.0	255.0.0.0	213.34.12.4	213.34.12.3	15
116.0.0.0	255.0.0.0	213.34.12.4	213.34.12.3	13
129.13.0.0	255.255.0.0	198.21.17.6	198.21.17.5	2
198.21.17.0	255.255.255.0	198.21.17.5	198.21.17.5	1
198.21.17.5	255.255.255.255	127.0.0.1	127.0.0.1	1
198.21.17.255	255.255.255.255	198.21.17.5	198.21.17.5	1
213.34.12.0	255.255.255.0	213.34.12.3	213.34.12.3	1
213.34.12.3	255.255.255.255	127.0.0.1	127.0.0.1	1
213.34.12.255	255.255.255.255	213.34.12.3	213.34.12.3	1
224.0.0.0	224.0.0.0	198.21.17.6	198.21.17.6	1
224.0.0.0	224.0.0.0	213.34.12.3	213.34.12.3	1
255.255.255.255	255.255.255.255	198.21.17.6	198.21.17.6	1

Таблица 18.6. Таблица маршрутизации аппаратного маршрутизатора

Адрес назначения	Маска	Шлюз	Метрика	Статус	TTL	Источник
198.21.17.0	255.255.255.0	198.21.17.5	0	Up	–	Подключена
213.34.12.0	255.255.255.0	213.34.12.3	0	Up	–	Подключена
56.0.0.0	255.0.0.0	213.34.12.4	14	Up	–	Статическая
116.0.0.0	255.0.0.0	213.34.12.4	12	Up	–	Статическая
129.13.0.0	255.255.0.0	198.21.17.6	1	Up	160	RIP

Таблица 18.7. Таблица маршрутизации Unix-маршрутизатора

Адрес назначения	Шлюз	Флаги	Число ссылок	Загрузка	Интерфейс
127.0.0.0	127.0.0.1	UH	1	154	lo0
Маршрут по умолчанию	198.21.17.7	UG	5	43270	le0

Адрес назначения	Шлюз	Флаги	Число ссылок	Загрузка	Интерфейс
198.21.17.0	198.21.17.5	U	35	246876	le0
213.34.12.0	213.34.12.3	U	44	132435	le1
129.13.0.0	198.21.1.7.6	UG	6	16450	le0
56.0.0.0	213.34.12.4	UG	12	5764	le1
116.0.0.0	213.34.12.4	UG	21	23544	le1

ПРИМЕЧАНИЕ

Заметим, что поскольку между структурой сети и таблицей маршрутизации нет однозначного соответствия, то и для каждого из приведенных вариантов таблицы можно предложить свои «подварианты», отличающиеся выбранным маршрутом к той или иной сети. В данном случае внимание концентрируется на существенных различиях в форме представления маршрутной информации разными реализациями маршрутизаторов.

Несмотря на достаточно заметные внешние различия, во всех трех «реальных» таблицах присутствуют все рассмотренные выше ключевые данные, которые необходимы протоколу IP для маршрутизации пакетов.

К таким данным, во-первых, относятся *адреса сети назначения* (столбцы адреса назначения в аппаратном маршрутизаторе и маршрутизаторе Unix или столбец сетевой адрес в маршрутизаторе ОС Windows 2000).

Вторым обязательным полем таблицы маршрутизации является *адрес следующего маршрутизатора* (столбцы «Шлюз» в аппаратном маршрутизаторе и маршрутизаторе Unix или столбец «Адрес шлюза» в маршрутизаторе ОС Windows 2000).

Третий ключевой параметр — *адрес порта*, на который нужно направить пакет, в некоторых таблицах указывается прямо (столбец «Интерфейс» в таблице маршрутизатора ОС Windows 2000), а в некоторых — косвенно. Так, в таблице Unix-маршрутизатора вместо адреса порта задается его условное наименование — le0 для порта с адресом 198.21.17.5, le1 для порта с адресом 213.34.12.3 и lo0 для внутреннего порта с адресом 127.0.0.1. В аппаратном маршрутизаторе поле, обозначающее выходной порт в какой-либо форме, вообще отсутствует. Это объясняется тем, что адрес выходного порта всегда можно косвенно определить по адресу следующего маршрутизатора. Например, определим по табл. 18.6 адрес выходного порта для сети 56.0.0.0. Из таблицы следует, что следующим маршрутизатором для этой сети будет маршрутизатор с адресом 213.34.12.4. Адрес следующего маршрутизатора должен принадлежать одной из непосредственно присоединенных к маршрутизатору сетей, и в данном случае это сеть 213.34.12.0. Маршрутизатор имеет порт, присоединенный к этой сети, и адрес этого порта 213.34.12.3 мы находим в столбце «Шлюз» второй строки таблицы маршрутизации, которая описывает непосредственно присоединенную сеть 213.34.12.0. Для непосредст-

венно присоединенных сетей адресом следующего маршрутизатора всегда является адрес собственного порта маршрутизатора. Таким образом, для сети 56.0.0 адресом выходного порта является 213.34.12.3.

Стандартным решением сегодня является использование поля маски в каждой записи таблицы, как это сделано в таблицах маршрутизатора ОС Windows 2000 и аппаратного маршрутизатора (столбцы «Маска»). Обработка масок при принятии решения маршрутизаторами будет рассмотрена далее. Отсутствие поля маски говорит о том, что либо маршрутизатор рассчитан на работу только с тремя стандартными классами адресов, либо для всех записей используется одна и та же маска, что снижает гибкость маршрутизации.

Поскольку в таблице маршрутизации Unix-маршрутизатора каждая сеть назначения упомянута только один раз, а, значит, выбор маршрута отсутствует, то поле метрики является необязательным параметром. В остальных двух таблицах поле метрики используется только для указания на то, что сеть подключена непосредственно. Метрика 0 для аппаратного маршрутизатора или 1 для маршрутизатора ОС Windows 2000 говорит маршрутизатору, что эта сеть непосредственно подключена к его порту, а другое значение метрики соответствует удаленной сети. Выбор метрики для непосредственно подключенной сети (1 или 0) является произвольным, главное, чтобы метрика удаленной сети отсчитывалась с учетом этого выбранного начального значения. В Unix-маршрутизаторе используется поле признаков, где флаг G (Gateway — шлюз) отмечает удаленную сеть, а его отсутствие — непосредственно подключенную.

Признак непосредственно подключенной сети говорит маршрутизатору, что пакет уже достиг своей сети, поэтому протокол IP активизирует ARP-запрос относительно IP-адреса узла назначения, а не следующего маршрутизатора.

Однако существуют ситуации, когда маршрутизатор должен обязательно хранить значение метрики для записи о каждой удаленной сети. Эти ситуации возникают, когда записи в таблице маршрутизации являются результатом работы некоторых протоколов маршрутизации, например протокола RIP. В таких протоколах новая информация о какой-либо удаленной сети сравнивается с информацией, содержащейся в таблице в данный момент, и если значение новой метрики лучше текущей, то новая запись вытесняет имеющуюся. В таблице Unix-маршрутизатора поле метрики отсутствует, и это значит, что он не использует протокол RIP.

Флаги записей присутствуют только в таблице Unix-маршрутизатора.

- U — маршрут активен и работоспособен. Аналогичный смысл имеет поле статуса в аппаратном маршрутизаторе.
- H — признак специфического маршрута к определенному хосту.
- G — означает, что маршрут пакета проходит через промежуточный маршрутизатор (шлюз). Отсутствие этого флага отмечает непосредственно подключенную сеть.

- D — означает, что маршрут получен из перенаправленного сообщения протокола ICMP. Этот признак может присутствовать только в таблице маршрутизации *конечного узла*. Признак означает, что конечный узел в какой-то предыдущей передаче пакета выбрал не самый рациональный следующий маршрутизатор на пути к данной сети, и этот маршрутизатор с помощью протокола ICMP сообщил конечному узлу, что все последующие пакеты к данной сети нужно отправлять через другой маршрутизатор.

В таблице Unix-маршрутизатора используется еще два поля, имеющих справочное значение. Поле числа ссылок показывает, сколько раз на данный маршрут ссылались при продвижении пакетов. Поле загрузки отражает количество байтов, переданных по данному маршруту.

В записях таблиц аппаратного маршрутизатора также имеется два справочных поля. Поле **времени жизни записи** (TTL) в данном случае никак не связано со временем жизни пакета. Здесь оно показывает время, в течение которого значение данной записи еще действительно. **Поле источника** говорит об источнике появления записи в таблице маршрутизации.

Источники и типы записей в таблице маршрутизации

Практически для всех маршрутизаторов существуют *три* основных источника появления записи в таблице.

- Одним из источников записей в таблицу маршрутизации является **программное обеспечение стека TCP/IP**, которое при инициализации маршрутизатора автоматически заносит в таблицу несколько записей, в результате чего создается так называемая минимальная таблица маршрутизации. Программное обеспечение формирует записи о *непосредственно подключенных сетях* и маршрутах по умолчанию, информация о которых появляется в стеке при ручном конфигурировании интерфейсов компьютера или маршрутизатора. К таким записям в приведенных примерах относятся записи о сетях 213.34.12.0 и 198.21.17.0, а также запись о маршруте по умолчанию в Unix-маршрутизаторе и запись 0.0.0.0 в маршрутизаторе ОС Windows 2000. Программное обеспечение также автоматически заносит в таблицу маршрутизации записи *об адресах особого назначения*. В приведенных примерах таблица маршрутизатора ОС Windows 2000 содержит наиболее полный набор записей такого рода. Несколько записей в этой таблице связаны с особым адресом 127.0.0.0. Записи с адресом 224.0.0.0 требуются для обработки групповых адресов. Кроме того, в таблицу могут быть занесены адреса, предназначенные для обработки широковещательных рассылок (например, записи 8 и 11 содержат адрес отправки широковещательного сообщения в соответствующих подсетях, а последняя запись в таблице — адрес ограниченной широковещательной рассылки). Заметим, что в некоторых таблицах записи об особых адресах вообще отсутствуют.

- Еще одним источником записей в таблице является **администратор**, непосредственно формирующий записи с помощью некоторой системной утилиты, например программы route, имеющейся в операционных системах Unix и Windows 2000. В аппаратных маршрутизаторах также всегда имеется команда для ручного задания записей таблицы маршрутизации. Заданные вручную записи всегда являются *статическими*, то есть они не имеют срока жизни. Эти записи могут быть как постоянными, то есть сохраняющимися при перезагрузке маршрутизатора, так и временными, хранящимися в таблице только до выключения устройства. Часто администратор вручную заносит запись о маршруте по умолчанию. Таким же образом в таблицу маршрутизации может быть внесена запись о специфическом для узла маршруте.
- И, наконец, третьим источником записей могут быть **протоколы маршрутизации**, такие как RIP или OSPF. Такие записи всегда являются *динамическими*, то есть имеют ограниченный срок жизни.

Программные маршрутизаторы Windows 2000 и Unix не показывают источник появления той или иной записи в таблице, а аппаратный маршрутизатор использует для этой цели поле источника. В приведенном в табл. 18.6 примере первые две записи созданы программным обеспечением стека на основании данных о конфигурации портов маршрутизатора — это показывает признак «Подключена». Следующие две записи обозначены как статические — это означает, что их ввел вручную администратор. Последняя запись является следствием работы протокола RIP, поэтому в ее поле «TTL» имеется значение 160.

Пример IP-маршрутизации без масок

Рассмотрим на примере IP-сети (рис. 18.4) процесс продвижения пакета в составной сети. При этом будем считать, что все узлы сети, рассматриваемой в примере, имеют адреса, основанные на классах. Особое внимание будет уделено взаимодействию протокола IP с протоколами разрешения адресов ARP и DNS.

Итак, пусть пользователю компьютера cit.mgu.com, находящегося в сети Ethernet1, необходимо установить связь с FTP-сервером. Пользователю известно символьное имя сервера unix.mgu.com. Поэтому он набирает на клавиатуре команду обращения к FTP-серверу по имени:

```
> ftp unix.mgu.com
```

Эта команда включает обязательные три этапа.

1. Передача от клиента DNS-запроса для определения IP-адреса узла назначения.
2. Передача от сервера DNS-ответа.
3. Передача пакета от FTP-клиента к FTP-серверу.

Давайте рассмотрим каждый из них подробно.

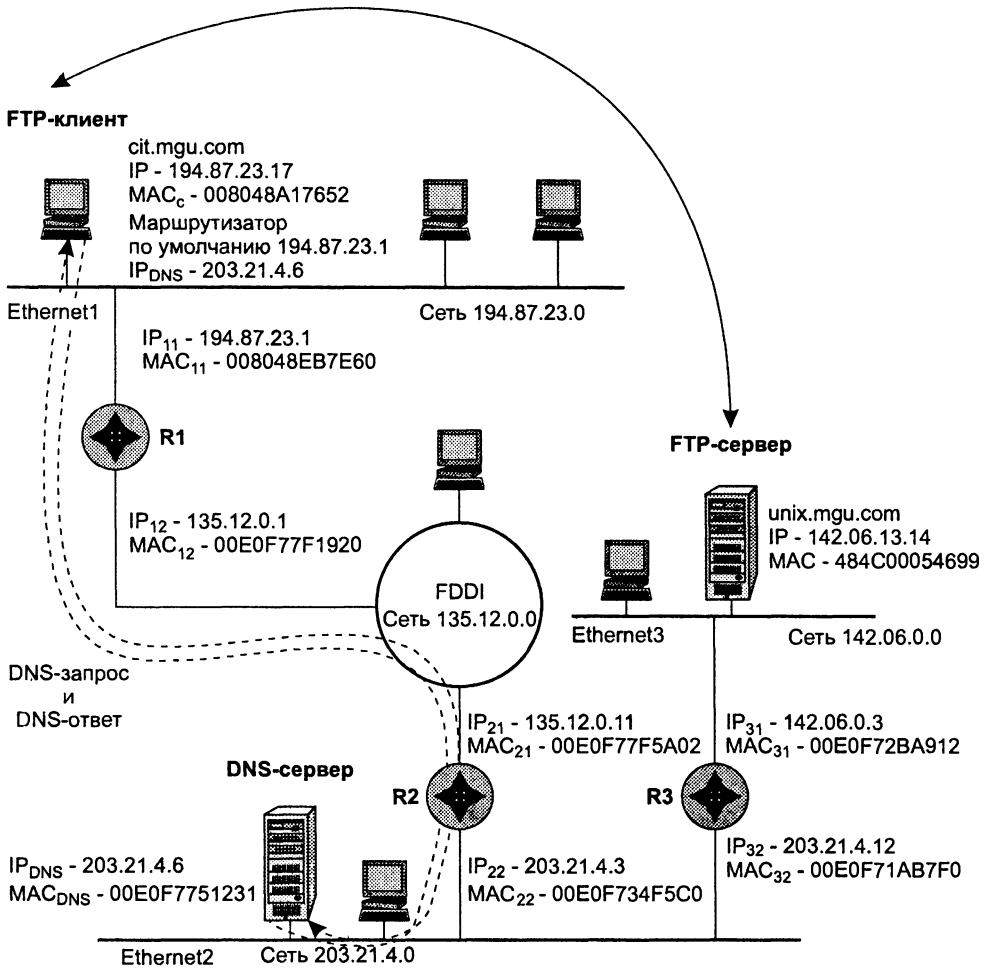


Рис. 18.4. Пример IP-маршрутизации

Передача DNS-запроса

1. FTP-клиент передает запрос к работающей на этом же компьютере клиентской части протокола DNS, который в свою очередь формирует к DNS-серверу запрос, интерпретируемый примерно так: «Какой IP-адрес соответствует символьному имени unix.mgu.com?». Запрос упаковывается в UDP-дейтаграмму, затем в IP-пакет (рис. 18.5). В заголовке пакета в качестве адреса назначения указывается IP-адрес DNS-сервера — 203.21.4.6. Этот адрес известен программному обеспечению клиентского компьютера, так как он входит в число его конфигурационных параметров.
2. Прежде чем поместить IP-пакет в кадр Ethernet, необходимо выяснить, надо ли этот пакет маршрутизировать через составную сеть, или он адресован узлу, находящемуся в той же сети, что и отправитель. Для этого модуль IP сравнивает номера сетей в адресах отправителя и получателя, то есть

194.87.23.17 и 203.21.4.6. В результате сравнения выясняется, что пакет направляется в другую сеть, следовательно, его необходимо передать маршрутизатору. Поскольку в сети Ethernet1 имеется только один маршрутизатор (R1), все конечные узлы этой сети вместо полноценной таблицы маршрутизации пользуются адресом 194.87.23.1 этого маршрутизатора в качестве маршрутизатора по умолчанию.

Заголовок IP		Заголовок UDP	DNS-запрос
IP-адрес отправителя	IP-адрес получателя		
194.87.23.17	203.21.4.6		

Рис. 18.5. IP-пакет с DNS-запросом

3. Чтобы сеть Ethernet1 смогла доставить пакет маршрутизатору R1, этот пакет необходимо поместить в поле данных кадра Ethernet и снабдить его MAC-адресом. Эта проблема решается с помощью протокола ARP. Пусть, например, нужного адреса в ARP-таблице нет, тогда клиентский узел отправляет широковещательный ARP-запрос: «Какой MAC-адрес соответствует IP-адресу 194.87.23.1?» Этот запрос получают все узлы сети Ethernet1, но ответ приходит только от интерфейса 1 маршрутизатора R1: «Я имею IP-адрес 194.87.23.1 и мой MAC-адрес 008048EB7E60». После получения этой информации компьютер cit.mgu.com отправляет по локальной сети пакет, упакованный в кадр Ethernet и имеющий поля, показанные на рис. 18.6.

Заголовок Ethernet		Заголовок IP		Заголовок UDP	DNS-запрос
MAC-адрес отправителя	MAC-адрес получателя	IP-адрес отправителя	IP-адрес получателя		
MAC_c - 00804A17652	MAC_{11} - 008048EB7E60	194.87.23.17	203.21.4.6		unix.mgu.com?

Рис. 18.6. Кадр Ethernet с DNS-запросом, отправленный с клиентского компьютера

4. Кадр принимается интерфейсом 1 маршрутизатора R1. Протокол Ethernet извлекает из этого кадра IP-пакет и передает его протоколу IP. Протокол IP находит в заголовке пакета адрес назначения 203.21.4.6 и просматривает записи своей таблицы маршрутизации. Пусть маршрутизатор R1 имеет в своей таблице маршрутизации следующую запись:

   203.21.4.0 135.12.0.11 135.12.0.1

   Эта запись говорит о том, что пакеты для сети 203.21.4.0 нужно передавать маршрутизатору 135.12.0.11, находящемуся в сети, подключенной к интерфейсу 135.12.0.1 маршрутизатора R1. Маршрутизатор R1 просматривает параметры интерфейса 135.12.0.1 и находит, что к нему подключена сеть FDDI. Так как максимально возможная длина дейтаграммы, которую можно поместить в поле данных кадра FDDI, больше, чем в поле данных кадра Ethernet, фрагментация IP-пакета не требуется. Поэтому маршрутизатор R1 формирует кадр формата FDDI.

5. На этом этапе модуль IP маршрутизатора R1 должен определить MAC-адрес следующего маршрутизатора по известному IP-адресу 135.12.0.11. Для этого он обращается к протоколу ARP. Допустим, в этот раз в ARP-таблице присутствует запись о нужном адресе:

   135.12.0.11 — 00E0F77F5A02

   Теперь, зная MAC-адрес маршрутизатора R2 (00E0F77F5A02), маршрутизатор R1 отсылает в сеть FDDI кадр (рис. 18.7).

Заголовок FDDI		Заголовок IP		Заголовок UDP	DNS-запрос
MAC-адрес отправителя	MAC-адрес получателя	IP-адрес отправителя	IP-адрес получателя		
MAC_{12} - 00E0F77F1920	MAC_{21} - 00E0F77F5A02	194.87.23.17	203.21.4.6		unix.mgu.com?

Рис. 18.7. Кадр Ethernet с DNS-запросом, отправленный с маршрутизатора R1 маршрутизатору R2

6. Аналогично действует модуль IP на маршрутизаторе R2. Получив FDDI-кадр, он отбрасывает его заголовок, а из заголовка IP извлекает IP-адрес назначения и просматривает свою таблицу маршрутизации. Там он обнаруживает, что сеть назначения является непосредственно присоединенной к его второму интерфейсу. Поэтому он посылает ARP-запрос по сети Ethernet2: «Какой MAC-адрес соответствует IP-адресу конечного узла 203.21.4.6?» Получив ответ о MAC-адресе DNS-сервера — 00E0F7751231, маршрутизатор R2 отправляет в сеть Ethernet2 кадр (рис. 18.8).

Заголовок Ethernet		Заголовок IP		Заголовок UDP	DNS-запрос
MAC-адрес отправителя	MAC-адрес получателя	IP-адрес отправителя	IP-адрес получателя		
MAC_{21} - 00E0F734F5C0	MAC_{DNS} - 00E0F7751231	194.87.23.17	203.21.4.6		unix.mgu.com?

Рис. 18.8. Кадр Ethernet с DNS-запросом, отправленный с маршрутизатора R2

7. Сетевой адаптер DNS-сервера захватывает кадр Ethernet, обнаруживает совпадение MAC-адреса получателя, содержащегося в заголовке, со своим собственным адресом и направляет его модулю IP. После анализа полей заголовка IP из пакета извлекаются данные вышележащих протоколов. DNS-запрос передается программному модулю DNS-сервера. DNS-сервер просматривает свои таблицы, возможно, обращается к другим DNS-серверам и в результате формирует ответ, смысл которого состоит в следующем: *Символьному имени* unix.mgu.com соответствует IP-адрес 142.06.13.14.

ПРИМЕЧАНИЕ

Заметим, что во время всего путешествия пакета по составной сети от клиентского компьютера до DNS-сервера адреса получателя и отправителя в полях заголовка IP-пакета не изменились. Зато в заголовке каждого нового кадра, который переносил пакет от одного маршрутизатора к другому, аппаратные адреса изменялись на каждом отрезке пути.

Передача DNS-ответа

1. Стек TCP/IP, установленный на DNS-сервере, упаковывает DNS-ответ в UDP-дейтаграмму, затем — в IP-пакет (IP-адрес получателя ему известен из DNS-запроса). Протокол IP определяет, что пакет необходимо маршрутизировать.
2. Протокол IP просматривает таблицу маршрутизации, в результате определяет IP-адрес следующего маршрутизатора IP_{22} — 203.21.4.3.
3. Протокол ARP определяет MAC-адрес интерфейса маршрутизатора — 00E0F734F5C0.
4. IP-пакет размещается в поле данных кадра Ethernet и отправляется в сеть Ethernet2.
5. Маршрутизатор R2 получает кадр, выполняет шаги 2 и 3 и отправляет FDDI-кадр маршрутизатору R1.
6. Маршрутизатор R1 определяет по таблице маршрутизации, что поступивший пакет адресован в сеть, непосредственно подсоединенную к его интерфейсу. Поэтому IP обращается к протоколу ARP, чтобы получить MAC-адрес узла назначения, а не маршрутизатора.
7. В сеть Ethernet1 отправляется адресованный FTP-клиенту кадр (рис. 18.9).

Заголовок Ethernet		Заголовок IP		Заголовок UDP	DNS-ответ
MAC-адрес отправителя	MAC-адрес получателя	IP-адрес отправителя	IP-адрес получателя		
MAC_{11} - 008048EB7E60	MAC_c - 008048A17652	203.21.4.6	194.87.23.17		142.06.13.14

Рис. 18.9. Кадр Ethernet с DNS-запросом, отправленный с маршрутизатора R2

8. FTP-клиент извлекает из кадра DNS-ответ, теперь он может продолжить выполнение команды, в которой адрес FTP-сервера указан в виде IP-адреса:

```
> ftp 142.06.13.14
```

Передача пакета от FTP-клиента к FTP-серверу

Этот этап полностью аналогичен описанным выше шагам, относящимся к передаче по сети DNS-запроса и DNS-ответа. Однако для читателя будет весьма полезно описать этот процесс, обращая особое внимание на значения адресных полей заголовков кадров и заголовка вложенного IP-пакета.

Маршрутизация с использованием масок

Список ключевых слов: маска, демилитаризованная зона, технология бесклассовой междоменной маршрутизации, префикс адреса, локализация адресов, подсеть, специфический маршрут, маски переменной длины, агрегирование адресов, перекрытие адресных пространств, CIDR.

Алгоритм маршрутизации усложняется, когда в систему адресации узлов вносятся дополнительные элементы — маски. В чем же причины отказа от хорошо себя зарекомендовавшего в течение многих лет метода адресации, основанного на классах? Основная из них — потребность в структуризации сетей в условиях дефицита нераспределенных номеров сетей.

Часто администраторы сетей испытывают неудобства, поскольку количества централизованно выделенных им номеров сетей недостаточно для того, чтобы структурировать сеть надлежащим образом, например, развести все слабо взаимодействующие компьютеры по разным сетям. В такой ситуации возможны два пути. Первый из них связан с получением от какого-либо центрального органа дополнительных номеров сетей. Второй способ, употребляющийся чаще, связан с использованием технологии масок, которая позволяет разделить одну сеть на несколько сетей.

Структуризация сети масками одинаковой длины

Допустим, администратор получил в свое распоряжение адрес класса B: 129.44.0.0. Он может организовать сеть с большим числом узлов, номера которых брать из диапазона 0.0.0.1–0.0.255.254. Всего получается $2^{16} - 2$ адреса (с учетом того, что адреса из одних нулей и одних единиц имеют специальное назначение и не годятся для адресации узлов). Однако ему не нужна одна большая неструктурированная сеть, производственная необходимость диктует администратору другое решение, в соответствии с которым сеть должна быть разделена на три отдельных подсети, при этом трафик в каждой подсети должен быть надежно локализован. Это позволит легче диагностировать сеть и проводить в каждой из подсетей особую политику безопасности. (Заметим, что разделение большой сети с помощью масок имеет еще одно преимущество — оно позволяет скрыть внутреннюю структуру сети предприятия от внешнего наблюдения и тем самым повысить ее безопасность.)

На рис. 18.10 показано разделение всего полученного администратором адресного диапазона на 4 равные части — каждая по 2^{14} адресов. При этом число разрядов, доступное для нумерации узлов, *уменьшилось* на два бита, а префикс (номер) каждой из четырех сетей стал *длиннее* на два бита. Следовательно, каждый из четырех диапазонов можно записать в виде IP-адреса с маской, состоящей из 18 единиц, или в десятичной нотации — 255.255.192.0.

129.44.0.0/18 (10000001 00101100 **00**000000 00000000)

129.44.64.0/18 (10000001 00101100 **01**000000 00000000)

129.44.128.0/18 (10000001 00101100 **10**000000 00000000)

129.44.192.0/18 (10000001 00101100 **11**000000 00000000)

Из приведенных записей видно, что администратор получает возможность использовать для нумерации подсетей два дополнительных бита (выделенных жирным шрифтом). Именно это позволяет ему сделать из одной централизованно выделенной сети четыре, в данном примере это 129.44.0.0/18, 129.44.64.0/18, 129.44.128.0/18, 129.44.192.0/18.

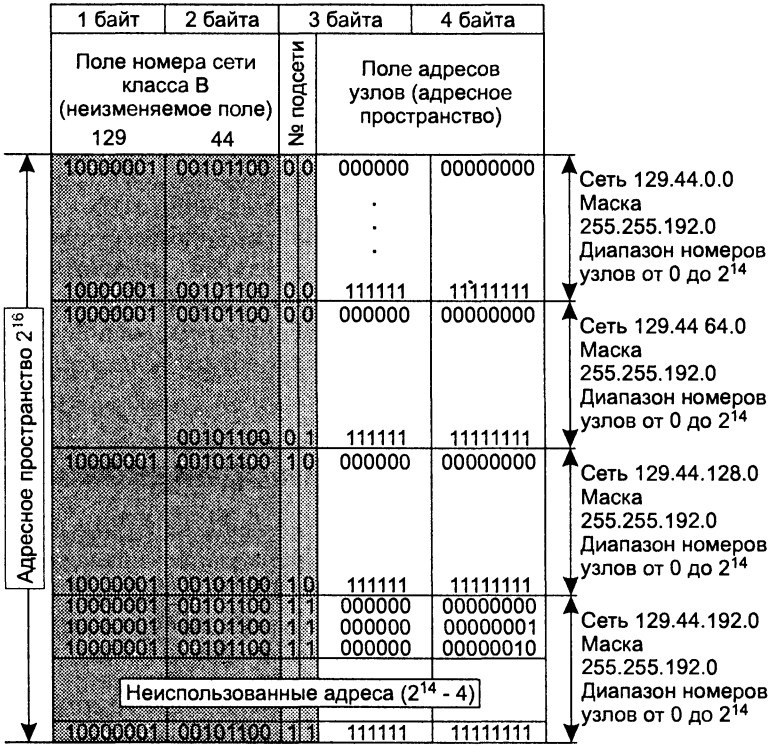


Рис. 18.10. Разделение адресного пространства сети класса B 129.44.0.0 на четыре равные части

ПРИМЕЧАНИЕ

Некоторые программные и аппаратные маршрутизаторы, следуя устаревшим рекомендациям RFC 950, не поддерживают номера подсетей, которые состоят либо только из одних нулей, либо только из одних единиц. Например, для такого типа оборудования номер сети 129.44.0.0 с маской 255.255.192.0, использованной в нашем примере, окажется недопустимым, поскольку в этом случае разряды в поле номера подсети имеют значение 00. По аналогичным соображениям недопустимым может оказаться номер сети 129.44.192.0 с тем же значением маски. Здесь номер подсети состоит только из единиц. Однако современные маршрутизаторы, поддерживающие рекомендации RFC 1878, свободны от этих ограничений.

Пример сети, построенной путем деления на 4 сети равного размера, показан на рис. 18.11. Весь трафик во внутреннюю сеть 129.44.0.0, направляемый из внешней сети, поступает через маршрутизатор R1. В целях структуризации информационных потоков во внутренней сети установлен дополнительный маршрутизатор R2. Каждая из вновь образованных сетей 129.44.0.0/18, 129.44.64.0/18, 129.44.128.0/18 и 129.44.192.0/18 подключена к соответственно сконфигурированным портам внутреннего маршрутизатора R2.

ПРИМЕЧАНИЕ

В одной из этих сетей (129.44.192.0/18), выделенной для создания соединения между внешним и внутренним маршрутизаторами, для адресации узлов были использованы всего два адреса — 129.44.192.1 (порт маршрутизатора R2) и 129.44.192.2 (порт маршрутизатора R1). Огромное число узлов в этой подсети не используются. Такой пример выбран исключительно в учебных целях, чтобы показать неэффективность сетей равного размера.

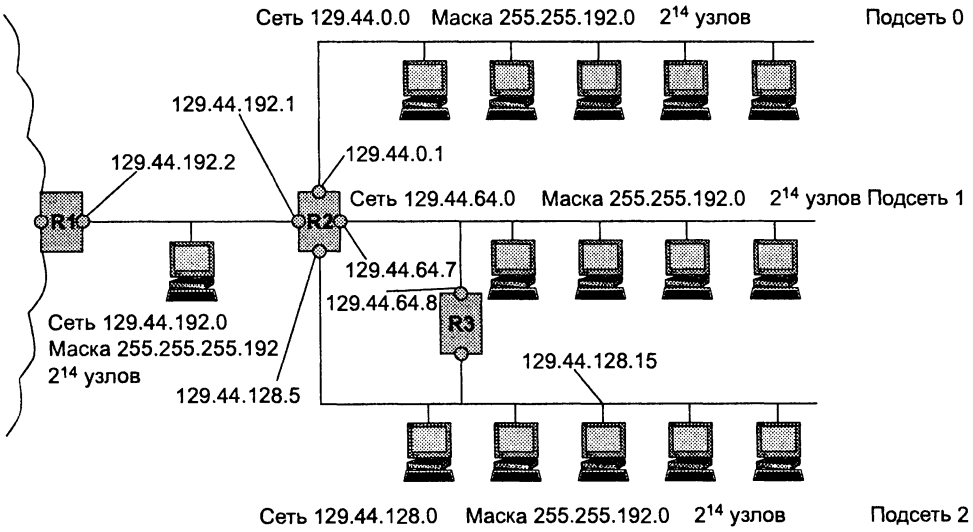


Рис. 18.11. Маршрутизация с использованием масок одинаковой длины

Извне сеть по-прежнему выглядит, как единая сеть класса B. Однако поступающий в сеть общий трафик разделяется локальным маршрутизатором R2 между четырьмя сетями. В условиях, когда механизм классов не действует, маршрутизатор должен иметь другое средство, которое позволило бы ему определять, какая часть 32-разрядного числа, помещенного в поле «Адрес назначения», является номером сети. Именно этой цели служит дополнительное поле маски, включенное в таблицу маршрутизации (табл. 18.8).

Таблица 18.8. Таблица маршрутизатора R2 в сети с масками одинаковой длины

Адрес назначения	Маска	Адрес следующего маршрутизатора	Адрес порта	Расстояние
129.44.0.0	255.255.192.0	129.44.0.1	129.44.192.2	Подключена
129.44.64.0	255.255.192.0	129.44.64.7	129.44.64.7	Подключена
129.44.128.0	255.255.192.0	129.44.128.5	129.44.128.5	Подключена
129.44.192.0	255.255.192.0	129.44.192.1	129.44.192.1	Подключена
0.0.0.0	0.0.0.0	129.44.192.2	129.44.192.1	–
129.44.128.15	255.255.255.255	129.44.64.8	129.44.64.7	–

Первые четыре записи в таблице соответствуют внутренним подсетям, непосредственно подключенным к портам маршрутизатора R2.

Запись 0.0.0.0 с маской 0.0.0.0 соответствует маршруту по умолчанию.

Последняя запись определяет специфический маршрут к узлу 129.44.128.15. В тех строках таблицы, в которых в качестве адреса назначения указан полный IP-адрес узла, маска имеет значение 255.255.255.255. В отличие от всех других узлов сети 129.44.128.0, к которым пакеты поступают с интерфейса 129.44.128.5 маршрутизатора R2, к данному узлу они будит приходить через маршрутизатор R3.

Просмотр таблиц маршрутизации с учетом масок

Алгоритм просмотра таблиц маршрутизации, содержащих маски, имеет много общего с описанным выше алгоритмом просмотра таблиц, не содержащих маски. Однако в нем имеются и существенные изменения.

1. Поиск следующего маршрутизатора для вновь поступившего IP-пакета протокол начинает с того, что *извлекает из пакета адрес назначения* (обозначим его IP_D). Затем протокол IP приступает процедуре просмотра таблицы маршрутизации, также состоящей из двух фаз, как и процедура просмотра таблицы, в которой столбец маски отсутствует.
2. *Первая фаза* состоит в *поиске специфического маршрута* для адреса IP_D. С этой целью из каждой записи таблицы, в которой маска имеет значение 255.255.255.255, извлекается адрес назначения и сравнивается с адресом из пакета IP_D. Если в какой-либо строке совпадение произошло, то адрес следующего маршрутизатора для данного пакета берется из данной строки.
3. *Вторая фаза* выполняется только в том случае, если во время первой фазы не произошло совпадения адресов. Она состоит в *поиске неспецифического маршрута*, общего для группы узлов, к которой относится и пакет с адресом IP_D. Для этого IP заново просматривает таблицу маршрутизации, причем с *каждой* записью производятся следующие действия:
   1) маска (обозначим ее M), содержащаяся в данной записи, «накладывается» на IP-адрес узла назначения IP_D, извлеченный из пакета: IP_D AND M;
   2) полученное в результате число сравнивается со значением, которое помещено в поле адреса назначения той же записи таблицы маршрутизации;
   3) если происходит совпадение, протокол IP соответствующим образом *отмечает эту строку*;
   4) если просмотрены не все строки, то IP-протокол аналогичным образом просматривает следующую строку, если все (включая строку о маршруте по умолчанию), то просмотр записей заканчивается и происходит переход к следующему шагу.
4. После просмотра всей таблицы маршрутизатор выполняет одно их трех действий:
   1) если не произошло ни одного совпадения и маршрут по умолчанию отсутствует, то пакет отбрасывается;

2) если произошло одно совпадение, то пакет отправляется по маршруту, указанному в строке с совпавшим адресом;
3) если произошло несколько совпадений, то все помеченные строки сравниваются и выбирается маршрут из той строки, в которой количество совпавших двоичных разрядов наибольшее (другими словами, в ситуации, когда адрес назначения пакета принадлежит сразу нескольким подсетям, маршрутизатор использует наиболее специфический маршрут).

ПРИМЕЧАНИЕ

Во многих таблицах маршрутизации запись с адресом 0.0.0.0 и маской 0.0.0.0 соответствует маршруту по умолчанию. Действительно, любой адрес в пришедшем пакете после наложения на него маски 0.0.0.0 даст адрес сети 0.0.0.0, что совпадает с адресом, указанным в записи. Поскольку маска 0.0.0.0 имеет нулевую длину, то этот маршрут считается самым неспецифическим и используется только при отсутствии совпадений с остальными записями из таблицы маршрутизации.

Проиллюстрируем, как маршрутизатор R2 (см. рис. 18.11) использует вышеописанный алгоритм для работы со своей таблицей маршрутизации (см. табл. 18.8). Пусть на маршрутизатор R2 поступает пакет с адресом назначения 129.44.78.200. Модуль IP, установленный на этом маршрутизаторе, прежде всего сравнит этот адрес с адресом 129.44.128.15, для которого определен специфический маршрут. Совпадения нет, поэтому модуль IP начинает последовательно обрабатывать все строки таблицы, накладывая маски и сравнивая результаты до тех пор, пока не найдет совпадения номера сети в адресе назначения и в строке таблицы. В результате определяется маршрут для пакета 129.44.78.200 — он должен быть отправлен на выходной порт маршрутизатора 129.44.64.7 в сеть 129.44.64.0, непосредственно подключенную к данному маршрутизатору.

Использование масок переменной длины

Во многих случаях более эффективным является разбиение сети на подсети разного размера. В частности, для подсети, которая связывает два маршрутизатора по двухточечной схеме, даже количество адресов сети класса C явно является избыточным.

На рис. 18.12 приведен другой пример распределения того же адресного пространства 129.44.0.0/16, что и в предыдущем примере. Здесь половина из имеющихся адресов (2^{15}) отведена для создания сети с адресом 129.44.0.0 и маской 255.255.128.0. Следующая порция адресов, составляющая четверть всего адресного пространства (2^{14}), назначена для сети 129.44.128.0 с маской 255.255.192.0.

Далее в пространстве адресов был «вырезан» небольшой фрагмент для создания сети, предназначенной для связывания внутреннего маршрутизатора R2 с внешним маршрутизатором R1. Для нумерации узлов в такой вырожденной сети достаточно отвести два двоичных разряда. Из четырех возможных комбинаций номеров узлов: 00, 01, 10 и 11 два номера имеют специальное назначение и не могут быть присвоены узлам, но оставшиеся два 10 и 01 позволяют адресовать порты маршрутизаторов. Поле номера узла в таком случае имеет два двоичных

разряда, маска в десятичной нотации имеет вид 255.255.255.252, а номер сети, как видно из рисунка, равен 129.44.192.0.

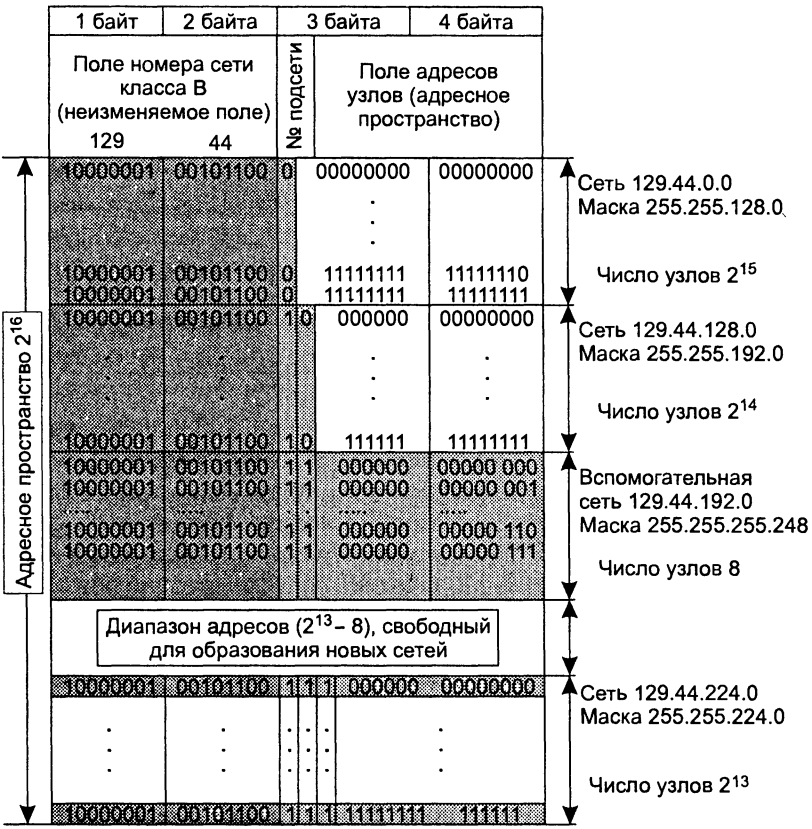


Рис. 18.12. Разделение адресного пространства сети класса B 129.44.0.0 на сети разного размера путем использования масок переменной длины

ПРИМЕЧАНИЕ

Глобальным связям между маршрутизаторами, соединенными по двухточечной схеме, не обязательно давать IP-адреса. Однако чаще всего такой вырожденной сети все же дают IP-адрес. Кроме прочего, это делается, например, для того, чтобы скрыть внутреннюю структуру сети и обращаться к ней по одному адресу входного порта маршрутизатора, в данном примере по адресу 129.44.192.1, применяя технику трансляции сетевых адресов (Network Address Translator, NAT[1]).

Оставшееся адресное пространство администратор может «нарезать» на разное количество сетей разного объема в зависимости от своих потребностей. Из оставшегося пула ($2^{14} - 4$) адресов администратор может образовать еще одну достаточно большую сеть с числом узлов 2^{13}. При этом свободными останутся почти столько же адресов ($2^{13} - 4$), которые также могут быть использованы для созда-

[1] О технологии NAT читайте в главе 20.

ния новых сетей. К примеру, из этого «остатка» можно образовать 31 сеть, каждая из которых равна размеру сети класса C, и к тому же еще несколько сетей меньшего размера. Ясно, что разбиение может быть другим, но в любом случае с помощью масок переменного размера администратор имеет больше возможностей рационально использовать все имеющиеся у него адреса.

На рис. 18.13 показан пример сети, структурированной с помощью масок переменной длины.

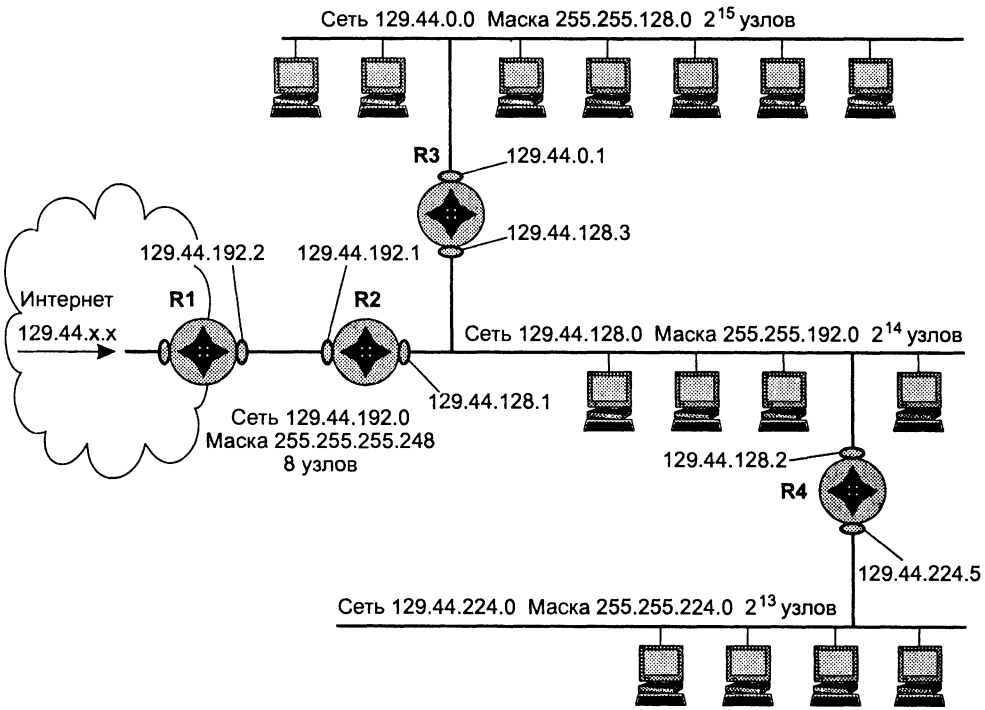


Рис. 18.13. Структуризация сети масками переменной длины

Давайте посмотрим, как маршрутизатор R2 обрабатывает поступающие на его интерфейсы пакеты (табл. 18.9).

Таблица 18.9. Таблица маршрутизатора R2 в сети с масками переменной длины

Адрес назначения	Маска	Адрес следующего маршрутизатора	Адрес порта	Расстояние
129.44.0.0	255.255.128.0	129.44.128.3	129.44.128.1	1
129.44.128.0	255.255.192.0	129.44.128.1	129.44.128.1	Подключена
129.44.192.0	255.255.255.248	129.44.192.1	129.44.192.1	Подключена
129.44.224.0	255.255.224.0	129.44.128.2	129.44.128.1	1
0.0.0.0	0.0.0.0	129.44.192.2	129.44.192.1	–

Пусть поступивший на R2 пакет имеет адрес назначения 129.44.192.15. Поскольку специфические маршруты в таблице отсутствуют, маршрутизатор переходит ко второй фазе — фазе последовательного анализа строк на предмет поиска совпадения с адресом назначения:

- (129.44.192.5) AND (255.255.128.0) = 129.44.128.0 — нет совпадения;
- (129.44.192.5) AND (255.255.192.0) = 129.44.192.0 — нет совпадения;
- (129.44.192.5) AND (255.255.255.248) = 129.44.192.0 — совпадение;
- (129.44.192.5) AND (255.255.224.0) = 129.44.192.0 — нет совпадения.

Таким образом, совпадение имеет место в одной строке. Пакет будет отправлен в непосредственно подключенную к данному маршрутизатору сеть на выходной интерфейс 129.44.192.1.

Если пакет с адресом 129.44.192.1 поступает из внешней сети, и маршрутизатор R1 не использует маски, пакет передается маршрутизатору R2, а потом снова возвращается в соединительную сеть. Очевидно, что такие передачи пакета не выглядят рациональными.

Маршрутизация будет более эффективной, если в таблице маршрутизации маршрутизатора R1 маршруты будут заданы масками переменной длины (табл. 18.10). Первая из приведенных двух записей говорит о том, что все пакеты, адреса которых начинаются с 129.44, должны быть переданы на маршрутизатор R2. Эта запись выполняет *агрегирование* адресов всех подсетей, созданных на базе одной сети 129.44.0.0. Вторая строка говорит о том, что среди всех возможных подсетей сети 129.44.0.0 есть одна (129.44.192.0/30), которой пакеты можно направлять непосредственно, а не через маршрутизатор R2.

Таблица 18.10. Фрагмент таблицы маршрутизатора R1

Адрес назначения	Маска	Адрес следующего маршрутизатора	Адрес порта	Расстояние
...				
129.44.0.0	255.255.0.0	129.44.192.1	129.44.191.2	2
129.44.192.0	255.255.255.192	129.44.192.2	129.44.192.2	Подключена
...				

ПРИМЕЧАНИЕ

В IP-пакетах при использовании механизма масок по-прежнему передается только IP-адрес назначения, а маска сети назначения не передается. Поэтому из IP-адреса пришедшего пакета невозможно выяснить, какая часть адреса относится к номеру сети, а какая — к номеру узла. Если маски во всех подсетях имеют один размер, то это не создает проблем. Если же для образования подсетей применяют маски переменной длины, то маршрутизатор должен каким-то образом узнавать, каким адресам сетей какие маски соответствуют. Для этого используются протоколы маршрутизации, переносящие между маршрутизаторами не только служебную информацию об адресах сетей, но и о масках, соответствующих этим номерам. К таким протоколам относятся протоколы RIPv2 и OSPF, а вот, например, протокол RIP маски не переносит и для масок переменной длины не подходит.

Перекрытие адресных пространств

Со сложностями использования масок администратор впервые сталкивается не тогда, когда начинает конфигурировать сетевые интерфейсы и создавать таблицы маршрутизации, а гораздо раньше — на этапе планирования сети. Планирование включает определение количества сетей, из которых будет состоять корпоративная сеть, оценку требуемого количества адресов для каждой сети, получение пула адресов от поставщика услуг, распределение адресного пространства между сетями. Последняя задача часто оказывается нетривиальной, особенно когда решается в условиях дефицита адресов.

Рассмотрим пример использования масок для организации *перекрывающихся адресных пространств*.

Пусть на некотором предприятии было принято решение обратиться к поставщику услуг для получения пула адресов, достаточного для создания сети, структура которой показана на рис. 18.14. Сеть клиента включает три подсети. Две из них — это внутренние сети отделов: сеть Ethernet на 600 пользователей и сеть Token Ring на 200 пользователей. Предприятие также предусматривает отдельную сеть на 10 узлов, главное назначение которой — предоставление информации в режиме открытого доступа для потенциальных клиентов. Такого рода участки корпоративной сети, в которых располагаются веб-серверы, FTP-серверы и другие источники публичной информации, называют **демилитаризованной зоной** (DeMilitarized Zone, DMZ). Еще одна сеть на два узла потребуется для связи с поставщиком услуг, общее число адресов, требуемых для адресации сетевых интерфейсов, составляет 812. Кроме того, необходимо, чтобы пул доступных адресов включал для каждой из сетей широковещательные адреса, состоящие только из единиц, а также адреса, состоящие только из нулей. Учитывая также, что в любой сети адреса всех узлов должны иметь одинаковые префиксы, становится очевидным, что минимальное количество адресов, необходимое клиенту для построения задуманной сети, может значительно отличаться от значения 812, полученного простым суммированием.

В данном примере поставщик услуг решает выделить клиенту непрерывный пул из 1024 адресов. Значение 1024 выбрано как наиболее близкое к требуемому количеству адресов, равному степени двойки ($2^{10} = 1024$). Поставщик услуг выполняет поиск области такого размера в имеющемся у него адресном пространстве — 131.57.0.0/16, часть которого, как показано на рис. 18.15, уже распределена. Обозначим распределенные участки и владеющих ими клиентов — S1, S2 и S3. Поставщик услуг находит среди нераспределенных еще адресов непрерывный участок размером 1024 адреса, начальный адрес которого кратен размеру данного участка. Таким образом, наш клиент получает пул адресов 131.57.8.0/22, обозначенный на рисунке через S.

Далее начинается самый сложный этап — распределение полученного от поставщика услуг адресного пула S между четырьмя сетями клиента. Прежде всего, администратор решил назначить для самой большой сети (Ethernet на 600 узлов) весь пул адресов 131.57.8.0/22, полученный от поставщика услуг (рис. 18.16).

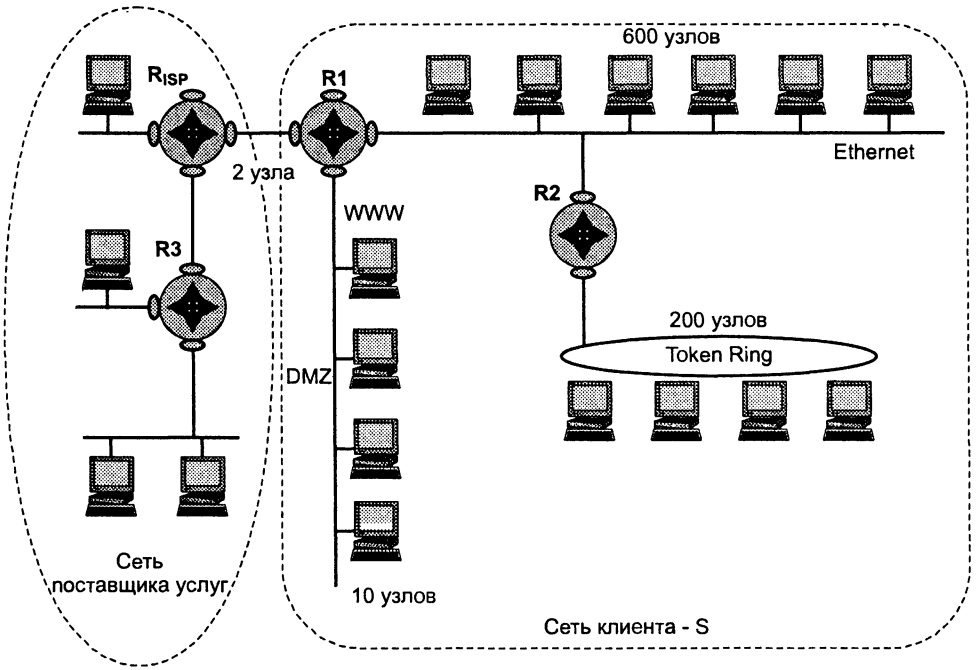


Рис. 18.14. Сети поставщика услуг и клиента

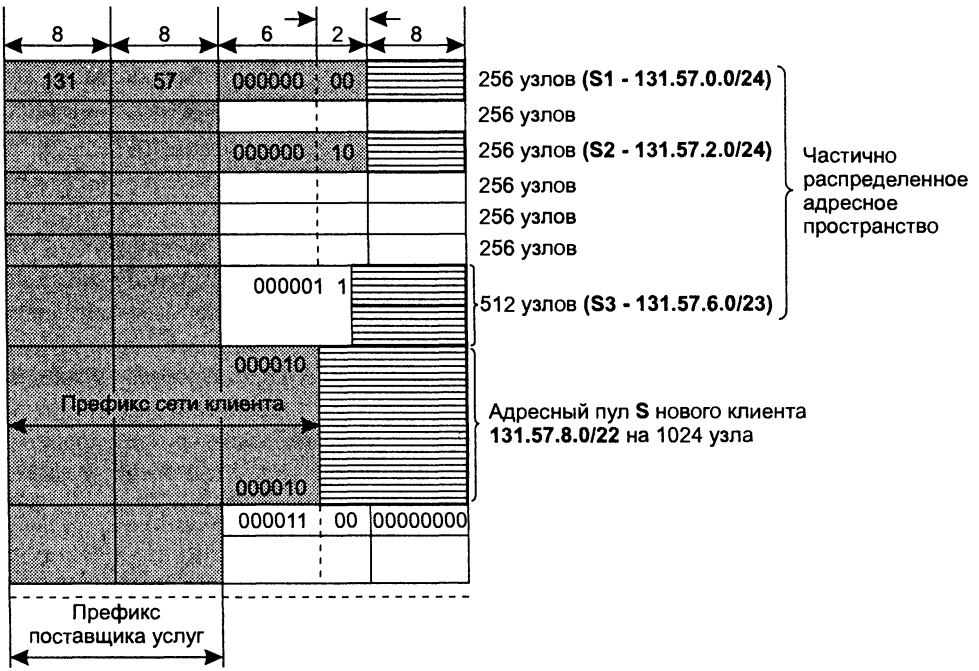


Рис. 18.15. Адресное пространство поставщика услуг

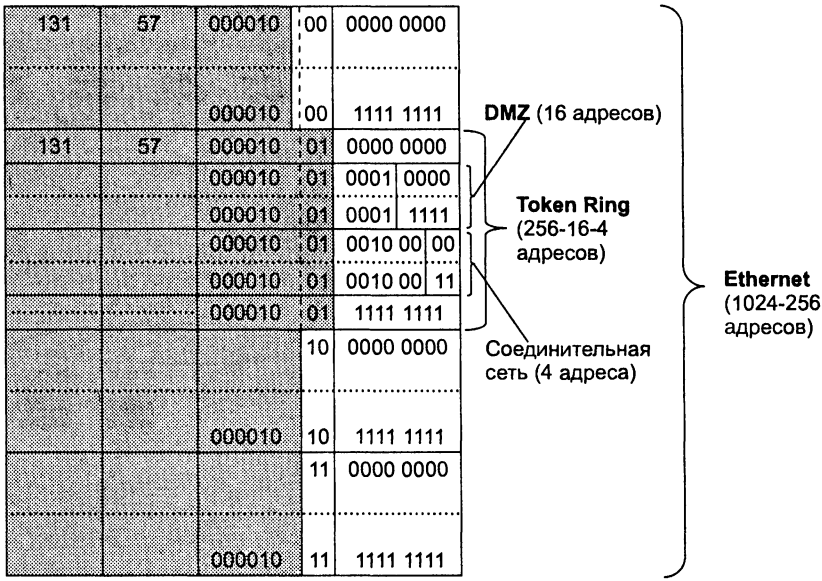


Рис. 18.16. Планирование адресного пространства для сетей клиента

Номер, назначенный для этой сети, совпадает с номером сети, полученным от поставщика услуг. А как же быть с оставшимися тремя сетями? Администратор учел, что для сети Ethernet требуется только 600 адресов, а из оставшихся 624 «выкроил» сеть Token Ring 131.57.9.0/24 на 250 адресов. Воспользовавшись тем, что для Token Ring требуется только 200 адресов, он «вырезал» из нее два участка: для DMZ-сети 131.57.9.16/28 на 16 адресов и для связывающей сети 131.57.9.32/30 на 4 адреса. В результате все сети клиента получили достаточное (а иногда и с избытком) количество адресов.

Следующий этап — это конфигурирование сетевых интерфейсов конечных узлов и маршрутизаторов. Каждому интерфейсу сообщается его IP-адрес и соответствующая маска. На рис. 18.17 показана сконфигурированная сеть клиента.

После конфигурирования сетевых интерфейсов должны быть созданы таблицы маршрутизации маршрутизаторов R1 и R2 клиента. Они могут быть сгенерированы автоматически или с участием администратора. Ниже приведена таблица маршрутизатора R2 (табл. 18.11).

Таблица 18.11. Таблица маршрутизатора R2

Адрес назначения	Маска	Адрес следующего маршрутизатора	Адрес выходного интерфейса	Расстояние
131.57.8.0	255.255.252.0	131.57.8.2	131.57.8.2	Подключена
131.57.9.0	255.255.255.0	131.57.9.1	131.57.9.1	Подключена
131.57.9.16	255.255.255.240	131.57.8.1	131.57.8.2	1
131.57.9.32	255.255.255.252	131.57.8.1	131.57.8.2	1

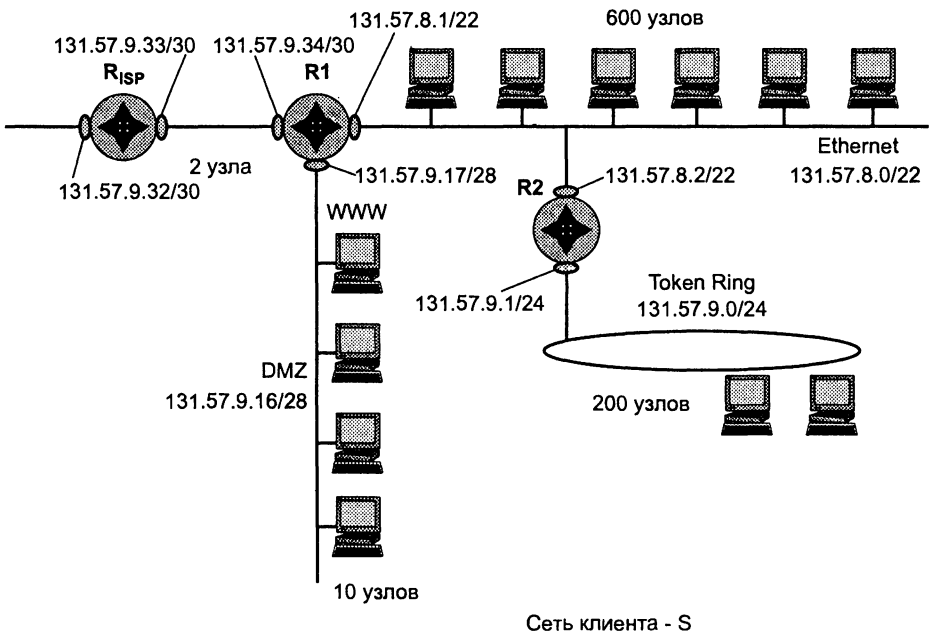


Рис. 18.17. Сконфигурированная сеть клиента

В данной таблице нет маршрута по умолчанию, а значит, все пакеты, адресованные сетям, адреса которых явно не указаны в таблице, будут отбрасываться маршрутизатором.

Пусть, например, на маршрутизатор R2 поступает пакет с адресом назначения 131.57.9.29. В результате просмотра таблицы получаем следующие результаты для каждой строки:

- (131.57.9.29) AND (255.255.252.0) = 131.57.8.0 — совпадение;
- (131.57.9.29) AND (255.255.255.0) = 131.57.9.0 — совпадение;
- (131.57.9.29) AND (255.255.255.240) = 131.57.9.16 — совпадение;
- (131.57.9.29) AND (255.255.255.252) = 131.57.9.28 — нет совпадения.

Поскольку при наличии нескольких совпадений выбирается маршрут из той строки, в которой совпадение адреса назначения с адресом из пакета имеет наибольшую длину, — определено, что пакет с адресом 131.57.9.29 направлен в DMZ-сеть.

CIDR

За последние несколько лет в Интернете многое изменилось: резко возросло число узлов и сетей, повысилась интенсивность трафика, изменился характер передаваемых данных. Из-за несовершенства протоколов маршрутизации обмен сообщениями об обновлении таблиц стал приводить к сбоям магистральных маршрутизаторов, происходящим из-за перегрузок при обработке большого объема служебной информации. Так, сегодня таблицы магистральных маршрутизаторов в Интернете могут содержать до нескольких сотен и даже тысяч маршрутов.

На решение этой проблемы направлена, в частности, и технология **бесклассовой междоменной маршрутизации** (Classless Inter-Domain Routing, CIDR).

Суть технологии CIDR заключается в следующем. Каждому поставщику услуг Интернета назначается *непрерывный* диапазон IP-адресов. При таком подходе все адреса каждого поставщика услуг имеют общую старшую часть — **префикс**, поэтому маршрутизация на магистралях Интернета может осуществляться на основе префиксов, а не полных адресов сетей. А это значит, вместо множества записей по числу сетей будет достаточно поместить *одну запись сразу для всех сетей, имеющих общий префикс*. Такое агрегирование адресов позволит уменьшить объем таблиц в маршрутизаторах всех уровней, а следовательно, ускорить работу маршрутизаторов и повысить пропускную способность Интернета.

Ранее мы рассматривали примеры, где администраторы корпоративных сетей использовали маски для деления непрерывного пула адресов, полученного от поставщика услуг, на несколько частей, чтобы использовать их для структуризации своей сети. Такой вариант использование масок называется *разделением на подсети* (subnetting).

Вместе с тем в процессе использования масок для разделения на подсети проявлялся и обратный эффект их применения — *эффект объединения подсетей*. Упрощенно говоря, для того чтобы направить весь суммарный трафик, адресованный из внешнего окружения в корпоративную сеть, разделенную на подсети, достаточно, чтобы во всех внешних маршрутизаторах наличествовала одна строка. В этой строке на месте адреса назначения должен быть указан *общий префикс для всех этих сетей*. Здесь мы имеем дело с операцией, обратной разделению на подсети, — операцией *агрегирования несколько сетей в одну более крупную* (supernetting).

Вернемся к рис. 18.15, на котором показано адресное пространство поставщика услуг с участками S1, S2, S3 и S, переданными в пользование четырем клиентам. Этот пример также иллюстрирует рис. 18.18. В результате агрегирования сетей клиентов в табл. 18.12 маршрутизатора R_{ISP} поставщика услуг для каждого клиента будет выделено по одной строке независимо от количества подсетей, организованных ими в своих сетях. Так, вместо четырех маршрутов к четырем сетям клиента S в таблице задан только один общий для всех них маршрут (выделенный жирным шрифтом).

Таблица 18.12. Таблица маршрутизатора R_{ISP} поставщика услуг

Адрес назначения	Маска	Следующий маршрутизатор	Номер выходного интерфейса	Расстояние
131.57.0.0 (S1)	255.255.255.0	–	1	Подключена
131.57.2.0 (S2)	255.255.255.0	R3	3	1
131.57.6.0 (S3)	255.255.254.0	R3	3	1
131.57.8.0 (S)	**255.255.252.0**	–	**2**	1
Маршрут по умолчанию	0.0.0.0	$R_{external}$	4	–

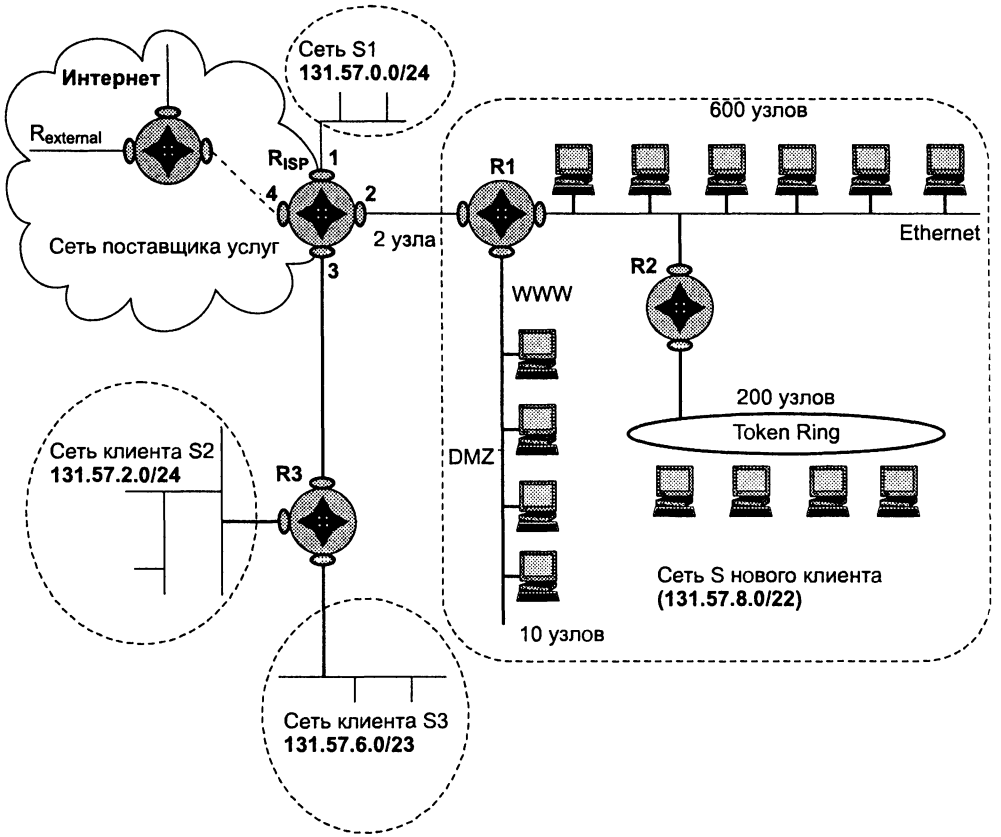


Рис. 18.18. Объединение подсетей

Итак, внедрение технологии CIDR позволяет решить две основные задачи.

- *Более экономное расходование адресного пространства.* Благодаря технологии CIDR поставщики услуг получают возможность «нарезать» блоки из выделенного им адресного пространства в точном соответствии с требованиями каждого клиента, при этом у клиента остается пространство для маневра на случай будущего роста.
- *Уменьшение числа записей в таблицах маршрутизации* за счет объединения маршрутов — одна запись в таблице маршрутизации может представлять большое количество сетей. Если все поставщики услуг Интернета будут придерживаться стратегии CIDR, то особенно заметный выигрыш будет достигаться в магистральных маршрутизаторах.

Необходимым условием эффективного использования технологии CIDR является **локализация адресов**, то есть назначение адресов, имеющих совпадающие префиксы, сетям, располагающимся территориально по соседству. Только в таком случае трафик может быть агрегирован.

К сожалению, сейчас распределение адресов носит во многом случайный характер. Кардинальный путь решения проблемы — перенумерование сетей. Однако эта процедура сопряжена с определенными временны́ми и материальными затратами, и для ее проведения пользователей нужно каким-либо образом стимулировать. В качестве таких стимулов рассматривается, например, введение оплаты за строку в таблице маршрутизации или же за количество узлов в сети. Первое требование подводит потребителя к мысли получить у поставщика услуг такой адрес, чтобы маршрутизация трафика в его сеть шла на основании префикса и номер его сети не фигурировал больше в магистральных маршрутизаторах. Требование оплаты каждого адреса узла также может подтолкнуть пользователя решиться на перенумерование с тем, чтобы получить ровно столько адресов, сколько ему нужно.

Технология CIDR уже успешно используется в текущей версии протокола IP (IPv4) и поддерживается такими протоколами маршрутизации, как OSPF, RIP-2, BGP4 (в основном магистральными маршрутизаторами Интернета). Особенности применения технологии CIDR в новой версии протокола IP (IPv6) будут рассмотрены далее.

Фрагментация IP-пакетов

Список ключевых слов: параметр MTU, идентификатор IP-пакета, поле TTL, смещение фрагмента, флаги MF и DF, сборка фрагментов, тайм-аут ожидания фрагмента.

Важной особенностью протокола IP, отличающей его от других сетевых протоколов (например, от сетевого протокола IPX), является его способность выполнять *динамическую фрагментацию* пакетов при передаче их между сетями с различными максимально допустимыми значениями длины поля данных кадров. Способность к фрагментации протокола IP во многом способствовала масштабируемости технологии TCP/IP.

MTU — параметр технологии

Прежде всего, отметим разницу между фрагментацией сообщений в узле-отправителе и динамической фрагментацией сообщений в транзитных узлах сети — маршрутизаторах. Практически во всех стеках протоколов есть протоколы, которые отвечают за деление (фрагментацию) сообщений прикладного уровня на такие части, которые укладывались бы в кадры канального уровня. Для этого они анализируют тип технологии нижнего уровня и определяют ее MTU (Maximum Transmission Unit — максимальная единица передачи).

В стеке TCP/IP эту задачу решает протокол TCP, который разбивает поток байтов, передаваемый ему с прикладного уровня, на сегменты нужного размера (например, на 1460 байт, если на нижнем уровне данной сети работает протокол Ethernet). Поэтому протокол IP *в узле-отправителе*, как правило, не использует свои возможности по фрагментации пакетов. А вот на маршрутизаторе, когда па-

кет необходимо передать из сети с большим в сеть с меньшим значением MTU, способности протокола IP выполнять фрагментацию становятся востребованными. Значения MTU различных технологий представлены в табл. 18.13.

Таблица 18.13. Значения MTU различных технологий

Технология	MTU
DIX Ethernet	1500 байт
Ethernet 802.3	1492 байт
Token Ring (IBM, 16 Мбит/с)	17 914 байт
Token Ring (802.5, 4 Мбит/с)	4464 байт
FDDI	4352 байт
X.25	576 байт

Из приведенной таблицы видно, что значения MTU для наиболее популярных технологий существенно отличаются, а это значит, что в современной сети, которой свойственна гетерогенность, фрагментация не является редким явлением.

Параметры фрагментации

Суть фрагментации — разбиение пакета, пришедшего из сети с большим значением MTU и направляемого в сеть с меньшим значением MTU, на более короткие *пакеты-фрагменты*. Фрагмент, путешествуя по сети, может вторично подвергнуться фрагментации на каком-либо из промежуточных маршрутизаторов. Каждый из фрагментов должен быть снабжен полноценным заголовком IP.

Некоторые из полей заголовка (идентификатор, TTL, флаги DF и MF, смещение) непосредственно предназначены для проведения последующей процедуры *сборки* фрагментов в исходное сообщение.

- Получатель фрагмента использует **идентификатор** для того, чтобы опознать все фрагменты одного и того же пакета. Модуль IP, отправляющий пакет, устанавливает в поле идентификатора значение, которое должно быть уникальным для данной пары отправителя и получателя в течение всего времени, пока данный пакет (или любой его фрагмент) может существовать в составной IP-сети. Модуль IP может обеспечивать уникальность идентификаторов, например, поддерживая таблицу, где каждая запись соотносится с каждым отдельным получателем, с которым осуществлялась связь, и содержит последнее значение времени жизни пакета в IP-сети. Однако поскольку поле идентификатора допускает 65 536 различных значений, некоторые реализации IP выбирают из этого диапазона идентификаторы случайным образом, полагаясь на высокую вероятность того, что идентификатор окажется уникальным в течение времени передачи пакета.
- Отправитель устанавливает в поле **TTL** время, в течение которого пакет может существовать в сети.

- Поле **смещения фрагмента** предоставляет получателю информацию о положении фрагмента в исходном пакете. Так, первый фрагмент будет иметь в поле смещения нулевое значение. В пакете, не разбитом на фрагменты, поле смещения также имеет нулевое значение.
- Флаг **MF** (More Fragments — больше фрагментов), установленный в 1, — признак того, что пришедший фрагмент не является последним. Модуль IP, отправляющий нефрагментированный пакет, устанавливает флаг MF в нуль.
- Флаг **DF** (Do not Fragment — не фрагментировать), установленный в 1, — признак того, что данный пакет не подлежит фрагментации ни при каких условиях. Если помеченный таким образом пакет не может достигнуть получателя без фрагментации, то модуль IP просто уничтожает пакет, а узлу-отправителю посылается диагностическое ICMP-сообщение.

ПРИМЕЧАНИЕ

Наличие возможности запретить фрагментацию позволяет в некоторых случаях ускорить работу приложений. Для этого вначале необходимо исследовать сеть, определить максимальный размер пакета, который сможет пройти весь путь без фрагментации, а затем использовать пакеты такого или меньшего размера для обмена данными. Данное средство позволяет также предотвратить фрагментацию в тех случаях, когда хост-получатель не имеет достаточных ресурсов для сборки фрагментов.

Процедуры фрагментации и сборки фрагментов

Рассмотрим вначале процедуру *фрагментации*. Прежде чем разделить прибывший пакет на фрагменты, модуль протокола IP, установленный на маршрутизаторе, организует несколько буферов для новых пакетов-фрагментов.

Затем он копирует в эти буферы содержимое некоторых полей заголовка IP из исходного пакета, создавая тем самым «заготовки» заголовков IP всех новых пакетов-фрагментов. Одни параметры заголовка IP копируются в заголовки всех фрагментов, а другие остаются лишь в заголовке первого фрагмента. Процесс фрагментации может изменить значения некоторых полей заголовков IP в пакетах-фрагментах по сравнению с заголовком IP исходного пакета. Так, каждый фрагмент имеет собственное значение контрольной суммы заголовка, смещения фрагмента и общей длины пакета. Во всех пакетах, кроме последнего, флаг MF устанавливается в единицу, а в последнем фрагменте — в нуль.

Содержимое поля данных каждого фрагмента формируется в результате деления содержимого поля данных исходного пакета. При этом должны быть выполнены два условия. Во-первых, размер фрагмента (заголовка в сумме с полем данных) не должен превосходить MTU нижележащей технологии. Во-вторых, размер поля данных каждого фрагмента, кроме самого последнего, должен быть кратным 8 байт. Размер последней части данных равен полученному остатку.

А теперь обсудим, как происходит *сборка* фрагментированного пакета. Сборка пакета происходит на *хосте назначения*.

ВНИМАНИЕ

Отметим, что IP-маршрутизаторы не собирают фрагменты пакетов в более крупные пакеты, даже если на пути встречается сеть, допускающая такое укрупнение. Это связано с тем, что отдельные фрагменты сообщения могут перемещаться по составной сети разными маршрутами, поэтому нет гарантии, что все фрагменты пройдут через какой-то один определенный маршрутизатор на их пути.

Итак, на хосте назначения для каждого фрагментированного пакета отводится отдельный буфер. В этот буфер принимающий протокол IP помещает IP-фрагменты, у которых совпадают IP-адреса отправителя и получателя, а также значения в полях идентификатора и протокола. Все эти признаки говорят модулю IP, что данные пакеты являются фрагментами одного исходного пакета. Собственно сборка заключается в размещении данных из каждого фрагмента в позицию, определенную смещением, указанным в заголовке фрагмента.

Когда первый фрагмент исходного пакета приходит на хост-получатель, этот хост запускает таймер, который определяет максимальное время ожидания прибытия остальных фрагментов данного пакета. В различных реализациях IP применяются разные правила выбора максимального времени ожидания. В частности, таймер может быть установлен на фиксированный период времени (от 60 до 120 секунд), рекомендуемый RFC. Как правило, этот интервал достаточен для доставки пакета от отправителя получателю. В других реализациях максимальное время ожидания определяется с помощью адаптивных алгоритмов измерения и статистической обработки временны́х параметров сети, позволяющих оценивать ожидаемое время прибытия фрагментов. Наконец, тайм-аут может быть выбран на базе значений TTL прибывающих фрагментов. Последний подход основан на том, что нет смысла ожидать, пока прибудут другие фрагменты пакета, если время жизни одного из прибывших фрагментов уже истекло.

Признаком окончания сборки является отсутствие незаполненных промежутков в поле данных и прибытие последнего фрагмента (с равным нулю флагом MF). После того как данные собраны, их можно передавать вышележащему протоколу, например TCP.

ВНИМАНИЕ

Если хотя бы один фрагмент пакета не успеет прийти на хост назначения к моменту истечения таймера, то никаких действий по дублированию отсутствующего фрагмента не предпринимается, а все полученные к этому времени фрагменты пакета отбрасываются! Хосту, пославшему исходный пакет, направляется ICMP-сообщение об ошибке. Такому поведению протокола IP вполне соответствует его кредо «с максимальными усилиями» — стараться по возможности, но никаких гарантий не давать.

Пример

Рассмотрим пример фрагментации на маршрутизаторе (рис. 18.19).

Пусть компьютер-отправитель связан с сетью, имеющей значение MTU 17 914 байт, например с сетью Token Ring. Транспортный уровень, как правило, знает MTU нижележащей технологии и соответствующим образом выбирает размер своих сегментов. Пусть в данном примере с транспортного уровня на уровень IP поступает сообщение

размером 6600 байт. Протокол IP формирует из этого сообщения поле данных IP-пакета и снабжает его заголовком. Обратим особое внимание на заполнение полей заголовка, связанных с фрагментацией. Во-первых, пакету присваивается уникальный идентификатор, например 12456. Во-вторых, поскольку пакет пока еще не был фрагментирован, в поле смещения помещается значение 0, признак MF также обнулен — это показывает, что пакет одновременно является и своим последним фрагментом. В-третьих, признак DF устанавливается в 1, это означает, что данный пакет можно фрагментировать. Общая величина IP-пакета составляет 6600 плюс 20 (размер заголовка IP), то есть 6620 байт, что умещается в поле данных кадра Token Ring. Далее модуль IP хоста-отправителя передает этот кадр своему сетевому интерфейсу, который отправляет кадры следующему маршрутизатору.

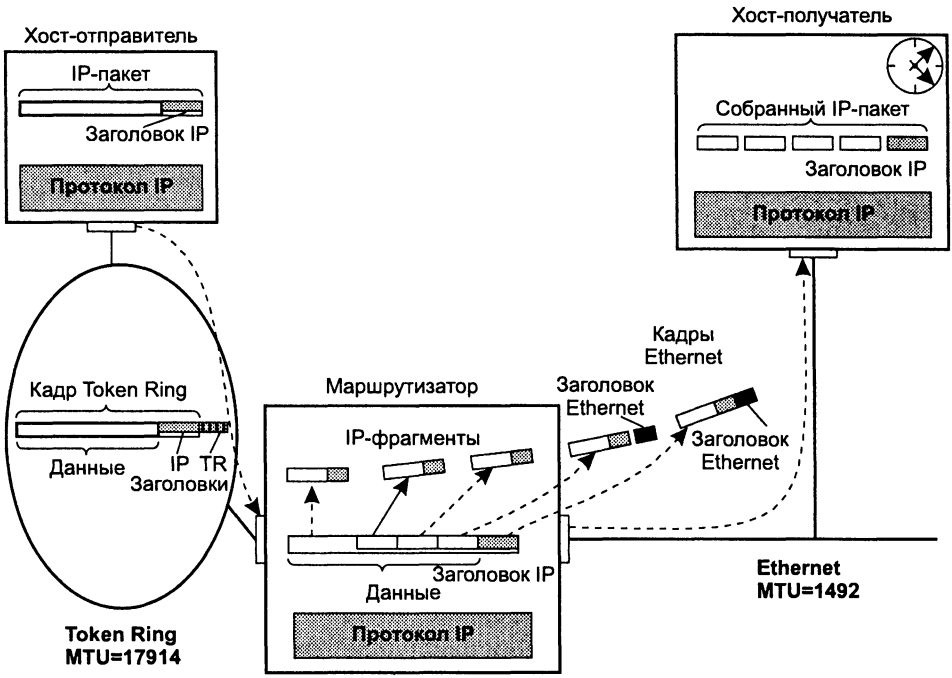


Рис. 18.19. Фрагментация

После того как наш кадр пройдет уровень сетевого интерфейса маршрутизатора и освободится от заголовка Token Ring, модуль IP по сетевому адресу определяет, что прибывший IP-пакет нужно передать в сеть Ethernet, которая имеет значение MTU, равное 1492, что значительно меньше размера поступившего на входной интерфейс пакета. Следовательно, IP-пакет необходимо фрагментировать. Маршрутизатор извлекает поле данных из пакета и делит его на 4 части по 1400 байт и 1 часть в 1000 байт (обратите внимание, что каждый фрагмент данных кратен 8). Затем модуль IP формирует новые IP-пакеты, четыре из которых имеют длину 1400 + 20 = 1420 байт, а один — 1000 + + 20 = 1020 байт, что меньше 1500 байт, поэтому они нормально помещаются в поле данных кадров Ethernet.

В результате хосту-получателю по сети Ethernet приходят пять IP-пакетов с одинаковым идентификатором 12456. Если время прихода фрагментов не превысило тайм-

аут, модуль IP может собирать исходное сообщение. При этом смещение укажет правильный порядок их сборки.

Протокол IPv6

Список ключевых слов: протокол IP нового поколения, префикс формата, индивидуальный адрес, групповой адрес, адрес произвольной рассылки, частный адрес, адрес локальной сети, не разделенной на подсети, адрес локальной сети, разделенной на подсети, глобальный агрегируемый уникальный адрес, поля TLA, NLA и SLA, идентификатор интерфейса, адрес обратной связи, неопределенный адрес, IPv4-совместимый IPv6-адрес, IPv4-отображенный IPv6-адрес, основной заголовок, заголовок маршрутизации, заголовок фрагментации, заголовок аутентификации, заголовок системы безопасности, специальные параметры, параметры получателя, протокол IPSec, магистраль 6Bone.

В начале 90-х годов стек протоколов TCP/IP столкнулся с серьезными проблемами. Именно в это время началось активное промышленное использование Интернета: переход к построению сетей предприятий на основе транспорта Интернета, применение веб-технологии для доступа к корпоративной информации, ведение электронной коммерции через Интернет, внедрение Интернета в индустрию развлечений (распространение видеофильмов, звукозаписей, интерактивные игры).

Все это привело к резкому росту числа узлов сети (в начале 90-х годов новый узел в Интернете появлялся каждые 30 секунд), изменению характера трафика и к ужесточению требований, предъявляемых к качеству обслуживания сетью ее пользователей.

Направления модернизации стека TCP/IP

Сообщество Интернета, а вслед за ним и весь телекоммуникационный мир, начали решать новые задачи путем создания новых протоколов для стека TCP/IP, таких как протокол резервирования ресурсов (RSVP), защищенный протокол IP (IPSec), протокол коммутации меток (MPLS) и т. п. Однако ведущим специалистам было ясно, что только за счет добавления новых протоколов технологию TCP/IP развивать нельзя — нужно решиться на *модернизацию сердцевины стека*, протокола IP. Некоторые проблемы нельзя было решить без изменения формата IP-пакета и логики обработки полей заголовка IP-пакетов. Наиболее очевидной проблемой такого рода была проблема дефицита IP-адресов, которую невозможно снять, не расширив размер полей адресов источника и приемника.

Критике стала все чаще подвергаться масштабируемость маршрутизации. Дело в том, что быстрый рост сети вызвал перегрузку маршрутизаторов, которые должны уже сегодня обрабатывать в своих таблицах маршрутизации информацию о нескольких десятках тысяч номеров сетей, да еще решать некоторые вспомогательные задачи, такие, например, как фрагментация пакетов. Некоторые из

предлагаемых решений данной проблемы также требовали внесения изменений в протокол IP.

Наряду с добавлением новых функций непосредственно в протокол IP необходимо было обеспечить его тесное взаимодействие с новыми протоколами — членами стека TCP/IP, что также требовало добавления в заголовок IP новых полей, обработку которых осуществляли бы эти протоколы. Например, для работы RSVP было желательно введение в заголовок IP поля метки потока, а для протокола IPSec — специальных полей для передачи данных, поддерживающих его функции обеспечения безопасности.

В результате сообщество Интернета после достаточно долгого обсуждения решило подвергнуть протокол IP серьезной переработке, выбрав в качестве основных целей модернизации:

- создание масштабируемой схемы адресации;
- сокращение объема работ, выполняемых маршрутизаторами;
- предоставление гарантий качества транспортных услуг;
- обеспечение защиты данных, передаваемых по сети.

Активные работы по модернизации протокола IP и разработке новых, ассоциированных с ним, протоколов начались в 1992 году. В это время сообществу Интернета были предложены несколько альтернативных вариантов протокола IP нового поколения: IPv7 (разработчик — Ullman), TUBA (Callon), ENCAPS (R. Hinden), SIP (S. Deering) и PIP (Francis).

В результате направления ENCAPS, SIP и PIP в 1993 году слились в единое предложение SIPP, которое в июле 1994 года на сессии сообщества Интернета было принято в качестве основы для создания **протокола IP нового поколения** (Next Generation Internet Protocol, IPng). Сейчас чаще для обозначения новой версии IP используется аббревиатура IPv6.

Документом, фиксирующим появление IPv6, стал RFC 1752. Базовый набор протоколов IPv6 был принят IETF в сентябре 1995 года. В августе 1998 года были приняты пересмотренные версии группы стандартов, определяющих как общую архитектуру IPv6 (RFC 2460), так и его отдельные аспекты, например систему адресации (RFC 2373).

Масштабируемая система адресации

Новая, шестая версия протокола IP (IPv6) внесла существенные изменения в систему адресации IP-сетей (RFC 2373). И, прежде всего, это коснулось *увеличения разрядности адреса.*

IPv6-адрес состоит из 128 бит, или 16 байт. Это дает возможность пронумеровать огромное количество узлов:

340 282 366 920 938 463 463 374 607 431 762 211 456.

Масштаб этого числа иллюстрирует, например, такой факт: если разделить это теоретически возможное количество IP-адресов между всеми жителями Земли (а их сегодня примерно 6 миллиардов), то на каждого из них придется невообра-

зимо, если не сказать бессмысленно большое количество IP-адресов — $5{,}7 \times 10^{28}$! Очевидно, что такое значительное увеличение длины адреса было сделано не только и даже не столько для снятия проблемы дефицита адресов.

Главной целью изменения системы адресации было не механическое увеличение адресного пространства, а повышение эффективности работы стека TCP/IP в целом.

Вместо прежних двух уровней иерархии адреса (номер сети и номер узла) в IPv6 имеется 4 уровня, из которых три уровня используются для идентификации сетей, а один — для идентификации узлов сети. За счет увеличения числа уровней иерархии в адресе новый протокол эффективно поддерживает технологию CIDR. Благодаря этому, а также усовершенствованной системе групповой адресации и введению нового типов адресов новая версия IP позволяет *снизить затраты на маршрутизацию.*

Произошли и чисто внешние изменения — разработчики стандарта предложили использовать вместо десятичной *шестнадцатеричную* форму записи IP-адреса. Каждые четыре шестнадцатеричные цифры отделяются друг от друга двоеточием. Вот как, например, может выглядеть адрес IPv6:

FEDC:0A98:0:0:0:0:7654:3210.

Если в адресе имеется длинная последовательность нулей, то запись адреса можно сократить. Например, приведенный выше адрес можно записать и так:

FEDC:0A98::7654:3210.

Сокращение в виде двух двоеточий (::) может употребляться в адресе только один раз. Можно также опускать незначащие нули в начале каждого поля адреса, например, вместо FEDC:0A98::7654:3210 можно писать FEDC:A98::7654:3210.

Для сетей, поддерживающих обе версии протокола (IPv4 и IPv6), разрешается использовать для младших 4 байт традиционную для IPv4 десятичную запись: 0:0:0:0:0:FFFF:129.144.52.38 или ::FFFF:129.144.52.38.

В новой версии IPv6 предусмотрено три основных типа адресов: индивидуальные адреса, групповые адреса и адреса произвольной рассылки. Тип адреса определяется значением нескольких старших битов адреса, которые названы **префиксом формата**.

- **Индивидуальный адрес** (unicast) определяет уникальный идентификатор отдельного интерфейса конечного узла или маршрутизатора. Назначение адреса этого типа совпадает с назначением уникальных адресов в версии IPv4 — с их помощью пакеты доставляются определенному интерфейсу узла назначения. В версии IPv6, в отличие от версии IPv4, отсутствует понятие класса сети (A, B, C и D) и связанное с ним фиксированное разбиение адреса на номер сети и номер узла по границам байтов. Индивидуальные адреса делятся на несколько подтипов для отражения специфики некоторых часто встречающихся в современных сетях ситуаций.
- **Групповой адрес** (multicast) IPv6 аналогичен по назначению групповому адресу IPv4. Он идентифицирует группу интерфейсов, относящихся, как правило, к разным узлам. Пакет с таким адресом доставляется *всем* интерфейсам с этим адресом. Групповые адреса используются в IPv6 для замены широкове-

щательных адресов — для этого вводится адрес особой группы, объединяющей все интерфейсы подсети.

- **Адрес произвольной рассылки** (anycast) — это новый тип адреса, который так же, как и групповой адрес, определяет группу интерфейсов. Однако пакет с таким адресом доставляется *любому* из интерфейсов группы, как правило, «ближайшему» в соответствии с метрикой, используемой протоколами маршрутизации. Синтаксически адрес произвольной рассылки ничем не отличается от индивидуального адреса и назначается из того же диапазона адресов. Адрес произвольной рассылки может быть назначен только интерфейсам маршрутизатора. Интерфейсы маршрутизаторов, входящие в одну группу произвольной рассылки, имеют индивидуальные адреса и, кроме того, общий адрес группы произвольной рассылки. Адреса такого типа ориентированы на маршрутизацию от источника, при которой маршрут прохождения пакета определяется узлом-отправителем путем указания IP-адресов всех промежуточных маршрутизаторов. Например, поставщик услуг может присвоить всем своим маршрутизаторам один и тот же адрес произвольной рассылки и сообщить его абонентам. Если абонент желает, чтобы его пакеты передавались через сеть этого поставщика услуг, то ему достаточно указать этот адрес в цепочке адресов маршрута от источника, и пакет будет передан через ближайший маршрутизатор данного поставщика услуг.

Так же как и в IPv4, в IPv6 имеются так называемые **частные адреса**, предназначенные для использования в автономных сетях. В отличие от версии IPv4 в версии IPv6 эти адреса представлены двумя разновидностями:

- **Адреса локальных сетей, не разделенных на подсети**, содержат только 64-разрядное поле идентификатора интерфейса, а остальные разряды, кроме префикса формата, должны быть нулевыми, поскольку потребность в номере подсети здесь отсутствует.
- **Адреса локальных сетей, разделенных на подсети**, содержат по сравнению с предыдущими адресами дополнительное двухбайтовое поле номера подсети.

Основным подтипом индивидуального адреса является **глобальный агрегируемый уникальный адрес**. Такие адреса могут агрегироваться для упрощения маршрутизации. В отличие от уникальных адресов узлов версии IPv4, которые состоят из двух полей — номера сети и номера узла, глобальные агрегируемые уникальные адреса IPv6 имеют более сложную структуру, включающую шесть полей (рис. 18.20).

3	13	8	24	16	64
FP	TLA		NLA	SLA	Идентификатор интерфейса

Рис. 18.20. Структура глобального агрегируемого уникального адреса в пакете IPv6

- **Префикс формата** (Format Prefix, FP) для этого типа адресов имеет размер три бита и значение 001.

Следующие три поля — агрегирования верхнего (Top-Level Aggregation, TLA), следующего (Next-Level Aggregation, NLA) и местного (Site-Level Aggregation, SLA) уровней — описывают три уровня идентификации сетей.

- **Поле TLA** предназначено для идентификации сетей самых крупных поставщиков услуг. Конкретное значение этого поля представляет собой общую часть адресов, которыми располагает данный поставщик услуг. Сравнительно небольшое количество разрядов, отведенных под это поле (13), выбрано специально для ограничения размера таблиц маршрутизации в магистральных маршрутизаторах самого верхнего уровня Интернета. Это поле позволяет перенумеровать 8196 сетей поставщиков услуг верхнего уровня, а значит, число записей, описывающих маршруты между этими сетями, также будет ограничено значением 8196, что ускорит работу магистральных маршрутизаторов. Следующие 8 разрядов зарезервированы на будущее для расширения при необходимости поля TLA.
- **Поле NLA** предназначено для нумерации сетей средних и мелких поставщиков услуг. Значительный размер поля NLA позволяет путем агрегирования адресов отразить многоуровневую иерархию поставщиков услуг.
- **Поле SLA** предназначено для адресации подсетей отдельного абонента, например подсетей одной корпоративной сети. Предполагается, что поставщик услуг назначает некоторому предприятию номер его сети, состоящий из фиксированного значения полей TLA и NLA, которые в совокупности являются аналогом номера сети версии IPv4. Остальная часть адреса — поля SLA и идентификатор интерфейса — поступает в распоряжение администратора корпоративной сети, который полностью берет на себя формирование адреса и не должен согласовывать этот процесс с поставщиком услуг. Причем поле идентификатора интерфейса имеет вполне определенное назначение — оно должно хранить физический адрес узла. На этом уровне также можно агрегировать адреса небольших подсетей в более крупные подсети, и размер поля SLA в 16 бит обеспечивает достаточную свободу и гибкость построения внутрикорпоративной иерархии адресов.
- **Идентификатор интерфейса** является аналогом номера узла в IPv4. Отличием версии IPv6 является то, что в общем случае идентификатор интерфейса просто *совпадает с его локальным (аппаратным) адресом*, а не представляет собой произвольно назначенный администратором номер узла. Идентификатор интерфейса имеет длину 64 бита, что позволяет поместить туда MAC-адрес (48 бит), адрес X.25 (до 60 бит), адрес конечного узла ATM (48 бит) или номер виртуального соединения ATM (до 28 бит), а также, вероятно, даст возможность использовать локальные адреса технологий, которые могут появиться в будущем. Такой подход в стиле протокола IPX *делает ненужным протокол ARP*, поскольку процедура отображения IP-адреса на локальный адрес становится тривиальной — она сводится к простому отбрасыванию старшей части адреса. Кроме того, в большинстве случаев *отпадает необходимость ручного конфигурирования* конечных узлов, так как младшую часть адреса — идентификатор интерфейса — узел узнает от аппаратуры (сетевого адаптера и т. п.), а старшую — номер подсети — ему сообщает маршрутизатор.

Очевидно, что при таком изобилии сетей, которое предоставляется клиенту в IPv6, совершенно теряет смысл операция использования масок для *разделения сетей* на подсети, в то время как обратная процедура — *объединение подсетей* — приобретает особое значение. Разработчики стандартов IPv6 считают, что агрегирование адресов является основным способом эффективного использования адресного пространства в новой версии протокола IP.

Пример

Пусть клиент получил от поставщика услуг пул адресов IPv6, определяемый следующим префиксом:

20:0A:00:C9:74:05/48.

Давайте проведем анализ этого числа. Поскольку его первые 3 бита равны 001, следовательно, это *глобальный агрегируемый уникальный адрес* (рис. 18.21).

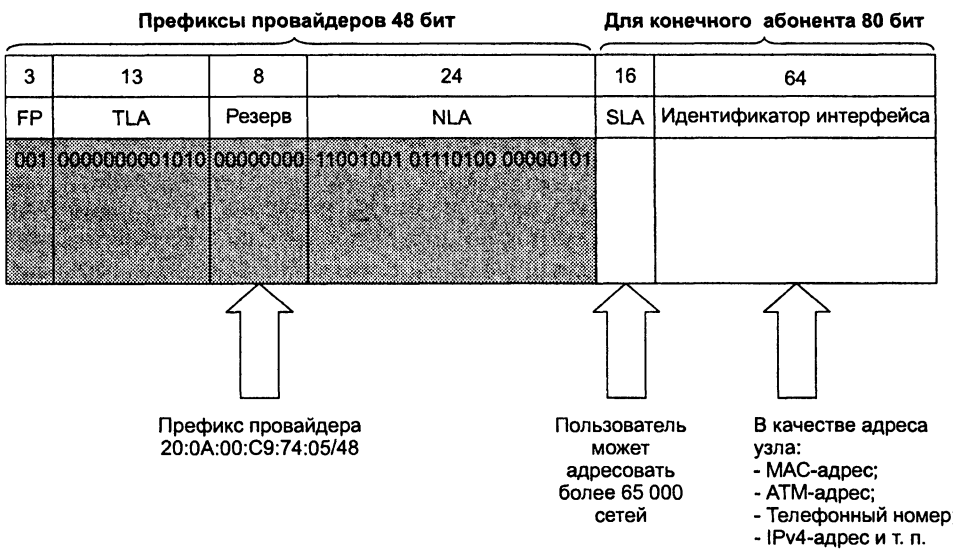


Рис. 18.21. Пример глобального агрегируемого адреса

Адрес этот принадлежит поставщику услуг верхнего уровня, у которого все сети имеют префикс 20:0A/16. Он может выделить поставщику услуг второго уровня некоторый диапазон адресов с общим префиксом, образованным его собственным префиксом, а также частью поля NLA. Длина поля NLA, отводимая под префикс, определяется маской, которую поставщик услуг верхнего уровня также должен сообщить своему клиенту — поставщику услуг второго уровня. Пусть в данном примере маска состоит из 32 единиц в старших разрядах, а результирующий префикс поставщика услуг второго уровня имеет вид:

20:0A:00:C9/32.

В распоряжении поставщика услуг второго уровня остается 16 разрядов поля NLA для нумерации сетей своих клиентов. В качестве клиентов могут выступать поставщики услуг третьего и более низких уровней, а также конечные абоненты — предприятия и организации. Пусть, например, следующий байт (01110100) в поле NLA поставщик

услуг использовал для передачи поставщику услуг более низкого (третьего) уровня, а тот, в свою очередь, использовал последний байт поля NLA для назначения пула адресов клиенту. Таким образом, с участием поставщиков услуг трех уровней был сформирован префикс 20:0A:00:C9:74:05/48, который получил клиент.

Протокол IPv6 оставляет в полном распоряжении клиента 2 байта (поле SLA) для нумерации сетей и 8 байт (полем идентификатора интерфейса) для нумерации узлов. Имея такой огромный диапазон номеров подсетей, администратор может использовать его по-разному. Он может выбрать простую плоскую организацию своей сети, назначая каждой имеющейся подсети определенное значение из диапазона в 65 535 адресов, игнорируя оставшиеся. В крупных сетях более эффективным способом (сокращающим размеры таблиц корпоративных маршрутизаторов) может оказаться иерархическая структуризация сети на основе *агрегирования адресов*. В этом случае используется та же технология CIDR, но уже не поставщиком услуг, а администратором корпоративной сети.

Помимо подробно рассмотренного выше глобального агрегируемого адреса, существуют и другие разновидности индивидуального адреса.

- **Адрес обратной петли** 0:0:0:0:0:0:0:1 играет в версии IPv6 ту же роль, что и адрес 127.0.0.1 в версии IPv4.
- **Неопределенный адрес**, состоящий из одних нулей, является аналогом адреса 0.0.0.0 протокола IPv4. Этот адрес может появляться в IP-пакетах только в качестве адреса источника, и это означает, что пакет послан до того, как узел изучил свой IP-адрес (например, до получения его от DHCP-сервера).

Предполагается, что довольно большое время будут сосуществовать островки Интернета, работающие по протоколу IPv6, и остальная часть Интернета, работающая на версии IPv4. Для того чтобы узлы, поддерживающие версию IPv6, могли использовать технику передачи пакетов IPv6 через сеть IPv4 в автоматическом режиме, разработан специальный подтип адресов, которые переносят IPv4-адрес в младших 4-х байтах IPv6-адреса, а в старших 12 байтах адреса содержат нули (рис. 18.22). Такие индивидуальные адреса делают очень простой процедуру преобразования адресов между двумя версиями протокола IP и называются **IPv4-совместимыми IPv6-адресами**.

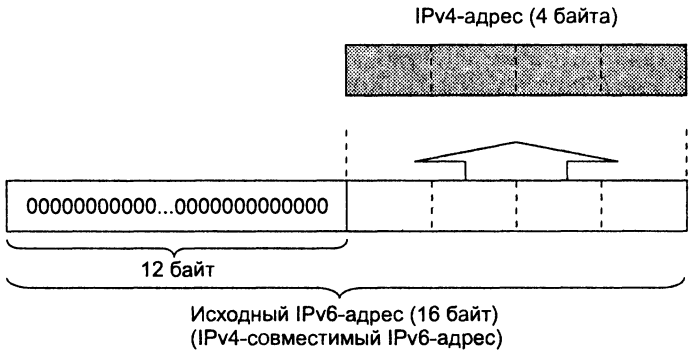


Рис. 18.22. Преобразование IPv6 в IPv4

Для решения обратной задачи — передачи IPv4-пакетов через части Интернета, работающие по протоколу IPv6, — предназначен **IPv4-отображенный IPv6-адрес**. Этот тип адреса по-прежнему содержит в 4-х младших байтах IPv4-адрес, в старших 10-ти байтах — нули, а в 5-м и 6-м байтах IPv6-адреса — единицы, которые показывают, что узел поддерживает только 4-ю версию протокола IP (рис. 18.23).

Работа по детализации подтипов IPv6-адресов еще далека от завершения. Сегодня определено назначение только 15 % адресного пространства IPv6, а оставшаяся часть адресов еще ждет своей очереди, чтобы найти применение для решения одной из многочисленных проблем Интернета.

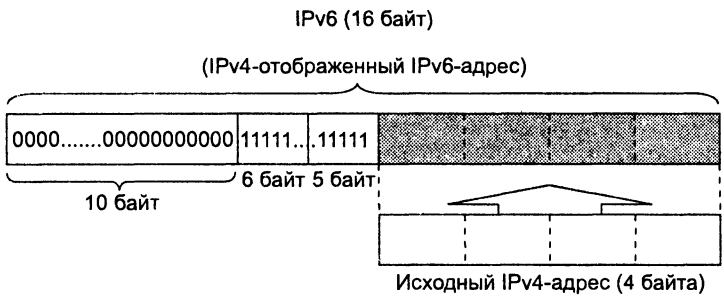


Рис. 18.23. Преобразование IPv4 в IPv6

Гибкий формат заголовка

Одной из основных целей изменения формата заголовка в IPv6 было снижение накладных расходов, то есть уменьшение объема служебной информации, передаваемой с каждым пакетом. Для этого в новом протоколе IP были введены понятия основного и дополнительного заголовков. Основной заголовок присутствует всегда, а дополнительные являются опциональными. Дополнительные заголовки могут содержать, например, информацию о фрагментации исходного пакета, полный маршрут следования пакета при маршрутизации от источника, информацию, необходимую для защиты передаваемых данных.

Основной заголовок имеет фиксированную длину в 40 байт, его формат показан на рис. 18.24.

Поле следующего заголовка соответствует по назначению полю протокола в версии IPv4 и определяет тип заголовка, который следует за данным. Каждый следующий дополнительный заголовок также содержит поле следующего заголовка. Если IP-пакет не содержит дополнительных заголовков, то в этом поле будет значение, закрепленное за протоколом TCP, UDP, RIP, OSPF или другим, определенным в стандарте IPv4.

В предложениях по поводу протокола IPv6 фигурируют пока следующие типы дополнительных заголовков:

- **заголовок маршрутизации** — указание полного маршрута при маршрутизации от источника;

- **заголовок фрагментации** — информация, относящаяся к фрагментации IP-пакета (поле обрабатывается только в конечных узлах);
- **заголовок аутентификации** — информация, необходимая для аутентификации конечных узлов и обеспечения целостности содержимого IP-пакетов;
- **заголовок системы безопасности** — информация, необходимая для обеспечения конфиденциальности передаваемых данных путем шифрования и дешифрирования;
- **специальные параметры** — параметры, необходимые для последовательной обработки пакетов на каждом ретрансляционном участке;
- **параметры получателя** — дополнительная информация для узла назначения.

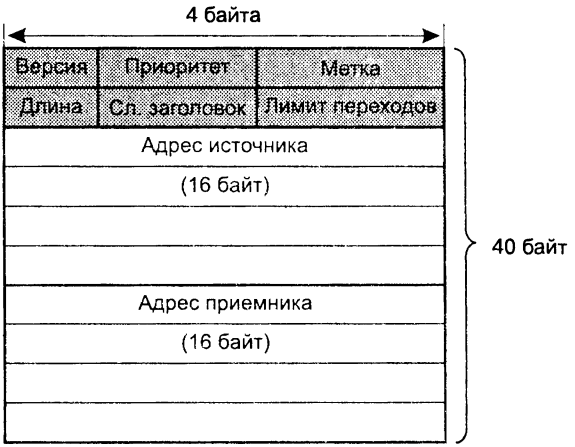


Рис. 18.24. Формат основного заголовка

Таким образом, IP-пакет может иметь, например, формат, показанный на рис. 18.25.

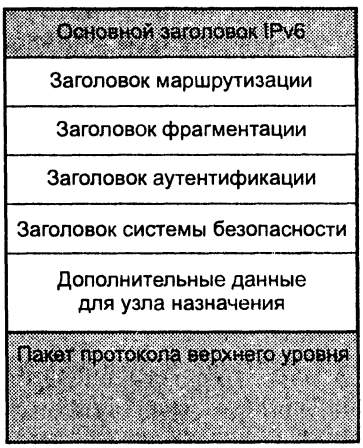


Рис. 18.25. Структура IPv6-пакета

Поскольку для маршрутизации пакета обязательным является только основной заголовок (почти все дополнительные заголовки обрабатываются только в конечных узлах), *это снижает нагрузку на маршрутизаторы.* С другой стороны, возможность использования большого количества дополнительных параметров *расширяет функциональность протокола IP* и делает его открытым для внедрения новых механизмов.

Снижение нагрузки на маршрутизаторы

Для того чтобы повысить производительность маршрутизаторов Интернета в части выполнения их основной функции — продвижения пакетов, в версии IPv6 предпринят ряд мер по освобождению маршрутизаторов от некоторых вспомогательных задач.

- *Перенесение функций фрагментации с маршрутизаторов на конечные узлы.* Конечные узлы в версии IPv6 обязаны найти минимальное значение MTU вдоль всего пути, соединяющего исходный узел с узлом назначения (эта техника под названием Path MTU Discovery уже используется в IPv4). Маршрутизаторы IPv6 не выполняют фрагментацию, а только посылают ICMP-сообщение «Слишком длинный пакет» конечному узлу, который должен уменьшить размер пакета.
- *Агрегирование адресов*, ведущее к уменьшению размера адресных таблиц маршрутизаторов, а значит, — к сокращению времени просмотра и обновления таблиц. При этом также сокращается служебный трафик, создаваемый протоколами маршрутизации.
- *Широкое использование маршрутизации от источника*, при которой узел-источник задает полный маршрут прохождения пакета через сети. Такая техника освобождает маршрутизаторы от необходимости просмотра адресных таблиц при выборе следующего маршрутизатора.
- *Отказ от обработки не обязательных параметров заголовка.*
- *Использование в качестве номера узла его MAC-адреса*, что избавляет маршрутизаторы от необходимости применять протокол ARP.

Новая версия протокола IP, являющаяся составной частью проекта IPv6, предлагает встроенные средства защиты данных. Размещение средств защиты на сетевом уровне делает их прозрачными для приложений, так как между уровнем IP и приложением всегда будет работать протокол транспортного уровня. Приложения переписывать при этом не придется. Новая версия протокола IP со встроенными средствами обеспечения безопасности называется **IPSec** (Security Internet Protocol — защищенный протокол IP). Возможности этого протокола подробно рассматриваются в главе 24.

Переход от версии IPv4 к версии IPv6 только начинается. Сегодня уже существуют фрагменты Интернета, в которых маршрутизаторы поддерживают обе версии протокола. Эти фрагменты объединены между собой через Интернет, образуя так называемую магистраль **6Bone**.

Выводы

Протокол IP решает задачу доставки сообщений между узлами составной сети. Поскольку он является дейтаграммным, то не дает никаких гарантий надежной доставки сообщений.

Важной особенностью протокола IP, отличающей его от других сетевых протоколов, например от сетевого протокола IPX, является его способность выполнять динамическую фрагментацию пакетов при передаче их между сетями с различными, максимально допустимыми значениями длины поля данных кадров (MTU).

Максимальная длина IP-пакета составляет 65 535 байт. Заголовок обычно имеет длину 20 байт и содержит информацию о сетевых адресах отправителя и получателя, параметры фрагментации, время жизни пакета, контрольную сумму и некоторые другие параметры.

Вид таблицы IP-маршрутизации зависит от конкретной реализации маршрутизатора. Несмотря на значительные внешние различия выводимых на экран таблиц, все они включают два обязательных поля, без которых невозможно выполнять маршрутизацию, — это поля адресов назначения и следующего маршрутизатора.

Записи в таблицу маршрутизации могут поступать из разных источников. Во-первых, в результате конфигурирования программное обеспечение стека TCP/IP заносит в таблицу записи о непосредственно подключенных сетях и маршрутизаторах по умолчанию, а также записи об особых адресах. Во-вторых, администратор вручную заносит записи о специфических маршрутах и о маршруте по умолчанию. В-третьих, протоколы маршрутизации автоматически заносят в таблицу динамические записи об имеющихся маршрутах.

Эффективным средством структуризации IP-сетей являются маски. Маски позволяют разделить одну сеть на несколько подсетей или объединить несколько сетей в одну более крупную сеть.

Значительная роль в будущем IP-сетей отводится технологии бесклассовой междоменной маршрутизации (CIDR), которая решает две основные задачи. Первая состоит в более экономном расходовании адресного пространства, вторая — в уменьшении числа записей в таблицах (одна запись может представлять множество сетей, объединенных общим префиксом).

В начале 90-х годов стек протоколов TCP/IP столкнулся с серьезными проблемами, которые нельзя было решить без изменения формата IP-пакета и логики обработки полей заголовка IP-пакетов. В результате сообщество Интернета решило создать новую версию протокола IP (IPv6), выбрав в качестве основных целей модернизации создание масштабируемой схемы адресации; повышение пропускной способности сети за счет сокращения работ, выполняемых маршрутизаторами; предоставление гарантий качества транспортных услуг; обеспечение защиты данных, передаваемых по сети.

Вопросы и задания

1. В чем проявляется ненадежность протокола IP?
2. Сравните таблицу моста или коммутатора с таблицей маршрутизатора. Каким образом формируются эти таблицы? Какую информацию содержат? От чего зависит их объем?
3. Рассмотрим маршрутизатор на магистрали Интернета. Какие записи содержатся в поле адреса назначения его таблицы маршрутизации? Варианты ответов:
   - номера всех сетей Интернета;
   - номера некоторых сетей Интернета;

- номера некоторых сетей и полные адреса некоторых конечных узлов Интернета, для которых определены специфические маршруты;
- специальные адреса типа 127.0.0.0 или 255.255.255.255.

4. Сколько записей о маршрутах по умолчанию может включать таблица маршрутизации?
5. Приведите примеры, когда может возникнуть необходимость в использовании специфических маршрутов?
6. Передается ли в IP-пакете маска в тех случаях, когда маршрутизация реализуется с использованием масок?
7. Какие преимущества дает технология CIDR? Что мешает ее широкому внедрению?
8. Имеется ли связь между длиной префикса непрерывного пула IP-адресов и числом адресов, входящих в этот пул?
9. Почему в записи о маршруте по умолчанию в качестве адреса сети назначения часто указывается 0.0.0.0 с маской 0.0.0.0?
10. Какие элементы сети могут выполнять фрагментацию? Варианты ответов:
    - только компьютеры;
    - только маршрутизаторы;
    - компьютеры, маршрутизаторы, мосты, коммутаторы;
    - компьютеры и маршрутизаторы.
11. Что произойдет, если при передаче пакета он был фрагментирован и один из фрагментов не дошел до узла назначения после истечения тайм-аута? Варианты ответов:
    - модуль IP узла-отправителя повторит передачу недошедшего фрагмента;
    - модуль IP узла-отправителя повторит передачу всего пакета, в состав которого входил недошедший фрагмент;
    - модуль IP узла-получателя отбросит все полученные фрагменты пакета, в котором потерялся один фрагмент, а модуль IP узла-отправителя не будет предпринимать никаких действий по повторной передаче данного пакета.
12. На рис. 18.26 показан компьютер с двумя сетевыми адаптерами, к которым подсоединены два сегмента. Компьютер работает под управлением Windows 2000. Может ли компьютер A в одном сегменте обмениваться данными с компьютером B, принадлежащем другому сегменту?
13. Может ли изменить ответ на предыдущий вопрос тот факт, что в сегментах используются разные канальные протоколы, например Ethernet и Token Ring?
14. Каково отношение администратора IPv6-сети к маскам? Варианты ответов:
    - полностью игнорирует как ненужное средство;
    - использует при объединении подсетей;
    - использует при разделении на подсети;
    - использует и при объединении подсетей, и при разделении на подсети.

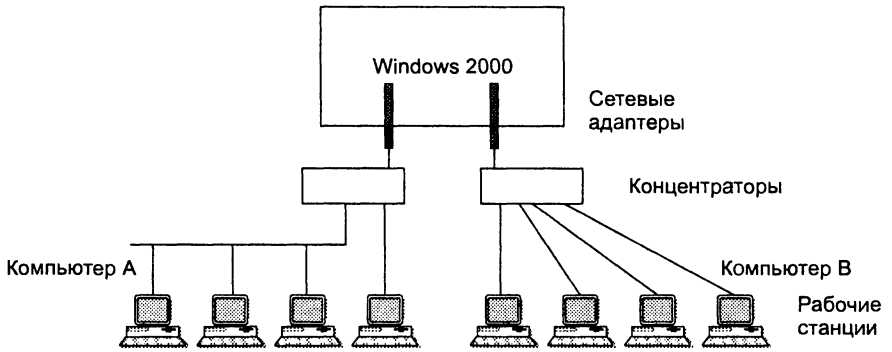


Рис. 18.26. Два сетевых сегмента, соединенные компьютером

15. Верно ли утверждение, что широковещательная рассылка является частным случаем групповой рассылки? Произвольной рассылки?
16. Может ли один сетевой интерфейс иметь одновременно несколько IPv6-адресов разных типов: уникальный адрес, адрес произвольной рассылки, групповой адрес?
17. В разделе «Перекрытие адресных пространств» приведен пример того, как администратор, планируя сеть своего предприятия, решил, что для сети Ethernet ему требуется 600 адресов, для сети Token Ring — 200, для DMZ —10 и для соединительной сети — 4. Решите ту же задачу по планированию сети, но для случая, когда в сети Token Ring планируется 300 рабочих мест. Какой пул адресов необходимо получить у поставщика услуг на этот раз? (Для определенности будем считать, что поставщик услуг выделит непрерывный пул адресов.) Как администратор распределит адреса между своими четырьмя сетями? Как будут выглядеть таблицы маршрутизации R1 и R2?

ГЛАВА 19 Базовые протоколы TCP/IP

Эту главу мы начнем с изучения протоколов TCP и UDP, исполняющих посредническую роль между приложениями и транспортной инфраструктурой сети. В то время как задачей уровня межсетевого взаимодействия, к которому относится протокол IP, является передача данных между сетевыми интерфейсами в составной сети, главная задача транспортного уровня, которую решают протоколы TCP и UDP, заключается в передаче данных между *прикладными процессами*, выполняющимися на компьютерах в сети.

Далее в главе рассматриваются протоколы маршрутизации, предназначенные для автоматического построения таблиц маршрутизации, на основе которых происходит продвижение пакетов сетевого уровня. Протоколы маршрутизации, в отличие от сетевых протоколов, таких как IP и IPX, не являются обязательными, так как таблица маршрутизации может строиться администратором сети вручную. Однако в крупных сетях со сложной топологией и большим количеством альтернативных маршрутов протоколы маршрутизации выполняют очень важную и полезную работу, автоматизируя построение таблиц маршрутизации, а также отыскивая новые маршруты при изменениях сети: отказах или появлении новых линий связи и маршрутизаторов.

Мы рассмотрим также протокол ICMP, являющийся средством оповещения отправителя о причинах недоставки его пакетов адресату. Помимо диагностики ICMP используется для *мониторинга* сети. Так, в основе популярных утилит мониторинга IP-сетей ping и traceroute лежат ICMP-сообщения.

Протоколы транспортного уровня TCP и UDP

Список ключевых слов: протокол управления передачей, протокол пользовательских дейтаграмм, мультиплексирование, демультиплексирование, порт, назначенный номер порта, хорошо известные номера портов, динамический номер порта, UDP-порт, TCP-порт, UDP-дейтаграмма, пользовательская дейтаграмма, UDP-сокет, порт источника, порт приемника, последовательный номер, подтвержденный номер, длина заголовка, резервные биты, кодовые биты, контрольная сумма, указатель срочности, параметры, заполнитель, логическое соединение, окно приема.

Как уже было отмечено, главная задача транспортного уровня заключается в передаче данных между *прикладными процессами*. Эту задачу решают **протокол управления передачей** (Transmission Control Protocol, TCP), описанный в RFC 793, и **протокол пользовательских дейтаграмм** (User Datagram Protocol, UDP), описанный в RFC 768. Протоколы TCP и UDP имеют много общего. Тот и другой обеспечивают интерфейс с вышележащим прикладным уровнем, передавая данные, поступающие на входной интерфейс хоста, соответствующему приложению. При этом оба протокола используют концепции «порт» и «сокет». Оба они также поддерживают интерфейс с нижележащим сетевым уровнем IP, упаковывая свои PDU в IP-пакеты. Протокольные сущности TCP и UDP, как и в случае протоколов прикладного уровня, устанавливаются только на конечных узлах. Однако, как мы увидим далее, различий между TCP и UDP гораздо больше, чем сходств.

Порты

Каждый компьютер может выполнять несколько процессов, более того, прикладной процесс тоже может иметь несколько точек входа, выступающих в качестве адреса назначения для пакетов данных. Поэтому после того, как пакет средствами протокола IP доставлен на сетевой интерфейс компьютера-получателя, данные необходимо переправить конкретному процессу-получателю.

Существует и обратная задача: пакеты, которые отправляют в сеть разные приложения, работающие на одном конечном узле, обрабатываются общим для них протоколом IP. Следовательно, в стеке должно быть предусмотрено средство «сбора» пакетов от разных приложений для передачи протоколу IP. Эту работу выполняют протоколы TCP и UDP.

Процедура приема данных протоколами TCP и UDP, поступающих от нескольких различных прикладных служб, называется **мультиплексированием**. Обратная процедура — процедура распределения протоколами TCP и UDP поступающих от сетевого уровня пакетов между набором высокоуровневых служб — называется **демультиплексированием** (рис. 19.1).

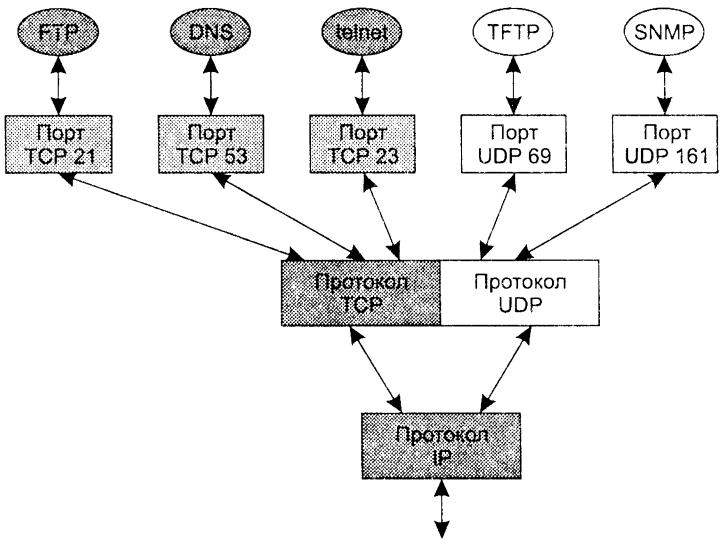


Рис. 19.1. Мультиплексирование и демультиплексирование на транспортном уровне

Протоколы TCP и UDP ведут для каждого приложения две очереди: очередь пакетов, поступающих к данному приложению из сети, и очередь пакетов, отправляемых данным приложением в сеть. Пакеты, поступающие на транспортный уровень, организуются операционной системой в виде множества очередей к точкам входа различных прикладных процессов. В терминологии TCP/IP такие системные очереди называются **портами**[1], причем входная и выходная очереди одного приложения рассматриваются как один порт. Для однозначной идентификации портов им присваивают номера. Номера портов используются для адресации приложений.

Если процессы представляют собой популярные общедоступные службы, такие как FTP, telnet, HTTP, TFTP, DNS и т. п., то за ними закрепляются *стандартные*, **назначенные номера**, также называемые **хорошо известными** (well-known) **номерами портов**. Эти номера закрепляются и публикуются в стандартах Интернета (RFC 1700, RFC 3232). Так, номер 21 закреплен за службой удаленного доступа к файлам FTP, а 23 — за службой удаленного управления telnet. Назначенные номера являются *уникальными в пределах Интернета* и назначаются приложениям *централизованно* из диапазона от 0 до 1023.

Для тех приложений, которые еще не стали столь распространенными, чтобы закреплять за ними стандартные номера, номера портов назначаются разработчиками этих приложений или операционной системой *локально* в ответ на поступление запроса от приложения. На каждом компьютере операционная система ведет список занятых и свободных номеров портов. При поступлении запроса от приложения, выполняемого на данном компьютере, операционная система выделяет ему первый свободный номер. Такие номера называют **динамическими**.

[1] Порты приложения не надо путать с портами (сетевыми интерфейсами) оборудования.

В дальнейшем все сетевые приложения должны адресоваться к данному приложению с указанием назначенного ему номера порта. После того как приложение завершит работу, выделенный ему локальный номер порта возвращается в список свободных и может быть назначен другому приложению. Динамические номера являются *уникальными в пределах каждого компьютера*, но при этом обычной ситуацией является совпадение номеров портов приложений, выполняемых на разных компьютерах. Как правило, клиентские части известных приложений (DNS, WWW, FTP, telnet и др.) получают динамические номера портов от ОС.

Все, что было сказано о портах, в равной степени относится к обоим протоколам транспортного уровня (TCP и UDP). В принципе нет никакой зависимости между назначением номеров для приложений, использующих протокол TCP, и приложений, работающих с протоколом UDP. Приложения, которые передают данные на уровень IP по протоколу UDP, получают номера, называемые **UDP-портами**. Аналогично приложениям, обращающимся к протоколу TCP, выделяются **TCP-порты**.

В том и другом случаях это могут быть как назначенные, так и динамические номера. Диапазоны чисел, из которых выделяются номера TCP- и UDP-портов, совпадают: от 0 до 1023 для назначенных и от 1024 до 65535 для динамических. Однако никакой связи между назначенными номерами TCP- и UDP-портов нет. Даже если номера TCP- и UDP-портов совпадают, они идентифицируют разные приложения. Например, одному приложению может быть назначен TCP-порт 1750, а другому — UDP-порт 1750. В некоторых случаях, когда приложение может обращаться по выбору к протоколу TCP или UDP (например, таким приложением является DNS), ему, исходя из удобства запоминания, назначаются совпадающие номера TCP- и UDP-портов (в данном примере — это номер 53).

Протокол UDP

Протокол UDP, являясь дейтаграммным протоколом, реализует сервис *по возможности*, то есть не гарантирует доставку своих сообщений, а, следовательно, никоим образом не компенсирует ненадежность дейтаграммного протокола IP.

Единица данных протокола UDP называется **UDP-дейтаграммой**, или **пользовательской дейтаграммой**. Каждая дейтаграмма переносит *отдельное пользовательское сообщение* (рис. 19.2). Это приводит к естественному ограничению: длина дейтаграммы UDP не может превышать длины поля данных протокола IP, которое, в свою очередь, ограничено размером кадра технологии нижнего уровня. Поэтому если UDP-буфер переполняется, то данные приложения отбрасываются.

Заголовок UDP, состоящий из четырех 2-байтовых полей, содержит номера портов отправителя и получателя, контрольную сумму и длину дейтаграммы.

Ниже приведен пример заголовка UDP с заполненными полями:

```
Source Port = 0x0035
Destination Port = 0x0411
Total length = 132 (0x84) bytes
Checksum = 0x5333
```

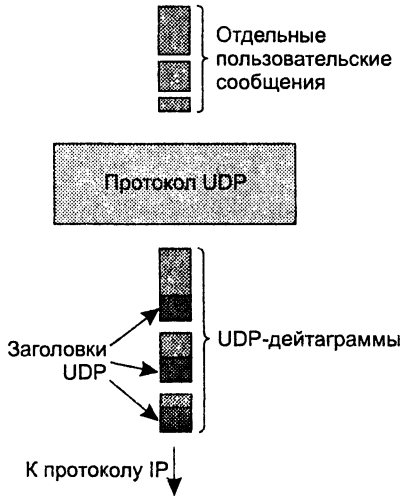


Рис. 19.2. Формирование дейтаграммы протокола UDP

В этой UDP-дейтаграмме в поле данных, длина которого, как следует из заголовка, равна (132 – 8) байт, помещено сообщение DNS-сервера. Это можно видеть по номеру порта источника (Source Port = 0x0035), что в шестнадцатеричном формате равно стандартному номеру DNS-сервера — 53.

Судя по простоте заголовка, протокол UDP очень сложным не является. Действительно, его функции сводятся к мультиплексированию и демультиплексированию данных между сетевым и прикладным уровнями.

Давайте рассмотрим, как протокол UDP решает задачу демультиплексирования. Казалось бы, для этой цели достаточно использовать номера портов. Кадры, несущие UDP-дейтаграммы, пребывают на сетевой интерфейс хоста, последовательно обрабатываются протоколами стека и, наконец, поступают в распоряжение протокола UDP. UDP извлекает из заголовка номер порта назначения и передает данные на соответствующий порт соответствующему приложению, то есть выполняет демультиплексирование.

Это решение выглядит очень логично и просто, однако оно неработоспособно в ситуации, когда на одном конечном узле выполняется *несколько копий* одного и того же приложения. Пусть, например, на одном хосте запущены два DNS-сервера, причем оба используют для передачи своих сообщений протокол UDP (рис. 19.3). DNS-сервер имеет хорошо известный UDP-порт 53. В то же время у каждого из DNS-серверов могут быть свои клиенты, собственные базы данных, собственные настройки. Когда на сетевой интерфейс данного компьютера придет запрос от DNS-клиента, в UDP-дейтаграмме будет указан номер порта 53, который в равной степени относится к обоим DNS-серверам — так кому же из них протокол UDP должен передать запрос? Чтобы снять неоднозначность, применяют следующий подход. Разным копиям одного приложения, даже установленным на одном компьютере, присваивают разные IP-адреса. В данном примере DNS-сервер 1 имеет IP-адрес IP_1, а DNS-сервер 2 — IP-адрес IP_2. Таким образом,

однозначно определяет прикладной процесс в сети (а тем более в пределах компьютера) пара (IP-адрес, номер порта UDP), называемая **UDP-сокетом** (UDP socket).

Таким образом, протокол UDP выполняет демультиплексирование на основе сокетов.

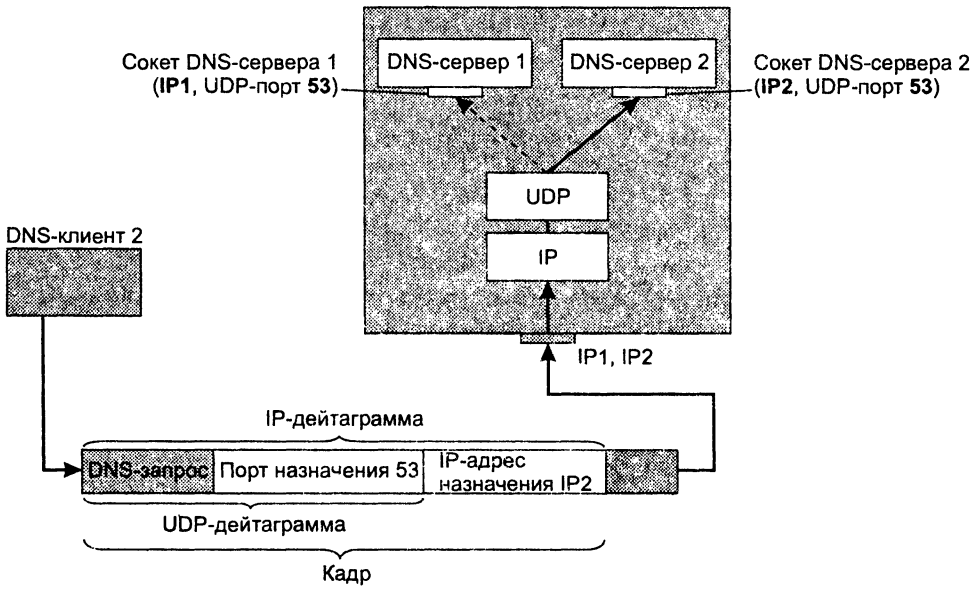


Рис. 19.3. Демультиплексирование протокола UDP на основе сокетов

ПРИМЕЧАНИЕ

Здесь мы должны уточнить ту упрощенную картину прохождения пакета вверх по стеку, которая была описана ранее. Действительно, как мы и отмечали в предыдущих главах, после обработки поступившего из сети пакета протоколом IP заголовок этого пакета отбрасывается, и «наверх» передается содержимое поля данных IP-пакета, например UDP-дейтаграмма. Однако мы упускали одну важную деталь — вместе с содержимым поля данных на транспортный уровень передается извлеченный из заголовка IP-адрес назначения.

Формат TCP-сегмента

Информация, поступающая к протоколу TCP от протоколов более высокого уровня, рассматривается протоколом TCP как *неструктурированный поток байтов*. Поступающие данные буферизуются средствами TCP. Для передачи на сетевой уровень из буфера «вырезается» некоторая непрерывная часть данных, которая называется сегментом[1] и снабжается заголовком (рис. 19.4).

[1] Заметим, что сегментом называют как единицу передаваемых данных в целом (поле данных и заголовок TCP), так и отдельно поле данных.

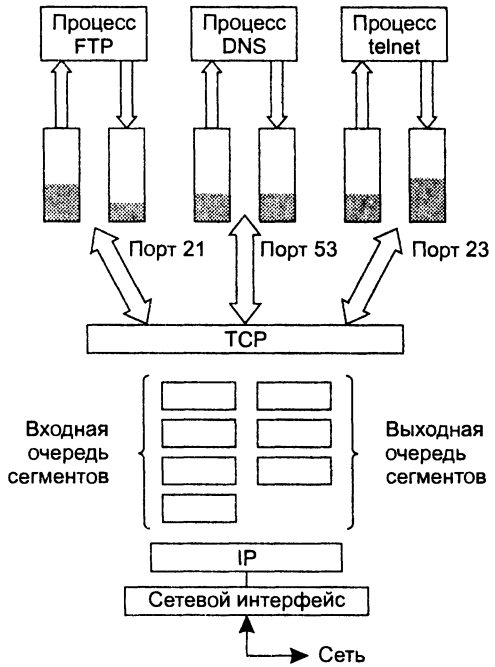


Рис. 19.4. Формирование TCP-сегментов из потока байтов

Заголовок TCP-сегмента содержит значительно больше полей, чем заголовок UDP, что отражает более развитые возможности первого протокола:

- **Порт источника** (sourc port) занимает 2 байта и идентифицирует процесс-отправитель.
- **Порт приемника** (destination port) занимает 2 байта и идентифицирует процесс-получатель.
- **Последовательный номер** (sequence number) занимает 4 байта и представляет собой номер байта, который определяет смещение сегмента относительно потока отправляемых данных (другими словами, номер первого байта данных в сегменте).
- **Подтвержденный номер** (acknowledgement number) занимает 4 байта и содержит максимальный номер байта в полученном сегменте, увеличенный на единицу. Именно это значение используется в качестве квитанции. Если установлен контрольный бит ACK, то это поле содержит следующий номер очереди, который отправитель данного сегмента желает получить в обратном направлении.
- **Длина заголовка** (hlen) занимает 4 бита и представляет собой длину заголовка TCP-сегмента, измеренную в 32-битовых словах. Длина заголовка не фиксирована и может изменяться в зависимости от значений, устанавливаемых в поле параметров.
- **Резерв** (reserved) занимает 6 бит.

- **Кодовые биты** (code bits) числом 6 содержат служебную информацию о типе данного сегмента. Положительное значение сигнализируется установкой этих битов в единицу:
  - **URG** — срочное сообщение;
  - **ACK** — квитанция на принятый сегмент;
  - **PSH** — запрос на отправку сообщения без ожидания заполнения буфера (протокол TCP может выжидать заполнения буфера перед отправкой сегмента, но если требуется срочная передача, то приложение сообщает об этом протоколу TCP с помощью данного бита);
  - **RST** — запрос на восстановление соединения;
  - **SYN** — сообщение, используемое для синхронизации счетчиков переданных данных при установлении соединения;
  - **FIN** — признак достижения передающей стороной последнего байта в потоке передаваемых данных.
- **Окно** (window) занимает 2 байта и задает количество байтов данных, ожидаемых отправителем данного сегмента, начиная с байта, номер которого указан в поле подтвержденного номера.
- **Контрольная сумма** (checksum) занимает 2 байта.
- **Указатель срочности** (urgent pointer) занимает 2 байта и указывает на конец данных, которые необходимо срочно принять, несмотря на переполнение буфера. Указатель срочности используется совместно с кодовым битом URG. То есть если какие-то данные необходимо переслать приложению-получателю вне очереди, то приложение-отправитель должно сообщить об этом протоколу TCP путем установки в единицу бита URG.
- **Параметры** (options) имеют переменную длину и могут вообще отсутствовать. Максимальная величина поля составляет 3 байта; оно используется для решения вспомогательных задач, например для выбора максимального размера сегмента. Поле параметров может располагаться в конце заголовка TCP, а его длина кратна 8 бит.
- **Заполнитель** (padding) может иметь переменную длину. Это фиктивное поле, используемое для доведения размера заголовка до целого числа 32-битовых слов.

Логические соединения — основа надежности TCP

Основным отличием TCP от UDP является то, что на протокол TCP возложена дополнительная задача — обеспечить *надежную доставку сообщений*, используя в качестве основы *ненадежный дейтаграммный протокол IP*.

На рис. 19.5 показаны сети, соединенные маршрутизаторами, на которых установлен протокол IP. Установленные на конечных узлах протокольные модули TCP решают задачу обеспечения надежного обмена данными путем установле-

ния между собой **логических соединений**. Благодаря логическому соединению TCP следит, чтобы передаваемые сегменты не были потеряны, не были продублированы и пришли к получателю в том порядке, в котором были отправлены.

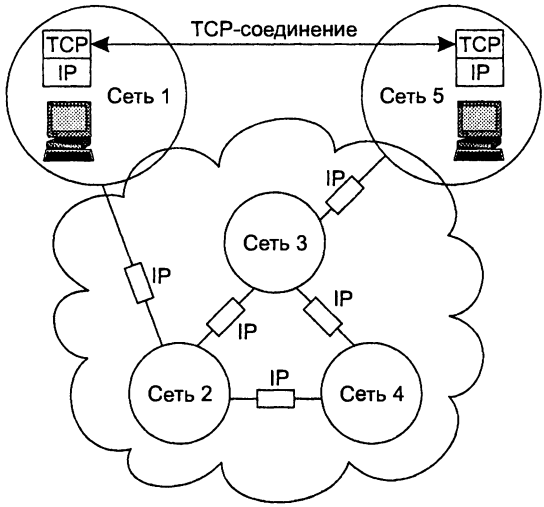


Рис. 19.5. TCP-соединение создает надежный логический канал между конечными узлами

При установлении логического соединения модули TCP договариваются между собой о параметрах процедуры обмена данными. В протоколе TCP каждая сторона соединения посылает противоположной стороне следующие параметры:

- максимальный размер сегмента, который она готова принимать;
- максимальный объем данных (возможно несколько сегментов), которые она разрешает другой стороне передавать в свою сторону, даже если та еще не получила квитанцию на предыдущую порцию данных (размер окна);
- начальный порядковый номер байта, с которого она начинает отсчет потока данных в рамках данного соединения.

В результате переговорного процесса модулей TCP с двух сторон соединения определяются параметры соединения. Одни из них остаются постоянными в течение всего сеанса связи, а другие адаптивно изменяются. В частности, в зависимости от загрузки буфера принимающей стороны, а также надежности работы сети динамически изменяется размер окна отправителя. Создание соединения означает также выделение операционной системой на каждой стороне соединения определенных системных ресурсов: для организации буферов, таймеров, счетчиков. Эти ресурсы будут закреплены за соединением с момента создания и до момента разрыва.

Логическое TCP-соединение однозначно идентифицируется парой сокетов.

Каждый сокет одновременно может участвовать в нескольких соединениях. Так, если (IP1, n1), (IP2, n2), (IP3, n3) — сокеты трех разных приложений, где IP1,

IP2, IP3 — их IP-адреса, а n1, n2, n3 — номера их TCP-портов, то возможно образование следующих соединений:

- соединение 1 — {(IP2, n2), (IP1, n1)};
- соединение 2 — {(IP1, n1), (IP3, n3)};
- соединение 3 — {(IP2, n2), (IP3, n3)}.

На рис. 19.6 показаны соединения 1 и 3, образованные сокетом (IP2, n2).

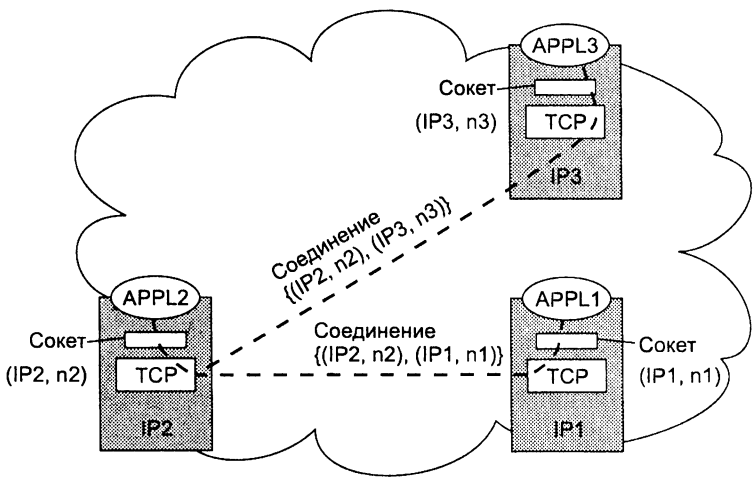


Рис. 19.6. Один сокет может участвовать в нескольких соединениях

А теперь рассмотрим на примере, как протокол TCP выполняет демультиплексирование. Пусть некий поставщик услуг оказывает услугу по веб-хостингу, то есть на его компьютере клиенты могут устанавливать свои веб-серверы. Веб-сервер основан на протоколе прикладного уровня HTTP, который, в свою очередь, использует протокол TCP. TCP ожидает запросы от веб-клиентов (браузеров), прослушивая хорошо известный порт 80.

На рис. 19.7 показан вариант хостинга с двумя веб-серверами — сервером www1.model.ru, имеющим IP-адрес IP1, и сервером www2.tour.ru с адресом IP2. К каждому из них может обращаться множество клиентов, причем клиенты могут одновременно работать как с сервером WWW1, так и с сервером WWW2. Работа каждого клиента требует сохранения прочитанных страниц, параметров и настроек сеанса связи, то есть *образования отдельного логического соединения*. Такое соединение создается протоколом TCP для каждой пары клиент-сервер.

На рисунке показаны два браузера, имеющие соответственно сокеты (IPk, nk) и (IPm, nm). Пользователь браузера k обращается одновременно к серверам WWW1 и WWW2. Наличие отдельных соединений для работы с каждым из этих серверов гарантирует разделение информационных потоков — у пользователя никогда не возникает вопроса, каким сервером ему была послана та или иная страница. Одновременно с пользователем браузера k с сервером WWW2 работает пользователь браузера m. И в этом случае отдельные логические соединения,

в рамках которых идет работа обоих пользователей, позволяют изолировать их информационные потоки. На рисунке показаны буферы, количество которых определяется не числом веб-серверов и не числом клиентов, а числом логических соединений. Сообщения в эти буферы направляются в зависимости от значений сокетов как отправителя, так и получателя.

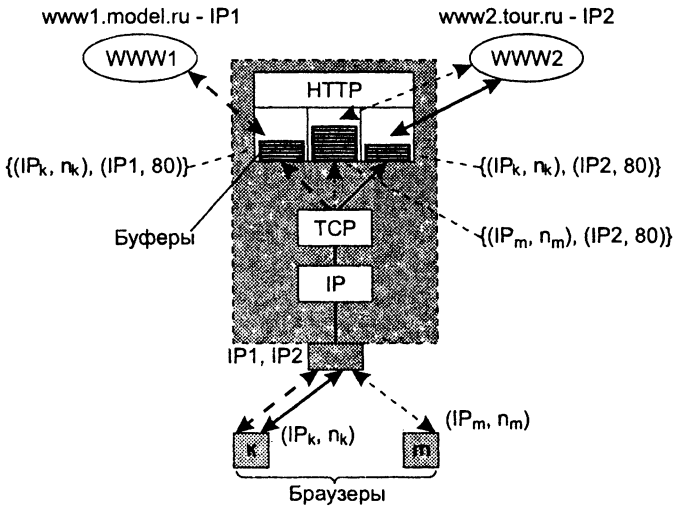


Рис. 19.7. Демультиплексирование TCP на основе соединений

Таким образом, протокол TCP осуществляет демультиплексирование на основе *соединений*.

Последовательный и подтвержденный номер

В рамках установленного соединения в протоколе TCP правильность передачи каждого сегмента должна подтверждаться квитанцией от получателя. *Квитирование* — это один из традиционных методов обеспечения надежной связи. В протоколе TCP используется частный случай квитирования — алгоритм скользящего окна.

Алгоритм скользящего окна в протоколе TCP имеет особенность. Хотя единицей передаваемых данных является *сегмент*, окно определено на множестве нумерованных *байтов* неструктурированного потока данных, поступающих с верхнего уровня и буферизуемых протоколом TCP.

При установлении соединения обе стороны договариваются о начальном номере байта, с которого будет вестись отсчет в течение всего данного соединения. У каждой стороны свой начальный номер. Идентификатором каждого сегмента является номер его первого байта. Нумерация байтов в пределах сегмента осуществляется так, что первый байт данных сразу вслед за заголовком имеет наименьший номер, а следующие за ним байты имеют следующие порядковые номера (рис. 19.8).

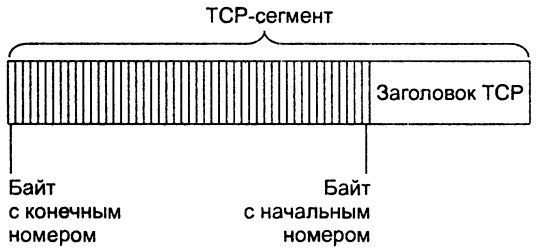


Рис. 19.8. Нумерация байтов в TCP-сегменте

Когда отправитель посылает TCP-сегмент, он в качестве идентификатора сегмента помещает в поле последовательного номера номер первого байта данного сегмента. Так, на рис. 19.9 идентификаторами сегментов являются номера 32600, 34060, 35520 и т. д. На основании этих номеров TCP-получатель не только отличает данный сегмент от других, но и позиционирует полученный фрагмент относительно общего потока байтов. Кроме того, он может сделать вывод, что полученный сегмент является дубликатом или что между двумя полученными сегментами пропущены данные и т. д.

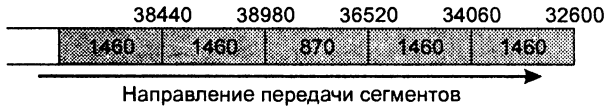


Рис. 19.9. Порядковый номер и номер квитанции

В качестве квитанции получатель сегмента отсылает ответное сообщение (сегмент), в которое помещает число (подтверждающий номер), на единицу превышающее максимальный номер байта в полученном сегменте. Для сегментов, изображенных на рис. 19.9, квитанцией о получении (подтвержденным номером) служат номера последнего байта каждого сегмента +1. Так для первого отправленного сегмента это будет число 34060, для второго — 35520 и т. д. Подтверждающий номер часто интерпретируют как номер следующего ожидаемого байта данных. Квитанция (подтверждение) в протоколе TCP посылается только в случае правильного приема данных, отрицательные квитанции не посылаются. Таким образом, отсутствие квитанции означает либо потерю сегмента, либо прием искаженного сегмента, либо потерю квитанции.

В протоколе TCP в одном и том же сегменте могут быть помещены и данные, которые посылает приложение другой стороне, и квитанция, которой модуль TCP подтверждает получение данных.

Окно приема

Протокол TCP является дуплексным, то есть в рамках одного соединения регламентируется процедура обмена данными в обе стороны. Каждая сторона одновременно выступает и как отправитель, и как получатель. У каждой стороны есть пара буферов: один — для хранения принятых сегментов, другой — для сег-

ментов, которые только еще предстоит отправить. Кроме того, имеется буфер для хранения копий сегментов, которые были отправлены, но квитанции о получении которых еще не поступили (рис. 19.10).

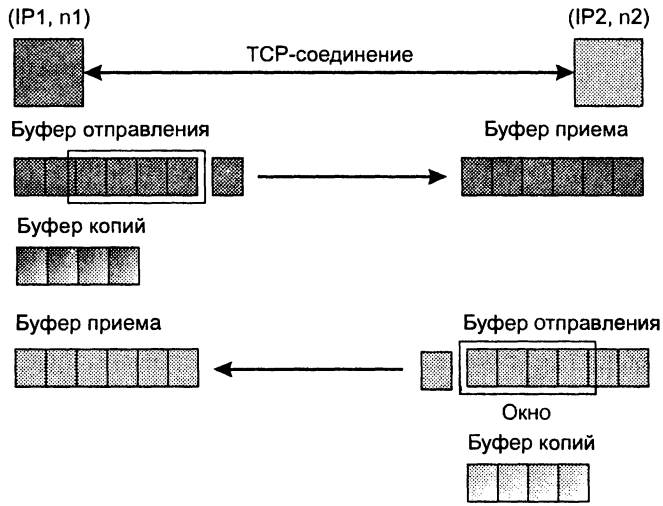


Рис. 19.10. Система буферов TCP-соединения

И при установлении соединения, и в ходе передачи обе стороны, выступая в роли получателя, посылают друг другу так называемые **окна приема**. Каждая из сторон, получив окно приема, «понимает», сколько байтов ей разрешается отправить с момента получения последней квитанции. Другими словами, посылая окна приема, обе стороны пытаются регулировать поток байтов в свою сторону, сообщая своему «визави», какое количество байтов (начиная с номера байта, о котором уже была выслана квитанция) они готовы в настоящий момент принять.

На рис. 19.11 показан поток байтов, поступающий с верхнего уровня в выходной буфер протокола TCP. Из потока байтов модуль TCP «нарезает» последовательность сегментов и готовит их для отправки другому сокету. Для определенности на рисунке принято направление перемещения данных справа налево. В этом потоке можно указать несколько логических границ. Первая граница отделяет сегменты, которые уже были отправлены и на которые уже пришли квитанции. По другую сторону этой границы располагается окно размером W байт. Часть байтов, входящих в окно, составляют сегменты, которые также уже отправлены, но квитанции на них пока не получены. Оставшаяся часть окна — это сегменты, которые пока не отправлены, но могут быть отправлены, так как входят в пределы окна. И наконец, последняя граница указывает на начало последовательности сегментов, ни один из которых не может быть отправлен до тех пор, пока не придет очередная квитанция и окно не будет сдвинуто вправо.

Если размер окна равен W, а последняя по времени квитанция содержала значение N, то отправитель может посылать новые сегменты до тех пор, пока в очередной сегмент не попадет байт с номером N + W. Этот сегмент выходит за рамки

окна, и передачу в таком случае необходимо приостановить до прихода следующей квитанции.

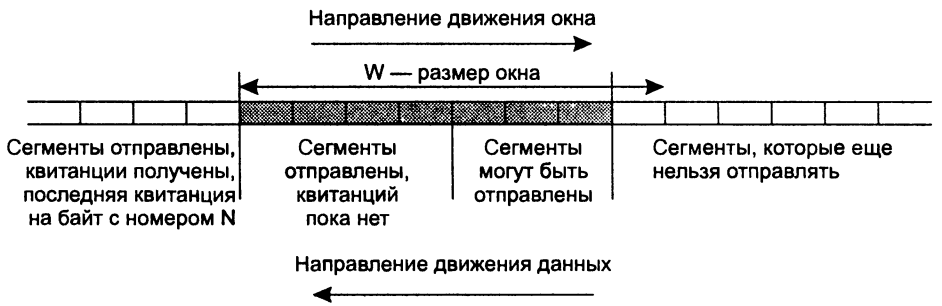


Рис. 19.11. Особенности реализации алгоритма скользящего окна в протоколе TCP

Накопительный принцип квитирования

Получатель может послать квитанцию, подтверждающую получение сразу нескольких сегментов, если они образуют непрерывный поток байтов. Например (рис. 19.12, *а*), если в буфер, плотно без пропусков заполненный потоком байтов до 2354 включительно, поочередно поступили сегменты (2355–3816), (3817–5275) и (5276–8400), где цифры в скобках означают номера первых и последних байтов каждого сегмента, то получателю достаточно отправить только одну квитанцию на все три сегмента, указав в ней в качестве номера квитанции значение 8401. Таким образом, процесс квитирования является накопительным.

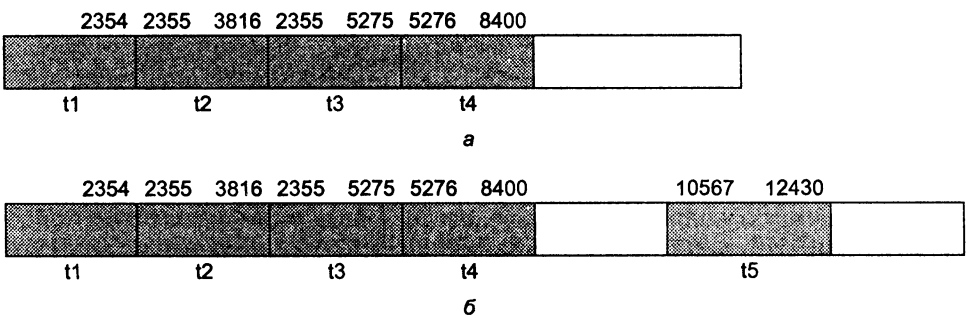


Рис. 19.12. Накопительный принцип квитирования: *а* — плотное заполнение буфера, в момент t4 передается квитанция, *б* — неплотное заполнение буфера, в момент t5 снова передается квитанция 8401

Однако сегменты могут прийти к получателю не в том порядке, в котором были посланы, то есть в приемном буфере может образоваться «прогалина» (рис. 19.12, *б*). Пусть, к примеру, после указанных выше трех сегментов вместо следующего по порядку сегмента (8401–10566) пришел сегмент (10567–12430). Очевидно, что послать в качестве номера квитанции значение 12431 нельзя, потому что это бы означало, что получены все байты вплоть до 12430. Поскольку в потоке байтов

образовался разрыв, получатель может только еще раз повторить квитанцию 8401, говоря тем самым, что все еще ожидает поступления потока байтов, начиная с 8401. Из этого примера видно, что, в отличие от многих других протоколов, протокол TCP подтверждает получение не отдельных блоков данных, а непрерывной последовательности байтов.

Время ожидания квитанции

Когда протокол TCP передает в сеть сегмент, он «на всякий случай» помещает его копию в очередь повторной передачи и запускает таймер. Когда приходит квитанция на этот сегмент, соответствующая копия удаляется из очереди. Если же квитанция не приходит до истечения срока, то сегмент посылается повторно. Может случиться так, что повторный сегмент придет тогда, когда исходный сегмент уже окажется на месте, тогда дубликат будет попросту отброшен.

Каким должно быть время ожидания (тайм-аут) очередной квитанции? От решения этой задачи зависит производительность протокола TCP. Тайм-аут не должен быть слишком коротким, чтобы по возможности исключить избыточные повторные передачи, снижающие полезную пропускную способность системы, но он не должен быть и слишком длинным, чтобы избежать длительных простоев, связанных с ожиданием несуществующей или «заблудившейся» квитанции.

При выборе величины тайм-аута должны учитываться скорость и надежность линий связи, их протяженность и многие другие факторы. В протоколе TCP тайм-аут определяется с помощью достаточно сложного адаптивного алгоритма, идея которого состоит в следующем. При каждой передаче засекается время от момента отправки сегмента до прихода квитанции о его приеме (время оборота). Получаемые значения времени оборота усредняются с весовыми коэффициентами, возрастающими от предыдущего замера к последующему. Это делается с тем, чтобы усилить влияние последних замеров. В качестве тайм-аута выбирается среднее время оборота, умноженное на некоторый коэффициент. Практика показывает, что значение этого коэффициента должно превышать 2. В сетях с большим разбросом времени оборота при выборе тайм-аута учитывается и дисперсия этой величины.

Управление окном приема

Размер окна приема связан с наличием в данный момент места в буфере данных у принимающей стороны. Поэтому в общем случае окна приема на разных концах соединения имеют разный размер. Например, можно ожидать, что сервер, вероятно обладающий большим буфером, пошлет клиентской станции окно приема больше, чем клиент серверу. В зависимости от состояния сети то одна, то другая сторона могут объявлять новые значения окон приема, динамически уменьшая и увеличивая их.

Варьируя величину окна, можно влиять на загрузку сети. Чем больше окно, тем бóльшая порция неподтвержденных данных может быть послана в сеть. Но если пришло бóльшее количество данных, чем может быть принято модулем TCP

данные будут отброшены. Это приведет к излишним пересылкам информации и ненужному увеличению нагрузки на сеть и модуль TCP.

С другой стороны, указание окна малого размера может ограничить передачу данных скоростью, которая определяется временем путешествия по сети каждого посылаемого сегмента. Чтобы избежать применения малых окон, в некоторых реализациях TCP предлагается получателю данных откладывать реальное изменение размеров окна до тех пор, пока свободное место не составит 20–40 % от максимально возможного объема памяти для этого соединения. Но и отправителю не стоит спешить с посылкой данных, пока окно принимающей стороны не станет достаточно большим. Учитывая эти соображения, разработчики протокола TCP предложили схему, согласно которой при установлении соединения заявляется большое окно, но впоследствии его размер существенно уменьшается. Существуют и другие прямо противоположные алгоритмы настройки окна, когда вначале выбирается минимальное окно, а затем, если сеть справляется с предложенной нагрузкой, его размер резко увеличивается.

Управлять размером окна приема может не только та сторона, которая посылает это окно, чтобы регулировать поток данных в свою сторону, но и вторая сторона — потенциальный отправитель данных. Если вторая сторона фиксирует ненадежную работу линии связи (регулярно запаздывают квитанции, часто требуется повторная передача), то она может по собственной инициативе уменьшить окно. В таких случаях действует правило: в качестве действующего размера окна выбирается минимальное из двух значений — значения, диктуемого приемной стороной, и значения, определяемого «на месте» отправителем.

Признаком перегрузки TCP-соединения является возникновение очередей на промежуточных узлах (маршрутизаторах) и на конечных узлах (компьютерах). При переполнении приемного буфера конечного узла «перегруженный» протокол TCP, отправляя квитанцию, помещает в нее новый, уменьшенный размер окна. Если он совсем отказывается от приема, то в квитанции указывается *окно нулевого размера*. Однако даже после этого приложение может послать сообщение на отказавшийся от приема порт. Для этого сообщение должно сопровождаться *указателем срочности*. В такой ситуации порт обязан принять сегмент, даже если для этого придется вытеснить из буфера уже находящиеся там данные. После приема квитанции с нулевым значением окна протокол-отправитель время от времени делает контрольные попытки продолжить обмен данными. Если протокол-приемник уже готов принимать информацию, то в ответ на контрольный запрос он посылает квитанцию с указанием ненулевого размера окна.

Как видно из нашего далеко не полного описания двух протоколов транспортного уровня стека TCP/IP, на один из них — TCP — возложена сложная и очень важная задача обеспечение надежной передачи данных через ненадежную сеть.

С другой стороны, функциональная простота протокола UDP обусловливает простоту алгоритма его работы, компактность и высокое быстродействие. Поэтому те приложения, в которых реализован собственный, достаточно надежный, механизм обмена сообщениями, основанный на установлении соединения, предпочитают для непосредственной передачи данных по сети использовать менее надежные, но более быстрые средства транспортировки, в качестве которых по

отношению к протоколу TCP и выступает протокол UDP. Протокол UDP может быть использован и в том случае, когда хорошее качество линий связи обеспечивает достаточный уровень надежности и без применения дополнительных приемов наподобие установления логического соединения и квитирования передаваемых пакетов. Заметим также, что поскольку протокол TCP основан на логических соединениях, он, в отличие от протокола UDP, *не годится для широковещательной и групповой рассылки.*

Протоколы маршрутизации

Список ключевых слов: время конвергенции, лавинная маршрутизация, маршрутизация, управляемая событиями, маршрутизация от источника, статическая маршрутизация, адаптивная маршрутизация, дистанционно-векторный алгоритм, алгоритм состояния связей, автономная система, внешний шлюз, внешний шлюзовой протокол, внутренний шлюзовой протокол, протокол BGPv4.

Классификация протоколов маршрутизации

Автоматически создаваемые таблицы маршрутизации обеспечивают рациональность маршрутов следования пакетов через сеть, при этом *критерии выбора маршрутов могут быть разными.* В IP-сетях сегодня применяются протоколы маршрутизации, в которых маршрут выбирается либо по критерию кратчайшего расстояния, где под расстоянием, проходимым пакетом, понимается количество промежуточных маршрутизаторов (хопов), либо по комплексному показателю, учитывающему также номинальную пропускную способность каналов между маршрутизаторами, надежность каналов или вносимые ими задержки.

Протокол маршрутизации должен создавать в маршрутизаторах *согласованные* друг с другом таблицы маршрутизации, то есть такие, которые обеспечат доставку пакета от исходной сети в сеть назначения за конечное число шагов. Можно представить и несогласованную пару таблиц, когда таблица маршрутизатора 1 показывает, что пакет для сети А нужно передать маршрутизатору 2, а таблица маршрутизатора 2 отправляет этот же пакет маршрутизатору 1. Современные протоколы маршрутизации обеспечивают согласованность таблиц, однако это их свойство не абсолютно — при изменениях в сети, например, при отказе каналов передачи данных или самих маршрутизаторов, возникают периоды нестабильной работы сети, вызванной временной несогласованностью таблиц разных маршрутизаторов. Протоколу маршрутизации обычно нужно некоторое время, которое называется **временем конвергенции**, чтобы после нескольких итераций обмена служебной информацией все маршрутизаторы сети внесли изменения в свои таблицы и в результате таблицы снова стали согласованными. Различные протоколы маршрутизации обладают *разным временем конвергенции.*

Следуя принципу масштабируемости, маршрутизация в Интернете функционирует в пределах автономных систем (Autonomous Systems, AS).

В результате маршрутизация в Интернете носит ярко выраженный *иерархический* характер. Внутри каждой автономной системы может применяться любой из существующих протоколов маршрутизации, в то время как между автономными системами всегда применяется один и тот же протокол, являющийся своеобразным языком «эсперанто», на котором автономные системы общаются между собой.

В IP-сетях в качестве внутренних шлюзовых протоколов, то есть протоколов, применяемых внутри автономных систем, сегодня активно используются три протокола — RIP, OSPF и IS-IS. Внешним шлюзовым протоколом, то есть протоколом выбора маршрута между автономными системами, сегодня является протокол BGP.

Маршрутизация без таблиц

Переходя к классификации протоколов маршрутизации, прежде всего, отметим, что существуют такие способы продвижения пакетов в составных сетях, которые вообще *не требуют наличия таблиц маршрутизации в маршрутизаторах.*

Наиболее простым способом передачи пакетов по сети является так называемая **лавинная маршрутизация**, когда каждый маршрутизатор передает пакет всем своим непосредственным соседям, кроме того, от которого его получил. Понятно, что это не самый рациональный способ, так как пропускная способность сети используется крайне расточительно, но он работоспособен (именно так мосты и коммутаторы локальных сетей поступают с кадрами, имеющими неизвестные адреса).

Другой вариант маршрутизации без таблицы — это **маршрутизация, управляемая событиями** (Event dependent routing), когда пакет к определенной сети назначения посылается по маршруту, уже приводившему ранее к успеху (для данного адреса назначения). Подобный метод маршрутизации использовался на этапе становления Интернета, когда перед отправкой пакета данных всем или нескольким соседям посылались эхо-запросы протокола ICMP, а затем на основании времени пришедших эхо-ответов выбирался сосед с минимальным временем отклика. Данный способ хорош для сетей, работающих по протоколам с предварительным установлением соединения. Запрос на установление соединения может посылаться нескольким соседям, а подтверждение установления соединения посылается тому соседу, который первым пришлет ответ.

Еще одним видом маршрутизации, не требующим наличия таблиц маршрутизации, является **маршрутизация от источника** (source routing). В этом случае отправитель помещает в пакет информацию о том, какие промежуточные маршрутизаторы должны участвовать в передаче пакета к сети назначения. На основе этой информации каждый маршрутизатор считывает адрес следующего маршрутизатора и, если он действительно является адресом его непосредственного соседа, передает ему пакет для дальнейшей обработки. Вопрос о том, как отправитель узнает точный маршрут следования пакета через сеть, остается открытым. Маршрут может задавать либо вручную администратор, либо автоматически узел-отправитель, но в этом случае ему нужно поддерживать какой-либо протокол маршрутизации, который сообщит ему о топологии и состоянии сети. Маршрутизация от источника опробовалась на этапе зарождения Интернета и сохра-

нилась как практически неиспользуемая возможность протокола IPv4. В IPv6 маршрутизация от источника является одним из стандартных режимов продвижения пакетов, существует даже специальный заголовок для реализации этого режима.

Адаптивная маршрутизация

В тех случаях, когда маршрутизация осуществляется на основании таблиц, различают *статическую* и *адаптивную* (*динамическую*) маршрутизацию.

При **статической маршрутизации** таблицы составляются и вводятся в память каждого маршрутизатора *вручную администратором сети*. Все записи в таблице имеют статус статических, что подразумевает бесконечный срок их жизни. При существенном изменении состояния сети администратору необходимо срочно внести изменения в соответствующие таблицы маршрутизации, иначе сеть будет работать некорректно.

При **адаптивной маршрутизации** все изменения конфигурации сети *автоматически* отображаются в таблицах маршрутизации *протоколами маршрутизации*. Эти протоколы основаны на сборе информации о топологии связей в сети, что позволяет им оперативно отрабатывать все текущие изменения. В таблицах маршрутизации при адаптивной маршрутизации обычно имеется информация об интервале времени, в течение которого данный маршрут будет оставаться действительным. Это время называют *временем жизни* (TTL) маршрута. Если по истечении времени жизни существование маршрута не подтверждается протоколом маршрутизации, то он считается нерабочим, пакеты по нему больше не посылаются.

Протоколы маршрутизации могут быть распределенными и централизованными.

При *распределенном* подходе в сети отсутствуют какие-либо выделенные маршрутизаторы, которые собирали бы и обобщали топологическую информацию: эта работа распределяется между всеми маршрутизаторами сети. Каждый маршрутизатор строит собственную таблицу маршрутизации, основываясь на данных, получаемых по протоколу маршрутизации от остальных маршрутизаторов сети.

При *централизованном* подходе в сети существует один маршрутизатор, который собирает всю информацию о топологии и состоянии сети от других маршрутизаторов. Затем этот выделенный маршрутизатор (который иногда называют *сервером маршрутов*) может построить таблицы маршрутизации для всех остальных маршрутизаторов сети, а затем распространить их по сети, чтобы каждый маршрутизатор получил собственную таблицу и в дальнейшем самостоятельно принимал решение о продвижении каждого пакета.

Применяемые сегодня в IP-сетях протоколы маршрутизации относятся к *адаптивным распределенным* протоколам.

Адаптивные алгоритмы маршрутизации должны отвечать нескольким важным требованиям. Во-первых, они должны обеспечивать *рациональность маршрута*.

Во-вторых, *алгоритмы должны быть достаточно простыми*, они не должны требовать слишком большого объема вычислений или порождать интенсивный служебный трафик. И наконец, алгоритмы маршрутизации должны обладать *свойством сходимости*, то есть всегда приводить к согласованному построению таблиц маршрутизации на всех маршрутизаторах сети за приемлемое время.

Адаптивные протоколы обмена маршрутной информацией, применяемые в настоящее время в вычислительных сетях, делятся на две группы:

- дистанционно-векторные алгоритмы (Distance Vector Algorithms, DVA);
- алгоритмы состояния связей (Link State Algorithms, LSA).

Дистанционно-векторные алгоритмы

В **дистанционно-векторных алгоритмах** (DVA) каждый маршрутизатор *периодически* и *широковещательно* рассылает по сети вектор, компонентами которого являются расстояния от данного маршрутизатора до всех известных ему сетей. Пакеты протоколов маршрутизации обычно называют *объявлениями*, так как с их помощью маршрутизатор объявляет остальным маршрутизаторам известные ему сведения о конфигурации сети. Расстояние в DVA обычно измеряют в числе хопов. Возможна и другая метрика, учитывающая не только число промежуточных маршрутизаторов, но и пропускную способность между соседними маршрутизаторами.

Получив от некоторого соседа вектор расстояний до известных тому сетей, маршрутизатор наращивает компоненты вектора на величину расстояния от себя до данного соседа. Кроме того, он дополняет вектор информацией об известных ему самому других сетях, о которых он узнал непосредственно (если они подключены к его портам) или из аналогичных объявлений других маршрутизаторов. Затем он снова рассылает новое значение вектора по сети. В конце концов, каждый маршрутизатор узнает через соседние маршрутизаторы информацию обо всех имеющихся в составной сети сетях и о расстояниях до них.

Затем он выбирает из нескольких альтернативных маршрутов к каждой сети тот маршрут, который обладает наименьшей метрикой. Ближайший маршрутизатор, который передал информацию о данном маршруте, отмечается в таблице маршрутизации как *следующий* (next hop).

Дистанционно-векторные алгоритмы хорошо работают только в небольших сетях. В больших сетях они засоряют линии связи интенсивным периодическим трафиком, к тому же изменения конфигурации не всегда корректно могут отрабатываться алгоритмом этого типа, так как маршрутизаторы не имеют точного представления о топологии связей в сети, а располагают только обобщенной информацией — вектором дистанций.

Наиболее распространенным протоколом, основанным на дистанционно-векторном алгоритме, является протокол RIP, который распространен в двух версиях — версия RIP IP работает с протоколом IP, а версия RIP IPX работает с протоколом IPX.

Алгоритмы состояния связей

Алгоритмы состояния связей (LSA) обеспечивают каждый маршрутизатор информацией, достаточной для построения точного графа связей сети. Все маршрутизаторы работают на основании одного и того же графа, что делает процесс маршрутизации более устойчивым к изменениям конфигурации.

Каждый маршрутизатор использует граф сети для нахождения оптимальных по некоторому критерию маршрутов до каждой из сетей, входящих в составную сеть.

Чтобы понять, в каком состоянии находятся линии связи, подключенные к его портам, маршрутизатор периодически обменивается короткими пакетами HELLO со своими ближайшими соседями.

В отличие от протоколов DVA объявления о состоянии связей протоколов LSA не повторяются периодически, а передаются только в том случае, когда с помощью сообщений HELLO было установлено изменение состояния какой-либо связи. В результате служебный трафик, создаваемый протоколами LSA, гораздо менее интенсивный, чем у протоколов DVA.

Протоколами, основанными на алгоритме состояния связей, являются протокол IS-IS стека OSI (этот протокол используется также в стеке TCP/IP), протокол OSPF стека TCP/IP и протокол NLSP стека Novell.

Применение нескольких протоколов маршрутизации

В одной и той же сети могут одновременно работать несколько разных протоколов маршрутизации (рис. 19.13). Это означает, что на некоторых (не обязательно всех) маршрутизаторах сети установлено и функционирует несколько протоколов маршрутизации, но при этом, естественно, через сеть взаимодействуют только одноименные протоколы. То есть если маршрутизатор 1 поддерживает, например, протоколы RIP и OSPF, маршрутизатор 2 — только RIP, а маршрутизатор 3 — только OSPF, то маршрутизатор 1 будет взаимодействовать с маршрутизатором 2 по протоколу RIP, с маршрутизатором 2 — по OSPF, а маршрутизаторы 2 и 3 вообще непосредственно друг с другом взаимодействовать не могут.

В маршрутизаторе, который поддерживает одновременно несколько протоколов, каждая запись в таблице является результатом работы одного из этих протоколов. Если о некоторой сети появляется информация от нескольких протоколов, то для однозначности выбора маршрута (а данные разных протоколов могут вести к разным рациональным маршрутам) устанавливаются приоритеты протоколов маршрутизации. Обычно предпочтение отдается протоколам LSA, как располагающим более полной информацией о сети по сравнению с протоколами DVA. В некоторых ОС в формах вывода на экран и печать в каждой записи таблицы маршрутизации имеется отметка о том, с помощью какого протокола маршрутизации эта запись получена. Но даже если эта отметка на экран и не выводится, она обязательно имеется во внутреннем представлении таблицы маршрутизации. По умолчанию каждый протокол маршрутизации, работающий на определенном маршрутизаторе, распространяет только ту информацию, которая была получена

этим маршрутизатором по данному протоколу. Таким образом, если о маршруте к некоторой сети маршрутизатор узнал от протокола RIP, то и распространять по сети объявления об этом маршруте он будет с помощью протокола RIP.

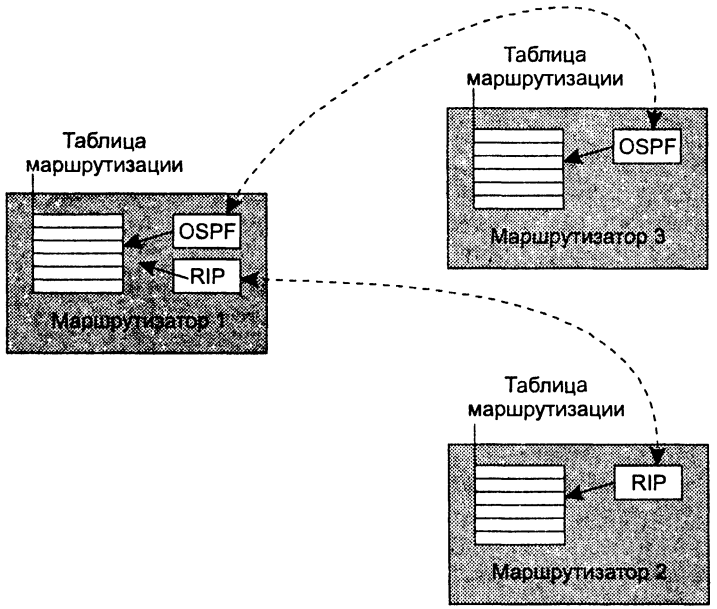


Рис. 19.13. Работа нескольких протоколов маршрутизации в одной сети

Однако возникает вопрос: «Каким образом маршрутизаторы, поддерживающие разные протоколы, обмениваются между собой маршрутной информацией, делая тем самым достижимыми все сети составной сети?» Для этого необходимо установить особый внутренний режим работы, называемый часто *перераспределением*. Такой режим позволяет одному протоколу маршрутизации использовать не только «свои», но и «чужие» записи таблицы маршрутизации, полученные с помощью другого протокола маршрутизации, указанного при конфигурировании.

Как видим, применение нескольких протоколов маршрутизации даже в пределах небольшой составной сети не является простым делом, от администратора требуется провести определенную работу по конфигурированию каждого маршрутизатора. Очевидно, что для крупных составных сетей требуется качественно иное решение. Сейчас мы перейдем к рассмотрению такого решения, которое было найдено для самой крупной на сегодня составной сети — для Интернета.

Внешние и внутренние шлюзовые протоколы

Интернет обладает не только организационной структурой, определяющей деление Интернета на сети различных поставщиков услуг (ISP). Интернет состоит также из автономных систем.

Автономная система — это совокупность сетей под единым административным управлением, обеспечивающим общую для всех входящих в автономную систему

маршрутизаторов политику маршрутизации. Обычно автономной системой управляет один поставщик услуг Интернета, самостоятельно выбирая, какие протоколы маршрутизации должны использоваться в некоторой автономной системе и каким образом между ними должно выполняться перераспределение маршрутной информации. Крупные поставщики услуг и корпорации могут представить свою составную сеть как набор нескольких автономных систем. Регистрация автономных систем происходит централизованно, как и регистрация IP-адресов и DNS-имен. Номер автономной системы состоит из 16 разрядов и никак не связан с префиксами IP-адресов сетей, входящих в нее.

В соответствии с этой концепцией Интернет выглядит как набор взаимосвязанных автономных систем, каждая из которых состоит из взаимосвязанных сетей (рис. 19.14).

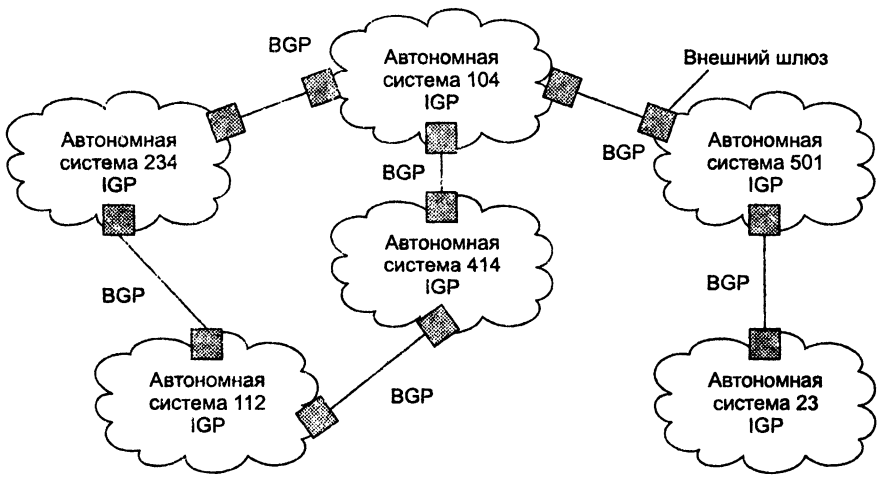


Рис. 19.14. Автономные системы Интернета

Основная цель деления Интернета на автономные системы — обеспечение многоуровневого подхода к маршрутизации. До введения автономных систем предполагался двухуровневый подход — то есть сначала *маршрут определялся как последовательность сетей*, а затем вел непосредственно к заданному узлу в конечной сети.

С появлением автономных систем появляется третий, верхний, уровень маршрутизации — теперь сначала *маршрут определяется как последовательность автономных систем*, затем — *как последовательность сетей*, а уж затем ведет к конечному узлу.

Автономные системы соединяются **внешними шлюзами**[1]. Что важно, между внешними шлюзами разрешается использовать только один протокол маршрутизации,

[1] Далее мы будем использовать термины «маршрутизатор» и «шлюз» как синонимы, чтобы отдать дань уважения традиционной терминологии Интернета и в то же время не забывать более современные термины.

причем не произвольный, а тот, который в данное время признается сообществом Интернета в качестве стандартного для внешних шлюзов. Такой протокол маршрутизации называется **внешним шлюзовым протоколом** (Exterior Gateway Protocol, EGP) и в настоящее время им является протокол BGP версии 4 (BGPv4). Все остальные протоколы (например, RIP, OSPF, IS-IS) являются **внутренними шлюзовыми протоколами** (Interior Gateway Protocols, IGP).

Внешний шлюзовой протокол отвечает за выбор *маршрута между автономными системами*. В качестве адреса следующего маршрутизатора указывается адрес точки входа в соседнюю автономную систему.

Внутренние шлюзовые протоколы отвечают за *маршрут внутри автономной системы*. В случае транзитной автономной системы эти протоколы указывают точную последовательность маршрутизаторов от точки входа в автономную систему до точки выхода из нее.

Автономные системы составляют магистраль Интернета. Концепция автономных систем скрывает от администраторов магистрали Интернета проблемы маршрутизации пакетов на более низком уровне — уровне сетей. Для администратора магистрали неважно, какие протоколы маршрутизации применяются внутри автономных систем, для него существует единственный протокол маршрутизации — **BGPv4**.

Протокол RIP

Список ключевых слов: протокол RIP, истечение времени жизни маршрута, тайм-аут, расщепление горизонта, триггерные обновления, замораживание изменений.

Протокол RIP (Routing Information Protocol — протокол маршрутной информации) является внутренним протоколом маршрутизации дистанционно-векторного типа, он представляет собой один из наиболее ранних протоколов обмена маршрутной информацией и до сих пор чрезвычайно распространен в вычислительных сетях ввиду простоты реализации.

Построение таблицы маршрутизации

Для IP имеются две версии RIP — RIPv1 и RIPv2. Протокол RIPv1 не поддерживает масок. Протокол RIPv2 передает информацию о масках сетей, поэтому он в большей степени соответствует требованиям сегодняшнего дня. Так как построение таблиц маршрутизации в обеих версиях 1 и 2 принципиально не отличается, в дальнейшем для упрощения записей будет описываться работа версии 1.

Для измерения расстояния до сети стандарты протокола RIP допускают различные виды метрик: хопы, значения пропускной способности, вносимые задержки, надежность сетей (то есть соответствующие признакам D, T и R в поле качества сервиса IP-пакета), а также любые комбинации этих метрик. Метрика должна обладать свойством *аддитивности* — метрика составного пути должна быть рав-

на сумме метрик составляющих этого пути. В большинстве реализаций RIP используется простейшая метрика — количество хопов, то есть количество промежуточных маршрутизаторов, которые нужно преодолеть пакету до сети назначения.

Рассмотрим процесс построения таблицы маршрутизации с помощью протокола RIP на примере составной сети, изображенной на рис. 19.15.

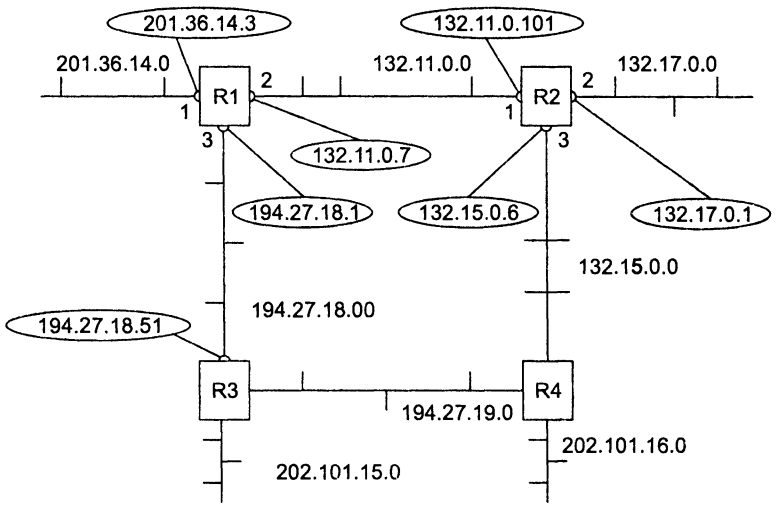


Рис. 19.15. Сеть, построенная на RIP-маршрутизаторах

Этап 1 — создание минимальной таблицы

В этой сети имеется восемь IP-сетей, связанных четырьмя маршрутизаторами с идентификаторами: R1, R2, R3 и R4. Маршрутизаторы, работающие по протоколу RIP, могут иметь идентификаторы, однако для протокола они не являются необходимыми. В RIP-сообщениях эти идентификаторы не передаются.

В исходном состоянии в каждом маршрутизаторе программным обеспечением стека TCP/IP автоматически создается минимальная таблица маршрутизации, в которой учитываются только непосредственно подсоединенные сети. На рисунке адреса портов маршрутизаторов в отличие от адресов сетей помещены в овалы.

Таблица 19.1 позволяет оценить примерный вид минимальной таблицы маршрутизации маршрутизатора R1.

Таблица 19.1. Минимальная таблица маршрутизации маршрутизатора R1

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
201.36.14.0	201.36.14.3	1	1
132.11.0.0	132.11.0.7	2	1
194.27.18.0	194.27.18.1	3	1

Минимальные таблицы маршрутизации в других маршрутизаторах будут выглядеть соответственно, например, таблица маршрутизатора R2 будет состоять из трех записей (табл. 19.2).

Таблица 19.2. Минимальная таблица маршрутизации маршрутизатора R2

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
132.11.0.0	132.11.0.101	1	1
132.17.0.0	132.17.0.1	2	1
132.15.0.0	132.15.0.6	3	1

Этап 2 — рассылка минимальной таблицы соседям

После инициализации каждого маршрутизатора он начинает посылать своим соседям сообщения протокола RIP, в которых содержится его минимальная таблица.

RIP-сообщения передаются в дейтаграммах протокола UDP и включают два параметра для каждой сети: ее IP-адрес и расстояние до нее от передающего сообщение маршрутизатора.

Соседями являются маршрутизаторы, которым данный маршрутизатор может передать IP-пакет по какой-либо своей сети, не пользуясь услугами промежуточных маршрутизаторов. Например, для маршрутизатора R1 соседями являются маршрутизаторы R2 и R3, а для маршрутизатора R4 — маршрутизаторы R2 и R3.

Таким образом, маршрутизатор R1 передает маршрутизаторам R2 и R3 следующие сообщения:

- сеть 201.36.14.0, расстояние 1;
- сеть 132.11.0.0, расстояние 1;
- сеть 194.27.18.0, расстояние 1.

Этап 3 — получение RIP-сообщений от соседей и обработка полученной информации

После получения аналогичных сообщений от маршрутизаторов R2 и R3 маршрутизатор R1 наращивает каждое полученное поле метрики на единицу и запоминает, через какой порт и от какого маршрутизатора получена новая информация (адрес этого маршрутизатора станет адресом следующего маршрутизатора, если эта запись будет внесена в таблицу маршрутизации). Затем маршрутизатор начинает сравнивать новую информацию с той, которая хранится в его таблице маршрутизации (табл. 19.3).

Записи с четвертой по девятую получены от соседних маршрутизаторов, и они претендуют на помещение в таблицу. Однако только записи с четвертой по седьмую попадают в таблицу, а записи восьмая и девятая — нет. Это происходит потому, что они содержат данные об уже имеющихся в таблице маршрутизатора R1 сетях, а расстояние до них больше, чем в существующих записях.

Таблица 19.3. Таблица маршрутизации маршрутизатора R1

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
201.36.14.0	201.36.14.3	1	1
132.11.0.0	132.11.0.7	2	1
194.27.18.0	194.27.18.1	3	1
132.17.0.0	132.11.0.101	2	2
132.15.0.0	132.11.0.101	2	2
194.27.19.0	194.27.18.51	3	2
202.101.15.0	194.27.18.51	3	2
~~132.11.0.0~~	~~132.11.0.101~~	~~2~~	~~2~~
~~194.27.18.0~~	~~194.27.18.51~~	~~3~~	~~2~~

Протокол RIP замещает запись о какой-либо сети только в том случае, если новая информация имеет лучшую метрику (расстояние в хопах меньше), чем имеющаяся. В результате в таблице маршрутизации о каждой сети остается только одна запись; если же имеется несколько записей, равнозначных в отношении расстояния путей к одной и той же сети, то все равно в таблице остается одна запись, которая пришла в маршрутизатор первая по времени. Для этого правила существует исключение — если худшая информация о какой-либо сети пришла от того же маршрутизатора, на основании сообщения которого была создана данная запись, то худшая информация замещает лучшую.

Аналогичные операции с новой информацией выполняют и остальные маршрутизаторы сети.

Этап 4 — рассылка новой таблицы соседям

Каждый маршрутизатор отсылает новое RIP-сообщение всем своим соседям. В этом сообщении он помещает данные обо всех известных ему сетях — как непосредственно подключенных, так и удаленных, о которых маршрутизатор узнал из RIP-сообщений.

Этап 5 — получение RIP-сообщений от соседей и обработка полученной информации

Этап 5 повторяет этап 3 — маршрутизаторы принимают RIP-сообщения, обрабатывают содержащуюся в них информацию и на ее основании корректируют свои таблицы маршрутизации.

Посмотрим, как это делает маршрутизатор R1 (табл. 19.4).

На этом этапе маршрутизатор R1 получает от маршрутизатора R3 информацию о сети 132.15.0.0, которую тот в свою очередь на предыдущем цикле работы по-

лучил от маршрутизатора R4. Маршрутизатор уже знает о сети 132.15.0.0, причем старая информация имеет лучшую метрику, чем новая, поэтому новая информация об этой сети отбрасывается.

Таблица 19.4. Таблица маршрутизации маршрутизатора R1

Номер сети	Адрес следующего маршрутизатора	Порт	Расстояние
201.36.14.0	201.36.14.3	1	1
132.11.0.0	132.11.0.7	2	1
194.27.18.0	194.27.18.1	3	1
132.17.0.0	132.11.0.101	2	2
132.15.0.0	132.11.0.101	2	2
~~132.15.0.0~~	~~194.27.18.51~~	~~3~~	~~3~~
194.27.19.0	194.27.18.51	3	2
~~194.27.19.0~~	~~132.11.0.101~~	~~2~~	~~3~~
~~202.101.15.0~~	~~194.27.18.51~~	~~3~~	~~2~~
202.101.16.0	132.11.0.101	2	3
~~202.101.16.0~~	~~194.27.18.51~~	~~3~~	~~3~~

О сети 202.101.16.0 маршрутизатор R1 узнает на этом этапе впервые, причем данные о ней приходят от двух соседей — от R3 и R4. Поскольку метрики в этих сообщениях указаны одинаковые, то в таблицу попадают данные, пришедшие первыми. В нашем примере считается, что маршрутизатор R2 опередил маршрутизатор R3 и первым переслал свое RIP-сообщение маршрутизатору R1.

Если маршрутизаторы периодически повторяют этапы рассылки и обработки RIP-сообщений, то за конечное время в сети установится корректный режим маршрутизации. Под корректным режимом маршрутизации здесь понимается такое состояние таблиц маршрутизации, когда все сети достижимы из любой сети с помощью некоторого рационального маршрута. Пакеты будут доходить до адресатов и не зацикливаться в петлях, подобных той, которая образуется на рис. 19.15, маршрутизаторами R1, R2, R3 и R4.

Очевидно, если в сети все маршрутизаторы, их интерфейсы и соединяющие их линии связи остаются работоспособными, то объявления по протоколу RIP можно делать достаточно редко, например один раз в день. Однако в сетях постоянно происходят изменения — меняется работоспособность маршрутизаторов и линий связи, кроме того, маршрутизаторы и линии связи могут добавляться в существующую сеть или же выводиться из ее состава.

Для адаптации к изменениям в сети протокол RIP использует ряд механизмов.

Адаптация RIP-маршрутизаторов к изменениям состояния сети

К новым маршрутам RIP-маршрутизаторы приспосабливаются просто — они передают новую информацию в очередном сообщении своим соседям и постепенно эта информация становится известна всем маршрутизаторам сети. А вот к изменениям, связанным с потерей какого-либо маршрута, RIP-маршрутизаторы адаптируются сложнее. Это связано с тем, что в формате сообщений протокола RIP нет поля, которое бы указывало на то, что путь к данной сети больше не существует.

Для уведомления о том, что некоторый маршрут недействителен, используются два механизма:

- истечение времени жизни маршрута;
- указание специального (бесконечного) расстояния до сети, ставшей недоступной.

Механизм **истечения времени жизни маршрута** основан на том, что каждая запись таблицы маршрутизации (как и записи таблицы продвижения моста/коммутатора), полученная по протоколу RIP, имеет время жизни (TTL). При поступлении очередного RIP-сообщения, которое подтверждает справедливость данной записи, таймер TTL устанавливается в исходное состояние, а затем из него каждую секунду вычитается единица. Если за время тайм-аута не придет новое сообщение об этом маршруте, он помечается как *недействительный*.

Время тайм-аута связано с периодом рассылки векторов по сети. В протоколе RIP IP период рассылки выбран равным 30 секундам, а в качестве тайм-аута выбрано шестикратное значение периода рассылки, то есть 180 секунд. Шестикратный запас времени нужен для уверенности в том, что сеть действительно стала недоступной, а не просто произошли потери RIP-сообщений (а это возможно, так как протокол RIP использует транспортный протокол UDP, который не обеспечивает надежной доставки сообщений). Если какой-либо маршрутизатор отказывает, переставая слать своим соседям сообщения о сетях, которые можно достичь через него, то через 180 секунд все записи, порожденные этим маршрутизатором, станут недействительными у его ближайших соседей. После этого процесс повторится уже для соседей ближайших соседей — они вычеркнут подобные записи уже через 360 секунд.

Как видно, сведения о недоступных через отказавший маршрутизатор сетях распространяются по сети не очень быстро. В этом заключается одна из причин выбора в качестве периода рассылки небольшой величины в 30 секунд. Механизм **тайм-аута** работает в тех случаях, когда маршрутизатор не может послать соседям сообщение об отказавшем маршруте, так как либо он сам неработоспособен, либо неработоспособна линия связи, по которой можно было бы передать сообщение.

Когда же сообщение послать можно, RIP-маршрутизаторы используют прием, заключающий в *указании бесконечного расстояния до сети, ставшей недоступной*. В протоколе RIP бесконечным условно считается расстояние равным 16 хопов. Получив сообщение, в котором расстояние до некоторой сети равно 16 (или 15,

что приводит к тому же результату, так как маршрутизатор наращивает полученное значение на 1), маршрутизатор должен проверить, исходит ли эта «плохая» информация о сети от того же маршрутизатора, сообщение которого послужило в свое время основанием для записи о данной сети в таблице маршрутизации. Если это тот же маршрутизатор, то информация считается достоверной и маршрут помечается как недоступный.

То, что за «бесконечное» расстояние принято столь небольшое число, вызвано тем, что в некоторых случаях отказы связей в сети вызывают длительные периоды некорректной работы RIP-маршрутизаторов, выражающейся в зацикливании пакетов в петлях сети. И чем меньше расстояние, используемое в качестве «бесконечного», тем такие периоды короче.

Пример

Рассмотрим случай зацикливания пакетов на примере сети, изображенной на рис. 19.15. Пусть маршрутизатор R1 обнаружил, что его связь с непосредственно подключенной сетью 201.36.14.0 потеряна (например, по причине отказа интерфейса 201.36.14.3). Маршрутизатор R1 отмечает в своей таблице маршрутизации, что сеть 201.36.14.0 недоступна. В худшем случае он обнаружит это сразу же после отправки очередных RIP-сообщений, так что до начала нового цикла его объявлений, в котором он должен сообщить соседям, что расстояние до сети 201.36.14.0 стало равным 16, остается почти 30 секунд. Каждый маршрутизатор работает на основании своего внутреннего таймера, не синхронизируя работу по рассылке объявлений с другими маршрутизаторами. Поэтому весьма вероятно, маршрутизатор R2 опередит маршрутизатор R1 и передаст ему свое сообщение раньше, чем R1 успеет передать новость о недостижимости сети 201.36.14.0. А в этом сообщении имеются данные, порожденные записью в таблице маршрутизации R2 (табл. 19.5).

Таблица 19.5. Таблица маршрутизации маршрутизатора R2

Номер сети	Адрес след. маршрутизатора	Порт	Расстояние
201.36.14.0	132.11.0.7	1	2

Эта запись, полученная от маршрутизатора R1, была корректна до отказа интерфейса 201.36.14.3; теперь она устарела, но маршрутизатор R2 об этом не знает.

Далее маршрутизатор R1 получает новую информацию о сети 201.36.14.0 — эта сеть достижима через маршрутизатор R2 с метрикой 2. Раньше R1 также получал эту информацию от R2, но игнорировал ее, так как его собственная метрика для 201.36.14.0 была лучше. Теперь R1 должен принять данные о сети 201.36.14.0, полученные от R2, и заменить запись в таблице маршрутизации о недостижимости этой сети (табл. 19.6).

Таблица 19.6. Таблица маршрутизации маршрутизатора R1

Номер сети	Адрес след. маршрутизатора	Порт	Расстояние
201.36.14.0	132.11.0.101	2	3

В результате в сети образуется маршрутная петля: пакеты, направляемые узлам сети 201.36.14.0, станут передаваться маршрутизатором R2 маршрутизатору R1, а маршру-

тизатор R1 будет возвращать их маршрутизатору R2. IP-пакеты будут циркулировать по этой петле до тех пор, пока не истечет время жизни каждого пакета. Рассмотрим периоды времени, кратные времени жизни записей в таблицах маршрутизаторов.

Время 0–180 с. После отказа интерфейса в маршрутизаторах R1 и R2 будут сохраняться некорректные записи. Маршрутизатор R2 по-прежнему снабжает маршрутизатор R1 своей записью о сети 201.36.14.0 с метрикой 2, так как ее время жизни не истекло. Пакеты зацикливаются.

Время 180–360 с. В начале этого периода у маршрутизатора R2 истекает время жизни записи о сети 201.36.14.0 с метрикой 2, так как маршрутизатор R1 в предыдущий период посылал ему сообщения о сети 201.36.14.0 с худшей метрикой, чем у R2, и они не могли подтверждать эту запись. Теперь маршрутизатор R2 принимает от маршрутизатора R1 запись о сети 201.36.14.0 с метрикой 3 и трансформирует ее в запись с метрикой 4. Маршрутизатор R1 не получает новых сообщений от маршрутизатора R2 о сети 201.36.14.0 с метрикой 2, поэтому время жизни его записи начинает уменьшаться. Пакеты продолжают зацикливаться.

Время 360–540 с. Теперь у маршрутизатора R1 истекает время жизни записи о сети 201.36.14.0 с метрикой 3. Маршрутизаторы R1 и R2 опять меняются ролями — R2 снабжает R1 устаревшей информацией о пути к сети 201.36.14.0, уже с метрикой 4, которую R1 преобразует в метрику 5. Пакеты продолжают зацикливаться.

Если бы в протоколе RIP не было выбрано расстояние 16 в качестве недостижимого, то описанный процесс длился бы бесконечно (вернее, пока не была бы исчерпана разрядная сетка поля расстояния, и при очередном наращивании расстояния было бы зафиксировано переполнение).

В результате маршрутизатор R2 на очередном этапе описанного процесса получает от маршрутизатора R1 метрику 15, которая после наращивания, превращаясь в метрику 16, фиксирует недостижимость сети. Период нестабильной работы сети длился 36 минут!

Ограничение в 15 хопов сужает область применения протокола RIP до сетей, в которых число промежуточных маршрутизаторов не может быть больше 15. Для более масштабных сетей нужно применять другие протоколы маршрутизации, например OSPF, или разбивать сеть на автономные области.

Приведенный пример хорошо иллюстрирует главную причину нестабильности маршрутизаторов, работающих по протоколу RIP. Эта причина коренится в самом принципе работы дистанционно-векторных протоколов — использовании информации, полученной из «вторых рук». Действительно, маршрутизатор R2 передает маршрутизатору R1 информацию о достижимости сети 201.36.14.0, за достоверность которой он сам не отвечает.

ПРИМЕЧАНИЕ

Не следует думать, что при любых отказах интерфейсов и маршрутизаторов в сетях возникают маршрутные петли. Если бы маршрутизатор R1 успел передать сообщение о недостижимости сети 201.36.14.0 раньше ложной информации маршрутизатора R2, то маршрутная петля не образовалась бы. Так что маршрутные петли даже без дополнительных методов борьбы с ними возникают в среднем не более чем в половине потенциально возможных случаев.

Методы борьбы с ложными маршрутами в протоколе RIP

Хотя протокол RIP не в состоянии полностью исключить в сети переходные состояния, когда некоторые маршрутизаторы пользуются устаревшей информацией о несуществующих маршрутах, имеется несколько методов, которые во многих случаях решают подобные проблемы.

Проблема с петлей, образующейся между соседними маршрутизаторами, надежно решается с помощью метода **расщепления горизонта**. Этот метод заключается в том, что маршрутная информация о некоторой сети, хранящаяся в таблице маршрутизации, никогда не передается тому маршрутизатору, от которого она получена.

Практически все сегодняшние маршрутизаторы, работающие по протоколу RIP, используют технику расщепления горизонта. Если бы маршрутизатор R2 в рассмотренном выше примере поддерживал технику расщепления горизонта, то он бы не передал маршрутизатору R1 устаревшую информацию о сети 201.36.14.0, так как получил ее именно от маршрутизатора R1.

Однако расщепление горизонта не помогает в тех случаях, когда петли образуются не двумя, а несколькими маршрутизаторами. Рассмотрим более детально ситуацию, которая возникнет в сети, приведенной на рис. 19.15, в случае потери связи маршрутизатора R1 с сетью 201.36.14.0. Пусть все маршрутизаторы этой сети поддерживают технику расщепления горизонта. Маршрутизаторы R2 и R3 не будут возвращать маршрутизатору в этой ситуации данные о сети 201.36.14.0 с метрикой 2, так как они получили эту информацию от маршрутизатора R1. Однако они будут передавать маршрутизатору информацию о достижимости сети 201.36.14.0 с метрикой 4 через себя, так как получили эту информацию по сложному маршруту, а не непосредственно от маршрутизатора R1. Например, маршрутизатор R2 получил эту информацию по цепочке R4-R3-R1, поэтому маршрутизатор R1 снова может быть обманут, пока каждый из маршрутизаторов в цепочке R3-R4-R2 не вычеркнет запись о достижимости сети 201.36.14.0.

Для предотвращения зацикливания пакетов по составным петлям при отказах связей применяются два других приема, называемые триггерными обновлениями и замораживанием изменений.

Прием **триггерных обновлений** состоит в том, что маршрутизатор, получив данные об изменении метрики до какой-либо сети, не ждет истечения периода передачи таблицы маршрутизации, а передает данные об изменившемся маршруте немедленно. Этот прием может во многих случаях предотвратить передачу устаревших сведений об отказавшем маршруте, но он перегружает сеть служебными сообщениями, поэтому триггерные объявления также делаются с некоторой задержкой. По этой причине возможна ситуация, когда регулярное обновление в каком-либо маршрутизаторе чуть опережает по времени приход триггерного обновления от предыдущего в цепочке маршрутизатора, и данный маршрутизатор успевает передать по сети устаревшую информацию о несуществующем маршруте.

Второй прием — **замораживание изменений** — позволяет исключить подобные ситуации. Он связан с введением тайм-аута на принятие новых данных о сети,

которая только что стала недоступной. Этот тайм-аут предотвращает принятие устаревших сведений о некотором маршруте от тех маршрутизаторов, которые находятся на некотором расстоянии от отказавшей связи и передают устаревшие сведения о ее работоспособности. Предполагается, что в течение тайм-аута «замораживания изменений» эти маршрутизаторы вычеркнут данный маршрут из своих таблиц, так как не получат о нем новых записей и не будут распространять устаревшие сведения по сети.

Протокол OSPF

Список ключевых слов: протокол OSPF, объявления о состоянии связей сети, область сети, граф связей сети, база данных топологии сети, алгоритм Дийкстры, сообщения HELLO, алгоритм состояния связей, синхронизация топологических БД, метрика, биты TOS.

Протокол OSPF (Open Shortest Path First — выбор кратчайшего пути первым) является достаточно современной реализацией алгоритма состояния связей (он принят в 1991 году) и обладает многими особенностями, ориентированными на применение в больших гетерогенных сетях.

Два этапа построения таблицы маршрутизации

Как и все протоколы маршрутизации, основанные на алгоритме состояния связей, OSPF разбивает процесс построения таблицы маршрутизации на два этапа.

На первом этапе каждый маршрутизатор строит граф связей сети, в котором вершинами графа являются маршрутизаторы и IP-сети, а ребрами — интерфейсы маршрутизаторов. Все маршрутизаторы для этого обмениваются со своими соседями той информацией о графе сети, которой они располагают к данному моменту. Этот процесс похож на процесс распространения векторов расстояний до сетей в протоколе RIP, однако сама информация качественно иная — это информация о топологии сети. Сообщения, с помощью которых распространяется топологическая информация, называются **объявлениями о состоянии связей сети** (Link State Advertisements, LSA). Кроме того, при передаче топологической информации OSPF маршрутизаторы ее не модифицируют, как это делают RIP-маршрутизаторы, а передают в неизменном виде. В результате все маршрутизаторы сети располагают идентичными сведениями о графе сети, которые хранятся *в базе данных о топологии сети.*

Второй этап состоит в нахождении оптимальных маршрутов с помощью полученного графа. Задача нахождения оптимального пути на графе является достаточно сложной и трудоемкой. В протоколе OSPF для ее решения используется итеративный алгоритм Дийкстры. Каждый маршрутизатор считает себя центром сети и ищет оптимальный маршрут до каждой известной ему сети. В каждом найденном таким образом маршруте запоминается только один шаг — до следующего маршрутизатора, в соответствии с принципом одношаговой маршрутизации. Данные об этом шаге и попадают в таблицу маршрутизации. Если

несколько маршрутов имеют одинаковую метрику до сети назначения, то в таблице маршрутизации запоминаются первые шаги всех этих маршрутов.

Сообщения HELLO и корректировка таблиц маршрутизации

Для того чтобы база данных о топологии сети соответствовала текущему состоянию сети, OSPF-маршрутизатрам необходимо постоянно отслеживать изменения состояния сети и вносить при необходимости коррективы в таблицу маршрутизации. Для контроля состояния связей и соседних маршрутизаторов OSPF-маршрутизаторы регулярно передают друг другу сообщения HELLO. Сообщения HELLO отправляются через каждые 10 секунд, чтобы повысить скорость адаптации маршрутизаторов к изменениям, происходящим в сети. Небольшой объем этих сообщений делает возможной такое частое тестирование состояния соседей и связей с ними. На основании принимаемых от непосредственных соседей сообщений HELLO маршрутизатор формирует записи о состоянии связях со своими непосредственными соседями в базе данных о топологии сети.

В том случае, когда сообщения HELLO перестают поступать от какого-либо непосредственного соседа, маршрутизатор делает вывод о том, что состояние связи изменилось с работоспособного на неработоспособное и делает соответствующую отметку в своей базе данных. Одновременно он отсылает всем непосредственным соседям объявление LSA об этом изменении, и те также корректируют свои базы данных и, в свою очередь, рассылают данное объявление LSA своим непосредственным соседям (естественно, кроме того соседа, от которого оно было получено). После корректировки графа сети каждый маршрутизатор заново ищет оптимальные маршруты и корректирует свою таблицу маршрутизации. Конвергенция таблиц маршрутизации к новому стабильному состоянию происходит очень быстро, это время состоит из времени передачи объявления LSA и времени работы алгоритма Дийкстры для нахождения новых маршрутов. Аналогичный процесс происходит и в том случае, когда в сети появляется новый сосед, объявляющий о себе с помощью своих сообщений HELLO, или новая связь.

Если же состояние сети не меняется, то объявления о связях не генерируются и таблицы маршрутизации не корректируются, что экономит пропускную способность сети и вычислительные ресурсы маршрутизаторов. Однако у этого правила есть исключение: каждые 30 минут OSPF-маршрутизаторы обмениваются всеми записями базы данных топологической информации, то есть синхронизируют их для более надежной работы сети. Так как этот период достаточно большой, то данное исключение незначительно сказывается на работе сети.

Связи и метрики

Маршрутизаторы соединены как с локальными сетями, так и непосредственно между собой глобальными двухточечными линиями связи, например каналами T1. Протокол OSPF в своих объявлениях распространяет информацию о связях двух типов: маршрутизатор-маршрутизатор и маршрутизатор-сеть.

Рассмотрим эти типы связи на примере сети, изображенной на рис. 19.16.

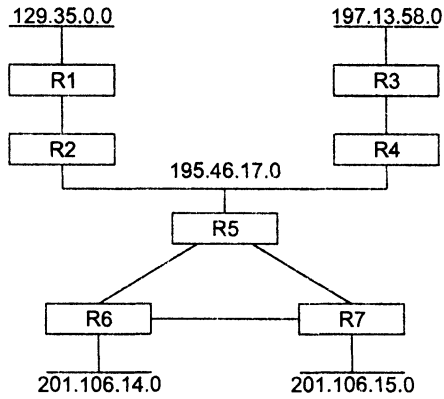


Рис. 19.16. Фрагмент сети OSPF

Данной сети соответствует граф, приведенный на рис. 19.17.

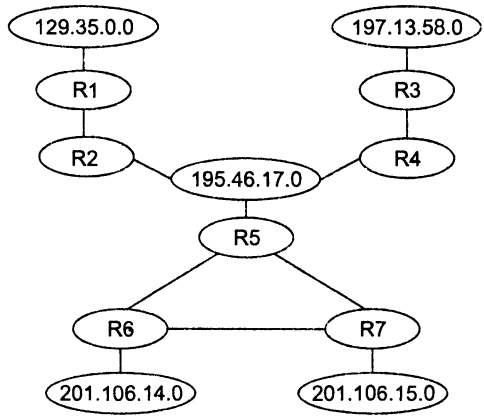


Рис. 19.17. Граф сети, построенный протоколом OSPF

Примером связи первого типа служит связь R3-R4, второго — связь R4-195.46.17.0/24 (здесь R3 и R4 также являются IP-адресами, но мы используем символьные идентификаторы, чтобы отличать эти вершины графа от сетей, для которых мы сохранили обычную нотацию IP-адресов). Если двухточечным линиям связи дать IP-адреса, то они станут дополнительными вершинами графа, как и локальные сети. Вместе с IP-адресом сети передается также информация о маске сети.

Каждая связь характеризуется метрикой. Протокол OSPF по умолчанию использует метрику, учитывающую пропускную способность каналов связи. Кроме того, допускается использование двух других метрик, учитывающих задержки и надежность передачи пакетов каналами связи. Для каждой из метрик протокол OSPF строит отдельную таблицу маршрутизации. Выбор нужной таблицы

происходит в зависимости от значений битов TOS в заголовке пришедшего IP-пакета.

Протокол OSPF поддерживает стандартные для многих протоколов (например, для протокола покрывающего дерева) значения расстояний для метрики, отражающей пропускную способность: так, для сети Ethernet она равна 10, для Fast Ethernet — 1, для канала T1 — 65, для канала 56 Кбит/с — 1785. При использовании высокоскоростных каналов, таких как Gigabit Ethernet или STM-16/64, администратору нужно задать другую шкалу скоростей, назначив единичное расстояние наиболее скоростному каналу.

При выборе оптимального пути на графе с каждым ребром графа связана метрика, которая добавляется к пути, если данное ребро в него входит. Пусть на приведенном примере маршрутизатор R5 связан с маршрутизаторами R6 и R7 каналами T1, а маршрутизаторы R6 и R7 связаны между собой каналом 56 Кбит/с. Тогда R7 определит оптимальный маршрут до сети 201.106.14.0 как составной, проходящий сначала через R5, а затем через R6, поскольку у этого маршрута метрика будет равна 65 + 65 = 130 единиц. Непосредственный маршрут через R6 не будет оптимальным, так как его метрика равна 1785. При использовании хопов был бы выбран маршрут через R6, что было бы не оптимально.

Протокол OSPF разрешает хранить в таблице маршрутизации несколько маршрутов к одной сети, если они обладают равными метриками. Если такие записи образуются в таблице маршрутизации, то маршрутизатор реализует режим баланса загрузки маршрутов, отправляя пакеты попеременно по каждому из маршрутов.

Области сети

К сожалению, вычислительная сложность протокола OSPF быстро растет с увеличением размерности сети. Для преодоления этого недостатка в протоколе OSPF вводится понятие **области сети** (не нужно путать с автономной системой Интернета). Маршрутизаторы, принадлежащие некоторой области, строят граф связей только для этой области, что сокращает размерность сети. Между областями информация о связях не передается, а пограничные для областей маршрутизаторы обмениваются только информацией об адресах сетей, имеющихся в каждой из областей, и *расстоянием от пограничного маршрутизатора до каждой сети*. При передаче пакетов между областями выбирается один из пограничных маршрутизаторов области, а именно тот, у которого расстояние до нужной сети меньше.

Протокол BGP

Список ключевых слов: пограничный шлюзовой протокол, автономные системы Интернета, магистраль Интернета, протокол BGPv4, внешний шлюз.

Пограничный шлюзовой протокол (Border Gateway Protocol, BGP) в версии 4 является сегодня основным протоколом обмена маршрутной информацией между автономными системами Интернета. Протокол BGP пришел на смену прото-

колу EGP[1], использовавшемуся в тот начальный период, когда Интернет имел единственную магистраль. Эта магистраль являлась центральной автономной системой, к которой присоединялись в соответствии с древовидной топологией все остальные автономные системы. Так как между автономными системами при такой структуре петли исключались, протокол EGP не предпринимал никаких мер для того, чтобы исключить зацикливание маршрутов.

BGPv4 успешно работает при любой топологии связей между автономными системами, что соответствует современному состоянию Интернета.

Поясним основные принципы работы BGP на примере (рис. 19.18).

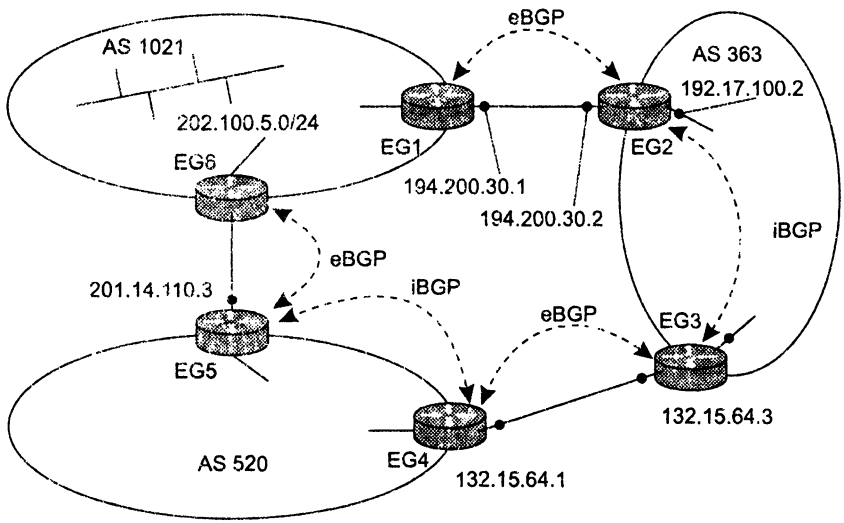


Рис. 19.18. Поиск маршрута между автономными системами с помощью протокола BGP

В каждой из трех автономных систем (AS 1021, AS 363 и AS 520) имеется несколько маршрутизаторов, исполняющих роль внешних шлюзов. На каждом из них работает протокол BGP, с помощью которого они общаются между собой.

Маршрутизатор взаимодействует с другими маршрутизаторами по протоколу BGP только в том случае, если администратор *явно* указывает при конфигурировании, что эти маршрутизаторы являются его *соседями*. Например, маршрутизатор EG1 в рассматриваемом примере будет взаимодействовать по протоколу BGP с маршрутизатором EG2 не потому, что эти маршрутизаторы соединены двухточечным каналом, а потому, что при конфигурировании маршрутизатора EG1 в качестве соседа ему был указан маршрутизатор EG2 (с адресом 194.200.30.2).

[1] EGP в данном случае является названием конкретного протокола маршрутизации. Напомним, что EGP служит также названием класса внешних шлюзовых протоколов, используемых для маршрутизации между автономными системами, что вносит некоторую путаницу.

Аналогично, при конфигурировании маршрутизатора EG2 его соседом был назначен маршрутизатор EG1 (с адресом 194.200.30.1).

Такой способ взаимодействия удобен в ситуации, когда маршрутизаторы, обменивающиеся маршрутной информацией, принадлежат разным поставщикам услуг (ISP). Администратор ISP может решать, с какими автономными системами он будет обмениваться трафиком, а с какими нет, задавая список соседей для своих внешних шлюзов. Протоколы RIP и OSPF, разработанные для применения внутри автономной системы, обмениваются маршрутной информацией со всеми маршрутизаторами, находящимися в пределах их непосредственной досягаемости (по локальной сети или через двухточечный канал). Это означает, что информация обо всех сетях появляется в таблице маршрутизации каждого маршрутизатора, так что каждая сеть оказывается достижимой для каждой. В корпоративной сети это нормальная ситуация, а в ISP-сетях нет, поэтому протокол BGP и исполняет здесь особую роль.

Для установления сеанса с указанными соседями BGP-маршрутизаторы используют протокол TCP (порт 179). При установлении BGP-сеанса могут применяться разнообразные способы аутентификации маршрутизаторов, повышающие безопасность работы автономных систем.

Основным сообщением протокола BGP является сообщение UPDATE (обновить), с помощью которого маршрутизатор сообщает маршрутизатору соседней автономной системы о достижимости сетей, относящихся к его собственной автономной системе. Само название этого сообщения говорит о том, что это триггерное объявление, которое посылается соседу только тогда, когда в автономной системе что-нибудь резко меняется: появляются новые сети или новые пути к сетям, или же напротив, исчезают существовавшие сети или пути.

В одном сообщении UPDATE можно объявить об одном новом маршруте или аннулировать несколько переставших существовать. Под маршрутом в BGP понимается последовательность автономных систем, которую нужно пройти на пути к указанной в адресе сети. Более формально информация о маршруте (BGP Route) к сети (Network/Mask_length) выглядит так:

BGP Route = AS_Path; NextHop; Network/Mask_length;

Здесь AS_Path — набор номеров автономных систем, NextHop — IP-адрес маршрутизатора, через который нужно передавать пакеты в сеть Network/Mask_length. Например, если маршрутизатор EG1 хочет объявить маршрутизатору EG2 о том, что в AS 1021 появилась новая сеть 202.100.5.0/24, то он формирует такое сообщение:

AS 1021; 194.200.30.1; 202.100.5.0/24

и передает его маршрутизатору EG2 автономной системы AS 363 (с которым у него, конечно, должен быть установлен BGP-сеанс).

Маршрутизатор EG2, получив сообщение UPDATE, запоминает в своей таблице маршрутизации информацию о сети 202.100.5.0/24 вместе с адресом следующего маршрутизатора 194.200.30.1 и отметкой о том, что эта информация была получена от протокола BGP. Маршрутизатор EG2 обменивается маршрутной информацией с внутренними шлюзами системы AS 363 по какому либо протоколу

группы IGP, например OSPF. Если у EG2 установлен режим перераспределения маршрутов BGP в маршруты OSPF, то все внутренние шлюзы AS 363 узнают о существовании сети 202.100.5.0/24 с помощью объявления OSPF, являющееся внешним. В качестве адреса следующего маршрутизатора маршрутизатор EG2 будет теперь объявлять адрес собственного внутреннего интерфейса, например, 192.17.100.2.

Однако для распространения сообщения о сети 202.100.5.0/24 в другие автономные системы, например, в AS 520, протокол OSPF использоваться не может. Маршрутизатор EG3, связанный с маршрутизатором EG4 автономной системы 520, должен пользоваться протоколом BGP, генерируя сообщение UPDATE нужного формата. Для решения этой задачи он не может использовать информацию о сети 202.100.5.0/24, полученную от протокола OSPF через один из своих внутренних интерфейсов, так как она имеет другой формат и не содержит, например, сведений о номере автономной системы, в которой находится эта сеть.

Проблема решается за счет того, что маршрутизаторы EG2 и EG3 также устанавливают между собой BGP-сеанс, хотя они и принадлежат одной и той же автономной системе. Такая реализация протокола BGP называется внутренней (Interior BGP, iBGP), в отличие от основной, внешней (Exterior BGP, eBGP). В результате маршрутизатор EG3 получает нужную информацию от маршрутизатора EG2 и передает ее внешнему соседу, маршрутизатору EG4. При формировании нового сообщения UPDATE маршрутизатор EG3 трансформирует сообщение, полученное от маршрутизатора EG2 за счет того, что добавляет в список автономных систем собственную автономную систему AS 520, а полученный адрес следующего маршрутизатора заменяет адресом собственного интерфейса:

AS 363, AS 1021; 132.15.64.3; 202.100.5.0/24.

Номера автономных систем позволяют исключать зацикливание сообщений UPDATE. Например, когда маршрутизатор EG5 передаст сообщение о сети 202.100.5.0/24 маршрутизатору EG6, то последний не будет его использовать, так как оно будет иметь вид:

AS 520, AS 363, AS 1021; 201.14.110.3; 202.100.5.0/24.

Так как в списке автономных систем уже есть номер собственной автономной системы, очевидно, что сообщение зациклилось.

Протокол BGP используется сегодня не только для обмена маршрутной информацией между AS.

Протокол ICMP

Список ключевых слов: протокол межсетевых управляющих сообщений, поля типа, кода, контрольной суммы, идентификатора и порядкового номера, эхо-запрос, эхо-ответ.

Протокол межсетевых управляющих сообщений (Internet Control Message Protocol, ICMP) играет в сети вспомогательную роль. Спецификация этого протокола содержится в RFC 792.

Существует ряд ситуаций, когда протокол IP не может доставить пакет адресату, например, когда истекает время жизни пакета, когда в таблице маршрутизации отсутствует маршрут к заданному в пакете адресу назначения, когда пакет не проходит проверку по контрольной сумме, когда шлюз не имеет достаточно места в своем буфере для передачи какого-либо пакета и т. д. и т. п. Как мы не раз отмечали, протокол IP работает по возможности (с максимальными усилиями), то есть не предпринимает мер для гарантированной доставки данных. Это свойство «необязательности» протокола IP компенсируется протоколами более высоких уровней, например TCP на транспортном уровне или в какой-то степени DNS на прикладном уровне. Они берут на себя обязанности по обеспечению надежности, применяя такие известные приемы, как нумерация сообщений, подтверждение доставки, повторная посылка данных.

Протокол ICMP служит дополнением протокола IP несколько другого рода. Он не предназначен для исправления возникших при передаче пакета проблем: если пакет потерян, ICMP не может послать его заново. Задача ICMP другая — он является средством оповещения отправителя о «несчастных случаях», произошедших с его пакетами. В то время как протокол IP посылает пакет и забывает о нем, протокол ICMP «отслеживает» передвижение пакета по сети и при отбрасывании пакета маршрутизатором передает сообщение об этом узлу-источнику, обеспечивая таким образом обратную связь между посланным пакетом и отправителем.

Пусть, например, протокол IP, работающий на каком-либо маршрутизаторе, обнаружил, что пакет для дальнейшей передачи по маршруту необходимо фрагментировать, но в пакете установлен признак DF (не фрагментировать). Протокол IP, обнаруживший, что он не может передать IP-пакет далее по сети, должен отправить *диагностическое* ICMP-сообщение узлу-источнику и только потом отбросить пакет.

Помимо диагностики ICMP также используется для *мониторинга* сети. Так, в основе популярных утилит для мониторинга IP-сетей ping и tracert лежат ICMP-сообщения. С помощью ICMP-сообщений приложение может определить маршрут перемещения данных, оценить работоспособность сети, определить время прохождения данных до заданного узла, сделать запрос о значении маски определенного сетевого интерфейса и т. п.

Заметим, что некоторые из пакетов могут исчезнуть в сети, не вызвав при этом никаких оповещений. В частности, протокол ICMP не предусматривает передачу сообщений о проблемах, возникающих при обработке IP-пакетов, несущих ICMP-сообщения об ошибках. (Это правило, однако, не действует для ICMP-запросов.) Такое решение было принято разработчиками протокола, чтобы не порождать «штормы» в сетях, когда количество сообщений об ошибках лавинообразно возрастает. По этой же причине ICMP-сообщения не передаются, если ошибка возникла при передаче какого-либо фрагмента, кроме первого, а также когда потерянный пакет имел широковещательный IP-адрес или был упакован в кадр с широковещательным адресом несущей технологии.

Поскольку IP-пакет содержит адрес отправителя, но не содержит никакой адресной информации о промежуточных маршрутизаторах, ICMP-сообщения направ-

ляются только конечным узлам. Здесь сообщения могут быть обработаны либо ядром операционной системы, либо протоколами транспортного и прикладного уровней, либо приложениями, либо просто проигнорированы. Важно, что обработка ICMP-сообщений не входит в обязанности протоколов IP и ICMP.

Типы ICMP-сообщений

Все типы ICMP-сообщений могут быть разделены на два класса:

- диагностические сообщения об ошибках;
- информационные сообщения типа запрос/ответ.

ICMP-сообщение инкапсулируется в поле данных IP-пакета (рис. 19.19).

Рис. 19.19. Инкапсуляция и формат ICMP-сообщения

Заголовок ICMP состоит из 8 байт; поля заголовка перечислены ниже.

- **Тип** (размером 1 байт) содержит код, определяющий тип сообщения. Основные типы сообщений перечислены в табл. 19.7.
- **Код** (размером 1 байт) более тонко дифференцирует тип ошибки.
- **Контрольная сумма**, подсчитанная для всего ICMP-сообщения, занимает 2 байта.

Заголовок также включает поле из 4 байт, содержимое которого зависит от значений полей типа и кода. В сообщениях типа запрос/ответ это поле содержит 2-байтовые подполя **идентификатора** и **порядкового номера** (см. далее). Числа из этих подполей дублируются из сообщения-запроса в сообщение-ответ. Идентификатор позволяет узлу-получателю сообщения определить, какому приложению направлен этот ответ, а порядковый номер используется приложением, чтобы связать ответ с соответствующим запросом (учитывая, что одно приложение может выдать несколько однотипных запросов). В сообщениях об ошибке это поле не используется и заполняется нулями.

Таблица 19.7. Возможные значения поля типа

Значение	Тип сообщения
0	Эхо-ответ
3	Узел назначения недостижим
4	Подавление источника
5	Перенаправление маршрута
8	Эхо-запрос

продолжение

Таблица 19.7 (*продолжение*)

Значение	Тип сообщения
11	Истечение времени дейтаграммы
12	Проблема с параметром пакета
13	Запрос отметки времени
14	Ответ отметки времени
17	Запрос маски
18	Ответ маски

Каждый тип ошибки может быть более точно охарактеризован кодом ошибки. Например, в табл. 19.8 приведены коды для сообщения о недостижимости узла назначения (ошибка типа 3 из предыдущей таблицы). Эти коды, которые могут быть указаны в сообщении этого типа, позволяют выявить множество различных причин данной ситуации. Недостижимость узла назначения может, в частности, быть вызвана временной неработоспособностью аппаратуры, неверным адресом назначения, отсутствием протокола прикладного уровня или открытого порта UDP/TCP в узле назначения.

Таблица 19.8. Коды, детализирующие причину ошибки о недостижимости узла назначения

Код	Причина
0	Сеть недостижима
1	Узел недостижим
2	Протокол недостижим
3	Порт недостижим
4	Требуется фрагментация, а бит DF установлен
5	Ошибка в маршруте, заданном источником
6	Сеть назначения неизвестна
7	Узел назначения неизвестен
8	Узел-источник изолирован
9	Взаимодействие с сетью назначения административно запрещено
10	Взаимодействие с узлом назначения административно запрещено
11	Сеть недостижима для заданного класса сервиса
12	Узел недостижим для заданного класса сервиса
13	Взаимодействие административно запрещено путем фильтрации

Формат поля данных ICMP-сообщения также зависит от значений полей типа и кода. Чтобы показать различия в форматах разных типов сообщений, мы рассмотрим в следующих разделах два примера:

- сообщения типа эхо-запрос и эхо-ответ;
- сообщение о недостижимости узла назначения.

Формат эхо-запроса/эхо-ответа и утилита ping

На рис. 19.20 показаны форматы эхо-запроса и эхо-ответа. Они отличаются друг от друга только значением поля типа (нули — для ответа, единицы — для запроса). В поле данных запроса отправитель помещает информацию, которую затем получает в ответе от узла назначения.

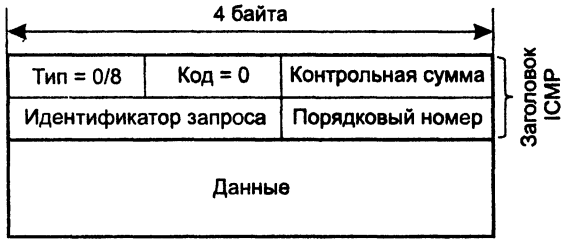


Рис. 19.20. Формат ICMP-сообщений типа эхо-запрос/эхо-ответ

Эхо-запрос и **эхо-ответ**, в совокупности называемые эхо-протоколом, представляют собой очень простое средство мониторинга сети. Компьютер или маршрутизатор посылает по составной сети эхо-запрос, указывая в нем IP-адрес узла, достижимость которого нужно проверить. Узел, получивший эхо-запрос, формирует и отправляет эхо-ответ отправителю запроса. Так как эхо-запрос и эхо-ответ передаются по сети внутри IP-пакетов, то их успешная доставка означает нормальное функционирование всей транспортной системы составной сети.

Во многих операционных системах используется утилита **ping**, предназначенная для тестирования достижимости узлов. Эта утилита обычно посылает серию эхо-запросов к тестируемому узлу и предоставляет пользователю статистику об утерянных эхо-ответах и среднем времени реакции сети на запросы. Утилита **ping** выводит на экран сообщения следующего вида обо всех поступивших ответах:

```
# ping server1.citmgu.ru
Pinging server1.citmgu.ru [193.107.2.200] with 64 bytes of data:
Reply from 193.107.2.200: bytes=64 time=256ms TTL= 123
Reply from 193.107.2.200: bytes=64 time=310ms TTL= 123
Reply from 193.107.2.200: bytes=64 time=260ms TTL= 123
Reply from 193.107.2.200: bytes=64 time=146ms TTL= 123
```

Из приведенной распечатки видно, что в ответ на тестирующие запросы, посланные узлу **server1.mgu.ru**, было получено 4 эхо-ответа. Длина каждого сообщения составляет 64 байта. В следующей колонке помещены значения времени оборота (RTT), то есть времени от момента отправки запроса до получения ответа на

этот запрос. Как видим, сеть работает достаточно нестабильно — время в последней строке отличается от времени во второй более чем в два раза. На экран выведено также оставшееся время жизни поступивших пакетов.

В зависимости от конкретной реализации утилиты ping, а также ее настроек (ключей) выводимые экранные формы могут отличаться. У утилиты ping обычно имеется несколько ключей, с помощью которых можно установить размер поля данных сообщения, начальное значение поля TTL, количество повторных передач пакетов, флаг DF.

В том случае, когда за установленное время тайм-аута ответы не приходят или протокол ICMP сообщает об ошибках, утилита ping выводит на экран соответствующие диагностические сообщения.

Формат сообщения об ошибке и утилита traceroute

На рис. 19.21 показан формат ICMP-сообщения об ошибке, в данном случае это сообщение о недостижимости узла назначения. Остальные ICMP-сообщения об ошибках имеют такой же формат и отличаются друг от друга только значениями полей типа и кода.

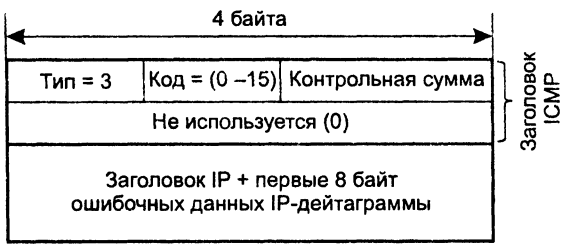


Рис. 19.21. Формат ICMP-сообщения об ошибке — недостижимости узла назначения

Когда маршрутизатор не может передать или доставить IP-пакет, он отсылает узлу, отправившему этот пакет, сообщение о недостижимости узла назначения. В поле типа помещается значение 3, а в поле кода — значение из диапазона 0–15, уточняющее причину, по которой пакет не был доставлен. Следующие за полем контрольной суммы 4 байт заголовка не используются и заполняются нулями.

Помимо причины ошибки, указанной в заголовке, в поле данных ICMP-сообщения всегда помещается заголовок IP и первые 8 байт данных того IP-пакета, который вызвал ошибку. Эта информация позволяет узлу-отправителю точнее установить причину ошибки, так как все протоколы стека TCP/IP, использующие для передачи своих сообщений IP-пакеты, содержат наиболее важную для анализа информацию в первых 8 байт своих сообщений. В частности, ими вполне могут оказаться первые 8 байт заголовка TCP или UDP, в которых содержится информация, идентифицирующая приложение, пославшее потерянный пакет. Следовательно, при разработке приложения можно предусмотреть встроенные средства реакции на сообщения о недоставленных пакетах.

Узел (или сеть) назначения может быть недостижим по причине временной неработоспособности аппаратуры из-за того, что отправитель указал неверный адрес назначения или маршрутизатор не имеет данных о пути к сети назначения. Недостижимость протокола и порта означает отсутствие реализации какого-либо протокола прикладного уровня в узле назначения или же отсутствие открытого порта протокола UDP или TCP в узле назначения.

Как было показано на примере утилиты ping, ICMP-сообщения эффективно используются для мониторинга сети. В частности, сообщения об ошибке истечения тайм-аута лежат в основе работы другой популярной утилиты traceroute для Unix, имеющей в Windows 2000 название tracert. Эта утилита позволяет проследить маршрут до удаленного хоста, определить RTT, IP-адрес и доменное имя для каждого промежуточного маршрутизатора (если это имя зарегистрировано в обратной зоне службы DNS). Такая информация полезна для локализации маршрутизатора, на котором обрывается путь пакета к удаленному хосту.

Утилита traceroute осуществляет трассировку маршрута путем посылки обычных IP-пакетов с адресом назначения, являющимся конечной точкой изучаемого маршрута. Суть метода трассировки состоит в том, что значение TTL первого отправляемого пакета установлено равным 1. Когда протокол IP первого маршрутизатора принимает этот пакет, то он в соответствии со своим алгоритмом уменьшает значение TTL на 1 и получает 0. Маршрутизатор отбрасывает пакет с нулевым временем жизни и возвращает узлу-источнику ICMP-сообщение об ошибке истечения тайм-аута вместе с заголовком IP и первыми 8 байтами потерянного пакета.

Получив ICMP-сообщение о причине недоставки пакета, утилита traceroute запоминает адрес первого маршрутизатора (который извлекает из заголовка IP-пакета, несущего ICMP-сообщение) и вычисляет для него RTT. Затем traceroute посылает следующий IP-пакет, но теперь со значением TTL, равным 2. Этот пакет благополучно проходит первый маршрутизатор, но «умирает» на втором, о чем немедленно отправляется аналогичное ICMP-сообщение об ошибке истечения тайм-аута. Утилита traceroute запоминает адрес и время для второго маршрутизатора и т. д. Такие действия выполняются с каждым маршрутизатором вдоль маршрута вплоть до узла назначения.

Мы рассмотрели работу утилиты traceroute весьма схематично, но и этого достаточно, чтобы оценить изящество идеи, лежащей в основе ее работы.

Ниже приведена копия экранной формы, выведенной утилитой tracert (Windows) при трассировке хоста ds.internic.net [198.49.45.29]:

```
1 311 ms 290 ms 261 ms 144.206.192.100
2 281 ms 300 ms 271 ms 194.85.73.5
3 2023 ms 290 ms 311 ms moscow-m9-2-S5.relcom eu.net [193.124.254.37]
4 290 ms 261 ms 280 ms MSK-M9-13 Relcom.EU.net [193.125.15.13]
5 270 ms 281 ms 290 ms MSK.RAIL-1-ATM0-155Mb.Relcom.EU.net [193 124.254.82]
6 300 ms 311 ms 290 ms SPB-RASCOM-1-E3-1-34Mb.Relcom.EU.net [193.124.254.78]
7 311 ms 300 ms 300 ms Hssi11-0.GW1.STK2.ALTER.NET [146.188.33.125]
8 311 ms 330 ms 291 ms 421.ATM6-0-0.CR2.STK2.Alter.Net [146.188.5.73]
9 360 ms 331 ms 330 ms 219 Hssi4-0.CR2.LND1.Alter.Net [146.188.2.213]
```

```
10 351 ms 330 ms 331 ms 412.Atm5-0.BR1.LND1.Alter.net [146.188.3.205]
11 420 ms 461 ms 420 ms 167.ATM8-0-0.CR1.ATL1.Alter.Net [137.39.69.182]12 461
ms 441 ms 440 ms 311.ATM12-0-0.BR1.ATL1.Alter.Net [137.39.21 73]13 451 ms 410
ms 431 ms atlantal-br1.bbnplanet.net [4.0.2.141]14 420 ms 411 ms 410 ms
viennal-br2.bbnplanet.net [4.0.3.154]15 411 ms 430 ms 2514 ms viennal-
nbr3.bbnplanet.net [4.0.3.150]16 430 ms 421 ms 441 ms viennal-
nbr2.bbnplanet.net [4.0.5.45]17 431 ms 451 ms 420 ms cambridge1-
br1.bbnplanet.net [4.0.5.42]18 450 ms 461 ms 441 мс cambridge1-
cr14.bbnplanet.net [4.0.3.94]19 451 мс 461 мс 460 мс attbcstoll.bbnplanet.net
[206.34.99.38]20 501 мс 460 мс 481 мс shutdown.ds.internic.net [198.49.45.29]
```

Последовательность строк соответствует последовательности маршрутизаторов, образующих маршрут к заданному узлу. Первое число в строке — число хопов до соответствующего маршрутизатора. Утилита traceroute тестирует каждый маршрутизатор трижды, поэтому следующие три числа в строке — это значения RTT, вычисленные путем посылки трех пакетов, время жизни которых истекло на этом маршрутизаторе. Если ответ от какого-либо маршрутизатора не приходит за заданное время, то вместо времени на экране печатается звездочка (*).

Далее идут IP-адрес и доменное имя (если оно имеется) маршрутизатора. Видно, что почти все интерфейсы маршрутизаторов поставщиков услуг Интернета зарегистрированы в службе DNS, а первые два, относящиеся к локальным маршрутизаторам, — нет.

Еще раз подчеркнем, что время, указанное в каждой строке, это не время прохождения пакетов между двумя соседними маршрутизаторами, а время, за которое пакет проделывает путь от источника до соответствующего маршрутизатора и обратно. Так как ситуация в Интернете с загрузкой маршрутизаторов постоянно меняется, то время достижимости маршрутизаторов не всегда нарастает монотонно, а может изменяться достаточно произвольным образом.

Выводы

В то время как задачей протокола IP является передача данных между сетевыми интерфейсами в составной сети, основная задача протоколов TCP и UDP заключается в передаче данных между прикладными процессами, выполняющимися на разных конечных узлах сети.

Основным отличием TCP от UDP является то, что на протокол TCP возложена дополнительная задача — обеспечение надежной доставки сообщений через составную сеть, все узлы которой используют для передачи сообщений ненадежный дейтаграммный протокол IP.

Протокол UDP является дейтаграммным протоколом, работающим без установления логического соединения по остаточному принципу (с максимальными усилиями). UDP не гарантирует доставку своих сообщений, а следовательно, не компенсирует ненадежность дейтаграммного протокола IP.

Системные очереди к точкам входа прикладных процессов называют портами. Порты идентифицируются номерами и однозначно определяют приложение в пределах компьютера. Приложения, которые используют протокол UDP, получают номера портов UDP, а приложения, обращающиеся к протоколу TCP, номера портов TCP.

Если процессы представляют собой популярные общедоступные службы, такие как FTP, telnet, HTTP, TFTP, DNS и т. п., то за ними централизовано закрепляются стандартные (назначенные) номера, называемые также общеизвестными номерами портов. Для тех служб, которые еще не стали столь распространенными, чтобы закреплять за ними стандартные номера, номера портов выделяются локальной операционной системой. Такие номера называют динамическими.

Сокетом прикладного процесса называется пара из IP-адреса и номер порта.

TCP решает задачу надежного обмена данными путем установления логических соединений. Соединение однозначно идентифицируется парой сокетов.

TCP-соединение является дуплексным, оно устанавливается в результате переговорного процесса о максимальном размере единицы передачи, максимальном объеме данных, которые можно передавать без получения подтверждения, о начальном порядковом номере байта, с которого начинается отсчет потока данных в рамках данного соединения. При создании соединения ОС выделяет каждой стороне определенные системные ресурсы для организации буферов, таймеров, счетчиков.

Процедура приема протоколами TCP/UDP данных, поступающих от различных прикладных служб, называется мультиплексированием. Обратная процедура распределения протоколом TCP/UDP поступающих от сетевого уровня пакетов между набором высокоуровневых служб называется демультиплексированием. Протокол UDP реализует демультиплексирование с помощью сокетов, а TCP — с помощью соединений.

Для управления потоком в рамках TCP-соединения используется специфический вариант алгоритма скользящего окна. Сторона-получатель передает стороне-отправителю размер окна приема в байтах. При этом она исходит из того, с какой скоростью сможет обрабатывать присылаемые данные. Однако управлять окном приема может и отправитель. Если отправитель фиксирует ненадежную работу линии связи, то он может по собственной инициативе уменьшить окно.

Протоколы маршрутизации генерируют для каждого маршрутизатора согласованные таблицы маршрутизации, то есть такие, которые позволят обеспечить доставку пакета по рациональному маршруту от исходной сети в сеть назначения за конечное число шагов. Для этого маршрутизаторы сети обмениваются специальной информацией о топологии составной сети.

Различают статическую и адаптивную (динамическую) маршрутизацию:

- при статической маршрутизации таблицы составляются и вводятся в память каждого маршрутизатора вручную администратором сети;
- адаптивная (динамическая) маршрутизация обеспечивает автоматическое обновление таблиц маршрутизации после изменения конфигурации сети.

Адаптивные протоколы маршрутизации делятся на две группы, каждая из которых связана с одним из следующих типов алгоритмов:

- в дистанционно-векторных алгоритмах каждый маршрутизатор периодически и широковещательно рассылает по сети вектор, компонентами которого являются расстояния от данного маршрутизатора до всех известных ему сетей;
- алгоритмы состояния связей обеспечивают каждый маршрутизатор информацией, достаточной для построения точного графа связей сети.

Протоколы маршрутизации Интернета делятся на внешние и внутренние. Внешние протоколы (EGP) переносят маршрутную информацию между автономными системами, а внутренние (IGP) применяются только в пределах определенной автономной системы.

Протокол RIP является наиболее старым протоколом маршрутизации сетей TCP/IP. Несмотря на его простоту, определенную простотой дистанционно-векторного алгоритма, RIP успешно работает в небольших сетях с количеством промежуточных маршрутизаторов не более 15.

RIP-маршрутизаторы при выборе маршрута обычно используют самую простую метрику — количество промежуточных маршрутизаторов между сетями, то есть хопов.

В сетях, работающих по протоколу RIP и имеющих петлевидные маршруты, могут наблюдаться достаточно длительные периоды нестабильной работы, когда пакеты «зацикливаются» в маршрутных петлях и не доходят до адресатов. Для борьбы с этими явлениями в RIP-маршрутизаторах предусмотрено несколько приемов (расщепление горизонта, замораживание изменений, триггерные обновления), которые сокращают в некоторых случаях периоды нестабильности.

Протокол OSPF был разработан для эффективной маршрутизации IP-пакетов в больших сетях со сложной топологией, включающей петли. Он основан на алгоритме состояния связей, который устойчив к изменениям топологии сети.

При выборе маршрута OSPF-маршрутизаторы используют метрику, учитывающую пропускную способность составных сетей.

Протокол OSPF разрешает хранить в таблице маршрутизации несколько маршрутов к одной сети, если они обладают равными метриками, что дает возможность маршрутизатору работать в режиме баланса загрузки маршрутов.

Протокол OSPF обладает высокой вычислительной сложностью, поэтому чаще всего работает на мощных аппаратных маршрутизаторах.

Протокол BGPv4 является сегодня протоколом обмена маршрутной информацией между автономными системами Интернета. BGPv4 устойчиво работает при любой топологии связей между автономными системами, что соответствует современной структуре Интернета.

Протокол ICMP играет в сети вспомогательную роль. Он используется для диагностики и мониторинга сети. Так, в основе работы популярных утилит мониторинга IP-сетей ping и tracert лежат ICMP-сообщения.

Вопросы и задания

1. Когда разработчики программного обеспечения предпочитают использовать протокол UDP, а когда TCP?
2. Какой объем данных (с точностью до одного байта) получен в течение TCP-сеанса отправителем TCP-сегмента, в заголовке которого в поле квитанции помещено значение 1845685? Известно, что первый полученный байт имел номер 50046.
3. Можно ли продвигать IP-пакеты, если в маршрутизаторе отсутствует таблица маршрутизации? Варианты ответов:
   - ❍ нет, это невозможно;
   - ❍ можно, если использовать маршрутизацию от источника;
   - ❍ можно, если в маршрутизаторе задан маршрут по умолчанию.
4. Можно ли обойтись в сети без протоколов маршрутизации?
5. В чем заключаются недостатки дистанционно-векторных протоколов маршрутизации? Варианты ответов:
   - ❍ интенсивный дополнительный трафик в крупных сетях;
   - ❍ выбранные маршруты не всегда обладают минимальной метрикой;

- большое время сходимости сети к согласованным таблицам маршрутизации.

6. В чем заключается основной принцип работы протоколов маршрутизации, основанных на алгоритмах состояния связей?
7. В чем разница между внутренними и внешними шлюзовыми протоколами?
8. Какая метрика используется в протоколе RIP?
9. По какой причине в протоколе RIP расстояние в 16 хопов между сетями полагается недостижимым? Варианты ответов:
   - из-за того, что поле, отведенное для хранения значения расстояния, имеет длину 4 двоичных разряда;
   - сети, в которых работает RIP, редко бывают большими;
   - для получения приемлемого времени сходимости алгоритма.
10. Какие существуют способы ускорения сходимости протокола RIP?
11. Назовите основные этапы построения таблицы маршрутизации по протоколу OSPF.
12. Какую роль играют сообщения HELLO в протоколе OSPF? Варианты ответов:
    - устанавливают соединение между двумя маршрутизаторами;
    - проверяют состояние линий связи и соседних маршрутизаторов;
    - информируют о том, что в сети работает протокол OSPF.
13. Какие типы метрик поддерживает протокол OSPF?
14. Для чего сеть маршрутизаторов, поддерживающих протокол OSPF, разбивают на области?
15. Что можно отнести к недостаткам протокола OSPF?
16. Почему протокол EGP больше не применяется в Интернете?
17. За счет какого механизма протокол BGP может работать в сетях, где между автономными системами существуют петли?
18. Какие параметры в объявлении, полученном от какой-либо автономной системы, меняет BGP-маршрутизатор, когда передает его в другую автономную систему?
19. В каких случаях ICMP-сообщение об ошибке не может быть послано при возникновении проблемы с IP-пакетом?
20. Кому адресовано ICMP-сообщение? Какой программный модуль обрабатывает его?
21. Каким образом ICMP-сообщение повышает надежность передачи данных по IP-сети?
22. Проведите с партнером сеанс моделирования работы протокола TCP. Для этого договоритесь с ним о максимальном размере сегмента, о начальных размерах буферов, о начальном значении порядкового номера, о размерах окна. Затем асинхронно начните «посылать» друг другу «сегменты» — карточки, на кото-

рых заполнены ключевые поля — номер первого байта, размер посылаемого сегмента, номер квитанции и, если требуется, новое значение размера окна. Время от времени «теряйте» карточки при передаче и выполняйте действия, соответствующие логике TCP. Не забудьте делать временные отметки на каждой копии отправленного сегмента, чтобы отслеживать приход квитанций. Поверьте, такое ваше перевоплощение в модуль TCP не только сделает его для вас более понятным, но и, что гораздо важнее, породит новые вопросы.

23. Сколько времени в худшем случае будет продолжаться период несогласованного состояния таблиц маршрутизации в сети, изображенной на рис. 19.15, после потери маршрутизатором R1 связи с сетью 201.36.14.0. Считайте, что все маршрутизаторы поддерживают механизм расщепления горизонта.
24. Предложите варианты метрики, которая одновременно учитывает пропускную способность, надежность и задержку линий связи.

ГЛАВА 20 Дополнительные функции маршрутизаторов IP-сетей

Основными функциями IP-маршрутизатора являются создание таблицы маршрутизации и продвижение IP-пакетов на основе данных этой таблицы. Для выполнения этих функций маршрутизатор должен поддерживать протокол IP, рассмотренный в главе 18, а также протоколы маршрутизации, с которыми мы познакомились в главе 19. Помимо этих базовых функций современные IP-маршрутизаторы поддерживают ряд важных более сложных возможностей, которые превращают IP-маршрутизаторы в гибкие и мощные многофункциональные устройства по обработке трафика. В этой главе мы рассмотрим наиболее важные из этих нетривиальных возможностей, наиболее часто используемые администраторами сетей.

Маршрутизатор является пограничным устройством, соединяющим сеть с внешним миром. Поэтому естественно возложить на него *функции по защите сети* от внешних атак. Эти функции IP-маршрутизаторы выполняют путем фильтрации пользовательского трафика в соответствии с разнообразными признаками, передаваемыми в IP-пакетах: адресами отправителя и получателя, идентификатором типа протокола, вложенным в IP-пакет, идентификатором типа приложения, генерирующего этот трафик. Подобная функциональность предотвращает проникновение нежелательного трафика во внутреннюю сеть и снижает вероятность атаки на ее хосты. Важную роль в защите внутренних ресурсов сети играет *технология трансляции сетевых адресов* (NAT), которая позволяет скрыть от внешних пользователей реальные адреса, используемые хостами сети.

Сравнительно новым свойством IP-сетей является *поддержка параметров качества обслуживания* (QoS). Отдельные механизмы, необходимые для контроля и предотвращения перегрузок, IP-маршрутизаторы поддерживают на протяжении уже долгого времени, однако стандарты систем обеспечения QoS были разработаны для IP-сетей только в конце 90-х. Существует две технологии поддержания параметров QoS для IP-сетей — интегрированное обслуживание и дифференцированное обслуживание. Первая обеспечивает качество обслуживания для отдельных потоков, а вторая разработана для агрегированных потоков, представляющих небольшое число классов трафика. Технология интегрированного обслуживания сегодня находит основное применение на периферии сети, в корпоративных сетях и сетях доступа, в то время как технология дифференцированного обслуживания начинает применяться на магистралях сетей. Причина такого разделения областей применения достаточно очевидна — обеспечение параметров QoS для каждого отдельного потока создает для маршрутизаторов дополнительную нагрузку, пропорциональную количеству обслуживаемых потоков. Через магистраль могут проходить сотни тысяч пользовательских потоков, поэтому реализация технологии интегрированного обслуживания может потребовать от магистрального маршрутизатора слишком высоких показателей вычислительной мощности и оперативной памяти.

Завершает главу рассмотрение функциональной структуры маршрутизаторов.

Фильтрация

Ключевые слова: фильтрация данных, фильтрация маршрутных объявлений, списки доступа маршрутизаторов, стандартный список доступа, метасимволы, ключевые слова списка доступа, ICMP, ping.

Протоколы IP-маршрутизации создают таблицы маршрутизации, на основе которых любой узел составной сети может обмениваться информацией с любым другим узлом. Благодаря этому принципу дейтаграммных сетей каждый пользователь Интернета может получать доступ к любому публичному сайту.

Напомним, что в сетях, основанных не технике виртуальных каналов, взаимодействие произвольных узлов невозможно без предварительной процедуры установления между ними виртуального канала.

Однако такая всеобщая достижимость узлов и сетей не всегда отражает потребности их владельцев. Поэтому многие маршрутизаторы поддерживают развитые средства фильтрации пользовательского трафика, а также фильтрации объявлений протоколов маршрутизации, что позволяет дифференцированно управлять достижимостью узлов.

Фильтрация пользовательского трафика

Под **фильтрацией** понимается нестандартная обработка IP-пакетов маршрутизаторами, приводящая к отбрасыванию некоторых пакетов или изменению их маршрута.

Фильтрация пользовательского трафика маршрутизаторами аналогична по принципу действия фильтрации, выполняемой коммутаторами локальных сетей (см. главу 15).

Условия фильтрации маршрутизаторов обычно существенно сложнее и учитывают гораздо больше признаков, чем коммутаторы локальных сетей. Например, это могут быть:

- IP-адрес источника и приемника;
- MAC-адреса источника и приемника;
- идентификатор интерфейса, с которого поступил пакет;
- тип протокола, сообщение которого несет IP-пакет (то есть TCP, UDP, ICMP или OSPF);
- номер порта TCP/UDP (то есть тип протокола прикладного уровня).

При наличии фильтра маршрутизатор сначала проверяет совпадение условия, описанного этим фильтром, с признаками пакета, и при положительной проверке выполняет над пакетом ряд нестандартных действий. Например, пакет может быть отброшен (drop); направлен к следующему маршрутизатору, отличающемуся от того, который указан в таблице маршрутизации; помечен, как вероятный

кандидат на отбрасывание при возникновении перегрузки. Одним из таких действий может быть и обычная передача пакета в соответствии с записями таблицы маршрутизации

Рассмотрим примеры фильтров, написанных на командном языке маршрутизаторов Cisco. Эти фильтры, называемые **списками доступа**, сегодня в IP-маршрутизаторах являются очень распространенным средством ограничения пользовательского трафика.

Наиболее простым является **стандартный список доступа**, который учитывает в качестве условия фильтрации только IP-адрес источника.

Общая форма такого условия выглядит следующим образом:

```
access-list номер_списка_доступа { deny | permit }
  { адрес_источника [ метасимволы_источника ] | any}
```

Стандартный список доступа определяет два действия с пакетом, который удовлетворяет описанному в фильтре условию: deny, то есть отбросить, и permit, то есть передать для стандартной обработки в соответствии с таблицей маршрутизации. Условием выбора того или иного действия в стандартном списке доступа является совпадение IP-адреса источника пакета с адресом источника, заданным в списке. Совпадение проверяется в том же стиле, что и при проверке таблицы маршрутизации, при этом **метасимволы** являются аналогом маски, но в несколько модифицированном виде. Двоичный нуль в поле метасимволов источника означает, что требуется совпадение значения этого разряда в адресе пришедшего пакета и в адресе, заданном в списке доступа. Двоичная единица означает, что совпадения в этом разряде не требуется. Практически, если вы хотите задать условие для всех адресов некоторой подсети, то должны использовать инвертированное значение маски этой подсети. Параметр any означает любое значение адреса — это просто более понятная и краткая форма записи значения 255.255.255.255 в поле метасимволов источника.

Пример стандартного списка доступа:

```
access-list 1 deny 192.78.46.0 0.0.0.255
```

Здесь:

- 1 — номер списка доступа;
- deny — действие с пакетом, который удовлетворяет условию данного списка доступа;
- 192.78.46.0 — адрес источника;
- 0.0.0.255 — метасимволы источника.

Этот фильтр запрещает передачу пакетов, у которых в старших трех байтах адреса источника имеется значение 192.78.46.0.

Список доступа может включать более одного условия. В этом случае он состоит из нескольких строк с ключевым словом access-list и одним и тем же номером списка доступа. Так, если мы хотим разрешить прохождение через маршрутиза-

тор пакетов хоста 192.78.46.12, запрещая передачу пакетов одному из хостов сети 192.78.46.0/24, то список доступа будет выглядеть следующим образом:

```
access-list 1 permit 192.78.46.12 0.0.0.0
access-list 1 deny 192.78.46.0 0.0.0.255
access-list 1 permit any
```

Условия списка доступа проверяются по очереди, если какое-либо из них дает совпадение, то выполняется действие permit или deny, определенное в этом условии. После этого остальные условия списка уже не проверяются. Считается по умолчанию, что в конце каждого списка имеется неявное условие вида:

```
[access-list 1 deny any]
```

Однако, если вам все же требуется пропускать все пакеты, не определенные явно в условиях, необходимо добавить в последней строке условие:

```
access-list 1 permit any
```

Список доступа можно применять к любому интерфейсу маршрутизатора и в любом направлении: если список применяется с ключевым словом in, то он действует на входящие в интерфейс пакеты, а если с ключевым словом out — то на выходящие. Например, написанный нами список доступа 1 можно применить к некоторому интерфейсу для обработки входящего трафика, используя следующую команду:

```
access-group 1 in
```

Существуют также и более мощные типы списков доступа для маршрутизаторов Cisco, например расширенные списки доступа. Общий формат этих списков следующий:

```
access-list номер_списка_доступа { deny | permit }
  { protocol | ключевое_слово_протокола }
  { адрес_источника [ метасимволы_источника ]] [ порт_источника ] | any }
  [ адрес_приемника [ метасимволы_приемника ]] [ порт_приемника ]
```

Пользуясь расширенными списками доступа, можно запретить прохождение во внутреннюю сеть предприятия FTP-пакетов. Как известно, FTP задействует для приема запросов от клиентов протокол TCP с хорошо известным портом 21. Для этого в список доступа нужно включить условие:

```
access-list 102 deny TCP any 21 any
```

Затем можно применить его к интерфейсу маршрутизатора, к которому подключена внутренняя сеть, с ключевым словом out.

Администраторы корпоративных сетей часто запрещают возможность трассировки извне внутренних хостов утилитой ping. Это делается с помощью условия:

```
access-list 101 deny ICMP any 192.78.46.8 0.0.0.0 eq 8
```

Как видно из условия, его синтаксис для протокола ICMP несколько отличается от общего синтаксиса расширенных списков доступа. Параметр eq 8 означает,

что запрещается передача ICMP-сообщений типа 8, соответствующего эхо-запросам, на основе которых разработана утилита ping.

Еще более гибким является язык фильтров программного маршрутизатора, работающего во многих версиях Unix. Синтаксис этого языка близок к синтаксису языка C, что позволяет строить весьма сложные логические конструкции с помощью условных операторов if, then, else.

Необходимо отметить, что фильтрация пользовательского трафика может существенно замедлять работу маршрутизатора, так как обработка каждого пакета требует проверки дополнительных условий.

Для того чтобы не создавать еще большую нагрузку на маршрутизатор и «не отвлекать» его от выполнения основных обязанностей, в фильтрах маршрутизаторов не используется информация о предыстории сеансов. Сколько бы ни было сложным условие фильтрации маршрутизатора, в нем учитываются только параметры *текущего* пакета и не могут учитываться параметры предыдущих пакетов, уже обработанных маршрутизатором. Это ограничение является главным отличием маршрутизаторов от брандмауэров, специальных программных систем, которые, используя информацию о предыстории сеансов, выполняют более качественную фильтрацию.

Фильтрация маршрутных объявлений

Для контроля достижимости узлов и сетей можно, наряду с фильтрацией пользовательского трафика, *ограничить распространение объявлений протоколов маршрутизации*. Такая мера предотвращает автоматическое появление в таблице маршрутизации записей о некоторых сетях. Этот способ требует гораздо меньших затрат вычислительной мощности маршрутизатора, так как маршрутные объявления поступают в маршрутизатор гораздо реже, чем пользовательские пакеты.

Пусть, например, маршрутизаторы Cisco должны ограничить распространение маршрутных объявлений о какой-нибудь сети. Для этого нужно включить описание данной сети в стандартный список доступа, а затем применить к интерфейсу специальную команду с ключевым словом distribute-list (вместо access-group, как в случае фильтрации пользовательского трафика).

Например, если администратор не хочет, чтобы информация о внутренних сетях предприятия 194.12.34.0/24 и 132.7.0.0/16 распространялась по внешним сетям, ему достаточно написать следующий стандартный список доступа:

```
access-list 2 deny 194 12.34.0 0.0.0.255
access-list 2 deny 132.7.0.0 0.0.255.255
access-list 2 permit any
```

После этого достаточно применить его к интерфейсу с помощью команды

```
distribute-list 2 out serial 1
```

Стандарты QoS в IP-сетях

Ключевые слова: система интегрированного обслуживания, система дифференцированного обслуживания, классы трафика, RSVP, QoS, ISP, микропоток, агрегированный поток, MPLS, алгоритм «ведра маркеров», алгоритм случайного раннего обнаружения, взвешенный алгоритм, профиль трафика, эталонный поток, маркер (токен), сглаживание, профилирование, пульсация, протокол резервирования ресурсов, PATH-сообщение, RESV-сообщение, резервирование ресурсов, дескриптор потока, спецификация фильтра, спецификация запроса приемника, спецификация трафика источника, быстрое продвижение, гарантированная доставка.

Технологии стека TCP/IP были созданы для эластичного трафика, который достаточно терпим к задержкам и вариациям задержек пакетов. Поэтому основное внимание разработчиков протоколов TCP/IP было сосредоточено на обеспечении надежной передачи трафика с помощью TCP. Тем не менее для борьбы с перегрузками на медленных линиях доступа в IP-маршрутизаторы со временем были встроены многие механизмы QoS, в том числе механизмы приоритетных и взвешенных очередей, профилирования трафика и обратной связи. Однако эти механизмы использовались каждым сетевым администратором по своему усмотрению, без какой-либо стройной системы. И только в середине 90-х годов начались работы по созданию стандартов QoS для IP-сетей, на основе которых можно было бы создать систему поддержки параметров QoS в масштабах составной сети и даже Интернета.

В результате были созданы две системы стандартов QoS для IP-сетей:

- **система интегрированного обслуживания** (Integrated Services, IntServ) ориентирована на предоставление гарантий QoS для потоков *конечных пользователей* (именно поэтому технология IntServ используется в основном на периферии сети);
- **система дифференцированного обслуживания** (Differentiated Services, DiffServ) делает то же самое *для классов трафика*, и, следовательно, ее предпочтительнее использовать на магистрали.

Обе системы используют все базовые элементы схемы поддержания параметров QoS, основанной на резервировании, то есть:

- кондиционирование трафика;
- сигнализацию для координации маршрутизаторов;
- резервирование пропускной способности интерфейсов маршрутизаторов для потоков и классов;
- приоритетные и взвешенные очереди.

Ни одна из этих технологий не решает проблемы инжиниринга трафика, так как пакеты по-прежнему направляются вдоль пути с наилучшей метрикой, выбираемому стандартным протоколом маршрутизации без учета реальной загрузки каналов передачи данных.

Модели качества обслуживания IntServ и DiffServ

Направление **IntServ** начало разрабатываться в IETF еще в начале 90-х годов и было первым направлением, в рамках которого проблема обеспечения параметров QoS в сетях TCP/IP начала решаться систематически. Базовая модель IntServ предполагает интегрированное взаимодействие маршрутизаторов сети по обеспечению требуемого качества обслуживания *вдоль всего пути микропотока* между конечными компьютерами.

Ресурсы маршрутизаторов (пропускная способность интерфейсов, размеры буферов) распределяются в соответствии с QoS-запросами приложений в пределах, разрешенных политикой QoS для данной сети. Эти запросы распространяются по сети сигнальным протоколом **RSVP**, который позволяет выполнять резервирование ресурсов для потоков данных.

Однако система IntServ обеспечения параметров QoS нашла довольно много противников, преимущественно среди поставщиков услуг Интернета (ISP). Дело в том, что при интегрированном обслуживании магистральные ISP-маршрутизаторы должны оперировать информацией о состоянии десятков тысяч микропотоков, проходящих через ISP-сети. Такая нетрадиционная нагрузка на маршрутизаторы требует коренного пересмотра их архитектуры и, естественно, ведет к резкому повышению стоимости IP-сетей и предоставляемых ими услуг.

Поэтому в конце 90-х была создана другая, более экономически эффективная технология QoS в IP-сетях, получившая название дифференцированного обслуживания (**DiffServ**). Она изначально была ориентирована на применение в пределах ISP-сетей, а конечные узлы, генерирующие микропотоки, в расчет не брались. Для технологии DiffServ поддержка параметров QoS начинается на пограничном маршрутизаторе ISP-сети, на который поступает большое количество микропотоков из сетей пользователей. Каждый пограничный маршрутизатор классифицирует и маркирует входящий трафик, разделяя его на небольшое число классов, обычно 3–4 (максимум — 8). Затем каждый маршрутизатор сети обслуживает классы трафика дифференцированно в соответствии с произведенной маркировкой, выделяя каждому классу определенное количество ресурсов. Резервирование ресурсов маршрутизаторов производится статически, чаще всего вручную администратором сети. Роль сигнального протокола играют метки принадлежности пакетов к тому или иному классу.

Ответственность за согласованное обслуживание трафика всеми маршрутизаторами сети несет администратор, так как он принимает решение, какие пропускную способность и величину буфера выделить каждому классу на каждом интерфейсе каждого маршрутизатора.

Модель DiffServ существенно снижает нагрузку на маршрутизаторы ISP-сети, так как требует хранить информацию о состоянии только небольшого количества классов. Кроме того, эта модель удобна для поставщиков услуг тем, что позволяет поддерживать параметры QoS автономно, только в пределах своих сетей. Однако за эти преимущества приходится платить, и, прежде всего, отказом от

гарантии сквозной поддержки параметров QoS. Даже если каждый поставщик услуг обеспечит дифференцированное обслуживание в своей сети, общая картина получится фрагментированной, так как за каждый фрагмент отвечает отдельный администратор, и согласование параметров резервирования остается исключительно субъективной процедурой, не поддерживаемой никакими протоколами.

Несмотря на большое внимание, уделяемое в последнее время дифференцированному обслуживанию как простому средству, с помощью которого можно уже сейчас обеспечить повышение качества обслуживания в Интернете без больших затрат, существуют и другие точки зрения. Так, Лоуренс Робертс (Lawrence G. Roberts), один из тех, кто стоял у истоков Интернета, высказал весьма резкое отрицательное отношение к попыткам упрощенно решить проблему поддержания качества обслуживания в Интернете (см. статью Lawrence G. Roberts «Judgment Call», в журнале *Data Communications* за апрель 1999 года).

Ведутся также работы по *комбинированному применению технологий IntServ и DiffServ*. Каждая технология в этих моделях работает в своей области, IntServ — в сетях доступа, где количество микропотоков относительно невелико, а DiffServ — в магистральных сетях. Еще одним компонентом, дополняющим DiffServ, является технология MPLS, позволяющая решать задачи инжиниринга трафика в IP-сетях. Эта технология рассматривается в части V, посвященной глобальным сетям, так как она появилась в результате «скрещивания» IP с такой популярной технологией глобальных сетей, как ATM, и ее проще изучать после рассмотрения ATM.

Обе технологии (IntServ и DiffServ) опираются на одни и те же базовые механизмы QoS. В частности, в IP-маршрутизаторах для профилирования и формирования трафика применяется *алгоритм ведра маркеров*.

Алгоритм ведра маркеров

Алгоритм ведра маркеров позволяет оценить и ограничить среднюю скорость и величину пульсации потока пакетов. Этот алгоритм основан на сравнении потока пакетов с некоторым эталонным потоком. Эталонный поток представлен маркерами, заполняющими условное ведро маркеров (рис. 20.1).

Под маркером в данном случае понимается некий абстрактный объект, носитель «порции» информации, используемый для построения эталонного потока. Генератор маркеров периодически с постоянным интервалом w направляет очередной маркер в ведро с ограниченным объемом b байт. Все маркеры имеют одинаковый объем m байт, а генерация маркеров происходит так, что ведро заполняется со скоростью r бит в секунду. Нетрудно подсчитать, что r = 8m/w. Эта скорость r и является максимальной средней скоростью для трафика пакетов, а объем ведра соответствует максимальному размеру пульсации потока пакетов. Если ведро заполняется маркерами «до краев» (то есть суммарный объем маркеров в ведре становится равным b), то поступление маркеров временно пре-

кращается. Фактически ведро маркеров представляет собой счетчик, который наращивается на m каждые w секунд.

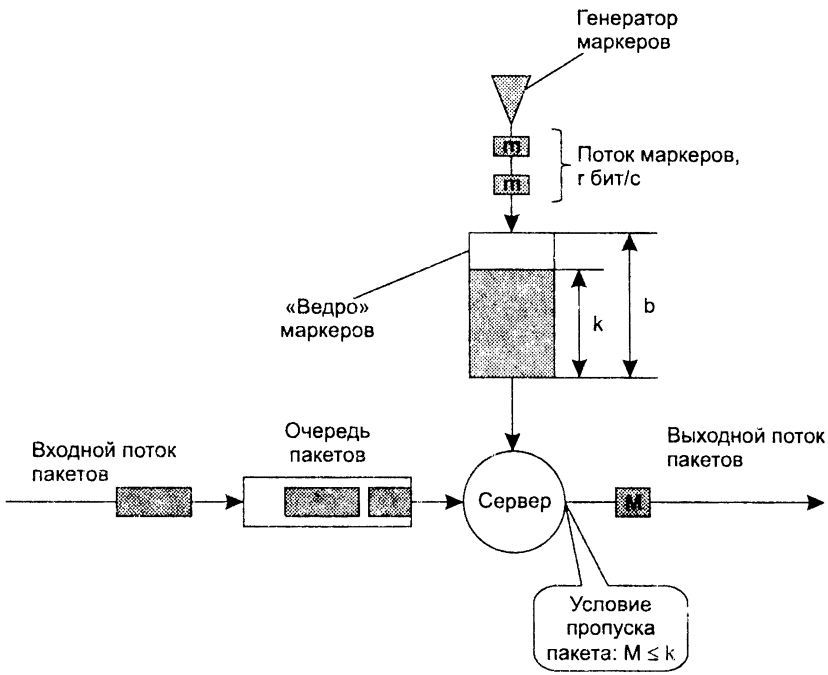


Рис. 20.1. Алгоритм ведра маркеров

При применении алгоритма ведра маркеров профиль трафика определяется **средней скоростью** r и **объемом пульсации** b.

Сравнение эталонного и реального потоков выполняет сервер — абстрактное устройство, которое имеет два входа. Вход 1 связан с очередью пакетов, а вход 2 — с ведром маркеров. Сервер также имеет выход, на который он передает пакеты из входной очереди пакетов. Вход 1 сервера моделирует входной интерфейс маршрутизатора, а выход — выходной интерфейс.

Пакет из входной очереди продвигается сервером на выход только в том случае, если к моменту его поступления на сервер «ведро» заполнено маркерами до уровня не ниже M байт, где M — объем пакета.

При продвижении пакета из ведра удаляются маркеры общим объемом в M байт (с точностью до размера одного маркера, то есть до m байт).

Если же ведро заполнено недостаточно, то пакет обрабатывается одним из двух описанных ниже нестандартных способов, выбор которых зависит от цели применения алгоритма.

- Если алгоритм ведра маркеров применяется для *сглаживания* трафика, то пакет просто задерживается в очереди на некоторое дополнительное время, ожидая поступления в ведро нужного числа маркеров. Таким образом, даже если в результате пульсации в систему приходит большая группа пакетов, из очереди пакеты выходят более равномерно, в темпе, задаваемом генератором маркеров.
- Если же алгоритм ведра маркеров используется для *профилирования* трафика, то пакет отбрасывается, как не соответствующий профилю. Более мягким решением может быть повторная маркировка пакета, понижающая его статус при дальнейшем обслуживании. Например, пакет может быть помечен особым признаком (удалять при необходимости), в результате чего при перегрузках маршрутизаторы будут отбрасывать этот пакет в первую очередь. При дифференцированном обслуживании пакет может быть переведен в другой класс, который обслуживается с более низким качеством.

ПРИМЕЧАНИЕ

Алгоритм ведра маркеров допускает пульсацию трафика в определенных пределах. Пусть пропускная способность выходного интерфейса, который моделируется выходом сервера, равна R. Это значит, что сервер не может передавать данные на выход со скоростью, превышающей R бит/с. Можно показать, что на любом интервале времени t средняя скорость исходящего с сервера потока равна минимуму из двух величин: R и r + b/t. При больших значениях t скорость выходного потока стремится к r — это и говорит о том, что алгоритм обеспечивает желаемую среднюю скорость. В то же время в течение небольшого времени t пакеты могут выходить из сервера со скоростью, большей r. Если r + b/t < R, то они выходят из сервера со скоростью r + b/t, в противном случае интерфейс ограничивает эту скорость до величины R. Период времени t соответствует пульсации трафика. Эта ситуация наблюдается тогда, когда в течение некоторого времени пакеты не поступали в сервер, так что ведро полностью заполнилось маркерами (то есть времени, большего, чем b/r). Если после этого на вход сервера поступит большая последовательность пакетов, следующих один за другим, то эти пакеты будут передаваться на выход со скоростью выходного интерфейса R также один за другим, без интервалов. Максимальное время такой пульсации составляет b/(R – r) секунд, после чего обязательно наступит пауза, необходимая для наполнения опустевшего ведра. Объем пульсации составляет Rb/(R – r) байт. Из приведенного соотношения видно, что алгоритм ведра маркеров начинает плохо работать, если средняя скорость r выбирается близкой к пропускной способности выходного интерфейса. В этом случае пульсация может продолжаться очень долго, что обесценивает алгоритм.

Случайное раннее обнаружение

Механизм профилирования TCP-трафика, названный **случайным ранним обнаружением** (Random Early Detection, RED), разработан сообществом Интернета для предотвращения перегрузок на магистралях Интернета.

RED работает с TCP и основан на свойстве TCP, которое заключается в том, что при потерях пакетов источник трафика замедляет передачу пакетов в сеть. В алгоритме RED используется два конфигурируемых порога уровня перегрузки (рис. 20.2). Когда уровень перегрузки не превышает первого (нижнего) порога, то пакеты не отбрасываются. Когда уровень перегрузки находится между двумя

порогами, пакеты отбрасываются с линейно возрастающей вероятностью из диапазона от 0 до конфигурируемой величины (максимальной вероятности отбрасывания пакета). Максимальная вероятность отбрасывания действует при достижении второго (верхнего) порога. Когда же перегрузка превышает второй порог, пакеты начинают отбрасываться с вероятностью 100 %.

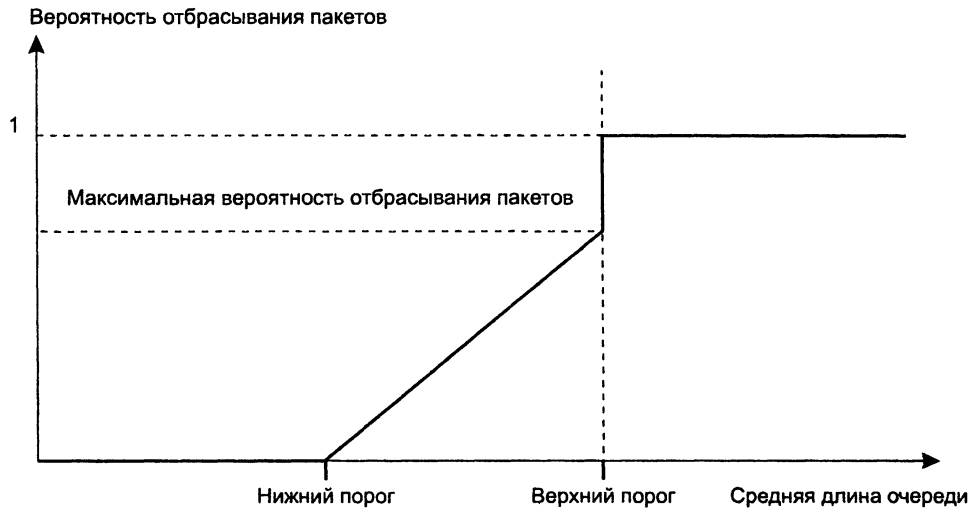


Рис. 20.2. Вероятность отбрасывания пакетов алгоритмом RED

В качестве показателя перегрузки используется вычисляемое среднее значение длины очереди пакетов, относящейся к определенному TCP-сеансу.

ПРИМЕЧАНИЕ

Заметим, что для UDP-трафика механизм RED неприменим, так как протокол UDP работает без установления логического соединения и, следовательно, потерь пакетов не замечает.

В том случае, когда нужно обеспечить разные параметры обратной связи для разных классов трафика, применяется **взвешенный алгоритм RED** (Weighted RED, WRED). Этот вариант алгоритма RED позволяет задавать для каждого класса трафика свои значения нижнего и верхнего порогов, а также вероятность отбрасывания пакетов. Обычно механизмы WRED и WFQ применяются совместно, обеспечивая надежную доставку TCP-трафика с гарантированной скоростью.

Интегрированное обслуживание и протокол RSVP

Интегрированное обслуживание основано на резервировании ресурсов маршрутизаторов вдоль пути следования потока данных от одного конечного узла

(точнее, приложения) до другого (рис. 20.3). Приложение должно использовать соответствующий интерфейс API, чтобы передать запрос о резервировании ресурсов для определенного потока. Подобное резервирование является *однонаправленным*, так что если гарантированное качество обслуживания должно быть обеспечено для двустороннего обмена, потребуется две операции резервирования.

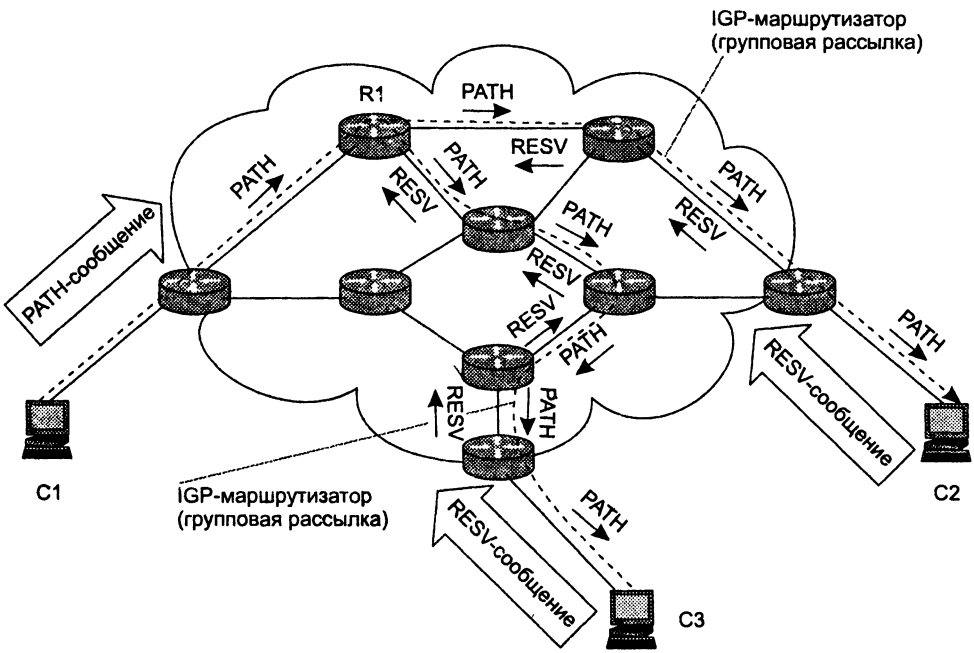


Рис. 20.3. Резервирование ресурсов по протоколу RSVP

Резервирование в модели IntServ выполняется с помощью **протокола резервирования ресурсов** (RSVP). Это сигнальный протокол, во многом подобный *сигнальным протоколам телефонных сетей*. Однако специфика дейтаграммных пакетных сетей естественно накладывает свой отпечаток. Так, параметры коммутации в IP-сетях не являются атрибутом резервирования, потому что IP-пакеты в любом случае (при резервировании или без него) будут передаваться маршрутизаторами на основе записей таблицы маршрутизации.

Ниже описана процедура резервирования необходимых ресурсов сети с помощью протокола RSVP, а в табл. 20.1 сведены воедино все упоминаемые в этом описании типы сообщений.

1. Источник данных (компьютер C1 на рис. 20.3) посылает получателям по уникальному или групповому (как на рисунке) адресу специальное **PATH-сообщение**, в котором указывает рекомендуемые параметры для качественного приема своего трафика: верхние и нижние границы пропускной способности, задержки и вариации задержки. Эти параметры составляют **спецификацию трафика источника**. PATH-сообщение передается маршрутизаторами сети

в направлении ко всем указанным в групповом адресе получателям. В качестве параметров трафика применяются параметры алгоритма ведра маркеров, то есть средняя скорость и глубина ведра. Кроме того, дополнительно могут быть заданы максимально допустимая скорость и предельные размеры пакетов потока.

2. Каждый маршрутизатор, поддерживающий протокол RSVP, получив PATH-сообщение, фиксирует «состояние пути», которое включает предыдущий адрес источника PATH-сообщения, то есть последний по времени шаг в обратном направлении (ведущий к источнику). Это необходимо для того, чтобы ответ приемника прошел по тому же пути, что и PATH-сообщение.

3. После получения PATH-сообщения приемник отправляет в обратном направлении маршрутизатору, от которого он получил это сообщение, **запрос на резервирование ресурсов** (RESerVation Request, RESV), то есть **RESV-сообщение**. На рис. 20.3 показано два приемника, компьютеры C2 и C3. В дополнение к спецификациям трафика источника C1 (которые содержат параметры для качественного приема его трафика: верхние и нижние границы пропускной способности, задержки и вариации задержки) RESV-сообщение дополнительно включает **спецификацию запроса приемника**, в которой указываются требуемые *приемнику* параметры качества обслуживания, и **спецификацию фильтра**, которая определяет, к каким пакетам сеанса применять данное резервирование (например, по типу транспортного протокола и номеру порта). Вместе спецификации запроса и фильтра представляют собой **дескриптор потока**, для которого выполняется резервирование. Запрашиваемые параметры QoS в спецификации запроса могут отличаться от указанных в спецификации трафика. Например, если приемник решает принимать не все посылаемые источником пакеты, а только их часть (что указывается в спецификации фильтра), то ему нужна, соответственно, меньшая пропускная способность.

4. Каждый маршрутизатор, поддерживающий протокол RSVP вдоль восходящего пути, получив RESV-сообщение, выполняет две проверки: во-первых, имеются ли у маршрутизатора ресурсы, необходимые для поддержания запрашиваемого уровня QoS, а во-вторых, имеет ли пользователь право на резервирование ресурсов. Если запрос не может быть удовлетворен (из-за недостатка ресурсов или ошибки авторизации), маршрутизатор возвращает сообщение об ошибке отправителю. Если запрос принимается, то маршрутизатор посылает RESV-сообщение далее вдоль маршрута следующему маршрутизатору, а данные о требуемом уровне QoS передаются механизмам маршрутизатора, ответственным за управлением трафиком.

5. Прием маршрутизатором запроса на резервирование ресурсов означает также передачу параметров QoS на обработку в соответствующие блоки маршрутизатора. Конкретный способ обработки параметров QoS маршрутизатором в протоколе RSVP не описывается, но обычно она заключается в том, что маршрутизатор проверяет наличие свободной пропускной способности и емкости памяти для нового резервирования. При положительном результате проверки

маршрутизатор запоминает новые параметры резервирования и вычитает их из счетчиков соответствующих свободных ресурсов.

6. Когда последний в обратном направлении маршрутизатор получает RESV-сообщение и принимает запрос, то он посылает подтверждающее сообщение узлу-источнику. При групповом резервировании учитывается тот факт, что в точках разветвления дерева доставки несколько резервируемых потоков сливаются в один. Так, в маршрутизаторе R1 в рассматриваемом примере сливаются RESV-сообщения от приемников C2 и C3. Если для всех резервируемых потоков запрашивается одинаковая пропускная способность, то она требуется и для общего потока, а если запрашиваются различные величины пропускной способности, то для общего потока выбирается максимальная.
7. После установления состояния резервирования в сети источник начинает отправлять данные, которые обслуживаются на всем пути к приемнику (приемникам) с заданным качеством обслуживания.

Таблица 20.1. Таблица сообщений протокола резервирования ресурсов (RSVP)

Типы сообщений	Содержание сообщений
PATH-сообщение от источника к приемнику	Спецификация трафика источника
Спецификация трафика источника	Рекомендуемые параметры для качественного приема своего трафика: верхние и нижние границы пропускной способности, задержки и вариации задержки, параметры алгоритма ведра маркеров, то есть средняя скорость и глубина ведра, дополнительно могут быть заданы максимально допустимая скорость и предельные размеры пакетов потока
Спецификация фильтра	Определяет, к каким пакетам сеанса применять данное резервирование (например, по типу транспортного протокола и номеру порта)
Спецификация запроса приемника	Требуемые приемнику параметры качества обслуживания
Дескриптор потока	Спецификация фильтра плюс спецификация запроса приемника
RESV-сообщение — запрос на резервирование ресурсов	Спецификация трафика источника плюс дескриптор потока

Нужно подчеркнуть, что описанная схема выполняет резервирование только в одном направлении. Для того чтобы в рамках пользовательского сеанса данные передавались с заданным качеством обслуживания также и в обратном направлении, нужно, чтобы приемник и источник поменялись местами и выполнили RSVP-резервирование еще раз.

Для того чтобы параметры резервирования можно было применить затем к трафику данных, необходимо, чтобы RSVP-сообщения и пакеты данных следовали

через сеть *одним и тем же маршрутом*. Это можно обеспечить, если передавать RSVP-сообщения на основе тех же записей таблиц маршрутизации, которые применяются для пользовательского трафика.

ВНИМАНИЕ

Если для передачи RSVP-сообщений будет использоваться традиционная схема выбора маршрута из таблиц маршрутизации, то при этом потеряется возможность полноценного решения задач инжиниринга трафика, так как не все возможные маршруты будут задействованы для резервирования, а только кратчайший маршрут, выбранный в соответствии с некоторой метрикой протокола маршрутизации.

Резервирование можно отменить прямо или косвенно. Прямая отмена выполняется по инициативе источника или приемника с помощью соответствующих сообщений протокола RSVP. Неявная отмена происходит по тайм-ауту: состояние резервирования имеет срок жизни, как, например, и динамические записи в таблицах маршрутизации, и приемник по протоколу RSVP должен периодически подтверждать резервирование. Если же подтверждающие сообщения перестают поступать, то резервирование отменяется по истечении его срока жизни. Такое резервирование называется мягким.

Для протокола RSVP в настоящее время разработано большое количество расширений, которые делают его пригодным не только для работы в рамках архитектуры RSVP. Одними из наиболее важных являются расширения, относящиеся к инжинирингу трафика. Эти расширения применяются в технологии MPLS, рассматриваемой в части V.

Дифференцированное обслуживание

Дифференцированное обслуживание (DiffServ) опирается на ту же обобщенную модель QoS, что и интегрированное обслуживание, однако в качестве объектов обслуживания рассматриваются не отдельные потоки, а классы трафика.

Напомним, что **классом трафика** называется совокупность поступающих на обработку пакетов, обладающих общими признаками, например все пакеты голосовых приложений или все пакеты с MTU в определенных пределах.

В отличие от потока класс трафика не различает пакеты в зависимости от их маршрута, и рис. 20.4 иллюстрирует это отличие. Так, маршрутизатор R1 относит все потоки, требующие приоритетного обслуживания и втекающие в его интерфейс i1, к одному классу, независимо от их дальнейшего маршрута. Маршрутизатор R2 оперирует уже с другим составом приоритетного класса, так как в него вошли не все потоки интерфейса i1 маршрутизатора R1.

Обычно в сети DiffServ поддерживается дифференцированное обслуживание небольшого количество классов трафика, например двух (чувствительного к задержкам и эластичного) или трех (к первым двум прибавляется класс, требующий гарантированной доставки пактов с определенным минимумом скорости

трафика). Небольшое количество классов определяет масштабируемость этой модели, так как маршрутизаторы не должны запоминать состояния каждого потока пользователя. Высокая степень масштабирумости DiffServ обеспечивается также тем, что каждый маршрутизатор самостоятельно принимает решение о том, как он должен обслуживать тот или иной класс трафика, не согласовывая свои действия с другими маршрутизаторами. Такой подход назван *независимым поведением маршрутизаторов* (Per Hop Behavior, PHB). Так как в модели DiffServ маршруты пакетов не отслеживаются, то здесь не используется сигнальный протокол резервирования ресурсов, подобный RSVP модели IntServ. Вместо этого маршрутизаторы сети выполняют статическое резервирование ресурсов для каждого из поддерживаемых сетью классов.

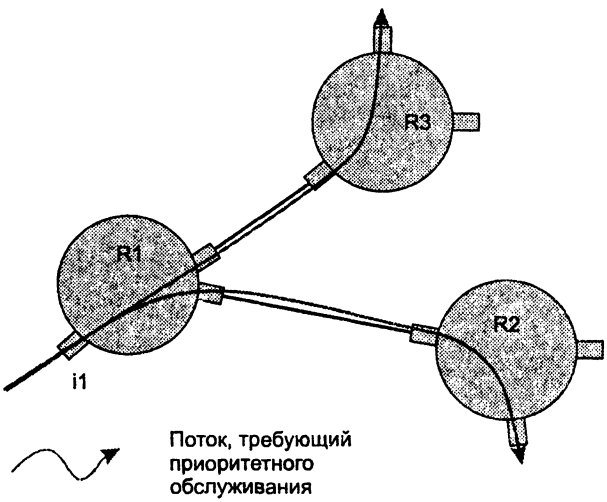


Рис. 20.4. В модели DiffServ объектами обслуживания являются классы трафика, а не потоки

В качестве признака принадлежности IP-пакета к определенному классу в DiffServ используется метка, переносимая полем приоритета IP-пакета (ToS-байт), которое с появлением стандартов DiffServ было переопределено и названо DS-байтом. Как показано на рис. 20.5, DS-байт переопределяет значения битов ToS-байта, как это было определено ранее в соответствующих спецификациях (RFC 791, RFC 1122, RFC 1349).

В настоящее время используются только старшие 6 бит DS-байта, причем только старшие 3 из них требуются для определения класса трафика (что дает не более 8 различных классов). Младший бит (из используемых шести) DS-байта обычно переносит признак IN — индикатор того, что пакет «вышел» из профиля трафика (аналогично признакам DE и технологии Frame Relay и CPL в технологии ATM). Промежуточные два бита обычно описывают различные варианты обслуживания пакетов внутри одного класса трафика.

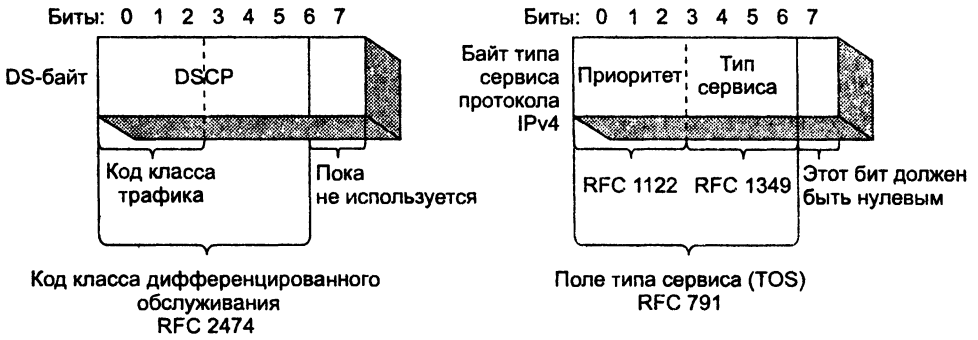


Рис. 20.5. Соответствие битов DS-байта битам поля типа сервиса

Маршрутизатор, поддерживающий модель DiffServ, должен обеспечивать классификацию, маркирование, измерение и кондиционирование трафика, его обслуживание в приоритетной или взвешенной очереди и сглаживание.

Хотя маркировкой пакетов может заниматься каждый маршрутизатор сети, в модели дифференцированного обслуживания основным вариантом считается маркировка пакетов на границе сети, поддерживающей модель DiffServ и находящейся под административным контролем одной организации. Такая сеть называется DiffServ-доменом. При выходе пакетов за пределы DiffServ-домена маркировка снимается, так что другой домен может назначить ее заново. Пограничные маршрутизаторы DiffServ исполняют роль контрольно-пропускных пунктов домена, проверяя входящий трафик и определяя, имеет ли он право на дифференцированное обслуживание.

Модель DiffServ подразумевает существование соглашения об уровне обслуживания (SLA) между доменами с общей границей. Соглашение SLA определяет критерии политики предоставления сервиса, профиль трафика, а также гарантируемые параметры QoS. Ожидается, что трафик будет формироваться и сглаживаться в выходных точках домена в соответствии с SLA, а во входной точке домена будет кондиционироваться в соответствии с правилами политики. Любой трафик «вне профиля» (например, выходящий за верхние границы полосы пропускания, указанной в SLA) не получает гарантий обслуживания (или же оплачивается по повышенной стоимости в соответствии с SLA). Правила политики предоставления сервиса могут включать время дня, адреса источника и приемника, транспортный протокол, номера портов. В том случае, когда соблюдаются правила политики и трафик удовлетворяет оговоренному профилю, домен DiffServ должен обеспечить при обслуживании этого трафика параметры QoS, зафиксированные в SLA.

На сегодняшний день IETF разработаны два стандарта пошагового продвижения пакетов для схемы PHB, которые представляют два разных варианта обслуживания.

- *Быстрое продвижение* (Expedited Forwarding, EF) характеризуется одним значением кода (10111) и представляет собой высший уровень качества обслужи-

вания, обеспечивая минимум задержек и вариаций задержек. Любой трафик, интенсивность которого превышает указанную в профиле, отбрасывается.

- *Гарантированная доставка* (Assured Forwarding, AF). Имеется четыре класса трафика и три уровня отбрасывания пакетов в каждом классе — всего 12 различных типов трафика. Каждому классу трафика выделяется определенный минимум пропускной способности и размер буфера для хранения его очереди. Трафик, параметры которого превышают указанные в профиле, доставляется с меньшей степенью вероятности, чем трафик, удовлетворяющий условиям профиля. Это означает, что качество его обслуживания может быть понижено, но он не обязательно будет отброшен.

На основе этих **пошаговых спецификаций** и соответствующих соглашениях SLA могут быть построены **сервисы** для конечных пользователей «из конца в конец» — это EF-сервис и AF-сервис соответственно.

Основное назначение EF-сервиса — предоставление качества обслуживания, сопоставимого с качеством обслуживания выделенных каналов, поэтому этот сервис называется также *сервисом виртуальных выделенных каналов.*

Поскольку EF-сервис допускает полное вытеснение другого трафика (например, при его реализации с помощью приоритетной очереди), то его реализация должна включать некоторые средства ограничения влияния EF-трафика на другие классы трафика, например, с путем ограничения скорости EF-трафика на входе маршрутизатора по алгоритму ведра маркеров. Максимальная скорость EF-трафика и, возможно, величина пульсаций должны устанавливаться сетевым администратором.

Четыре класса AF-сервиса ориентированы на гарантированную доставку, но без минимизации уровня задержек пакетов, как это оговорено для EF-сервиса. Гарантированная доставка выполняется в том случае, когда входная скорость трафика не превышает отведенной данному классу минимальной пропускной способности. Реализация классов AF-трафика хорошо сочетается с EF-сервисом — EF-трафик может обслуживаться по приоритетной схеме, но с ограничением интенсивности входного потока. Оставшаяся пропускная способность распределяется между классами AF-трафика в соответствии с алгоритмом взвешенного обслуживания, который обеспечивает необходимую пропускную способность, но не минимизацию задержек. Реализация AF-сервиса предполагает (но не требует) взвешенного обслуживания для каждого класса с резервированной полосой пропускания, а также применения обратной связи (в форме RED).

Относительная простота определяет недостатки дифференцированного обслуживания. Главным недостатком является сложность предоставления количественных гарантий пользователям. Поясним это на примере сети, изображенной на рис. 20.6.

Обслуживание классов трафика подразумевает, что пограничные маршрутизаторы выполняют профилирование трафика без учета адреса назначения пакетов. Обычно для входных интерфейсов пограничных маршрутизаторов задается некоторый порог допустимой нагрузки для трафика каждого класса. Например, пусть наша сеть поддерживает трафик двух классов, реализуя особое обслуживание и обслуживание с максимальными усилиями, причем порог для трафика с особым обслуживанием установлен в 20 % пропускной способности для каждого входного интерфейса каждого пограничного маршрутизатора. Кроме того,

предположим для упрощения рассуждений, что все интерфейсы маршрутизаторов сети имеют одинаковую пропускную способность.

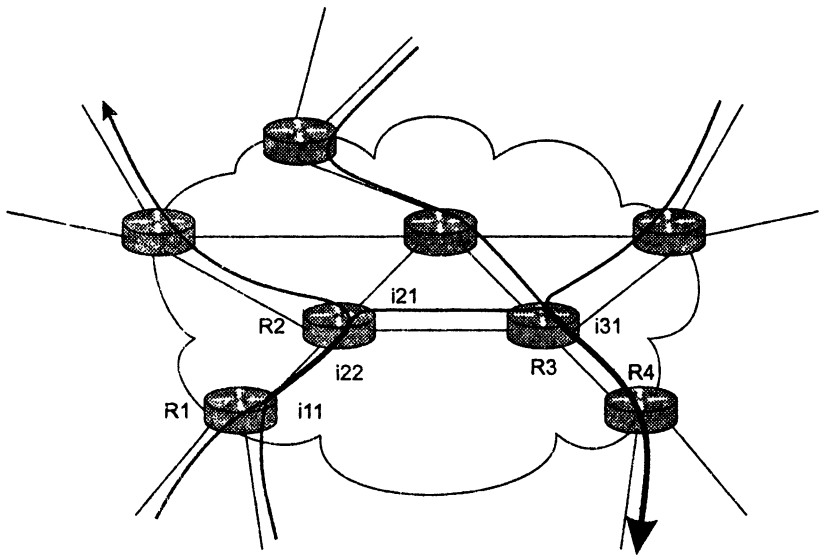


Рис. 20.6. Неопределенность уровня обслуживания в модели DiffServ

Несмотря на такое достаточно жесткое ограничение, интерфейсы маршрутизаторов сети оказываются под воздействием разной нагрузки. На рис. 20.6 для упрощения ситуации показаны только потоки, требующие особого обслуживания. Так, выходной интерфейс i11 маршрутизатора R1 обслуживает два таких потока и нагружен на 40 %, в то время как выходной интерфейс i21 маршрутизатора R2 — только один из них, так как второй поток уходит через другой выходной интерфейс. Выходной же интерфейс i31 маршрутизатора R3 перегружен, обслуживая три таких потока, так что его коэффициент использования равен 60 %. Учитывая факторы, влияющие на образование очередей (см. главу 7), мы знаем, что коэффициент использования является наиболее существенным фактором и значения в районе 50 % являются критическими. Поэтому в интерфейсе i31 возникают длинные очереди пакетов класса особого обслуживания, которые снижают качество такого обслуживания, так как приводят к длительным задержкам и их вариациям, а также потерям пакетов. Кроме того, страдает трафик класса обслуживания с максимальными усилиями, проходящий через этот интерфейс, так как ему достается только 40 % пропускной способности интерфейса.

Мы несколько утрировали картину, так как обычно интерфейсы магистральных маршрутизаторов являются более скоростными, чем пограничных, так что их коэффициент использования оказывается ниже, чем сумма коэффициентов использования входных интерфейсов, как в нашем примере. Для того чтобы снизить вероятность перегрузки внутренних интерфейсов магистральных маршрутизаторов и выходных интерфейсов пограничных маршрутизаторов, можно также уменьшить допустимый порог нагрузки входных интерфейсов трафиком особого обслуживания, например, до 5 %.

Однако все эти меры не дают гарантии, что все интерфейсы всех маршрутизаторов сети будут работать в нужном диапазоне значений коэффициента использования, а следовательно, обеспечивать заданное качество обслуживания. Для того чтобы дать такие гарантии, необходимо «улучшить» модель DiffServ и применять методы инжиниринга трафика, то есть контролировать не классы, а потоки трафика, в данном случае агрегированные. Под *агрегированным* понимается поток, состоящий из пакетов одного класса, имеющих общую часть пути через сеть. Эта общая часть не обязательно включает полный путь, от входного интерфейса одного из пограничных маршрутизаторов до выходного интерфейса другого пограничного маршрутизатора. Достаточно, чтобы пакеты проходили хотя бы два общих интерфейса, чтобы считать их агрегированным потоком, как, например, в случае потока, проходящего через интерфейсы i11 и i22 (см. рис. 20.6).

Затем, зная путь прохождения каждого агрегированного потока через сеть, можно проверить, имеются ли достаточные ресурсы вдоль пути для каждого потока, например, не превышают ли коэффициенты использования интерфейсов заданного порога. Для этого нужно провести профилирование с учетом адреса назначения пакетов. Однако реализация такого подхода в IP-сетях сталкивается с несколькими трудностями. Во-первых, в технологии DiffServ не предусмотрен сигнальный протокол, такой как, например, RSVP в технологии IntServ. Это означает, что все проверки наличия ресурсов у маршрутизаторов для каждого агрегированного потока нужно выполнять в автономном режиме, вручную или с помощью какого-то специального программного обеспечения. Во-вторых, для проведения таких расчетов нужно знать пути потоков через сеть. Такие пути определяются таблицами маршрутизации, которые строятся протоколом маршрутизации, например RIP или OSPF (либо их комбинацией, если в сети используются несколько протоколов маршрутизации IGP-класса), или вручную. Поэтому для ручного или автоматизированного расчетов нужно знать таблицы маршрутизации всех маршрутизаторов сети и следить за их изменениями, а это непросто, учитывая, что отказы линий связи или маршрутизаторов приводят к перестройке таблиц. Нужно также учитывать, что маршрутизаторы могут применять методы балансировки нагрузки, разделяя агрегированный поток на несколько подпотоков, что также усложняет расчеты.

«Улучшенная» версия DiffServ, обеспечивающая учет адресов назначения, повышает качество услуг оператора связи, но вместе с тем усложняет саму идею метода, в основе которого лежит идея независимого обслуживания классов трафика каждым маршрутизатором сети.

Трансляция сетевых адресов

Ключевые слова: безопасность, дефицит адресов, трансляция сетевых адресов, частные адреса, традиционная технология NAT, NAT-устройство, базовая трансляция сетевых адресов, трансляция сетевых адресов и портов, маршрутные объявления, исходящие и входящие сеансы связи.

Маршрутизация в составной сети осуществляется на основе тех адресов назначения, которые помещены в заголовки пакетов. Как правило, эти адреса остаются

неизменными с момента их формирования отправителем до момента поступления на узел получателя. Однако из этого правила есть исключения. Например, в широко применяемой сегодня технологии **трансляции сетевых адресов** (Network Address Translation, NAT) предполагается продвижение пакета во внешней сети (в Интернете) на основании адресов, отличающихся от тех, которые используются для маршрутизации пакета во внутренней (корпоративной) сети.

Причины подмены адресов

Одной из наиболее популярных причин использования технологии NAT является дефицит IP-адресов. Если по каким-либо причинам предприятию, у которого имеется потребность подключения к Интернету, не удается получить у поставщика услуг необходимого количества глобальных IP-адресов, то оно может прибегнуть к технологии NAT. В этом случае для адресации внутренних узлов используются специально зарезервированные для этих целей **частные адреса**. Мы уже рассказывали о них в главе 17.

Для того чтобы узлы с частными адресами могли связываться между собой через Интернет или с узлами, имеющими глобальные адреса, необходимо использовать технологию NAT.

Технология NAT также оказывается полезной, когда предприятие из соображений безопасности желает скрыть адреса узлов своей сети, чтобы не дать возможности злоумышленникам составить представление о структуре и масштабах корпоративной сети, а также о структуре и интенсивности исходящего и входящего трафиков.

Традиционная технология NAT

Технология трансляции сетевых адресов имеет несколько разновидностей, наиболее популярная из которых — **традиционная технология NAT** – позволяет узлам из частной сети прозрачным для пользователей образом получать доступ к узлам внешних сетей. Подчеркнем, что в данном варианте NAT решается проблема организации только тех сеансов связи, которые *исходят* из частной сети. Направление сеанса в данном случае определяется положением инициатора: если обмен данными инициируется приложением, работающем на узле внутренней сети, то сеанс называется исходящим, несмотря на то, что в его рамках в сеть могут поступать данные извне[1].

Идея технологии NAT состоит в следующем. Пусть сеть предприятия образует тупиковый домен, узлам которого присвоены частные адреса (рис. 20.7). На маршрутизаторе, связывающем сеть предприятия с внешней сетью, установлено программное обеспечение NAT. Это NAT-устройство динамически отображает набор частных адресов {IP*} на набор глобальных адресов {IP}, полученных предприятием от поставщика услуг и присвоенных внешнему интерфейсу маршрутизатора предприятия.

---

[1] Традиционная технология NAT в виде исключения допускает сеансы обратного направления, заранее выполняя статическое взаимно однозначное отображение внутренних и внешних адресов для некоторого ограниченного набора узлов.

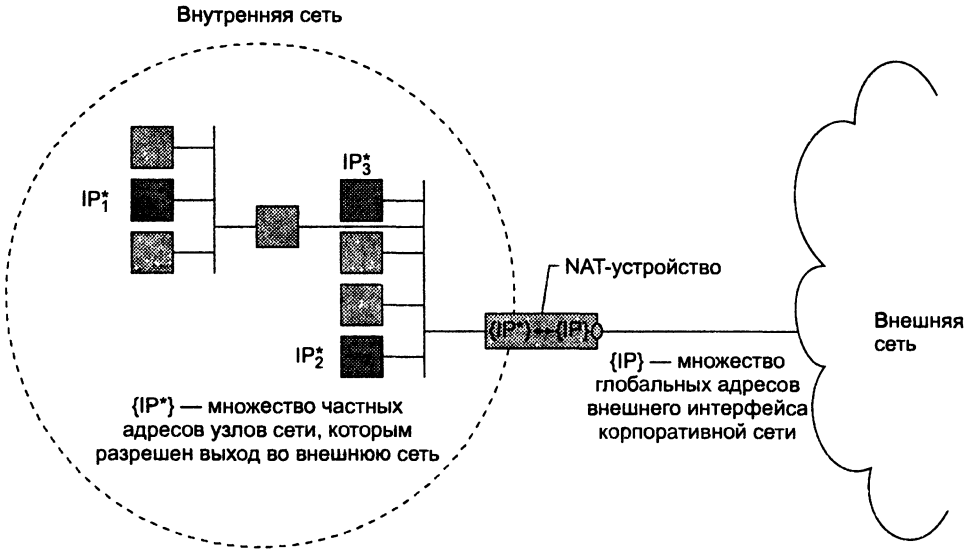


Рис. 20.7. Схема действия традиционной технологии NAT

Важным для работы NAT-устройства является правило распространения маршрутных объявлений через границы частных сетей. Объявления протоколов маршрутизации о внешних сетях «пропускаются» пограничными маршрутизаторами во внутренние сети и обрабатываются внутренними маршрутизаторами. Обратное утверждение неверно — маршрутизаторы внешних сетей не получают объявлений о внутренних сетях, объявления о них отфильтровываются при передаче на внешние интерфейсы. Поэтому внутренние маршрутизаторы «знают» маршруты ко всем внешним сетям, а внешним маршрутизаторам ничего не известно о существовании частных сетей.

Традиционная технология NAT подразделяется на технологии **базовой трансляции сетевых адресов** (Basic Network Address Translation, Basic NAT) и **трансляции сетевых адресов и портов** (Network Address Port Translation, NAPT). В технологии Basic NAT для отображения используются только IP-адреса, а в технологии NAPT — еще и так называемые транспортные идентификаторы, в качестве которых чаще всего выступают TCP- и UDP-порты.

Базовая трансляция сетевых адресов

Если количество локальных узлов, которым необходимо обеспечить выход во внешнюю сеть, меньше или равно имеющегося количества глобальных адресов, то для каждого частного адреса гарантировано однозначное отображение на глобальный адрес. В каждый момент времени количество внутренних узлов, которые получают возможность взаимодействовать с внешней сетью, ограничивается количеством адресов в глобальном наборе. Понятно, что в такой ситуации целью трансляции является не столько решение проблемы дефицита адресов, сколько обеспечение безопасности.

Частные адреса некоторых узлов могут отображаться на глобальные адреса *статически*. К таким узлам можно обращаться извне, используя закрепленные за ними глобальные адреса. Соответствие внутренних адресов внешним задается таблицей, поддерживаемой маршрутизатором или другим устройством (например, брандмауэром), на котором установлено программное обеспечение NAT.

В нескольких тупиковых доменах могут быть совпадающие частные адреса. Например, в сетях A и B на рис. 20.8 для внутренней адресации применяется один и тот же блок адресов 10.0.1.0/24.

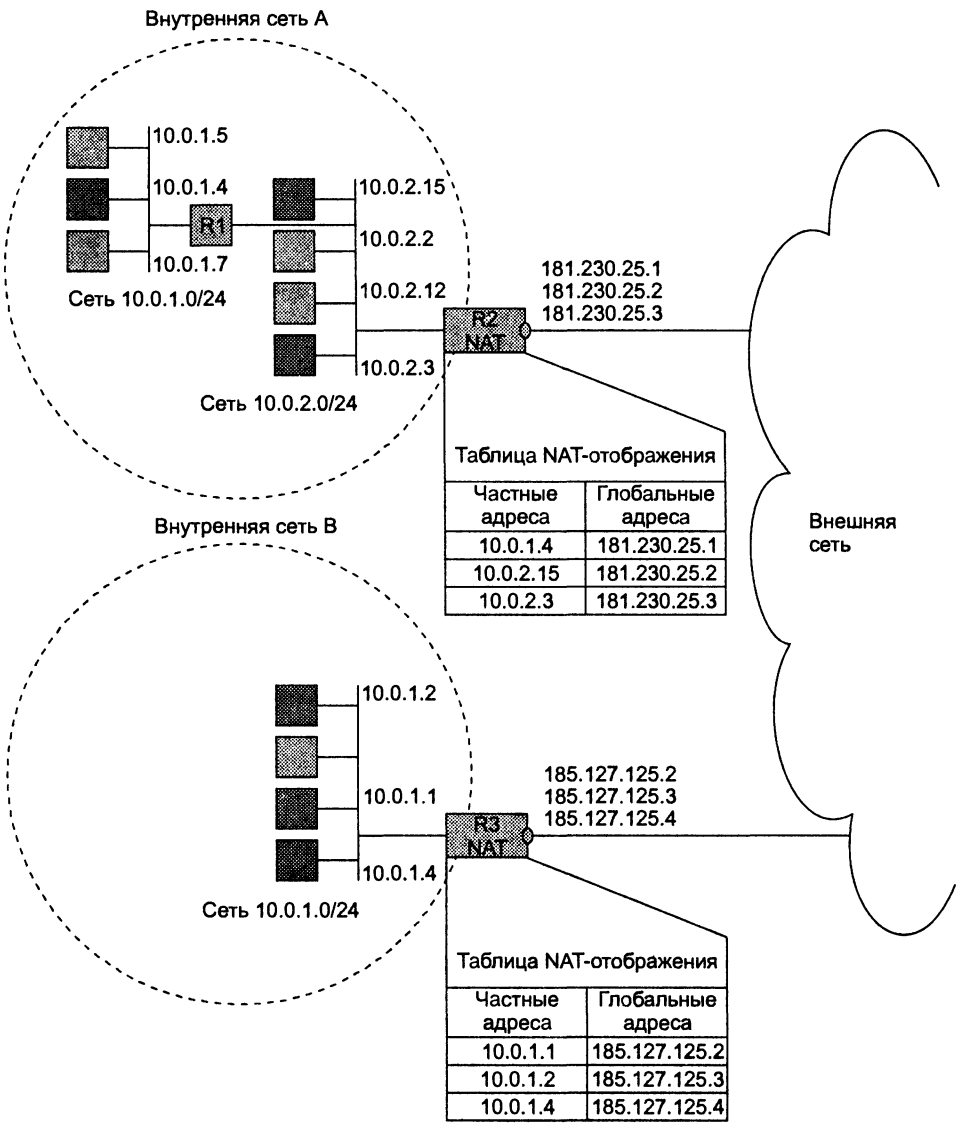


Рис. 20.8. Базовая трансляция сетевых адресов для исходящих сеансов

В то же время адреса внешних интерфейсов обеих сетей (181.230.25.1/24, 181.230.25.2/24 и 181.230.25.3/24 в сети А и 185.127.125.2/24, 185.127.125.3/24 185.127.125.4/24 в сети В) уникальны глобально, то есть никакие другие узлы в составной сети их не используют. В данном примере в каждой из сетей только три узла имеют возможность «выхода» за пределы сети своего предприятия. Статическое соответствие частных адресов этих узлов глобальным адресам задано в таблицах пограничных устройств обеих сетей.

Когда узел 10.0.1.4 сети А посылает пакет хосту 10.0.1.2 сети В, то он помещает в заголовок пакета в качестве адреса назначения глобальный адрес 185.127.125.3/24. Узел-источник направляет пакет своему маршрутизатору R1 по умолчанию, которому известен маршрут к сети 185.127.125.0/24. Маршрутизатор передает пакет на пограничный маршрутизатор R2, которому также известен маршрут к сети 185.127.125.0/24. Перед отправкой пакета протокол NAT, работающий на данном пограничном маршрутизаторе, используя таблицу отображения, заменяет в поле адреса источника частный адрес 10.0.1.4 соответствующим ему глобальным адресом 181.230.25.1/24. Когда пакет после путешествия по внешней сети поступает на внешний интерфейс NAT-устройства сети В, глобальный адрес назначения 185.127.125.3/24 преобразуется в частный адрес 10.0.1.2. Пакеты, передаваемые в обратном направлении, проходят аналогичную процедуру трансляции адресов.

Заметим, что в описанной операции не требуется участия узлов отправителя и получателя, то есть она прозрачна для пользователей.

Трансляция сетевых адресов и портов

Пусть некоторая организация имеет частную IP-сеть и глобальную связь с поставщиком услуг Интернета. Внешнему интерфейсу пограничного маршрутизатора R2 назначен глобальный адрес, а остальным узлам сети организации назначены частные адреса. NAPT позволяет *всем* узлам внутренней сети одновременно взаимодействовать с внешними сетями, используя единственный зарегистрированный IP-адрес. Возникает законный вопрос, каким образом внешние пакеты, поступающие *в ответ* на запросы из частной сети, находят узел-отправитель, ведь в поле адреса источника всех пакетов, отправляющихся во внешнюю сеть, помещается один и тот же адрес — адрес внешнего интерфейса пограничного маршрутизатора?

Для однозначной идентификации узла отправителя привлекается дополнительная информация. Если в IP-пакете находятся данные протокола UDP или TCP, то в качестве такой информации выступает номер UDP- или TCP-порта соответственно. Но и это не вносит полной ясности, поскольку из внутренней сети может исходить несколько запросов с совпадающими номерами портов отправителя, а значит, опять возникает вопрос об однозначности отображения единственного глобального адреса на набор внутренних адресов. Решение состоит в том, что при прохождении пакета из внутренней во внешнюю сеть каждой паре {внутренний частный адрес; номер TCP- или UDP-порта отправителя} ставится в соответствие пара {глобальный IP-адрес внешнего интерфейса; назначенный номер TCP- или UDP-порта}. Назначенный номер порта выбирается произволь-

но, однако должно быть выполнено условие его уникальности в пределах всех узлов, получающих выход во внешнюю сеть. Соответствие фиксируется в таблице.

Эта модель при наличии единственного зарегистрированного IP-адреса, полученного от поставщика услуг, удовлетворяет требованиям по доступу к внешним сетям большинства сетей средних размеров.

На рис. 20.9 приведен пример, когда в тупиковой сети А используются внутренние адреса из блока 10.0.0.0. Внешнему интерфейсу маршрутизатора этой сети поставщиком услуг назначен адрес 181.230.25.1.

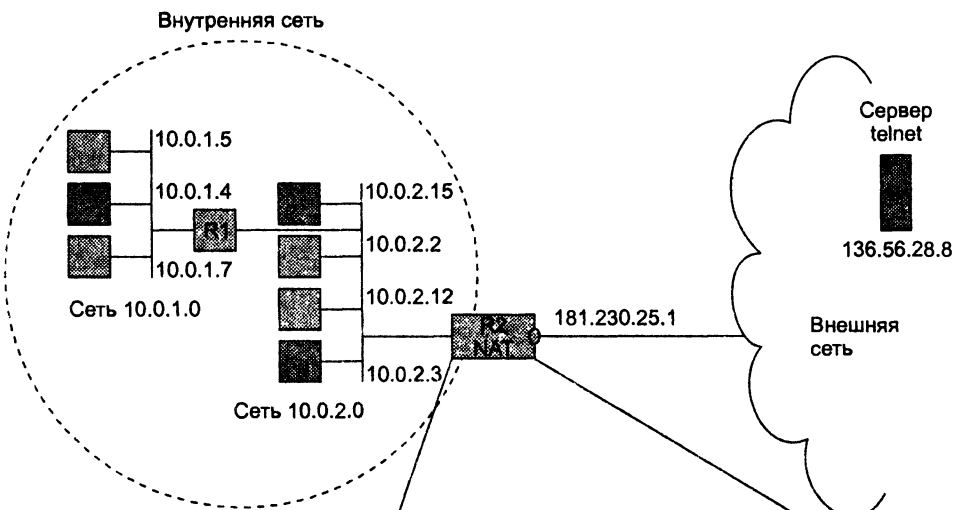


Таблица NAT-отображения

Частный адрес	Порт	Глобальный адрес	Назначенный порт
10.0.1.4	1245	181.230.25.1	3451
10.0.2.15	1245	181.230.25.1	3452
10.0.2.3	1045	181.230.25.1	3455

Рис. 20.9. Трансляция сетевых адресов и портов для исходящих TCP- и UDP-сеансов

Когда хост 10.0.1.4 внутренней сети посылает во внешнюю сеть пакет серверу telnet, то он в качестве адреса назначения использует его глобальный адрес 136.56.28.8. Пакет поступает маршрутизатору R1, который знает, что путь к сети 136.56.0.0/16 идет через пограничный маршрутизатор R2. Модуль NAPT маршрутизатора R2 транслирует адрес 10.0.1.4 и порт TCP 1245 источника в глобально уникальный адрес 181.230.25.1 и уникально назначенный TCP-порт, в приведенном примере — 3451. В таком виде пакет отправляется во внешнюю сеть и достигает сервера telnet. Когда получатель генерирует ответное сообщение, то он в качестве адреса назначения указывает единственный зарегистрированный глобальный адрес внутренней сети, являющийся адресом внешнего интерфейса NAPT-устройства. В качестве номера порта получателя сервер помещает назначенный

номер TCP-порта, взятый из поля порта отправителя пришедшего пакета. При поступлении ответного пакета на NAPT-устройство внутренней сети именно по номеру порта в таблице трансляции выбирается нужная строка. По ней определяется внутренний IP-адрес соответствующего узла и действительный номер порта. Эта процедура трансляции полностью прозрачна для конечных узлов.

ВНИМАНИЕ

Заметьте, что в таблице имеется еще одна запись с номером порта 1245, такая ситуация вполне возможна: операционные системы на разных компьютерах независимо присваивают номера портов клиентским программам. Именно для разрешения такой неоднозначности и привлекаются уникальные назначенные номера портов.

В технологии NAPT разрешаются только исходящие из частной сети TCP- и UDP-сеансы. Однако возникают ситуации, когда нужно обеспечить доступ к некоторому узлу внутренней сети извне. В простейшем случае, когда служба зарегистрирована, то есть ей присвоен хорошо известный номер порта (например, WWW или DNS), и, кроме того, эта служба представлена во внутренней сети в единственном экземпляре, задача решается достаточно просто. Служба и узел, на котором она работает, однозначно определяются хорошо известным зарегистрированным номером порта службы.

Завершая рассмотрение технологии NAT, заметим, что помимо традиционной технологии NAT существуют и другие ее варианты, например двойной трансляции сетевых адресов, при которой модифицируются оба адреса — и источника, и приемника (в отличие от традиционной технологии NAT, когда модифицируется только один адрес). Двойная трансляция сетевых адресов необходима, когда частные и внешние адресные пространства имеют коллизии. Наиболее часто это происходит, когда внутренний домен имеет некорректно назначенные публичные адреса, которые принадлежат другой организации. Подобная ситуация может возникнуть из-за того, что сеть организации была изначально изолированной и адреса назначались произвольно, причем из глобального пространства. Или же такая коллизия может быть следствием смены поставщика услуг, причем организация хотела бы сохранить старые адреса для узлов внутренней сети.

Маршрутизаторы

Ключевые слова: маршрутизатор, таблица маршрутизации, интерфейсы маршрутизатора, получение доступа к среде, формирование битовых сигналов, прием кадра, подсчет контрольной суммы, ARP, магистральные, региональные, корпоративные, пограничные маршрутизаторы, маршрутизаторы операторов связи, программные маршрутизаторы, маршрутизаторы локальных сетей и удаленных офисов, мультипроцессорная организация, операционная система маршрутизатора.

Функции маршрутизаторов

Основная функция маршрутизатора — чтение заголовков пакетов сетевых протоколов, принимаемых и буферизуемых по каждому порту (например, IPX, IP, AppleTalk или DECnet), и принятие решения о дальнейшем маршруте следования пакета по его сетевому адресу, включающему, как правило, номера сети и узла.

Функции маршрутизатора могут быть разбиты на три группы в соответствии с уровнями модели OSI (рис. 20.10).

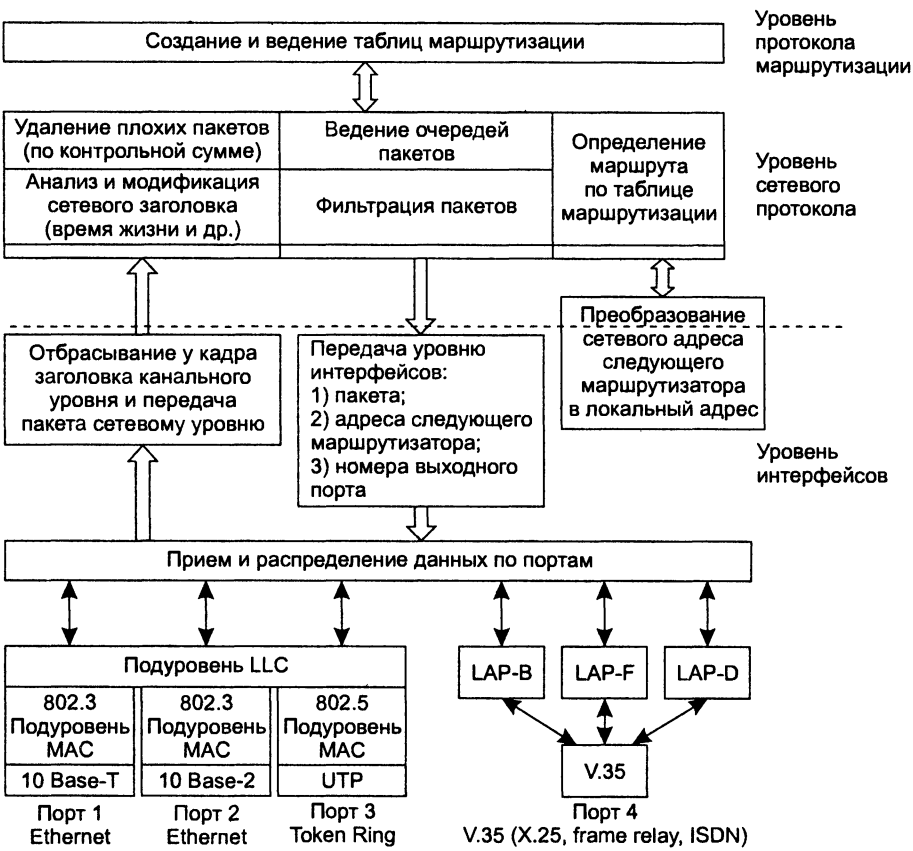


Рис. 20.10. Функциональная модель маршрутизатора

Уровень интерфейсов

На нижнем уровне маршрутизатор, как и любое устройство, подключенное к сети, обеспечивает физический интерфейс со средой передачи, включая согласование уровней электрических сигналов, линейное и логическое кодирование, оснащение определенным типом разъема. В разных моделях маршрутизаторов часто предусматриваются различные наборы физических интерфейсов, представляющих собой комбинацию портов для подсоединения локальных и глобальных сетей.

С каждым интерфейсом для подключения локальной сети неразрывно связан определенный протокол канального уровня, например семейства Ethernet, Token Ring, FDDI. Интерфейсы для присоединения к глобальным сетям чаще всего определяют только некоторый стандарт физического уровня, над которым в маршрутизаторе могут работать различные протоколы канального уровня. Например, глобальный порт может поддерживать интерфейс V.35, над которым могут работать различные протоколы канального уровня: PPP (передает трафик протокола IP и других сетевых протоколов), LAP-B (используемый в сетях X.25), LAP-F (используемый в сетях frame relay), LAP-D (используемый в сетях ISDN), ATM. Разница между интерфейсами локальных и глобальных сетей объясняется тем, что технологии локальных сетей определяют стандарты как физического, так и канального уровней, которые могут применяться только вместе.

Интерфейсы маршрутизатора выполняют полный набор функций физического и канального уровней по передаче кадра, включая получение доступа к среде (если это необходимо), формирование битовых сигналов, прием кадра, подсчет его контрольной суммы и передачу поля данных кадра верхнему уровню при корректном значении контрольной суммы.

ПРИМЕЧАНИЕ

Как и любой конечный узел, каждый порт маршрутизатора имеет собственный аппаратный адрес (в локальных сетях MAC-адрес), по которому ему другие узлы направляют кадры, требующие маршрутизации.

Перечень физических интерфейсов, которые поддерживает та или иная модель маршрутизатора, является его важнейшей потребительской характеристикой. Маршрутизатор должен поддерживать все протоколы канального и физического уровней, используемые в каждой из сетей, к которым он будет непосредственно присоединен. На рис. 20.10 показана функциональная модель маршрутизатора с четырьмя портами, реализующими следующие физические интерфейсы: 10Base-T и 10Base-2 для двух портов Ethernet, UTP для Token Ring, а также интерфейс V.35, над которым могут работать протоколы LAP-B, LAP-D или LAP-F, обеспечивая подключение к сетям X.25, ISDN и frame relay.

Кадры, которые поступают на порты маршрутизатора, после обработки соответствующими протоколами физического и канального уровней освобождаются от заголовков канального уровня. Извлеченные из поля данных кадра пакеты передаются модулю сетевого протокола.

Уровень сетевого протокола

Сетевой протокол, в свою очередь, извлекает из пакета заголовок сетевого уровня, *анализирует и корректирует содержимое его заголовка.* Прежде всего проверяется контрольная сумма, и если пакет пришел поврежденным, то он отбрасывается. Кроме того, выполняется проверка на превышение времени жизни пакета (время, которое провел пакет в сети). Если превышение имело место, то пакет также отбрасывается. На этом этапе вносятся корректировки в содержимое некоторых полей, например наращивается время жизни пакета, пересчитывается контрольная сумма.

На сетевом уровне выполняется одна из важнейших функций маршрутизатора — *фильтрация трафика*. Пакет сетевого уровня, находящийся в поле данных кадра, для мостов/коммутаторов представляется неструктурированной двоичной последовательностью. Маршрутизаторы же, программное обеспечение которых содержит модуль сетевого протокола, способны производить анализ *отдельных полей пакета*. Они оснащаются развитыми средствами пользовательского интерфейса, которые позволяют администратору без особых усилий задавать сложные правила фильтрации. Маршрутизаторы, как правило, позволяют также анализировать структуру сообщений транспортного уровня, поэтому фильтры могут не пропускать в сеть сообщения определенных прикладных служб, например службы telnet, анализируя поле типа протокола в транспортном сообщении.

Однако основной функцией сетевого уровня маршрутизатора является *определение маршрута пакета*. По номеру сети, извлеченному из заголовка пакета, модуль сетевого протокола находит в таблице маршрутизации строку, содержащую сетевой адрес следующего маршрутизатора, и номер порта, на который нужно передать данный пакет, чтобы он двигался в правильном направлении.

Перед тем как передать сетевой адрес следующего маршрутизатора на канальный уровень, необходимо преобразовать его в локальный адрес той технологии, которая используется в сети, содержащей следующий маршрутизатор. Для этого сетевой протокол обращается к *протоколу разрешения адресов*.

С сетевого уровня пакет, локальный адрес следующего маршрутизатора и номер порта маршрутизатора передаются вниз, канальному уровню. На основании указанного номера порта осуществляется коммутация с одним из интерфейсов маршрутизатора, средствами которого выполняется упаковка пакета в кадр соответствующего формата. В поле адреса назначения заголовка кадра помещается локальный адрес следующего маршрутизатора. Готовый кадр отправляется в сеть.

Уровень протокола маршрутизации

Сетевые протоколы активно используют в своей работе таблицу маршрутизации, но ни ее построением, ни поддержанием ее содержимого не занимаются. Эти функции выполняют протоколы маршрутизации. На основании этих протоколов маршрутизаторы обмениваются информацией о топологии сети, а затем анализируют полученные сведения, определяя наилучшие по тем или иным критериям маршруты. Результаты анализа и составляют содержимое таблиц маршрутизации.

Помимо перечисленных выше функций на маршрутизаторы могут быть возложены и другие обязанности, например операции, связанные с фрагментацией.

Классификация маршрутизаторов по областям применения

По областям применения маршрутизаторы делятся на несколько классов (рис. 20.11).

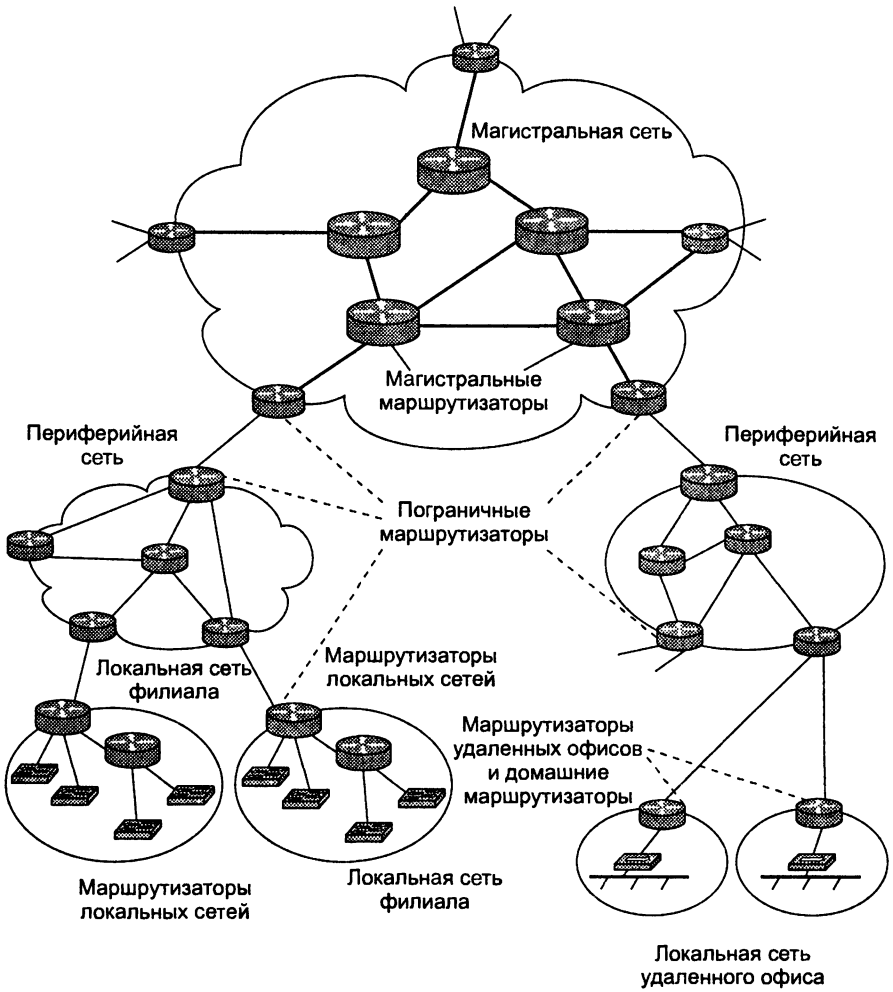


Рис. 20.11. Классы маршрутизаторов

Магистральные маршрутизаторы предназначены для построения магистральной сети оператора связи или крупной корпорации. Маршрутизаторы магистрали оперируют с агрегированными информационными потоками, переносящими данные большого количества пользовательских соединений.

Для решения этой задачи магистральные маршрутизаторы оснащаются высокоскоростными интерфейсами, такими как ATM 155/622 Мбит/с, Gigabit Ethernet и 10G Ethernet, а также интерфейсами SONET/SDH со скоростями от 155 Мбит/с до 10 Гбит/с. Для создания отказоустойчивой топологии магистральной сети магистральные маршрутизаторы должны поддерживать несколько таких интерфейсов.

Очевидно, для того чтобы не создавать «узких мест» в магистральной сети, магистральный маршрутизатор должен обладать очень высокой производительностью.

Например, если маршрутизатор оснащен 8 интерфейсами 10 Гбит/с (Ethernet или SDH), то его общая производительность должна составлять 80 Гбит/с. Для достижения такой производительности магистральные маршрутизаторы обладают распределенной внутренней архитектурой, подобной архитектуре коммутаторов локальных сетей, которая была рассмотрена в главе 15. Каждый порт или группа портов оснащается *собственным процессором*, который самостоятельно выполняет продвижение IP-пакетов на основании локальной копии таблицы маршрутизации. Для передачи пакетов между портами используется *коммутирующий блок* на основе разделяемой памяти, общей шины или коммутатора каналов. Общие задачи, включая построение таблицы маршрутизации, хранение конфигурационных параметров, удаленное управление маршрутизатором и т. п., решает *центральный блок управления*.

Понятно, что функции продвижения IP-пакетов существенно сложнее, чем продвижения кадров Ethernet и других технологий локальных сетей. Поэтому процессоры портов обычно не нагружают дополнительными функциями, такими как фильтрация трафика или трансляция адресов. Даже обеспечение параметров QoS не всегда реализуется таким процессором в полном объеме — обычно дело ограничивается поддержанием очередей, а до профилирования трафика не доходит. Это связано с тем, что магистральный маршрутизатор работает внутри сети и не взаимодействует с внешним миром, а значит, и не выполняет пограничные функции, требующие фильтрации и профилирования. Другими словами, основная задача магистрального маршрутизатора — передача пакетов между своими интерфейсами с как можно большей скоростью.

Большое количество интерфейсов, характерное для магистрального маршрутизатора, позволяет строить избыточные топологии, приближающиеся к полносвязной схеме, и тем самым обеспечивать отказоустойчивость сети. Однако и сам магистральный маршрутизатор должен обладать высокой надежностью. Надежность и отказоустойчивость маршрутизатора достигается за счет избыточных модулей, таких как центральные процессоры, процессоры портов, источники питания.

Пограничные маршрутизаторы, называемые также **маршрутизаторами доступа**, соединяют магистральную сеть с периферийными сетями. Эти маршрутизаторы образуют особый слой, который выполняет функции приема трафика от внешних по отношению к магистрали сетей.

Периферийная сеть часто находится под автономным административным управлением. Это может быть сеть клиента оператора связи, непосредственно присоединенная к его магистрали, или же сеть регионального отделения крупной корпорации, обладающей собственной магистралью.

В любом случае трафик поступает на интерфейсы пограничного маршрутизатора от сети, которую администратор магистрали не может контролировать, — трафик нужно фильтровать и профилировать. Поэтому к пограничному маршрутизатору предъявляются другие требования, нежели к магистральному. На первый план выступают его способности к *максимальной гибкости при фильтрации* и *профилировании трафика*. Кроме того, очень важно, чтобы производительность

пограничного маршрутизатора не снижалась при выполнении этих дополнительных функций. Интерфейсы пограничного маршрутизатора менее скоростные, чем магистрального, но более разнообразные, так как ему приходится присоединять к магистрали сети различных технологий.

Деление маршрутизаторов на магистральные и пограничные не является строгим и четким. Такое деление просто отражает ту область, где применение маршрутизатора предпочтительно, где в наибольшей степени проявляются его преимущества. В то же время любой маршрутизатор можно применять не только в его профильной области. Так, магистральный маршрутизатор, оснащенный низкоскоростными портами, может одновременно играть роль пограничного. А маршрутизатор, хорошо исполняющий роль пограничного для крупной сети, может быть магистральным маршрутизатором для сети меньшего масштаба, где его интерфейсы вполне справятся с нагрузкой на магистраль.

Деление маршрутизаторов на магистральные и пограничные отражает только один аспект их применения, а именно их положение относительно собственной и внешних сетей. Понятно, что существуют и другие аспекты. Так, маршрутизаторы можно разделить на **маршрутизаторы операторов связи** и **корпоративные маршрутизаторы**.

Основным отличием корпоративных маршрутизаторов является их *высокая надежность*, а также *поддержка полного набора функций*, необходимых для коммерческой работы в Интернете, начиная от протокола BGP и кончая системами регистрации пользовательских потоков данных, что необходимо для биллинговых схем. Требования к надежности объясняются высокой стоимостью простоя маршрутизатора при оказании коммерческих услуг. Требования к надежности услуг передачи данных постоянно растут, пользователи Интернета и виртуальных частных сетей хотят, чтобы эти услуги были такими же надежными, как услуги телефонной сети. Поэтому, когда мы говорим о том, что готовность некоторых моделей маршрутизаторов достигла рубежа 0,999 и стремится к показателям телефонного оборудования в 0,99999, то в первую очередь это относится к маршрутизаторам операторов связи, как магистральным, так и пограничным. Корпоративные маршрутизаторы предназначены для применения в пределах корпоративной сети, поэтому требования к надежности здесь ниже, а функциональность для работы в Интернете в качестве самостоятельной автономной системы не требуется.

Конечно, характеристики маршрутизаторов операторов связи и корпоративных маршрутизаторов в значительной степени зависят от масштаба и специфики оператора связи или корпорации. Для международного оператора связи, в иерархии ISP относящегося к категории Tiar 1, сегодня требуются магистральные маршрутизаторы с интерфейсами 10 Гбит/с, которые в недалеком будущем будут заменены маршрутизаторами с портами DWDM, работающими с 40 волнами и обеспечивающими общую скорость порта 400 Гбит/с. Пограничные маршрутизаторы такого оператора также будут относиться к лучшим маршрутизаторам этого класса по производительности, работая с портами доступа со скоростями от 622 Мбит/с до 2,5 Гбит/с.

Менее крупным операторам связи, то есть региональным и локальным, такие высокопроизводительные маршрутизаторы не требуются, так как объемы переда-

ваемого ими трафика гораздо меньше. Поэтому магистральный маршрутизатор такого оператора может ограничиться поддержкой интерфейсов 2–155 Мбит/с, а пограничный должен, кроме того, обеспечивать коммутируемый доступ абонентов через телефонные сети. В небольших сетях магистральных маршрутизаторов может не быть вообще, такая сеть будет состоять из одних (или даже одного) пограничных маршрутизаторов.

Аналогичная картина наблюдается и в корпоративных сетях, где также применяются маршрутизаторы различной производительности и надежности. Например, крупные корпорации могут применять магистральные и пограничные маршрутизаторы, близкие по характеристикам к маршрутизаторам операторов связи категории Tiar 1. Однако более обычной является ситуация, когда в корпоративных сетях применяется оборудование с характеристиками на один уровень ниже. Это значит, что крупные многонациональные корпорации применяют оборудование, которое обычно используется региональными операторами и т. д., по нисходящей.

Маршрутизаторы региональных отделений соединяют региональные отделения между собой и с магистральной сетью. Сеть регионального отделения, так же как и магистральная сеть, может состоять из нескольких локальных сетей. Такой маршрутизатор обычно представляет собой некоторую упрощенную версию магистрального корпоративного маршрутизатора.

Если он выполнен на основе шасси, то количество слотов его шасси меньше (4–5). Возможен также конструктив с фиксированным количеством портов. Поддерживаемые интерфейсы локальных и глобальных сетей менее скоростные. Это наиболее обширный класс выпускаемых маршрутизаторов, характеристики которых могут приближаться к характеристикам магистральных маршрутизаторов, а могут и опускаться до характеристик маршрутизаторов удаленных офисов.

Маршрутизаторы удаленных офисов соединяют, как правило, единственную локальную сеть удаленного офиса с магистральной сетью или сетью регионального отделения по глобальной связи.

Как правило, интерфейс локальной сети — это Ethernet 10/100 Мбит/с, а интерфейс глобальной сети — выделенная линия со скоростью 64 Кбит/с, 1,544 или 2 Мбит/с. Маршрутизатор удаленного офиса может поддерживать работу по коммутируемой телефонной линии в качестве резервной связи для выделенного канала. Существует очень большое количество типов маршрутизаторов удаленных офисов. Это объясняется как массовостью потенциальных потребителей, так и специализацией такого типа устройств, проявляющейся в поддержке какого-либо конкретного типа глобальной связи. Например, существуют маршрутизаторы, работающие только в ISDN-сети, существуют модели только для аналоговых выделенных линий и т. п.

Чем меньше требований предъявляется к производительности маршрутизатора, тем более вероятно, что он выполнен по классической схеме первых маршрутизаторов (и мостов локальных сетей), то есть схемы на основе единственного центрального процессора и лишенных процессоров портов. Такая схема гораздо дешевле, но ее производительность полностью определяется производительностью процессора и не масштабируется с ростом числа портов.

Программный маршрутизатор, являясь одной из популярных реализаций такой схемы, представляет собой программный модуль универсальной операционной системы семейства Unix или Windows.

И только появление в глобальных сетях высокоскоростных технологий, таких как ATM, SONET/SDH, DWDM, привело к резкому повышению требований к производительности маршрутизаторов, в результате чего представители наиболее совершенного класса маршрутизаторов повсеместно перешли на *многопроцессорные схемы с коммутирующим блоком*, успешно опробованные на коммутаторах локальных сетей.

Маршрутизаторы локальных сетей предназначены для разделения крупных локальных сетей на подсети. Это особый класс маршрутизаторов, которые, как правило, не имеют интерфейсов глобальных сетей.

Многие маршрутизаторы этого типа ведут свое происхождение от коммутаторов локальных сетей, что и дало им второе название — **коммутаторы 3-го уровня**. Коммутаторы 3-го уровня выполняют все функции маршрутизаторов, но, кроме того, могут работать и как обычные коммутаторы локальных сетей, то есть коммутаторы 2-го уровня. Режим работы (маршрутизатор или коммутатор) зависит от параметров конфигурации. Возможен также комбинированный режим работы, когда несколько портов коммутатора 3-го уровня имеют один и тот же IP-адрес сети (рис. 20.12). В этом случае передача пакетов между группой портов, принадлежащих одной сети, выполняется в режиме коммутации, на канальном уровне, то есть на основе MAC-адресов. Если же порты принадлежат разным IP-сетям, то тогда коммутатор выполняет маршрутизацию между сетями. Выбор режима передачи пакета определяется конфигурированием IP-адресов портов и, соответственно, компьютеров.

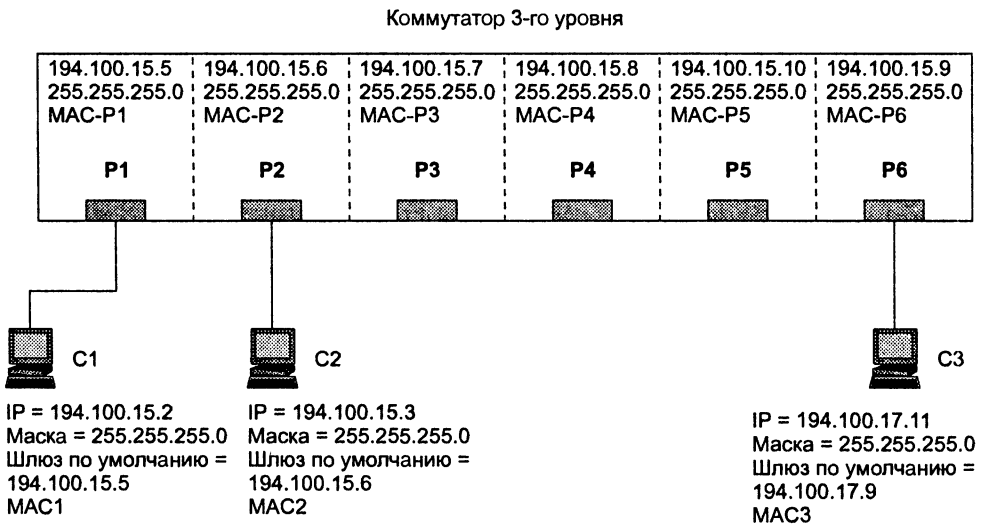


Рис. 20.12. Комбинированный режим работы коммутатора 3-го уровня

Пример

Например, если два компьютера (C1 и C2 на рис. 20.12) имеют адреса, принадлежащие одной сети, то при обмене информацией они не будут передавать пакеты маршрутизатору по умолчанию, а будут использовать протокол ARP для того, чтобы узнать MAC-адрес компьютера назначения. Пусть компьютеру C1 требуется передать пакет компьютеру C2. Коммутатор 3-го уровня передает кадр ARP-запроса компьютера C1 с широковещательным MAC-адресом всем портам, принадлежащим одной IP-сети, то есть портам P1, P2, P3 и P4. Компьютер C2 распознает свой IP-адрес (194.100.15.3) в этом запросе и отвечает направленным кадром с MAC-адресом назначения компьютера C1 (MAC1), помещая в ответ собственный MAC-адрес (MAC2). После этого компьютер C1 направляет IP-пакет компьютеру C2, помещая его в кадр с адресом назначения MAC2. Коммутатор 3-го уровня передает этот кадр с порта P1 на порт P2 в соответствии с алгоритмом моста на основе таблицы продвижения 2-го уровня. Аналогичным образом будет работать коммутатор 3-го уровня. В случае когда компьютеры принадлежат разным IP-сетям, поведение компьютера-отправителя диктует коммутатору 3-го уровня способ продвижения пакета. Если, например, компьютер C1 отправляет пакет компьютеру C3, находящемуся в другой сети, то он обязан передать пакет маршрутизатору по умолчанию, а не пытаться с помощью ARP узнать MAC-адрес компьютера назначения. Поэтому компьютер C1 делает ARP-запрос о MAC-адресе известного ему маршрутизатора по умолчанию, которым для него является порт P1 с IP-адресом IP-R1. После получения MAC-адреса порта P1 (MAC-P1) компьютер C1 посылает ему IP-пакет для компьютера C3 (то есть по IP-адресу назначения 194.100.17.11), оформив его как кадр Ethernet с адресом назначения MAC-P1. Получив кадр с собственным MAC-адресом, коммутатор 3-го уровня обрабатывает его по схеме маршрутизации, а не коммутации.

Коммутаторы 3-го уровня поддерживают технику VLAN, являясь основным типом устройств для соединения отдельных виртуальных сетей в составную IP-сеть. Обычно каждой виртуальной сети присваивается номер IP-сети, так что передача внутри сетей идет на основе MAC-адресов, а между сетями — на основе IP-адресов. В представленном на рис. 20.12 примере сети порты P1–P4 могут принадлежать одной виртуальной сети, а порты P5, P6 — другой.

Выводы

IP-маршрутизаторы позволяют фильтровать пользовательский трафик на основе различных признаков, включающих адреса источника и назначения, тип протокола, который переносят IP-пакеты, номера UPD- и TCP-портов и некоторые другие. Это свойство маршрутизаторов широко используется для защиты сетей от атак злоумышленников и ограничения доступа легальных пользователей.

Фильтрация маршрутных объявлений обеспечивает управление связностью сетей в целом, предотвращая появление записей об определенных сетях в таблицах маршрутизации.

IP-маршрутизаторы уже долгое время поддерживают многие механизмы QoS: приоритетные и взвешенные очереди, профилирование трафика, обратную связь для TCP-трафика. Однако только в середине 90-х годов, когда Интернет начал переносить чувствительный к задержкам трафик, начались работы по созданию системы стандартов QoS для IP-сетей.

Сегодня существует две системы стандартов QoS для IP-сетей — IntServ и DiffServ. Первая обеспечивает гарантированное качество обслуживания микропотоков, используя сигнальный

протокол RSVP для резервирования ресурсов маршрутизаторов. Недостатком такого подхода является большая нагрузка на магистральные маршрутизаторы, которые должны хранить информацию состояния тысяч пользовательских потоков.

В технологии DiffServ используется агрегированный подход, когда качество обслуживания обеспечивается для небольшого количества классов трафика. Это существенно снижает нагрузку на маршрутизаторы. Кроме того, технология DiffServ основана на модели независимого поведения маршрутизаторов (PHB), когда каждый маршрутизатор самостоятельно решает, какие ресурсы он должен выделить каждому классу, что также упрощает работу маршрутизаторов и дает возможность реализовать дифференцированное обслуживание в пределах сети поставщика услуг. Однако упрощенный подход технологии DiffServ ведет к снижению уровня гарантий параметров QoS, то есть к повышению вероятности возникновения в сети ситуаций, когда параметры QoS будут выходит за требуемые для клиента границы.

Типичный маршрутизатор представляет собой программируемое вычислительное устройство, которое работает под управлением специализированной операционной системы, оптимизированной для выполнения операций построения таблиц маршрутизации и продвижения пакетов на их основе.

Маршрутизатор часто строится по мультипроцессорной схеме, причем используется симметричное мультипроцессирование, асимметричное мультипроцессирование и их сочетание. Наиболее рутинные операции обработки пакетов выполняются программно специализированными процессорами или аппаратно большими интегральными схемами. Более высокоуровневые действия выполняют программно универсальные процессоры.

Маршрутизаторы можно классифицировать различными способами. Их можно разделить на магистральные и пограничные маршрутизаторы (по положению относительно границ сети), на маршрутизаторы операторов связи и корпоративные маршрутизаторы (в зависимости от типа предприятия, владеющего сетью). Маршрутизаторы, работающие в корпоративной сети, принято также делить на собственно корпоративные маршрутизаторы (работающие в центральной сети предприятия), маршрутизаторы региональных подразделений и маршрутизаторы удаленных офисов. Существует также специальный класс маршрутизаторов локальных сетей, которые не поддерживают WAN-интерфейсы (их обычно называют коммутаторами 3-го уровня).

Технология трансляции сетевых адресов (NAT) позволяет предприятию решить проблему дефицита IP-адресов, а также повысить безопасность сети путем сокрытия адресов узлов своей сети за счет использования во внутренней сети частных адресов, которые при выходе пакета во внешнюю сеть транслируются в глобальные IP-адреса.

Традиционная технология NAT подразделяется на базовую технологию NAT (Basic NAT), использующую для отображения только IP-адреса, и технологию NAPT. В последней для отображения привлекаются еще и так называемые транспортные идентификаторы, которыми чаще всего являются номера TCP- и UDP-портов.

Вопросы и задания

1. Какие параметры пакета могут использоваться при фильтрации трафика маршрутизатором? Варианты ответов:
   - IP-адрес источника;
   - протокол, переносимый в IP-пакете;
   - номер TCP- или UDP-порта;
   - IP-адрес источника предыдущего пакета.

2. Чем результат фильтрации объявлений маршрутизации отличается от результата фильтрации пользовательского трафика?
3. Какую смысловую нагрузку несет термин «интегрированные» в названии технологии IntServ?
4. За счет какого параметра можно ограничить пульсацию входного потока пакетов, профилируемого по алгоритму ведра маркеров?
5. Почему в методе случайного раннего обнаружения (RED) вероятность отбрасывания пакетов зависит от усредненной длины очереди, а не от текущей?
6. Поясните основные этапы резервирования ресурсов маршрутизаторов по протоколу RSVP.
7. В чем состоит принципиальное ограничение технологии IntServ? Варианты ответов:
   - ее нельзя применять для групповой адресации;
   - маршрутизатор должен хранить информацию о состоянии каждого потока;
   - конечные узлы должны периодически повторять резервирование.
8. Почему в технологии DiffServ не используется сигнальный протокол?
9. В чем отличие между EF- и AF-сервисом?
10. Благодаря каким своим особенностям технология DiffServ нашла поддержку среди операторов связи? Варианты ответов:
    - она может быть реализована в пределах сети оператора независимо от сетей других операторов;
    - маршрутизаторы работают с классами трафика, что не создает большой дополнительной нагрузки на маршрутизаторы;
    - она предусматривает автоматизацию расчетов параметров QoS.
11. В чем назначение технологии NAT? Варианты ответов:
    - отражение DOS-атак;
    - решение проблемы дефицита адресов в протоколе IPv4;
    - защита внутреннего адресного пространства сети предприятия.
12. Какие дополнительные признаки пакетов используются в технологии NAT для отображения множества внутренних адресов на один глобальный адрес?
13. Заполните столбец «Назначенный порт» в табл. 20.2.

Таблица 20.2. Пример NAT-таблицы

Частный адрес	Порт отправителя	Глобальный адрес	Назначенный порт
10.0.25.1	1035	193.55.13.79	
10.0.25.2	1035	193.55.13.79	
10.0.25.3	1035	193.55.13.79	
10.0.25.2	1047	193.55.13.79	
10.0.25.1	1047	193.55.13.79	

14. Перечислите основные варианты архитектуры маршрутизаторов.
15. По каким критериям обычно классифицируют маршрутизаторы?
16. В чем состоят особенности коммутаторов 3-го уровня?
17. Составьте список или списки доступа для маршрутизатора Cisco, соединяющего предприятие с Интернетом (рис. 20.13). Список доступа должен обеспечивать:
    - обмен пользователей сети 194.100.12.0/24 (кроме пользователя 194.100.12.25) только с узлами сетей 132.22.0.0/16 и 201.17.200.0/24 (обмен информацией с Интернетом для этих пользователей запрещен);
    - пользователю 194.100.12.25 разрешен любой обмен без каких-либо ограничений;
    - доступ к серверам сети 201.17.200.0/24 из Интернета возможен только по протоколам FTP и HTTP, а доступ к этим серверам по протоколу ICMP должен быть запрещен.

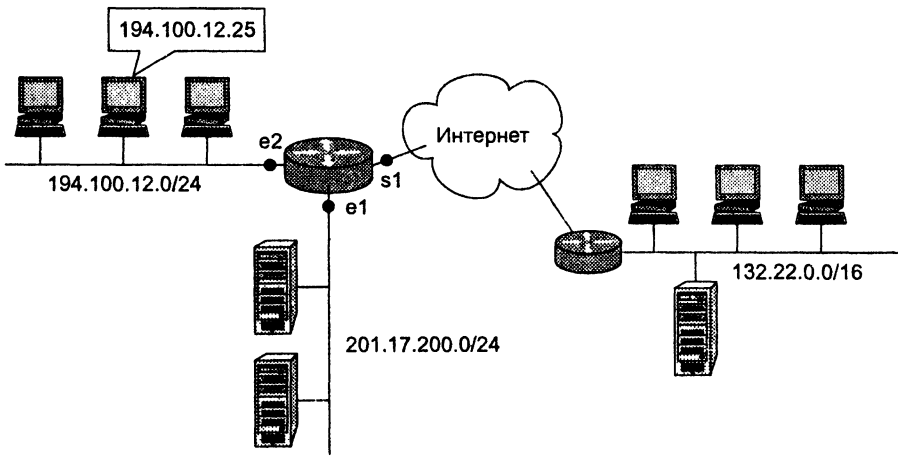


Рис. 20.13. Фильтрация трафика с помощью маршрутизатора

18. При изучении технологии NAT мы упростили реальную картину. В частности, мы не затрагивали проблемы, которые могут возникнуть при поступлении во внутреннюю сеть ICMP-сообщений об ошибках. Предложите свой вариант алгоритма, который должен применяться протоколом NAT при появлении на внешнем интерфейсе ICMP-сообщения. (*Подсказка*: прежде чем передавать ICMP-сообщение далее, протокол NAT должен внести исправления не только в заголовок IP, но и в поле данных ICMP.)
19. Почему для UDP-трафика неприменим механизм RED?

Часть V

Технологии глобальных сетей

Технология IP, которую мы рассматривали в предыдущей части книги, позволяет строить составные сети различного типа, как локальные, так и глобальные. Существуют также технологии, которые были разработаны специально для глобальных сетей. Эти технологии реализованы в сетях X.25, которые представляют сегодня только исторический интерес, а также в активно применяемых сегодня сетях Frame Relay и ATM. Объединяет все перечисленные технологии то, что они основаны на технике виртуальных каналов, которой посвящена глава 21.

Техника виртуальных каналов является альтернативой дейтаграммному способу продвижения пакетов, на котором основаны сети Ethernet и IP. Конкуренция между этими двумя основными принципами передачи данных существует давно, практически с момента появления первых сетей с коммутацией пакетов. Коммерческие глобальные сети достаточно долго, вплоть до революционного распространения Интернета в середине 90-х, отдавали предпочтение технике виртуальных каналов. Основная причина состоит в том, что эта техника обеспечивает гораздо более высокую степень контроля над соединениями между пользователями сети и путями прохождения потоков информацией через узлы сети. В результате оператор получает возможность рационального распределения ресурсов между пользователями в соответствии с теми услугами, на которые они подписались. Проблема обеспечения параметров QoS также проще решается при применении виртуальных каналов. Безусловно, у этого подхода есть и недостатки, основной из которых — большие затраты времени и труда на установление каждого виртуального соединения.

Дейтаграммный метод взаимодействия узлов сети, напротив, отличаясь простотой связи любого узла сети с любым другим узлом, дает оператору весьма ограниченный контроль над распределением ресурсов между пользователями. В современных сетях компромисс достигается за счет сочетания обоих методов. В составной глобальной сети значительная часть образующих ее сетей работает на основе техники виртуальных каналов, то есть является сетями Frame Relay или ATM. В то же время объединение этих сетей происходит на основе дейтаграммного протокола IP. Такое многослойное построение WAN дает необходимый результат, но приводит к достаточно сложной организации сети и частичному дублированию функций каждым из слоев. Например, протоколы маршрутизации работают как в ATM-сетях, так и на лежащем выше уровне IP. Попытки более тесной интеграции дейтаграммного метода с методом виртуальных каналов привели к созданию технологии MPLS. В этой технологии протоколы маршрутизации стека TCP/IP используются для исследования топологии сети и нахождении рациональных маршрутов, а продвигаются пакеты на основе техники виртуальных каналов. Вопросы использования технологии IP для построения глобальных сетей рассматриваются в главе 22.

Обеспечение высокоскоростного доступа к сетевой магистрали представляет собой сегодня масштабную проблему. Действительно, скорость нужно повысить на миллионах линий связи, соединяющих помещения пользователей с ближайшим центральными офисами операторов связи. Поэтому традиционные для магистрали решения, основанные на использовании оптического волокна и требующие прокладки новых кабелей к домам и офисным зданиям, для обеспечения массового доступа чаще всего экономически не оправданы. Более эффективными

являются технологии, в которых задействуется существующая кабельная инфраструктура (например, линии ADSL, работающие на абонентских окончаниях телефонной сети), или кабельные модемы, использующие системы кабельного телевидения. Альтернативным решением является беспроводной доступ, как мобильный, так и фиксированный. В главе 23 изучаются различные подходы к организации высокоскоростного доступа пользователей к сетевой магистрали.

Часть, а вместе с ней и книга, завершается главой 24, которая посвящена обеспечению безопасности транспортной системы сети. Здесь рассматриваются различные типы виртуальных частных сетей (VPN), в частности подробно описываются сети, построенные на основе защищенного протокола IP (IPSec), а также одна из наиболее популярных современных технологий виртуальных частных сетей — технология MPLS.

- Глава 21. Виртуальные каналы в глобальных сетях
- Глава 22. Технология IP в глобальных сетях
- Глава 23. Удаленный доступ
- Глава 24. Защита сетевого трафика

ГЛАВА 21 Виртуальные каналы в глобальных сетях

Технологии X.25, Frame Relay и ATM значительно отличаются функциональными характеристиками. В то же время все они используют технику виртуальных каналов, которая является разновидностью техники, ориентированной на установление соединения. В главе 2 этот механизм был рассмотрен в общих чертах, теперь мы познакомимся с его деталями и особенностями реализации в каждой из перечисленных технологий.

Технологии виртуальных каналов рассматриваются в данной главе в хронологической последовательности. Технология **X.25** появилась на заре эры компьютерных сетей, практически одновременно с сетью ARPANET, давшей начало Интернету и дейтаграммному протоколу IP. В сетях X.25 виртуальные каналы используются для надежной передачи данных, что в 70-80 годы, когда эта технология была очень популярна, было весьма актуально, так как многие линии связи были аналоговыми и не могли сами по себе обеспечить надежную передачу цифровых данных. Поэтому так ценилась способность X.25 к восстановлению искаженных и потерянных пакетов.

Распространение высокоскоростных и надежных цифровых каналов в середине 80-х годов привело к тому, что функции технологии X.25 по обеспечению надежной передачи данных стали избыточными. Результатом этого стало появление принципиально новой технологии глобальных сетей **Frame Relay**. Особенность этой технологии состоит в том, что, освободившись от многих ненужных в современном телекоммуникационном мире функций, она выполняет только тот минимум, который необходим для доставки кадров адресату. Вместе с тем перечень возможностей Frame Relay был расширен за счет функции поддержки параметров QoS для эластичного трафика. Первоначально разработчики стандартов Frame Relay не планировали использования данной технологии для переноса чувствительного к задержкам трафика, поэтому уровень задержек и джиттер не входят в число параметров, гарантируемых пользователю. Тем не менее передача голоса с высоким качеством в сетях Frame Relay возможна — для этого коммутаторы сети должны обеспечивать приоритезацию трафика.

Технология **ATM** предоставляет своим пользователям разнообразный и интегрированный набор транспортных услуг. В отличие от технологий X.25 и Frame Relay, ATM была изначально задумана как технология, в равной степени ориентированная на передачу трафика всех существующих типов: компьютерных данных, голоса, видео, управления объектами и т. п. Фиксированный небольшой размер кадра, называемого здесь ячейкой, позволяет минимизировать задержки трафика реального времени. Однако платой за высокое качество услуг оказываются техническая сложность и высокая стоимость ATM-сети, а также проблемы обработки ячеек на сверхвысоких скоростях, таких как, например, 2,5 и 10 Гбит/с. Тем не менее ATM является очень популярной технологией и с ее возможностями по инжинирингу трафика и поддержке параметров QoS пока не сравнилась ни одна из существующих технологий.

Техника виртуальных каналов

Ключевые слова: коммутируемый виртуальный канал, постоянный виртуальный канал, сигнальный протокол, таблица маршрутизации, таблица коммутации, система управления сетью, кратковременный и долговременный потоки, агрегированный поток.

Существует два типа виртуальных каналов.

- **Коммутируемый виртуальный канал** (Switched Virtual Circuit, SVC), создание которого происходит по инициативе конечного узла сети с помощью автоматической процедуры.
- **Постоянный виртуальный канал** (Permanent Virtual Circuit, PVC), его создание происходит заранее, причем коммутаторы настраиваются вручную администратором сети, возможно, с привлечением централизованной системы сетевого администрирования и некоторого служебного протокола (пока чаще всего — фирменного).

Аббревиатуры SVC/PVC часто интерпретируются еще и как *Switched/Permanent Virtual Channel* (коммутируемый/постоянный виртуальный канал) или *Switched/Permanent Virtual Connection* (коммутируемое/постоянное виртуальное соединение).

Рассмотрим сначала процесс создания коммутируемого виртуального канала, то есть канала SVC.

Коммутируемые виртуальные каналы

Процедура создания коммутируемого виртуального канала подобна процедуре установления соединения в телефонных сетях, кратко рассмотренной в главе 3. В телефонных сетях протокол, реализующий такую процедуру, называется **сигнальным протоколом**, поэтому и протоколы установления виртуального соединения в сетях с коммутацией пакетов также часто называют сигнальными.

Создание коммутируемого виртуального канала требует наличия в коммутаторах таблиц маршрутизации, аналогичных таблицам маршрутизации дейтаграммных сетей, например IP-сетей. При этом неважно, как они были построены, автоматически, с помощью какого-либо протокола маршрутизации, или же вручную. Пример такой таблицы приведен на рис. 21.1.

Рисунок иллюстрирует процесс прокладки виртуального канала между узлами N1, A1 и N2, A2 через сеть, представленную здесь двумя коммутаторами S1 и S2. На рисунке помечены три описанных ниже этапа выполнения этой процедуры.

1. Установление виртуального канала начинается с того, что узел-инициатор N1, A1 генерирует специальный пакет — запрос на установление логического соединения с узлом N2, A2. В нашем обобщенном примере этот запрос назван Call Setup (такое же название он носит и в некоторых конкретных сиг нальных протоколах, например в Q.933 для Frame Relay и Q.2931 для ATM). Упрощенно можно сказать, что запрос содержит пару: многоразрядный адрес

узла назначения и начальное значение идентификатора виртуального канала (VCI). Пусть в нашем примере в самом начале запрос Call Setup имеет следующий начальный вид:

(102, 132456.8112).

Здесь 102 — начальное значение VCI, а 132456.8112 — адрес узла назначения, старшая часть которого — номер подсети, младшая часть — номер узла[1].

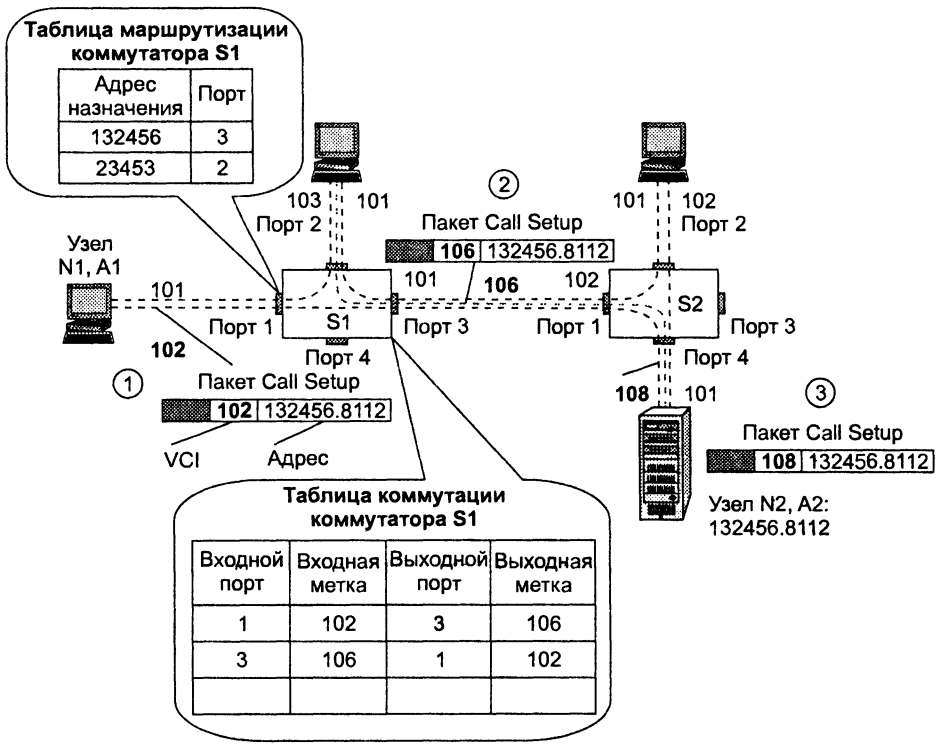


Рис. 21.1. Установление виртуального канала

Присвоенный виртуальному каналу номер 102 имеет локальное значение для порта компьютера, через который устанавливается соединение. Так как через порт уже проходит виртуальный канал с номером 101, то программное обеспечение сигнального протокола, работающее на конечном узле, просто выбрало первый свободный не используемый в данный момент на данном порту номер из разрешенного диапазона. *Такой подход гарантирует уникальную идентификацию виртуальных каналов в пределах каждого порта.*

Узел-инициатор должен выбрать коммутатор сети, которому целесообразно передать запрос на установление канала. Такой выбор может происходить на

[1] В примере для определенности используются 3-байтовые адреса подсетей и 2-байтовые адреса конечных узлов, на практике в глобальных сетях на основе виртуальных каналов часто применяются более длинные адреса.

основе таблицы маршрутизации узла-отправителя, но если узел соединен с сетью единственным портом, как в приведенном примере, то таблица маршрутизации узлу, естественно, не требуется. Попав в буфер порта 1 коммутатора S1, пакет Call Setup обрабатывается в соответствии со своим адресом назначения и значениями **таблицы маршрутизации**. Запись с адресом 132456 говорит, что пакет нужно передать на порт 3.

ПРИМЕЧАНИЕ

Заметим, что в приведенной таблице маршрутизации нет информации об адресе следующего коммутатора — в отличие от таблиц IP-сетей. Это связано с тем, что в глобальных сетях коммутаторы всегда связаны двухточечными линиями связи (физическими каналами), не поддерживающими множественное подключение, типичное для технологий локальных сетей, поэтому номер выходного порта однозначно определяет следующий коммутатор.

2. После определения для пакета Call Setup выходного порта коммутатор S1 генерирует для него новое значение номера виртуального канала, а именно 106. Этот номер был выбран потому, что на участке сети от порта 3 коммутатора S1 до порта 1 коммутатора S2 он однозначно идентифицирует устанавливаемый виртуальный канал. Именно это обстоятельство имелось в виду, когда ранее отмечалось, что идентификаторы виртуальных каналов имеют *локальный характер*. После изменения значения идентификатора виртуального канала пакет Call Setup приобретает вид (106, 132456.8112) и передается через выходной порт 3 коммутатора S1 на входной порт 1 коммутатора S2.

   Одновременно с продвижением пакета коммутатор создает **таблицу коммутации** (не путайте с упомянутой выше таблицей маршрутизации). Таблица коммутации потребуется впоследствии, когда виртуальный канал будет установлен и по нему начнут передаваться пользовательские данные, причем уже без адресов узлов назначения. Каждая запись таблицы коммутации состоит из четырех основных полей:

   - номера входного порта;
   - входной метки (SVC) в поступающих на входной порт пакетах;
   - номера выходного порта;
   - выходной метки (SVC) в передаваемых через выходной порт пакетах.

   В таблице коммутации запись 1-102-3-106 означает, что все пакеты, которые поступят на порт 1 с идентификатором виртуального канала 102, будут продвигаться на порт 3, а в поле идентификатора виртуального канала появится новое значение — 106.

   Виртуальные каналы могут быть однонаправленными и двунаправленными. В рассматриваемом примере создается двунаправленный канал, поэтому коммутатор создает еще одну запись в таблице коммутации — для продвижения пакетов в обратном направлении, от узла N2, A2 к узлу N1, A1. Эта запись является зеркальной по отношению к первой записи, так что пакет, имеющий метку 106 и поступивший на порт 3 коммутатора S1, получит при выходе из порта 1 первоначальное значение метки, а именно 102. В результате узел N1, A1 правильно распознает принадлежность пришедшего пакета виртуаль-

ному каналу, несмотря на постоянные смены номеров в процессе путешествия пакета по сети.

3. Процедуру установления виртуального канала продолжает коммутатор S2. По адресу назначения, указанному в запросе, и по своей таблице маршрутизации (на рисунке она не показана) он определяет выходной порт и передает на него запрос, обновляя при этом поле идентификатора виртуального канала. В данном случае коммутатор назначил пакету Call Setup номер виртуального канала 108. В результате запрос приходит в конечный узел в виде (108, 132456.8112). Конечный узел, получив запрос, может его принять или отвергнуть. О положительном решении, то есть об установлении виртуального канала, он сообщает служебным пакетом Connect, который проходит по сети в обратном направлении, используя «зеркальные» записи в таблице коммутации.

После получения подтверждения Connect конечные узлы могут начать пользоваться проложенным виртуальным каналом, посылая по нему пользовательские данные. Отправляемые узлом N1, A1 ячейки продвигаются на основе значения идентификатора виртуального канала, который обычно имеет небольшую длину, например, в технологии X.25 он занимает всего полтора байта, в то время как длина адреса узла в сети X.25 достигает 16 байт.

По существу, в сетях на основе коммутируемых виртуальных каналов используются два режима работы сети.

- При прокладке канала SVC запрос на установление соединения передается по сети в стандартном режиме маршрутизации с глобальными (для всей сети) адресами назначения и информацией о полной топологии сети. То есть протоколы установления виртуальных каналов (сигнальные протоколы) работают на сетевом уровне модели OSI.
- После установления соединения сеть начинает работать на основе локальных меток и локальных таблиц коммутации, что позволяет отнести такой режим к канальному уровню модели OSI, а коммуникационные устройства — к классу коммутаторов (стандартное название для устройств этого уровня).

Постоянные виртуальные каналы

Постоянный виртуальный канал (PVC) не прокладывается динамически узлами сети. Вместо этого администратор сети заранее создает таблицы коммутации вручную. Администратор может делать это локально, подключаясь к коммутатору, например, с помощью интерфейса SS-232 и используя свой ноутбук как виртуальный терминал. Конечно, это — не самый удобный способ конфигурирования таблиц коммутации для таких распределенных систем, как глобальные сети. Поэтому обычно администратор использует ту или иную **систему управления сетью**. Администратор вводит в систему данные о том, через какие узлы должен проходить виртуальный канал, система взаимодействует с коммутаторами сети, автоматически выбирая нужные значения меток и создавая записи в таблицах коммутации. К сожалению, системы управления сетью часто несовместимы, так

как любая из них — это сложное приложение, которое реализуется разными производителями по-разному.

В результате автоматизировать установление канала PVC можно только в пределах части сети, работающей на оборудовании одного производителя, а «сшивать» части PVC на границе сетей приходится вручную.

Очевидно, что при создании PVC таблицы маршрутизации становятся ненужными, так как путь выбирается администратором.

Для того чтобы приготовить к использованию созданный постоянный виртуальный канал, администратор должен ввести в конечные узлы, для которых канал создавался, метки каждого конца канала. Например, если бы виртуальный канал на рис. 21.1 был проложен как постоянный, тогда администратор компьютера N1, A1 должен был ввести в него метку 102, а администратор компьютера N2, A2 — метку 108.

Если технология виртуальных каналов поддерживает только каналы PVC, то это дает основание считать ее исключительно технологией канального уровня. Примером такого рода является технология Frame Relay, в которой долгое время существовали только каналы PVC, так что ее по праву считали канальной технологией. И хотя сегодня сети Frame Relay поддерживают оба типа каналов, ее попрежнему часто называют канальной технологией, имея в виду режим продвижения данных. Технология ATM с самого начала своего существования поддерживала обе разновидности виртуальных каналов, тем не менее ее по той же причине чаще всего относят к технологии второго уровня.

Сравнение технологий виртуальных каналов и дейтаграмм

Техника виртуальных каналов имеет свои достоинства и недостатки по сравнению с дейтаграммной техникой.

В отличие от дейтаграммных протоколов, таких, например, как IP, протоколы с поддержкой виртуальных каналов требуют предварительного установления соединения, что вносит дополнительную задержку перед передачей данных. Эта задержка особенно сказывается при передаче небольших объемов данных — так называемых *кратковременных* потоков, когда время установления виртуального канала может быть соизмеримым со временем передачи данных.

Дейтаграммные сети, в которых этап установления соединения отсутствует, более эффективно справляются с передачей **кратковременных потоков** данных. Сети, поддерживающие виртуальные каналы, больше подходят для передачи **долговременных потоков**.

Однако следует учесть, что время, затраченное на установление виртуального канала, компенсируется последующей быстрой передачей всего потока пакетов. Перемещение пакетов в сети с поддержкой виртуальных каналов ускоряется за счет двух факторов.

- Решение о продвижении пакета принимается быстрее благодаря меньшему размеру таблиц коммутации.
- Существенно сокращается адресная часть в пакетах, следовательно, снижается доля служебной информации и возрастает полезная скорость передачи данных. Адреса конечных узлов в глобальных сетях обычно имеют достаточно большую длину — 14–15 десятичных цифр, которые занимают до 20 байт в служебном поле пакета. Номер же виртуального канала обычно не превышает 10–12 бит.

Наиболее эффективным по критерию производительности являются каналы PVC. Значительную часть работы по маршрутизации пакетов администратор сети уже выполнил, вручную проложив виртуальные каналы, коммутаторам остается только быстро продвигать пакеты на основе готовых таблиц коммутации портов. Постоянный виртуальный канал подобен выделенному каналу в том смысле, что не требуется устанавливать или разрывать соединение, обмен пакетами по PVC может происходить в любой момент времени. Отличие PVC от выделенной линии состоит в том, что пользователь не имеет тех гарантий относительно действительной пропускной способности канала. Зато использование PVC обычно намного дешевле, чем аренда выделенной линии, за счет того, что пользователь делит пропускную способность сети с другими пользователями.

Постоянные виртуальные каналы выгодно использовать для передачи **агрегированных потоков** трафика, состоящих из большого количества индивидуальных потоков абонентов сети. В этом случае виртуальный канал прокладывается не между конечными абонентами, а между граничными участками сетевой магистрали, на которой данный агрегированный поток существует, например, от одного пограничного маршрутизатора сети оператора связи до другого. В силу закона больших чисел агрегированные потоки обладают высокой степенью устойчивости, так что для них эффективно задействовать постоянные виртуальные каналы.

Еще одно преимущество дейтаграммных сетей состоит в том, что они достаточно быстро адаптируются к изменениям топологии, вызванным, например, отказом маршрутизатора или линии связи. При возникновении такой ситуации пакеты просто направляются по новому пути (здесь, правда, нужно учесть время установления новой конфигурации в таблицах маршрутизации). А при отказе коммутатора или линии связи на пути виртуального канала соединение разрывается, и виртуальный канал нужно прокладывать заново в обход отказавших участков сети.

Сети X.25

Ключевые слова: интерфейс между пользователем и сетью, интерфейс между сетями, устройство PAD, международный номер, код идентификации сети, номер национального терминала, синхронные интерфейсы X.21 и X.21 bis, протокол сбалансированного доступа к линии связи, протокол X.25/3, пакет данных.

Стандарт X.25 был разработан комитетом CCITT в 1974 году и пересматривался несколько раз. Стандарт описывает не внутреннее устройство сети X.25, а только пользовательский интерфейс с сетью, что является характерной особенностью глобальных сетей. Интерфейс этого типа называют **интерфейсом между пользователем и сетью** (User-to-Network Interface, UNI). Внутреннее же устройство сети может быть произвольным, эта часть оставлена на усмотрение оператора сети. На практике коммутаторы глобальных сетей взаимодействуют по таким же протоколам, аналогичным протоколам UNI. Для взаимодействия между собой сетей различных операторов связи обычно разрабатывается **интерфейс между сетями** (Network-to-Network Interface, NNI), который часто является модифицированной версией интерфейса UNI.

Технология сетей X.25 имеет несколько существенных признаков, отличающих ее от других технологий.

- X.25 наилучшим образом подходит для передачи трафика низкой интенсивности, характерного для алфавитно-цифровых терминалов 70–80 годов, и в меньшей степени соответствует более высоким требованиям трафика локальных сетей.
- В структуре сети имеется специальное устройство **PAD** (Packet Assembler Disassembler), предназначенное для сборки нескольких низкоскоростных старт-стопных потоков байтов от алфавитно-цифровых терминалов в пакеты, передаваемые по сети и направляемые компьютерам для обработки. Наличие устройства PAD выдает возраст технологии X.25, так как в начале 70-х алфавитно-цифровые терминалы представляли собой неинтеллектуальные устройства, которые могли выполнять только примитивные операции. Все же операции по выполнению функций протоколов канального и сетевого уровней выполняли за них устройства PAD.
- Протоколы трехуровневого стека протоколов X.25 на канальном и сетевом уровнях работают с установлением соединения, управляют потоками данных и исправляют ошибки. Такая избыточность функций, обеспечивающих надежную передачу данных, объясняется ориентацией технологии на ненадежные линии связи с интенсивностью битовых ошибок в диапазоне 10^{-3}–10^{-4}.
- Сетевой уровень рассчитан на работу только с одним протоколом канального уровня и не может подобно протоколу IP объединять разнородные сети.

Сеть X.25 состоит из коммутаторов, расположенных в различных географических точках и соединенных высокоскоростными выделенными линиями (рис. 21.2). Выделенные линии могут быть как цифровыми, так и аналоговыми.

Устройства PAD могут быть встроенными или удаленными. Встроенное устройство PAD обычно расположено в стойке коммутатора. Терминалы получают доступ к нему по телефонной сети с помощью модемов. Удаленное устройство PAD подключается к коммутатору через выделенную линию связи X.25. К удаленному устройству PAD терминалы подключаются по асинхронному интерфейсу, обычно по интерфейсу RS-232C. Одно устройство PAD обычно обеспечивает доступ для 8, 16 или 24 асинхронных терминалов. Терминалы не имеют конечных адресов сети X.25. Адрес присваивается порту PAD, который подключен к коммутатору пакетов X.25 выделенной линией.

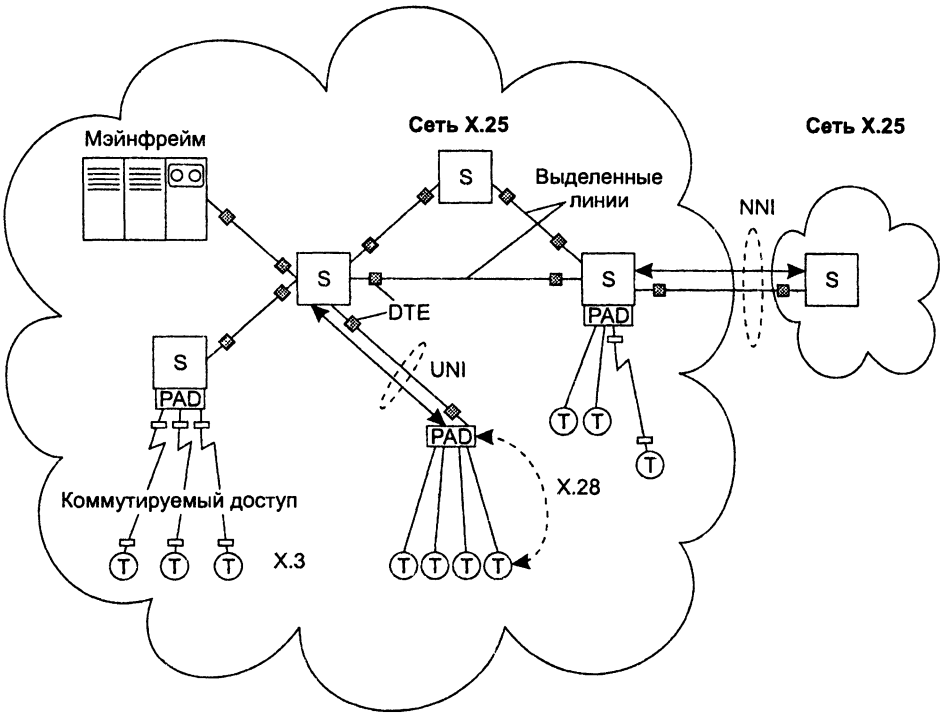


Рис. 21.2. Структура сети X.25

Компьютеры и локальные сети обычно подключаются к сети X.25 непосредственно через адаптер X.25 или маршрутизатор, поддерживающий на своих интерфейсах протоколы X.25.

Адресация в сетях X.25

Если сеть X.25 работает автономно, то администратор может использовать адреса любой длины (в пределах формата поля адреса) и назначать адресам произвольные значения. Максимальная длина поля адреса в пакете X.25 составляет 16 байт.

Рекомендация X.121 комитета CCITT определяет международную систему нумерации адресов для сетей передачи данных общего пользования. Если предполагается, что сеть X.25 будет обмениваться данными с другими сетями X.25, то при назначении адресов ее узлам следует придерживаться стандарта X.121.

Адреса X.121, называемые также **международными номерами** (International Data Numbers, IDN), имеют разную длину, которая может доходить до 14 десятичных знаков.

- Первые 4 цифры IDN называют **кодом идентификации сети** (Data Network Identification Code, DNIC). Код DNIC поделен на две части:

- первая часть (3 десятичных цифры) определяет страну, в которой находится сеть;
- вторая (1 десятичная цифра) — номер сети X.25 в данной стране.

Таким образом, внутри каждой страны можно организовать только 10 сетей X.25. Если же требуется перенумеровать больше, чем 10 сетей для одной страны, проблема решается тем, что одной стране дается несколько кодов. Например, Россия имела до 1995 года один код — 250, а в 1995 году ей был выделен еще один код — 251.

- Остальные цифры называются **номером национального терминала** (National Terminal Number, NTN). Эти цифры позволяют идентифицировать определенное устройство DTE в сети X.25.

Стек протоколов сети X.25

Стандарты сетей X.25 описывают три уровня протоколов (рис. 21.3).

- *На физическом уровне* определены **синхронные интерфейсы X.21** и **X.21 bis** к оборудованию передачи данных — либо к устройствам DSU/CSU, если выделенная линия является цифровой, либо к синхронному модему, если линия аналоговая. Протокол физического уровня *линии связи* не оговорен, и это дает возможность использовать линии разных стандартов.
- *На канальном уровне* используется подмножество протокола HDLC, обеспечивающее возможность автоматической передачи в случае ошибок в линии. На канальном уровне обычно используется **протокол сбалансированного доступа к линии связи** (Link Access Protocol — Balanced, LAP-B). Этот протокол обеспечивает сбалансированный режим работы, то есть оба узла, участвующих в соединении, равноправны. По протоколу LAP-B устанавливается соединение между пользовательским оконечным оборудованием (компьютером, IP- или IPX-маршрутизатором) и коммутатором сети. Хотя стандарт это и не оговаривает, но по протоколу LAP-B также обычно устанавливаются соединения на канальном уровне между непосредственно связанными коммутаторами сети. LAP-B является протоколом, который ориентирован на соединение и использует алгоритм скользящего окна для надежной передачи кадров между двумя непосредственно соединенными устройствами. В отличие от TCP реализация этого алгоритма в протоколе LAB-P проще. Нумеруются в данном случае не байты, а кадры. Окно не может изменяться динамически, имея фиксированный размер в 8 или 128 кадров. LAP-B относится к семейству протоколов высокоуровневого управления линией связи (HDLC), работа которых более подробно рассматривается в главе 22.
- *На сетевом уровне* (в стандарте он назван не сетевым, а *пакетным*) определен протокол **X.25/3** для обмена пакетами между оконечным оборудованием и сетью передачи данных. Соединения LAP-B обеспечивают надежную связь между *двумя соседними узлами*, но не дают возможности обмениваться информацией конечным узлам между собой. Для установления виртуального соединения между *конечными узлами* используется протокол X.25/3.

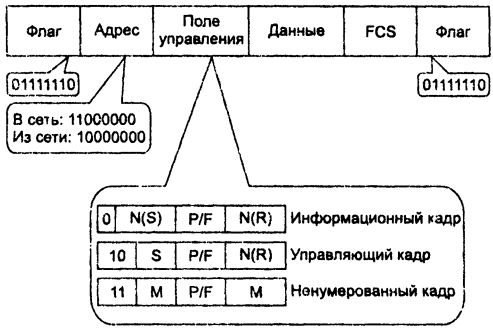


Рис. 21.3. Стек протоколов сети X.25

Остановимся подробнее на работе протокола X.25/3. Основными функциями его являются:

- установление и разрыв виртуального канала (соединения) между конечными абонентами сети;
- маршрутизация пакетов;
- управление потоком пакетов.

Для установления виртуального канала узел-инициатор соединения посылает в сеть пакет сигнализации Call Request, вложенный в кадр LAP-B.

ПРИМЕЧАНИЕ

В отличие от других сетей на основе техники виртуальных каналов, в стеке X.25 отсутствует отдельный протокол сигнализации. Когда это необходимо, его функции берет на себя протокол X.25/3, переходя в специальный режим.

В пакете Call Request задаются адреса получателя и отправителя в формате X.121. Пакет Call Request принимается коммутатором сети и маршрутизируется на основании таблицы маршрутизации. Протокол маршрутизации для сетей X.25 не определен, поэтому таблицы маршрутизации здесь всегда формируются вручную.

Перемещаясь по маршруту от одного коммутатора к другому, пакет Call Request побуждает их генерировать новые записи в таблицах коммутации и назначать ему новые значения меток. Таким образом создается новый виртуальный канал. Начальное значение номера виртуального канала задает пользователь, помещая его в пакет Call Request в поле LCN (Logical Channel Number — номер логического канала) — аналоге поля VCI, которое мы не раз упоминали при описании принципа установления виртуальных каналов.

После установления виртуального канала конечные узлы обмениваются пакетами другого формата — **пакетами данных**. В пакетах данных адреса получателя и отправителя уже не указываются, в качестве адресной информации используется только короткий номер LCN.

Отличием технологии X.25 от рассматриваемых далее технологий Frame Relay и ATM является то, что она представляет собой технологию сетевого уровня. Дей-

ствительно, после установления виртуального канала в сети X.25 данные передаются протоколом сетевого, а не канального уровня.

Сети Frame Relay

Ключевые слова: технология ISDN, услуги Frame Relay и Frame Switching, протокол LAP-F (Q.922), протоколы LAP-F core и LAP-F control, протоколы LAP-D (Q.921) и Q.933, поля DLCI, C/R, DE, FECN и BECN, согласованная скорость передачи данных, согласованная величина пульсации, дополнительная величина пульсации, признак готовности к удалению.

Сети **Frame Relay** гораздо лучше подходят для передачи пульсирующего трафика компьютерных сетей по сравнению с сетями X.25. Правда, это преимущество проявляется только тогда, когда линии связи приближаются по качеству к линиям связи локальных сетей, а для глобальных линий такое качество обычно достижимо только при использовании волоконно-оптических кабелей.

Технология Frame Relay была сначала стандартизована комитетом CCITT (ITU-T) как одна из служб сетей ISDN. Технология **ISDN** является первым широкомасштабным проектом по созданию всемирной универсальной сети, предоставляющей все основные виды услуг телефонных сетей и сетей передачи данных. К сожалению, этот амбициозный проект не достиг поставленной цели, и сегодня сети нового поколения строятся уже на основе других технологий, в частности IP. В то же время в ходе реализации проекта было достигнуто несколько хотя и не таких глобальных, но тем не менее очень важных целей. К ним можно причислить и создание технологии Frame Relay, которая сегодня является уже независимой от ISDN технологией[1].

В рекомендациях I.122, вышедших в свет в 1988 году, услуги по передаче данных входили в число дополнительных услуг пакетного режима ISDN. При пересмотре этих рекомендаций в 1992–93 гг. появились стандарты на две новые услуги: **Frame Relay** и **Frame Switching**. Разница между ними состоит в том, что Frame Switching обеспечивает гарантированную доставку кадров, а Frame Relay — доставку по возможности.

Простая и в то же время эффективная для волоконно-оптических линий связи технология Frame Relay сразу привлекла внимание ведущих телекоммуникационных компаний и организаций по стандартизации. В ее становлении и стандартизации помимо CCITT (ITU-T) активное участие принимали форум по ретрансляции кадров (Frame Relay Forum, FRF) и комитет T1S1 института ANSI. Технология же Frame Switching так и осталась всего лишь стандартом, никогда не имевшим широкого распространения.

Стандарты Frame Relay, подготовленные и ITU-T/ANSI, и FRF, определяют два типа виртуальных каналов — постоянные (PVC) и коммутируемые (SVC). Это соответствует потребностям пользователей, так как для соединений, по которым

[1] Мы кратко рассмотрим сети ISDN в главе 22.

трафик передается почти всегда, больше подходят постоянные каналы, а для соединений, требующихся только несколько часов в месяц, — коммутируемые. Однако производители оборудования Frame Relay и поставщики услуг сетей Frame Relay начали с поддержки только постоянных виртуальных каналов. Это, естественно, значительно обеднило технологию. Оборудование, поддерживающее коммутируемые виртуальные каналы, появилось на рынке с большой задержкой. Именно поэтому технология Frame Relay часто ассоциируется только с постоянными виртуальными каналами.

Стек протоколов Frame Relay

Стек протоколов Frame Relay устроен значительно проще, чем стек технологии X.25. Разработчики технологии Frame Relay, учитывая высокое качество каналов связи на оптическом волокне, появившихся в конце 80-х годов, посчитали возможным не включать в протоколы стека функции обеспечения надежности. Если же, несмотря на малую вероятность такого события, ошибка все же происходит, то технология Frame Relay игнорирует эту ситуацию, оставляя работу по восстановлению утерянных или искаженных кадров протоколам верхних уровней, таким как TCP.

Именно благодаря низкой протокольной избыточности технология Frame Relay обеспечивает высокую пропускную способность и небольшие времена задержки кадров.

ПРИМЕЧАНИЕ

Параллельно была разработана технология Frame Switching, которая, как и X.25, обеспечивает надежную передачу кадров на канальном уровне и может применяться в тех случаях, когда каналы обладают недостаточно высоким уровнем качества или же к канальному уровню по каким-то причинам предъявляются требования надежной передачи кадров. На практике технология Frame Switching не нашла своего применения, но поскольку стек протоколов Frame Relay создавался с учетом существования технологии Frame Switching, мы далее все же кратко остановимся на ней.

На рис. 21.4 показан стек протоколов технологий Frame Relay и Frame Switching в том виде, в котором они описаны в рекомендациях ITU-T. Протоколы слоя управления выполняют работу по установлению виртуального соединения, а протоколы слоя данных передают кадры по уже установленному виртуальному соединению.

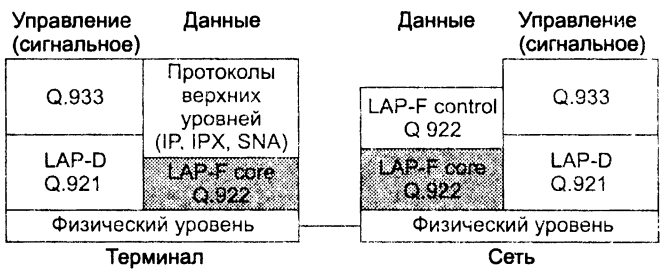


Рис. 21.4. Стек протоколов технологий Frame Relay и Frame Switching

На канальном уровне сетей Frame Relay работает протокол **LAP-F** (Link Access Procedure for Frame mode bearer services), называемый в рекомендациях ITU-T аббревиатурой **Q.922**. Существует две версии этого протокола.

- Протокол **LAP-F core** является той «рабочей лошадкой», которая трудится во всех сетях Frame Relay. Этот протокол обеспечивает минимум средств, позволяющих построить сеть Frame Relay. Правда, в этом случае сеть будет предоставлять только услуги постоянных виртуальных каналов.
- Протокол **LAP-F control**, обеспечивающий восстановление кадров по алгоритму скользящего окна, необходим для того, чтобы сеть оказывала услуги Frame Switching (коммутации кадров).

Оба протокола (LAP-F core и LAP-F control) относятся к протоколам канального уровня, обеспечивая передачу кадров между двумя соседними коммутаторами.

На физическом уровне сеть Frame Relay может использовать линии связи технологии PDH/SDH или ISDN.

Теперь рассмотрим слой управления, выполняющий функции установления динамически коммутируемых каналов SVC. Коммутаторы сети должны поддерживать два протокола слоя управления — на канальном уровне LAP-D (который называется также **Q.921**) и **Q.933** на сетевом. Протокол LAP-D в сетях Frame Relay обеспечивает надежную передачу сигнальных кадров между соседними коммутаторами.

Протокол Q.933 использует адреса конечных узлов, между которыми устанавливается виртуальный канал. Эти адреса обычно задаются в формате телефонных адресов, соответствующих стандарту E.164. Адрес состоит из 15 десятичных цифр, которые делятся, как и обычные телефонные номера, на поля кода страны (от 1 до 3 цифр), кода города и номера абонента. К адресу добавляется до 40 цифр *подадреса*, которые требуются для нумерации терминальных устройств, если у одного абонента их несколько.

Протокол автоматического составления таблиц маршрутизации для технологии Frame Relay не определен, поэтому может использоваться фирменный протокол производителя оборудования, или же таблицы могут составляться вручную.

ВНИМАНИЕ

Основное преимущество Frame Relay по сравнению X.25 состоит в следующем. В то время как в сетях X.25 после установления соединения пользовательские данные передаются протоколами канального и сетевого уровней, в сетях Frame Relay после установления виртуального соединения данные передаются только с помощью протокола канального уровня, что значительно снижает накладные расходы.

Технологию Frame Relay чаще всего относят к технологиям *канального* уровня, ставя во главу угла процедуры передачи пользовательских данных и опуская процедуры установления виртуального канала, которые выполняются с привлечением протокола *сетевого* уровня.

По виртуальным каналам Frame Relay могут передаваться данные различных протоколов. Спецификация RFC 1490 определяет методы инкапсуляции в кадры Frame Relay пакетов сетевых протоколов, таких как IP и IPX, протоколов локальных сетей, например Ethernet, а также протокола SNA.

Структура кадра протокола LAP-F приведена на рис. 21.5.

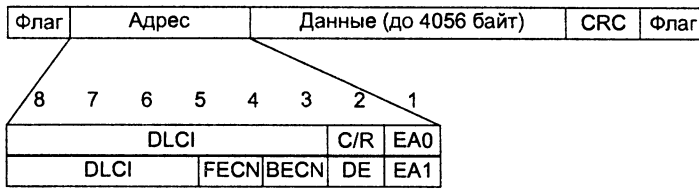


Рис. 21.5. Формат кадра LAP-F

Поле **DLCI** (Data Link Connection Identifier — идентификатор соединения уровня канала данных) состоит из 10 бит, что позволяет задействовать до 1024 виртуальных соединений. Поле DLCI может занимать и большее число разрядов — этим управляют признаки расширения адреса EA0 и EA1 (аббревиатура EA как раз и означает Extended Address, то есть расширенный адрес). Если бит расширения адреса установлен в ноль, то признак называется EA0 и означает, что в следующем байте имеется продолжение поля адреса, а если бит расширения адреса равен 1, то поле называется EA1 и означает окончание поля адреса. Десятиразрядный формат DLCI является основным, но при использовании трех байтов для адресации поле DLCI имеет длину 16 бит, а при использовании четырех байтов — 23 бита.

Стандарты Frame Relay распределяют DLCI-адреса между пользователями и сетью следующим образом:

- 0 — используется для виртуального канала локального интерфейса администрирования (LMI);
- 1–15 — зарезервированы;
- 16–991 — используются абонентами для нумерации каналов PVC и SVC;
- 992–1007 — используются сетевой транспортной службой;
- 1008–1022 — зарезервированы;
- 1023 — используется для управления канальным уровнем.

Таким образом, в любом интерфейсе Frame Relay для оконечных устройств пользователя отводится 976 DLCI-адресов.

Поле данных может иметь размер до 4056 байт.

Поле **C/R** переносит признак команды (Command) или ответа (Response). Этот признак является унаследованным и используется в протокольных операциях HDLC.

Поля **DE**, **FECN** и **BECN** используются протоколом для управления трафиком и поддержания заданного качества обслуживания виртуального канала.

Поддержка параметров QoS

Для каждого виртуального соединения определяется несколько параметров, связанных со скоростью передачи данных и влияющих на качество обслуживания.

- **Согласованная скорость передачи данных** (Committed Information Rate, CIR) — скорость, с которой сеть будет передавать данные пользователя.
- **Согласованная величина пульсации** (Committed Burst Size, Bc) — максимальное количество байтов, которое сеть будет передавать от данного пользователя за интервал времени T, называемый временем пульсации, соблюдая согласованную скорость CIR.
- **Дополнительная величина пульсации** (Excess Burst Size, Be) — максимальное количество байтов, которое сеть будет пытаться передать сверх установленного значения Bc за интервал времени T.

Эти параметры являются однонаправленными, то есть виртуальный канал может поддерживать разные значения CIR, Bc и Be для каждого направления.

Если приведенные выше величины определены, то время T определяется следующей формулой:

$$T = Bc/CIR.$$

Можно задать значения CIR и T, тогда производной величиной станет величина пульсации Bc. Обычно для контроля пульсаций трафика выбирается время T, равное 1–2 секунды при передаче данных, и в диапазоне десятков-сотен миллисекунд при передаче голоса.

Соотношение между параметрами CIR, Bc, Be и T иллюстрирует рис. 21.6 (R — скорость в канале доступа; f_1–f_5 — кадры).

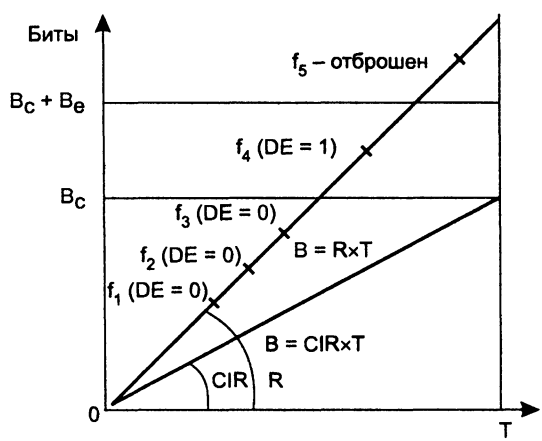


Рис. 21.6. Реакция сети на поведение пользователя

Основным параметром, по которому абонент и сеть заключают соглашение при установлении виртуального канала, является согласованная скорость передачи данных. Для постоянных виртуальных каналов это соглашение является частью

контракта на пользование услугами сети. При установлении коммутируемого виртуального канала SVC соглашение о качестве обслуживания заключается автоматически с помощью протокола Q.933 — требуемые параметры CIR, Bc и Be передаются в пакете запроса на установление соединения.

Скорость передачи данных измеряется на контрольном интервале времени T, на котором проверяются условия соглашения. В общем случае пользователь не должен в этом интервале передавать в сеть данные со средней скоростью, превосходящей CIR. Если же он нарушает соглашение, то сеть не гарантирует доставку кадра и помечает этот кадр **признаком готовности к удалению** (Discard Eligibility, DE), равным 1. Однако кадры, отмеченные таким признаком, удаляются из сети только в том случае, если коммутаторы сети испытывают перегрузки. Если же перегрузок нет, то кадры с признаком DE = 1 доставляются адресату.

Такое щадящее поведение сети соответствует случаю, когда общее количество данных, переданных пользователем в сеть за период T, не превышает значения Bc + Be. Если же этот порог превышен, то кадр не помечается признаком DE, а немедленно удаляется.

Рисунок 21.6 иллюстрирует случай, когда за интервал времени T в сеть по виртуальному каналу поступило 5 кадров. Средняя скорость поступления данных в сеть составила на этом интервале R бит/с, и она оказалась выше CIR. Кадры f_1, f_2 и f_3 доставили в сеть данные, суммарный объем которых не превысил порог Bc, поэтому эти кадры ушли дальше транзитом с признаком DE = 0. Данные кадра f_4, прибавленные к данным кадров f_1, f_2 и f_3, уже превысили порог Bc, но еще не превысили порога Bc + Be, поэтому кадр f_4 также ушел дальше, но уже с признаком DE = 1. Данные кадра f_5, прибавленные к данным предыдущих кадров, превысили порог Bc + Be, поэтому этот кадр был удален из сети.

Для контроля соглашения о параметрах качества обслуживания все коммутаторы сети Frame Relay поддерживают алгоритм «дырявого ведра» (leaky bucket). Этот алгоритм относится к тому же классу алгоритмов, что и уже рассмотренный нами в главе 20 алгоритм «ведра маркеров». Он также позволяет контролировать среднюю скорость и пульсацию трафика, однако делает это несколько иначе.

Алгоритм поддерживает счетчик C поступивших от пользователя байтов. Каждые T секунд этот счетчик уменьшается на величину Bc (или же сбрасывается в 0, если значение счетчика меньше, чем Bc). Все кадры, данные которых не увеличили значение счетчика свыше порога Bc, пропускаются в сеть со значением признака DE = 0. Кадры, данные которых привели к значению счетчика, большему Bc, но меньшему Bc + Be, также передаются в сеть, но уже с признаком DE = 1. И наконец, кадры, которые привели к значению счетчика, большему Bc + Be, отбрасываются коммутатором.

Пользователь может договориться о поддержании не всех параметров качества обслуживания для данного виртуального канала, а только некоторых.

Например, можно использовать только параметры CIR и Bc. Этот вариант дает более качественное обслуживание, так как кадры никогда не отбрасываются коммутатором сразу. Коммутатор только помечает кадры, которые превышают порог Bc за время T, признаком DE = 1. Если сеть не сталкивается с перегрузками,

то кадры такого канала всегда доходят до конечного узла, даже если пользователь постоянно нарушает договор с сетью.

Популярен еще один вид заказа на обслуживание, при котором оговаривается только порог Be, а скорость CIR полагается равной нулю. Все кадры такого канала сразу же отмечаются признаком DE = 1, но отправляются в сеть, а при превышении порога Be отбрасываются. Контрольный интервал времени T в этом случае вычисляется как Be/R, где R — скорость доступа к каналу.

Как видно из описания, алгоритм дырявого ведра более «строго» контролирует пульсации трафика, чем алгоритм ведра маркеров (см. главу 20). Алгоритм ведра маркеров разрешает трафику в периоды пониженной активности накапливать объем пульсации, а затем использовать эти накопления в периоды всплесков трафика. В алгоритме дырявого ведра такой возможности нет, так как счетчик C сбрасывается в ноль принудительно в конце каждого периода T независимо от того, сколько байтов поступило от пользователя в сеть в течение этого периода.

На рис. 21.7 приведен пример сети Frame Relay с пятью удаленными региональными отделениями корпорации. Обычно доступ к сети осуществляется по каналам с пропускной способностью, большей, чем CIR. Но при этом пользователь платит не за пропускную способность канала, а за заказанные величины CIR, Bc и Be. Так, при использовании в качестве линии доступа канала T1 и заказа обслуживания со скоростью CIR, равной 128 Кбит/с, пользователь будет платить только за скорость 128 Кбит/с, а скорость канала T1 в 1,544 Мбит/с окажет влияние на верхнюю границу возможной пульсации Bc + Be.

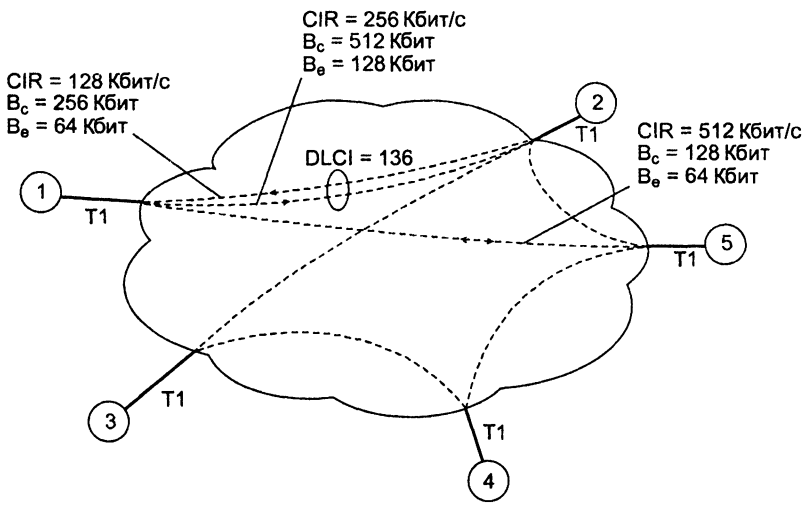


Рис. 21.7. Пример обслуживания в сети Frame Relay

Параметры качества обслуживания могут быть разными для разных направлений виртуального канала. Так, на рисунке абонент 1 соединен с абонентом 2 виртуальным каналом с DLCI = 136. При направлении от абонента 1 к абоненту 2 канал имеет среднюю скорость 128 Кбит/с с пульсациями Bc = 256 Кбит (интервал T составил 1 с) и Be = 64 Кбит. А при передаче кадров в обратном направле-

нии средняя скорость уже может достигать значения 256 Кбит/с с пульсациями Bc = 512 Кбит и Be = 128 Кбит.

Механизм резервирования средней пропускной способности и максимальной пульсации является основным механизмом обеспечения параметров QoS в сетях Frame Relay.

Соглашения должны заключаться таким образом, чтобы сумма средних скоростей передачи данных в виртуальных каналах не превосходила возможностей портов коммутаторов. При заказе постоянных каналов за это отвечает администратор, а при установлении коммутируемых виртуальных каналов — программное обеспечение коммутаторов. При правильно взятых на себя обязательствах сеть борется с перегрузками путем удаления кадров с признаком DE = 1 и кадров, превысивших порог Bc + Be.

В технологии Frame Relay определен еще и дополнительный (необязательный) механизм управления потоком. Это механизм оповещения конечных пользователей о том, что в коммутаторах сети возникли перегрузки (переполнение необработанными кадрами). Бит FECN (Forward Explicit Congestion Notification — прямое явное уведомление о перегрузке) кадра извещает об этом принимающую сторону. На основании значения этого бита принимающая сторона должна с помощью протоколов более высоких уровней (TCP/IP, SPX и т. п.) известить передающую сторону о том, что та должна снизить интенсивность отправки пакетов в сеть.

Бит BECN (Backward Explicit Congestion Notification — обратное явное уведомление о перегрузке) извещает о переполнении в сети передающую сторону и является рекомендацией немедленно снизить скорость передачи. Бит BECN обычно отрабатывается на уровне устройств доступа к сети Frame Relay — маршрутизаторов, мультиплексоров и устройств CSU/DSU. Протокол Frame Relay не требует от устройств, получивших кадры с установленными битами FECN и BECN, немедленного прекращения передачи в данном направлении, как, например, происходит в сетях X.25. Эти биты должны служить указанием для протоколов более высоких уровней (TCP, SPX, NCP и т. п.) о снижении темпа передачи пакетов. Так как регулирование потока и принимающей, и передающей сторонами инициируется в разных протоколах по-разному, то разработчики протоколов Frame Relay учли оба направления снабжения предупреждающей информацией о переполнении сети.

Технология ATM

Ключевые слова: протокол PNNI, уровень адаптации ATM, подуровень сегментации и реассемблирования, подуровень конвергенции, протоколы AAL1, AAL2, AAL3/4 и AAL5, протокол ATM, номер виртуального соединения, идентификатор виртуального пути, идентификатор виртуального канала, коммутация виртуального пути, коммутация виртуального канала, протокол Q.2931, услуги категорий CBR, rtVBR, nrtVBR, ABR и UBR, служебная ячейка администрирования ресурсов, петля обратной связи.

Технология ATM (Asynchronous Transfer Mode — асинхронный режим передачи) была разработана как единый универсальный транспорт для нового поколения сетей с интегрированным обслуживанием, которые называются также широкополосными сетями ISDN (Broadband ISDN, B-ISDN). По сути, ATM стала второй попыткой построения универсальной сети после неудачи ISDN. В отличие от технологии Frame Relay, которая изначально предназначалась только для передачи эластичного компьютерного трафика, цели разработчиков ATM были значительно шире.

Технология ATM должна была обеспечивать:

- передачу трафика любого типа, как компьютерного, так и мультимедийного (голос, видео, управление в реальном времени), причем для каждого вида трафика качество обслуживания должно соответствовать его потребностям;
- иерархию скоростей передачи данных, от десятков мегабит до нескольких гигабит в секунду с гарантированной пропускной способностью для критически важных приложений;
- возможность использования имеющейся инфраструктуры линий связи и физических протоколов (PDH, SDH, High-speed LAN);
- взаимодействие с унаследованными протоколами локальных и глобальных сетей (IP, SNA, Ethernet, ISDN).

Необходимо сразу подчеркнуть, что большая часть этих целей была достигнута, и с середины 90-х годов ATM является работающей технологией, обеспечивающей наиболее полную и последовательную поддержку параметров QoS для пользователей сети. Кроме того, ATM, как и любая технология на основе техники виртуальных каналов, предоставляет широкие возможности по решению задач инжиниринга трафика.

Разработку стандартов ATM осуществляет большое количество производителей телекоммуникационного оборудования и операторов связи, входящих в форум ATM, а также комитеты ITU-T и ANSI.

Несмотря на очевидные успехи технологии ATM, которая работает на многих магистралях крупнейших операторов связи, опыт эксплуатации показал и ее ограничения. Так, технология ATM не вытеснила все остальные технологии и не стала единственной транспортной технологией телекоммуникационных сетей, хотя в середине 90-х годов казалось, что благодаря очевидным технологическим достоинствам ATM это неминуемо должно произойти. Теоретически, ATM может использоваться непосредственно прикладным уровнем протоколов, так что сеть может работать без протоколов IP и TCP/UDP. ATM обладает для этого многими качествами: поддержкой всех видов трафика, масштабируемостью и собственным сложным протоколом маршрутизации. Однако это возможно только в том случае, если сеть является технологически однородной, когда все сети всех поставщиков услуг поддерживают ATM. Очевидно, такой подход противоречит принципу составных сетей, согласно которому каждая сеть может поддерживать собственную транспортную технологию, а общий сетевой уровень объединяет эти сети в единую сеть.

Поэтому на практике протокол IP, начавший доминировать на сетевом уровне в середине 90-х годов, по-прежнему используется для объединения сетей, а ATM

остается одной из технологий, на основе которой работают многие сети, образующие составную сеть. Проблемы взаимодействия ATM с IP рассматриваются в главе 22.

Основные принципы технологии ATM

Сеть ATM имеет классическую иерархическую структуру крупной территориальной сети — конечные станции соединяются индивидуальными линиями связи с коммутаторами нижнего уровня, которые, в свою очередь, соединяются с коммутаторами более высоких уровней. Коммутаторы ATM с момента рождения этой технологии поддерживают как каналы PVC, так и каналы SVC. Для сетей ATM определен протокол маршрутизации **PNNI** (Private NNI — частный интерфейс NNI), с помощью которого коммутаторы могут строить таблицы маршрутизации автоматически, причем с учетом требований инжиниринга трафика. В публичных сетях ATM обычно используются адреса в стандарте E.164, что делает простым взаимодействие этих сетей с телефонными сетями. Адреса ATM имеют иерархическую структуру, подобно телефонным номерам или IP-адресам, которая обеспечивает масштабируемость сетей ATM до любого уровня, даже общемирового.

В больших сетях применяется понятие агрегированного виртуального пути, который объединяет виртуальные каналы, имеющие в сети ATM общий маршрут между исходным и конечным узлами или общую часть маршрута между некоторыми двумя коммутаторами сети. Это свойство также обеспечивает масштабируемость сетей ATM, так как позволяет существенно сократить количество виртуальных соединений, которые поддерживает магистральный коммутатор, а значит, повысить эффективность его работы.

Стандарт ATM не вводит свои спецификации на реализацию физического уровня. Здесь он основывается на технологии SDH/SONET, принимая ее иерархию скоростей. В соответствии с этим начальная скорость доступа пользователя сети — это скорость STM-1/OC-3 155 Мбит/с. Магистральное оборудование ATM работает и на более высоких скоростях STM-4 622 Мбит/с и STM-16 2,5 Гбит/с. Существует также оборудование ATM, которое поддерживает скорости PDH, такие как 2 или 34/45 Мбит/с.

Однако все перечисленные пока характеристики технологии ATM не свидетельствуют о том, что это некая «особенная» технология, а скорее представляют ее как достаточно развитую, но в то же время достаточно типичную технологию глобальных сетей, основанную на технике виртуальных каналов.

Главным свойством ATM, которое отличает ее от других технологий, является комплексная поддержка параметров QoS для всех основных видов трафика.

Для достижения этого свойства разработчики ATM тщательно проанализировали все типы трафика и провели его классификацию. Мы уже познакомились с этой классификацией в главе 7, когда рассматривали требования различных приложений к QoS. Напомним, что ATM разбивает весь трафик на 5 классов: A,

B, C, D и X. Первые четыре класса представляют трафик типовых приложений, которые отличаются устойчивым набором требований к задержкам и потерям пакетов, а также тем, генерируют они трафик с постоянной (CBR) или переменной (VBR) битовой скоростью. Класс X зарезервирован для уникальных приложений, набор характеристик и требований которых не относится ни к одному из первых четырех классов.

Однако на какое количество классов мы бы ни разбивали существующий трафик, принципиальная задача от этого не меняется — нужно найти решение для успешного сосуществования в одном канале и эластичных, и чувствительных к задержкам классов трафика. Требования этих классов почти всегда противоречат друг другу. Одним из таких противоречий является требование к размеру кадра.

Эластичный трафик выигрывает от увеличения размера кадра, так как при этом уменьшаются накладные расходы на служебную информацию. Мы видели на примере Ethernet, что скорость передачи пользовательской информации может изменяться почти в два раза при изменении размера поля данных от его минимальной величины в 46 байт до максимальной в 1500 байт. Конечно, размер кадра не может увеличиваться до бесконечности, так как при этом теряется сама идея коммутации пакетов. Тем не менее для эластичного трафика при современном уровне скоростей размер кадра в несколько тысяч байтов является вполне приемлемым.

Напротив, чувствительный к задержкам трафик обслуживается лучше при использовании кадров небольшого размера в несколько десятков байтов. При применении больших кадров начинают проявляться два нежелательных эффекта:

- ожидание низкоприоритетных кадров в очередях;
- задержка пакетизации.

Рассмотрим эти эффекты на примере голосового трафика.

Мы знаем, что *время ожидания кадра в очереди* можно уменьшить, если обслуживать кадры чувствительного к задержкам трафика в приоритетной очереди. Однако если размер кадра может меняться в широком диапазоне (например, от 29 до 4500 байт, как в технологии FDDI), то даже при придании чувствительным к задержкам кадрам высшего приоритета обслуживания в коммутаторах время ожидания компьютерного пакета может все равно оказаться недопустимо высоким. Например, пакет в 4500 байт будет передаваться в выходной порт на скорости 2 Мбит/с (максимальная скорость работы порта коммутатора Frame Relay) 18 мс. При совмещении трафика за это время необходимо через тот же порт передать 144 замера голоса. Прерывать передачу пакета в сетях нежелательно, так как при распределенном характере сети накладные расходы на оповещение соседнего коммутатора о прерывании пакета, а потом — о возобновлении передачи пакета с прерванного места оказываются слишком большими.

Задержка пакетизации — это время, в течение которого первый замер голоса ждет момента окончательного формирования пакета и отправки его по сети. Механизм образования этой задержки иллюстрирует рис. 21.8.

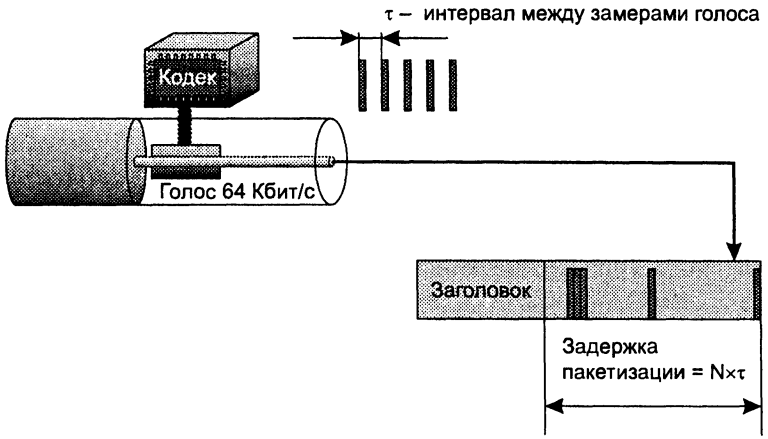


Рис. 21.8. Задержка пакетизации

Кодек делает замеры голоса через одинаковые интервалы времени. На рисунке в качестве примера он делает это с частотой 8 кГц, то есть через каждые 125 мкс. Если мы используем для передачи голоса кадры Ethernet максимального размера, то в один кадр помещается 1500 замеров голоса, так как каждый замер кодируется одним байтом данных. В результате первый замер, помещенный в кадр Ethernet, вынужден ждать отправки кадра в сеть (1500 – 1) × 125 = 187 375 мкс, или около 187 мс. Это весьма большая задержка для голосового трафика, рекомендации ITU-T говорят о величине 150 мс как о максимально допустимой задержке голоса. Важно отметить, что задержка пакетизации не зависит от битовой скорости протокола, она зависит только от частоты работы кодека и размера поля данных кадра. Это отличает ее от задержки ожидания, которая снижается с возрастанием битовой скорости.

Кадр ATM в 53 байта с полем данных 48 байт явился результатом компромисса между требований эластичного и чувствительного к задержкам трафиков. Другими словами, можно сказать, что компромисс был достигнут между телефонистами и компьютерщиками — первые настаивали на размере поля данных в 32 байта, а вторые — в 64 байта. Небольшой и фиксированный размер кадра ATM дал ему специальное название — *ячейка*.

При размере поля данных в 48 байт одна ячейка ATM обычно переносит 48 замеров голоса, которые делаются с интервалом в 125 мкс. Поэтому первый замер должен ждать примерно 6 мс, прежде чем ячейка будет отправлена по сети. Именно по этой причине телефонисты боролись за уменьшение размера ячейки, так как 6 мс — это задержка, близкая к пределу, за которым начинаются нарушения качества передачи голоса. При выборе размера ячейки в 32 байта задержка пакетизации составила бы 4 мс, что гарантировало бы более качественную передачу голоса. А стремление компьютерных специалистов увеличить поле данных до 64 байт вполне понятно — при этом повышается полезная скорость передачи данных. Избыточность служебных данных при использовании 48-байтного поля данных составляет 10 %, а при использовании 32-байтного поля данных она сразу повышается до 16 %.

Для пакета, состоящего из 53 байт, при скорости в 155 Мбит/с время передачи кадра на выходной порт составляет менее 3 мкс. Так что эта задержка не очень существенна для трафика, пакеты которого должны передаваться каждые 125 мкс.

Чтобы пакеты содержали адрес узла назначения и в то же время процент служебной информации не превышал размер поля данных пакета, в технологии ATM применен стандартный для WAN прием — передача ячеек в соответствии с техникой виртуальных каналов. Общая длина номера виртуального канала составляет 24 бит, что вполне достаточно для обслуживания большого количества виртуальных соединений каждым портом коммутатора глобальной (может быть всемирной) сети ATM.

Нужно отметить, что использование в ATM ячеек такого небольшого размера, создающих отличные условия для качественного обслуживания чувствительного к задержкам трафика, имеет и обратную сторону. Платой за качество является высокий уровень нагрузки на ATM-коммутаторы при работе на высоких скоростях. Напомним, что объем работы, который выполняет коммутатор или маршрутизатор любой технологии, прямо пропорционален количеству обрабатываемых в единицу времени пакетов, или кадров. Очевидно, что использование ячеек размеров с полем данных 48 байт приводит к колоссальному росту объема работы для ATM-коммутатора по сравнению с, например, коммутатором Ethernet, работающим с кадрами 1500 байт. Из-за этого обстоятельства ATM-коммутаторы долго не могли превзойти границу скорости интерфейсов в 622 Мбит/с и сравнительно недавно стали поддерживать интерфейсы 2,5 Гбит/с.

Выбор для передачи данных любого типа небольшой ячейки фиксированного размера еще не решает задачу совмещения разнородного трафика в одной сети, а только создает предпосылки для ее решения. Для полного решения этой задачи технология ATM привлекает и развивает идеи *резервирования пропускной способности* и *качества обслуживания*, реализованные в технологии Frame Relay.

В технологии ATM для каждого класса трафика определен набор количественных параметров, которые приложение должно задать. Например, для трафика класса A необходимо указать постоянную скорость, с которой приложение будет посылать данные в сеть, а для трафика класса B — максимально возможную скорость, среднюю скорость и максимально возможную пульсацию. Для голосового трафика можно не только указать на важность синхронизации между передатчиком и приемником, но и количественно задать верхние границы задержек и вариации задержек ячеек.

В технологии ATM поддерживается следующий набор основных количественных параметров для трафика виртуального соединения:

- *пиковая скорость передачи ячеек* (Peak Cell Rate, PCR);
- *средняя скорость передачи ячеек* (Sustained Cell Rate, SCR);
- *минимальная скорость передачи ячеек* (Minimum Cell Rate, MCR);
- *максимальная величина пульсаций* (Maximum Burst Size, MBS);
- *доля потерянных ячеек* (Cell Loss Ratio, CLR);
- *задержка передачи ячеек* (Cell Transfer Delay, CTD);
- *вариация задержек ячеек* (Cell Delay Variation, CDV).

Параметры скорости измеряются в ячейках в секунду, максимальная величина пульсаций — в ячейках, а временные параметры — в секундах. Максимальная величина пульсаций определяет количество ячеек, которое приложение может передать с пиковой скоростью при заданной средней скорости. Доля потерянных ячеек является отношением потерянных ячеек к общему количеству отправленных ячеек по данному виртуальному соединению. Так как виртуальные соединения являются дуплексными, то для каждого направления соединения могут быть заданы разные значения параметров.

В технологии ATM принят не совсем традиционный подход к трактовке качества обслуживания (QoS). Обычно качество обслуживания трафика характеризуется параметрами пропускной способности (здесь это RCR, SCR, MCR, MBS), параметрами задержек пакетов (CTD и CDV), а также параметрами надежности передачи пакетов (CLR). В ATM скоростные характеристики называют *параметрами трафика* и не включают их в число параметров качества обслуживания, хотя, по существу, они таковыми являются. Параметрами QoS в ATM являются только CTD, CDV и CLR. Сеть старается обеспечить такой уровень обслуживания, чтобы поддерживались требуемые значения и для параметров трафика, и для задержек ячеек, и для доли потерянных ячеек.

Соглашение между приложением и сетью ATM называется трафик-контрактом. Основным его отличием от соглашений, применяемых в сетях Frame Relay, является выбор одного из нескольких определенных классов трафика, для которого наряду с параметрами пропускной способности трафика могут указываться параметры задержек ячеек, а также параметр надежности доставки ячеек. В сети Frame Relay класс трафика один, и он характеризуется только параметрами пропускной способности.

Если для приложения не критично поддержание параметров пропускной способности и QoS, то оно может отказаться от задания этих параметров, указав в запросе на установление соединения признак обслуживания с максимальными усилиями. Такой тип трафика получил название трафика с неопределенной битовой скоростью (UBR).

После заключения трафик-контракта, который относится к определенному виртуальному соединению, в сети ATM работает несколько протоколов и служб, обеспечивающих нужное качество обслуживания. Для трафика UBR сеть выделяет ресурсы «по возможности», то есть те, которые в данный момент не заняты виртуальными соединениями, заказавшими определенные параметры качества обслуживания.

Стек протоколов ATM

Стек протоколов ATM показан на рис. 21.9, а распределение протоколов по конечным узлам и коммутаторам ATM — на рис. 21.10.

Стек протоколов ATM соответствует нижним уровням семиуровневой модели ISO/OSI и включает уровень адаптации ATM, собственно уровень ATM и физический уровень. Прямого соответствия между уровнями протоколов технологии ATM и уровнями модели OSI нет.

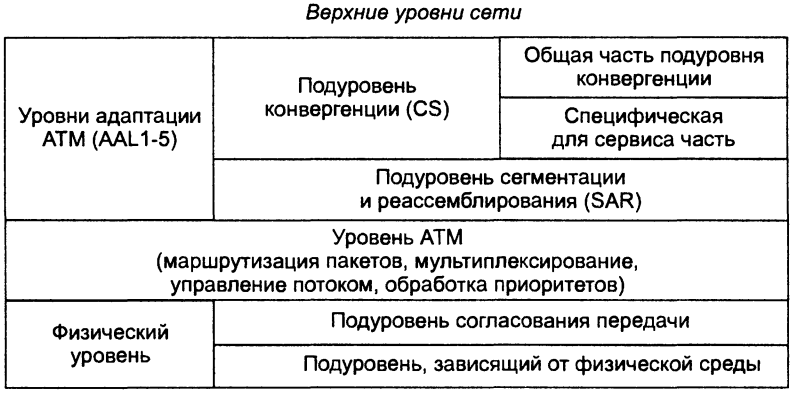


Рис. 21.9. Структура стека протоколов ATM

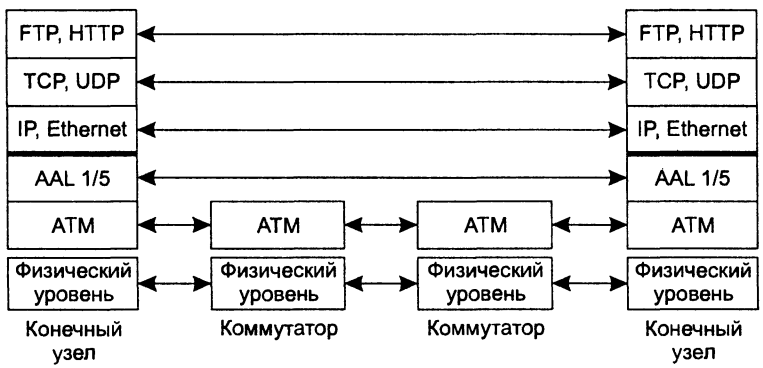


Рис. 21.10. Распределение протоколов по узлам и коммутаторам сети ATM

Уровень адаптации ATM

Уровень адаптации ATM (ATM Adaptation Layer, AAL) представляет собой набор протоколов AAL1–AAL5, которые преобразуют сообщения протоколов верхних уровней сети ATM в ячейки ATM нужного формата. Функции этих уровней достаточно условно соответствуют функциям транспортного уровня модели OSI, например функциям протокола TCP или UDP. Протоколы AAL при передаче пользовательского трафика работают только в конечных узлах сети, как и транспортные протоколы большинства технологий.

Каждый протокол уровня AAL обрабатывает пользовательский трафик определенного класса. На начальных этапах стандартизации каждому классу трафика соответствовал свой протокол AAL, который принимал в конечном узле пакеты от протокола верхнего уровня и заказывал с помощью соответствующего протокола нужные параметры трафика и качества обслуживания для данного виртуального канала. При развитии стандартов ATM такое однозначное соответствие между классами трафика и протоколами уровня AAL исчезло, и сегодня разре-

шается использовать для одного и того же класса трафика разные протоколы уровня AAL.

Уровень адаптации состоит из двух подуровней.

- **Подуровень сегментации и реассемблирования** (Segmentation And Reassembly, SAR) является нижним подуровнем AAL. Эта часть не зависит от типа протокола AAL (и, соответственно, от класса передаваемого трафика) и занимается разбиением (сегментацией) сообщения, принимаемого AAL от протокола верхнего уровня, на ячейки ATM, снабжением их соответствующим заголовком и передачей уровню ATM для отправки в сеть.
- **Подуровень конвергенции** (Convergence Sublayer, CS) — это верхний подуровень AAL. Этот подуровень зависит от класса передаваемого трафика. Протокол подуровня конвергенции решает такие задачи, как обеспечение временной синхронизации между передающим и принимающим узлами (для трафика, требующего такой синхронизации), контролем и возможным восстановлением битовых ошибок в пользовательской информации, контролем целостности передаваемого пакета компьютерного протокола (X.25, Frame Relay).

Протоколы AAL для выполнения своей работы используют служебную информацию, размещаемую в заголовках уровня AAL. После приема ячеек, пришедших по виртуальному каналу, подуровень SAR протокола AAL собирает посланное по сети исходное сообщение (в общем случае разбитое на несколько ячеек ATM) с помощью заголовков AAL, которые для коммутаторов ATM являются прозрачными, так как помещаются в 48-битном поле данных ячейки, как и полагается протоколу более высокого уровня. После сборки исходного сообщения протокол AAL проверяет служебные поля заголовка и концевика кадра AAL и на их основании принимает решение о корректности полученной информации.

Ни один из протоколов AAL при передаче пользовательских данных конечных узлов не занимается восстановлением потерянных или искаженных данных. Максимум, что делает протокол AAL, — уведомляет конечный узел о таком событии. Так сделано для ускорения работы коммутаторов сети ATM в расчете на то, что случаи потерь или искажений данных будут редкими. Восстановление потерянных данных (или игнорирование этого события) отводится протоколам верхних уровней, не входящим в стек протоколов технологии ATM.

Протокол AAL1 обычно обслуживает трафик класса A с постоянной битовой скоростью (CBR), который характерен, например, для цифрового видео и цифровой речи, и чувствителен к временным задержкам. Этот трафик передается в сетях ATM таким образом, чтобы эмулировать обычные выделенные цифровые линии. Заголовок AAL1 занимает в поле данных ячейки ATM 1 или 2 байта, оставляя для передачи пользовательских данных соответственно 47 или 46 байт. В заголовке один байт отводится для нумерации ячеек, чтобы приемная сторона могла судить о том, все или не все посланные ячейки дошли до нее. При отправке голосового трафика временна́я отметка каждого замера известна, так как они следуют друг за другом с интервалом в 125 мкс, поэтому при потере ячейки можно скорректировать временную привязку байтов следующей ячейки, сдвинув ее на 125 × 46 мкс. Потеря нескольких байтов замеров голоса не так страшна, так как на приемной стороне воспроизводящее оборудование сглаживает сигнал.

В задачи протокола AAL1 входит сглаживание неравномерности поступления ячеек данных в узел назначения.

Протокол AAL2 был разработан для передачи трафика класса B, но при развитии стандартов его исключили из стека протоколов ATM, и сегодня трафик класса B передается с помощью протокола AAL1, AAL3/4 или AAL5.

Протокол AAL3/4 обрабатывает пульсирующий трафик — обычно характерный для трафика локальных сетей — с переменной битовой скоростью (VBR). Этот трафик обрабатывается так, чтобы не допустить потерь ячеек, но ячейки могут задерживаться коммутатором. Протокол AAL3/4 выполняет сложную процедуру контроля ошибок при передаче ячеек, нумеруя каждую составляющую часть исходного сообщения и снабжая каждую ячейку контрольной суммой. Правда, при искажениях или потерях ячеек этот уровень не занимается их восстановлением, а просто отбрасывает все сообщение — то есть все оставшиеся ячейки, так как для компьютерного трафика или компрессированного голоса потеря части данных является фатальной ошибкой. Протокол AAL3/4 представляет собой результат слияния протоколов AAL3 и AAL4, которые обеспечивали поддержку трафика компьютерных сетей соответственно с установлением соединения и без установления соединения. Однако ввиду большой близости используемых форматов служебных заголовков и логики работы протоколы AAL3 и AAL4 были впоследствии объединены.

Протокол AAL5 является упрощенным вариантом протокола AAL4 и работает быстрее, так как вычисляет контрольную сумму не для каждой ячейки сообщения, а для всего исходного сообщения и помещает ее в последнюю ячейку сообщения. Первоначально протокол AAL5 разрабатывался для передачи кадров сетей Frame Relay, но теперь он чаще всего применяется для передачи любого компьютерного трафика. Протокол AAL5 может поддерживать различные параметры качества обслуживания, кроме тех, которые связаны с синхронизацией передающей и принимающей сторон. Поэтому он обычно используется для поддержки всех классов трафика, относящегося к передаче компьютерных данных, то есть классов C и D. Некоторые производители оборудования с помощью протокола AAL5 обслуживают трафик CBR, оставляя задачу синхронизации трафика протоколам верхнего уровня. Протокол AAL5 работает не только в конечных узлах, но и в коммутаторах сети ATM. Однако там он выполняет служебные функции, не связанные с передачей пользовательских данных. В коммутаторах ATM протокол AAL5 поддерживает служебные протоколы более высоких уровней, занимающиеся установлением коммутируемых виртуальных соединений.

Существует определенный интерфейс между приложением, которому требуется передать трафик через сеть ATM, и уровнем адаптации AAL. С помощью этого интерфейса приложение (протокол компьютерной сети, модуль оцифровывания голоса) заказывает требуемую услугу, определяя тип трафика, его параметры, а также параметры QoS. Технология ATM допускает два варианта определения параметров QoS: первый — непосредственное задание их каждым приложением, второй — назначение их по умолчанию в зависимости от типа трафика. Последний способ упрощает задачу разработчика приложения, так как в этом случае выбор максимальных значений задержек доставки ячеек и вариации задержек перекладывается на плечи администратора сети.

Самостоятельно обеспечить требуемые параметры трафика и QoS протоколы AAL не могут. Для выполнения соглашений трафик-контракта требуется согласованная работа коммутаторов сети вдоль всего виртуального соединения. Эта работа выполняется протоколом ATM, обеспечивающим передачу ячеек различных виртуальных соединений с заданным уровнем качества обслуживания.

Протокол ATM

Протокол ATM занимает в стеке протоколов ATM примерно то же место, что и протокол IP в стеке TCP/IP или протокол LAP-F в стеке протоколов технологии Frame Relay. Протокол ATM передает ячейки через коммутаторы при установленном и настроенном виртуальном соединении, то есть на основании готовых таблиц коммутации портов.

Протокол ATM выполняет коммутацию по **номеру виртуального соединения**, который в технологии ATM разбит на две части:

- **идентификатор виртуального пути** (Virtual Path Identifier, VPI);
- **идентификатор виртуального канала** (Virtual Channel Identifier, VCI).

Помимо решения этой основной задачи протокол ATM выполняет ряд функций по контролю за соблюдением трафик-контракта со стороны пользователя сети, маркировке ячеек-нарушителей, отбрасыванию ячеек-нарушителей при перегрузке сети, а также управлению потоком ячеек для повышения производительности сети (естественно, при соблюдении условий трафик-контракта для всех виртуальных соединений).

Формат ячеек протокола ATM представлен на рис. 21.11.

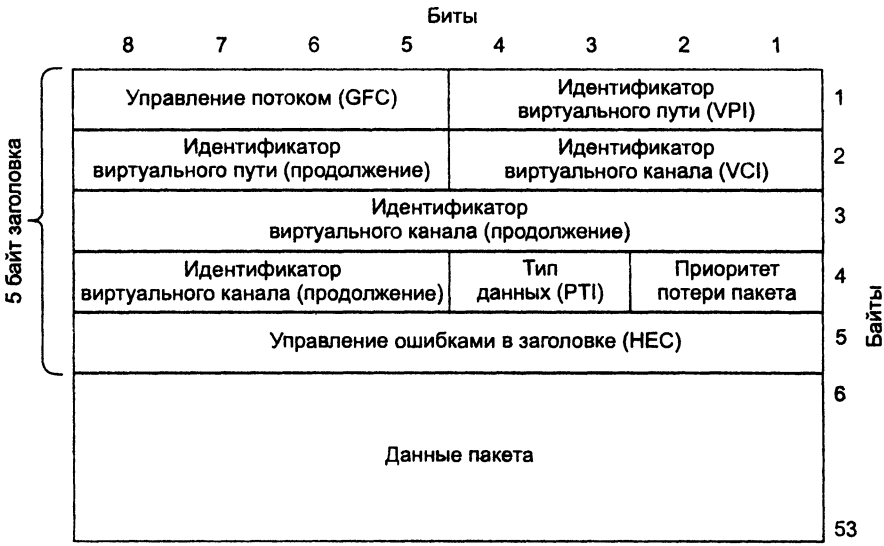


Рис. 21.11. Формат ATM-ячейки

Поле *обобщенного управления потоком* (Generic Flow Control, GFC) используется только при взаимодействии конечного узла и первого коммутатора сети. В настоящее время его точные функции не определены.

Поля *идентификатора виртуального пути* (VPI) и *идентификатора виртуального канала* (VCI) занимают соответственно 1 и 2 байта. Эти поля задают номер виртуального соединения, разделенный на старшую (VPI) и младшую (VCI) части.

Поле *идентификатора типа данных* (Payload Type Identifier, PTI) состоит из 3 бит и задает тип данных, переносимых ячейкой, — пользовательские или управляющие (например, управляющие установлением виртуального соединения). Кроме того, один бит этого поля используется для указания о перегрузке в сети. Это бит EFCI (Explicit Forward Congestion Identifier — прямой явный идентификатор перегрузки), который играет ту же роль, что и бит FECN в технологии Frame Relay, то есть передает информацию о перегрузке по направлению потока данных.

Поле *приоритета потери кадра* (Cell Loss Priority, CLP) играет в данной технологии ту же роль, что и поле DE в технологии Frame Relay — в нем коммутаторы ATM отмечают ячейки, которые нарушают соглашения о параметрах качества обслуживания, чтобы удалить их при перегрузках сети. Таким образом, ячейки с полем CLP = 0 являются для сети высокоприоритетными, а ячейки с полем CLP = 1 — низкоприоритетными.

Поле *управления ошибками в заголовке* (Header Error Control, HEC) содержит контрольную сумму, вычисленную для заголовка ячейки. Контрольная сумма вычисляется с помощью техники корректирующих кодов Хэмминга, поэтому она позволяет не только обнаруживать ошибки, но и исправлять все одиночные ошибки, а также некоторые двойные. Кроме того, поле HEC обеспечивает не только обнаружение и исправление ошибок в заголовке, но и нахождение границы начала кадра в потоке байтов кадров SDH, которые являются предпочтительным физическим уровнем технологии ATM, или же в потоке битов физического уровня, основанного на ячейках. Указателей, позволяющих в поле данных кадра STS-n (STM-n) технологии SONET/SDH обнаруживать границы ячеек ATM (подобных тем указателям, которые используются для определения, например, границ виртуальных контейнеров подканалов T1/E1), не существует. Поэтому коммутатор ATM вычисляет контрольную сумму для последовательности из пяти байтов, находящихся в поле данных кадра STM-n, и, если вычисленная контрольная сумма говорит о корректности заголовка ячейки ATM, первый байт становится границей ячейки. Если же это не так, то происходит сдвиг на один байт и операция продолжается. Таким образом, технология ATM выделяет асинхронный поток ячеек ATM в синхронных кадрах SDH или потоке битов физического уровня, основанного на ячейках.

Рассмотрим методы коммутации ячеек ATM на основе пары чисел VPI/VCI. Коммутаторы ATM могут работать в двух режимах.

- **Коммутация виртуального пути.** В этом режиме коммутатор выполняет продвижение ячейки только на основании значения поля VPI, а значение поля VCI он игнорирует. Обычно так работают магистральные коммутаторы территориальных сетей. Они доставляют ячейки из одной пользовательской сети в другую на основании *только старшей части номера виртуального канала*,

что соответствует идее агрегирования адресов. В результате один виртуальный путь соответствует целому набору виртуальных каналов, коммутируемых как единое целое.

- **Коммутация виртуального канала.** После доставки ячейки в локальную сеть ATM ее коммутаторы начинают коммутировать ячейки с учетом как поля VPI, так и поля VCI, но при этом им хватает для коммутации только младшей части номера виртуального соединения, так что фактически они работают с VCI, оставляя VPI без изменения. Этот режим и называется режимом коммутации виртуального канала.

Для создания коммутируемого виртуального канала в технологии ATM используются протоколы, не показанные на рис. 20.11. Подход здесь аналогичен подходу в сети ISDN — для установления соединения разработан отдельный **протокол Q.2931**, который весьма условно можно отнести к сетевому уровню. Этот протокол во многом похож на протоколы Q.931 и Q.933 (даже номером), но в него внесены, естественно, изменения, связанные с наличием нескольких классов трафика и дополнительных параметров качества обслуживания. Протокол Q.2931 опирается на достаточно сложный протокол канального уровня SSCOP, который обеспечивает надежную передачу пакетов Q.2931 в своих кадрах. В свою очередь, протокол SSCOP работает поверх протокола AAL5, который необходим для разбиения кадров SSCOP на ATM-ячейки и сборки этих ячеек в кадры при доставке кадра SSCOP в коммутатор назначения.

ПРИМЕЧАНИЕ

Протокол Q.2931 появился в стеке протоколов технологии ATM после принятия версии интерфейса UNI 3.1, а до этого в версии UNI 3.0 вместо него использовался протокол Q.93B. Из-за несовместимости протоколов Q.2931 и Q.93B версии пользовательского интерфейса UNI 3.0 и UNI 3.1 также несовместимы. Версия UNI 4.0 обратно совместима с UNI 3.1, так как основана на тех же служебных протоколах, что и версия UNI 3.1.

Виртуальные соединения, образованные с помощью протокола Q.2931, бывают симплексными (однонаправленными) и дуплексными.

Протокол Q.2931 позволяет также устанавливать двухточечные виртуальные соединения и виртуальные соединения с одним отправителем и несколькими получателями. Первый случай поддерживается во всех технологиях, основанных на виртуальных каналах, а второй характерен для технологии ATM и является аналогом групповой рассылки с одним ведущим (передающим) узлом. При установлении соединения с одним отправителем и несколькими получателями ведущим считается узел, который является инициатором этого соединения. Сначала этот узел устанавливает виртуальное соединение всего с одним узлом, а затем путем специального вызова добавляет к соединению по одному новому члену. Ведущий узел становится вершиной дерева соединения, а остальные узлы — листьями этого дерева. Сообщения, которые посылает ведущий узел, принимают все листья соединения, но сообщения, которые посылает какой-либо лист (если соединение дуплексное), принимает только ведущий узел.

Пакеты протокола Q.2931, предназначенные для установления коммутируемого виртуального канала, имеют те же названия и то же назначение, что и пакеты протокола Q.933, рассмотренные при изучении технологии Frame Relay, но структура их полей, естественно, другая.

Адрес конечного узла в коммутаторах ATM 20-байтный.

- При работе в публичных сетях используется адрес стандарта E.164. Адрес имеет гибкий формат и может делиться на части для обеспечения иерархической маршрутизации между сетями и подсетями. Он поддерживает больше уровней иерархии, чем IPv4-адрес, и похож в этом отношении на IPv6-адрес.
- Последние 6 байт адреса занимает поле идентификатора конечной системы (End System Identifier, ESI), которое имеет смысл MAC-адреса узла ATM, причем формат его также соответствует формату MAC-адреса.

ESI-адрес присваивается конечному узлу на предприятии-изготовителе в соответствии с правилами IEEE, то есть три первых байта содержат код предприятия, а остальные три — порядковый номер, за уникальность которого отвечает данное предприятие.

При работе в частных сетях ATM обычно применяются адреса близкого к E.164 формата с небольшим изменениями.

Конечный узел при подключении к коммутатору ATM выполняет так называемую процедуру регистрации. При этом конечный узел сообщает коммутатору свой ESI-адрес, а коммутатор сообщает конечному узлу старшую часть адреса, то есть номер сети, в которой работает узел.

Кроме адресной части пакет CALL SETUP протокола Q.2931, с помощью которого конечный узел запрашивает установление виртуального соединения, включает также части, описывающие параметры трафика и требования QoS. При поступлении такого пакета коммутатор должен проанализировать эти параметры и решить, достаточно ли у него свободных ресурсов для обслуживания нового виртуального соединения. Если да, то новое виртуальное соединение принимается, и коммутатор передает пакет CALL SETUP дальше в соответствии с адресом назначения и таблицей маршрутизации, а если нет, запрос отвергается.

Категории услуг протокола ATM и управление трафиком

Для поддержания требуемого качества обслуживания различных виртуальных соединений и рационального использования ресурсов в сети на уровне протокола ATM реализовано несколько служб, предоставляющих для обслуживания пользовательского трафика услуги различных категорий. Эти службы являются внутренними службами сети ATM, они предназначены для поддержания пользовательского трафика различных классов совместно с протоколами AAL. Но в отличие от протоколов AAL, которые работают в конечных узлах сети, данные службы распределены по всем коммутаторам сети. Услуги этих служб разбиты на категории, которые в общем соответствуют классам трафика, поступающим на вход уровня AAL конечного узла. Услуги уровня ATM заказываются конечным узлом

через интерфейс UNI с помощью протокола Q.2931 при установлении виртуального соединения. Как и при обращении к уровню AAL, при заказе услуги необходимо указать категорию услуги, а также параметры трафика и параметры QoS. Эти параметры берутся из аналогичных параметров уровня AAL или же определяются по умолчанию в зависимости от категории услуги.

Всего на уровне протокола ATM определено пять категорий услуг:

- **CBR** (Constant Bit Rate) — услуги для трафика с постоянной битовой скоростью;
- **rtVBR** (real-time Variable Bit Rate) — услуги для трафика с переменной битовой скоростью, требующего соблюдения средней скорости передачи данных и синхронизации источника и приемника;
- **nrtVBR** (non real-time Variable Bit Rate) — услуги для трафика с переменной битовой скоростью, требующего соблюдения средней скорости передачи данных и не требующего синхронизации источника и приемника;
- **ABR** (Available Bit Rate) — услуги для трафика с переменной битовой скоростью, требующего соблюдения некоторой минимальной скорости передачи данных и не требующего синхронизации источника и приемника;
- **UBR** (Unspecified Bit Rate) — услуги для трафика, не предъявляющего требований к скорости передачи данных и синхронизации источника и приемника.

Названия большинства категорий услуг совпадают с названием типов пользовательского трафика, для обслуживания которого они разработаны, но необходимо понимать, что сами службы уровня ATM и их услуги — это внутренние механизмы сети ATM, которые экранируются от приложения уровнем AAL.

Услуги категории CBR предназначены для поддержания трафика синхронных приложений — голосового, эмуляции цифровых выделенных каналов и т. п. Когда приложение устанавливает соединение категории CBR, оно заказывает пиковую скорость передачи ячеек (PCR), являющуюся максимальной скоростью, которую может поддерживать соединение без риска потерять ячейку, а также параметры QoS: величины максимальной задержки ячеек (CTD), вариации задержек ячеек (CDV) и максимальной доли потерянных ячеек CLR.

Затем данные передаются по этому соединению с запрошенной скоростью — не с большей и, в большинстве случаев, не с меньшей, хотя уменьшение скорости приложением возможно, например, при передаче компрессированного голоса при выполнении услуги категории CBR. Любые ячейки, передаваемые станцией с большей скоростью, контролируются первым коммутатором сети и помечаются признаком CLP = 1. При перегрузках сети они могут просто отбрасываться сетью. Ячейки, которые запаздывают и не укладываются в интервал, оговоренный параметром вариации задержки (CDV), также считаются мало значащими для приложения и отмечаются признаком низкого приоритета CLP = 1.

Для соединений CBR нет ограничений на некоторую дискретность заказа скорости PCR, как, например, в каналах T1/E1, где скорость должна быть кратна 64 Кбит/с.

По сравнению со службой CBR, *службы VBR* требуют более сложной процедуры заказа соединения между сетью и приложением. В дополнение к пиковой скорости PCR приложение VBR заказывает еще и два других параметра: длительно

поддерживаемую скорость — SCR, которая представляет собой среднюю скорость передачи данных, разрешенную приложению, а также максимальный размер пульсаций — MBS. Максимальный размер пульсаций измеряется в количестве ATM-ячеек. Пользователь может превышать скорость вплоть до величины PCR, но только на короткие периоды времени, в течение которых передается объем данных, не превышающий MBS. Этот период времени называется *терпимостью к пульсациям* (Burst Tolerance, BT). Сеть вычисляет этот период как производный от трех заданных значений PCR, SCR и MBS.

Если скорость PCR наблюдается в течение периода времени, большего чем BT, то ячейки помечаются как «нарушители» — устанавливается признак CLP = 1.

Для *услуг категории rtVBR* задаются и контролируются те же параметры QoS, что и для услуг категории CBR, а услуги категории nrtVBR ограничиваются поддержанием параметров трафика. Сеть также поддерживает для обеих категорий услуг VBR определенный максимальный уровень доли потерянных ячеек CLR, который либо задается явно при установлении соединения, либо назначается по умолчанию в зависимости от класса трафика.

Для контроля параметров трафика и QoS в технологии ATM применяется так называемый *обобщенный алгоритм контроля скорости ячеек*, который может проверять соблюдение пользователем и сетью таких параметров, как PCR, CDV, SCR, BT, CTD и CDV. Он работает по модифицированному алгоритму дырявого ведра, применяемому в технологии Frame Relay.

Для многих приложений, которые могут быть чрезвычайно «взрывными» в отношении интенсивности трафика, невозможно точно предсказать параметры трафика, оговариваемые при установлении соединения. Например, обработка транзакций или трафик двух взаимодействующих локальных сетей непредсказуемы по своей природе — изменения интенсивности трафика слишком велики, чтобы заключить с сетью какое-либо разумное соглашение.

В отличие от CBR и обеих служб VBR, *служба UBR* не поддерживает ни параметры трафика, ни параметры качества обслуживания. Служба UBR предлагает только доставку с максимальными усилиями (по возможности) без каких-либо гарантий. Разработанная специально для превышения полосы пропускания служба UBR представляет собой частичное решение для тех непредсказуемых «взрывных» приложений, которые не готовы согласиться с фиксацией параметров трафика.

Главными недостатками услуг UBR являются отсутствие механизмов управления потоком данных и неспособность принимать во внимание другие типы трафика. Несмотря на перегрузку сети, соединения UBR будут продолжать передачу данных. Коммутаторы сети могут буферизовать некоторые ячейки поступающего трафика, но в некоторый момент буферы переполняются, и ячейки теряются. А так как для соединений UBR не оговаривается никаких параметров трафика и QoS, то их ячейки отбрасываются в первую очередь.

Служба ABR подобно службе UBR предоставляет возможность превышения полосы пропускания, но благодаря технике управления трафиком при перегрузке сети она дает некоторые гарантии сохранности ячеек.

ABR — это первый тип служб уровня ATM, который действительно обеспечивает надежный транспорт для пульсирующего трафика за счет того, что может находить неиспользуемые интервалы в общем трафике сети и заполнять их своими ячейками, если другим категориям служб эти интервалы не нужны.

Как и в службах CBR и VBR, при установлении соединения категории ABR оговаривается значение пиковой скорости (PCR). Однако соглашение о пределах изменения задержек передачи ячеек или о параметрах пульсаций не заключается. Вместо этого сеть и конечный узел заключают соглашение о требуемой минимальной скорости передачи (MCR). Это гарантирует приложению, работающему в конечном узле, небольшую пропускную способность, обычно минимально необходимую для того, чтобы приложение работало. Конечный узел соглашается не передавать данные со скоростью, выше пиковой, то есть PCR, а сеть соглашается всегда обеспечивать минимальную скорость передачи ячеек.

Если при установлении соединения ABR не задаются значения максимальной и минимальной скоростей, то по умолчанию считается, что PCR совпадает со скоростью линии доступа станции к сети, а скорость MCR считается равной нулю.

Трафик соединения категории ABR получает гарантированное качество услуг в отношении доли потерянных ячеек и пропускной способности. Что касается задержек передачи ячеек, то хотя сеть и старается свести их к минимуму, но гарантий по этому параметру не дает. Следовательно, служба ABR предназначена не для приложений реального времени, а для приложений, в которых поток данных не очень чувствителен к задержкам в передаче.

При передаче трафика CBR, VBR и UBR явное управление перегрузками в сети отсутствует. Вместо этого используется механизм отбрасывания ячеек-нарушителей, а узлы, пользующиеся услугами CBR и VBR, стараются не нарушать условия контракта под угрозой потери ячеек, поэтому они обычно не задействуют дополнительную пропускную способность, даже если она в данный момент доступна в сети.

Служба ABR позволяет воспользоваться резервами пропускной способности сети, так как сообщает конечному узлу о наличии в данный момент избыточной пропускной способности с помощью механизма обратной связи. Этот же механизм может помочь службе ABR снизить скорость передачи данных конечным узлом в сеть (вплоть до минимального значения MCR), если сеть испытывает перегрузку.

Узел, пользующийся услугами ABR, должен периодически посылать в сеть наряду с ячейками данных специальные **служебные ячейки администрирования ресурсов** (Resource Management, RM). RM-ячейки, которые узел отправляет вдоль потока данных, называются ячейками прямого администрирования ресурсов (Forward Resource Management, FRM), а ячейки, которые идут в обратном по отношению к потоку данных направлении, называются ячейками обратного администрирования ресурсов (Backward Resource Management, BRM).

Существует несколько **петель обратной связи**. Самая простая петля обратной связи — между конечными станциями. При ее наличии коммутатор сети извещает конечную станцию о перегрузке с помощью специального флага в поле прямого

управления перегрузками (флаг EFCI) ячейки данных, переносимой протоколом ATM. Затем конечная станция посылает через сеть сообщение, содержащееся в специальной ячейке BRM, говоря станции-отправителю о необходимости снизить скорость посылки ячеек в сеть.

В этом способе конечная станция несет основную ответственность за управление потоком, а коммутаторы играют пассивную роль в петле обратной связи, только уведомляя станцию-отправитель о перегрузке.

Такой простой способ имеет несколько очевидных недостатков. Конечная станция не узнает из BRM-сообщения, насколько нужно снизить скорость передачи данных в сеть. Поэтому она просто снизит скорость до величины MCR, хотя, возможно, это и не обязательно. Кроме того, при большой протяженности сети коммутаторы вынуждены продолжать буферизовать данные все время, пока уведомление о перегрузке будет путешествовать по сети, а для глобальных сетей это время может быть достаточно большим, и буферы могут переполниться, так что требуемый эффект достигнут не будет.

Разработаны и более сложные схемы управления потоком, в которых коммутаторы играют более активную роль, а узел-отправитель точнее узнает о возможной в данный момент скорости отправки данных в сеть.

В первой схеме узел-источник посылает в FRM-ячейке явное значение скорости передачи данных в сеть, которую он хотел бы поддерживать в данное время. Каждый коммутатор, через который проходит по виртуальному пути это сообщение, может снизить запрашиваемую скорость до некоторой величины, которую он способен поддерживать в соответствии с имеющимися у него свободными ресурсами (или оставить запрашиваемую скорость без изменения). Узел назначения, получив FRM-ячейку, отправляет ее обратно в виде BRM-ячейки, причем он тоже может снизить запрашиваемую скорость. Получив ответ в BRM-ячейке, узел-источник точно узнает, какая скорость отправки ячеек в сеть для него доступна.

Во второй схеме каждый коммутатор сети может работать и как источник, и как приемник. Как узел-источник он может сам генерировать FRM-ячейки и отправлять их по имеющимся виртуальным каналам. Как узел-приемник он может отправлять на основе получаемых FRM-ячеек BRM-ячейки в обратном направлении. Такая схема является более быстродействующей и полезной в протяженных территориальных сетях.

Как видно из описания, служба ABR предназначена не только для прямого поддержания требований к обслуживанию конкретного виртуального соединения, но и для более рационального распределения ресурсов сети между ее абонентами, что в конечном итоге также приводит к повышению качества обслуживания всех абонентов сети.

Коммутаторы сети ATM используют различные механизмы для поддержания требуемого качества обслуживания. Помимо описанных в стандартах комитета ITU-T и форума ATM механизмов заключения соглашения на основе параметров трафика и QoS с последующим отбрасыванием ячеек, не удовлетворяющих условиям соглашения, практически все производители оборудования ATM реа-

лизуют в своих коммутаторах несколько очередей ячеек, обслуживаемых с разными приоритетами.

Стратегия приоритетного обслуживания трафика основана на категориях услуг каждого виртуального соединения. До принятия спецификации ABR в большинстве коммутаторов ATM была реализована простая одноуровневая схема обслуживания, которая давала трафику CBR первый приоритет, трафику VBR — второй, а трафику UBR — третий. При такой схеме комбинация услуг CBR и VBR может потенциально заморозить трафик, обслуживаемый другим классом служб. Такая схема не будет правильно работать с трафиком ABR, так как не обеспечит его требования к минимальной скорости передачи ячеек. Для обеспечения этого требования должна быть выделена некоторая гарантированная полоса пропускания.

Чтобы поддерживать службу ABR, коммутаторы ATM должны реализовать двухуровневую схему обслуживания, которая бы удовлетворяла требованиям CBR, VBR и ABR. По этой схеме коммутатор предоставляет некоторую часть своей пропускной способности каждому классу служб. Трафик CBR получает часть пропускной способности, необходимую для поддержания пиковой скорости (PCR), трафик VBR получает часть пропускной способности, необходимую для поддержания средней скорости (SCR), а трафик ABR получает часть пропускной способности, достаточную для обеспечения требования минимальной скорости ячеек (MCR). Это гарантирует, что каждое соединение будет работать без потерь ячеек и не будет доставлять ABR-ячейки за счет трафика CBR или VBR. На втором уровне этого алгоритма трафик CBR и VBR может забрать всю оставшуюся пропускную способность сети, если это необходимо, так как соединения ABR уже получили свою минимальную пропускную способность, которая им гарантировалась.

Отдельной задачей, которую нужно решать для поддержки корректной работы описанных служб и, соответственно, обеспечения заданного уровня QoS для всех классов трафика, является оптимизация работы сети ATM методами инжиниринга трафика (TE). Использование в сетях ATM (как и в сетях Frame Relay) техники виртуальных каналов создает хорошие предпосылки для решения задачи TE. В технологии Frame Relay каких-либо автоматизированных процедур для динамического выбора маршрутов следования виртуальных путей в целях обеспечения сбалансированной загрузки ресурсов сети пока не существует. Вся работа по оптимизации маршрутов должна выполняться предварительно, с помощью каких-либо внешних программных систем моделирования или оптимизации сети, а затем вручную реализовываться за счет установления постоянных виртуальных каналов в соответствии с выбранными маршрутами.

В сетях ATM в выборе маршрута для виртуальных каналов и путей может участвовать протокол маршрутизации PNNI, который для прокладки новых виртуальных каналов учитывает не только номинальную, но и доступную в данный момент пропускную способность.

Пример

Рассмотрим крупную сеть ATM энергетической компании I-Energy. По масштабу и решаемым задачам эта компания похожа на компанию ABC-Power, с которой мы познако-

мились в главе 11. Напомним, что основными особенностями энергетической компании являются:

- рассредоточенность объектов (электростанций и распределительных станций) по большой территории, так что расстояния между узлами сети могут составлять десятки, а иногда и сотни километров;
- эффективное управление такой компанией требует наличия сервисов передачи данных различного типа, в том числе телефонных сервисов, специализированных диспетчерских голосовых сервисов, сервисов передачи телеметрической информации в реальном масштабе времени, сервисов компьютерной системы управления предприятием.

Первые три вида сервисов являются чувствительными к задержкам, а третий манипулирует в основном эластичными данными. Как мы знаем, совместить трафик этих двух классов очень сложно для сети с коммутацией пакетов, решение такой задачи требует применения технологий, поддерживающих параметры QoS и резервирование полосы пропускания.

Вы, очевидно, помните, что компания ABC-Power не стала подвергать себя риску внедрения новейших технологий, обеспечивающих показатели QoS, а выбрала более традиционный вариант. Она развернула первичную сеть SDH с магистралью 622 Мбит/с и сетью доступа 155 Мбит/с. Затем образовавшиеся физические каналы (соединения SDH) были использованы для создания нескольких независимых наложенных сетей: компьютерной, телефонной, диспетчерской и телеметрии. При таком подходе каждая наложенная сеть (и, соответственно, ее сервисы) получает в свое распоряжение каналы фиксированной пропускной способности. Преимуществом здесь является то, что пропускная способность каждого канала всегда находится в полном распоряжении соответствующей сети, так что данные передаются с нужной скоростью и без задержек. Как мы знаем, это общие свойства метода коммутации каналов, который применяется в SDH. Помним мы и о «врожденных» недостатках этого метода, а именно невозможности перераспределить пропускную способность каналов, даже если на протяжении какого-то времени она не используется той сетью, которой была выделена.

Компания I-Energy решила пойти другим путем. Она не стала создавать первичную сеть SDH, а имеющимися в ее распоряжении «темными» оптическими волокнами соединила коммутаторы ATM, установленные на всех объектах предприятия. Сеть ATM получилась крупномасштабной: более 40 узлов, общая протяженность оптического волокна составила около 1500 км, а наиболее протяженный участок между соседними коммутаторами ATM оказался равен 126 км. Магистраль сети образована коммутаторами ATM с интерфейсами 622 и 155 Мбит/с, а коммутаторы сети доступа оснащены в основном интерфейсами 2 Мбит/с, к которым присоединены учережденческо-производственные АТС (УПАТС) корпоративной телефонной сети и IP-маршрутизаторы компьютерной сети. Устройства доступа диспетчерской сети и сети телеметрии подключаются к сети ATM не непосредственно, а через УПАТС.

Телефонная сеть I-Energy насчитывает шесть крупных офисных АТС емкостью от 500 до 2000 номеров. Кроме того, имеются 12 станций меньшей емкости и 38 совсем небольших станций, установленных в относительно «малонаселенных» подразделениях организации.

УПАТС подключаются к коммутаторам ATM каналами E1. Для связи УПАТС верхнего уровня используются каналы PVC с полносвязной топологией. Двенадцать АТС следующего уровня имеют прямые каналы связи с тремя-четырьмя АТС верхнего уровня иерархии, а каждая из 38 станций самого нижнего уровня соединена с тремя-четырьмя УПАТС второго уровня. Все связи между УПАТС второго и третьего уровней также образованы с помощью PVC. Для обеспечения требуемых параметров QoS

все каналы PVC между УПАТС предоставляют услуги CBR с соответствующей величиной резервируемой пропускной способности. Однако в отличие от соединений SDH каналы PVC сети ATM полностью потребляют выделенную им пропускную способность только в периоды времени, когда трафик CBR действительно имеет максимальную интенсивность. Все остальное время неиспользуемая трафиком CBR пропускная способность выделяется компьютерному трафику, который обслуживается как трафик UBR.

Выводы

Техника виртуальных каналов заключается в разделении операций маршрутизации и коммутации пакетов. Первый пакет таких сетей содержит адрес вызываемого абонента и прокладывает виртуальный путь в сети, настраивая промежуточные коммутаторы. Остальные пакеты проходят по виртуальному каналу в режиме коммутации на основании номера виртуального канала.

Достоинствами техники виртуальных каналов являются ускоренная коммутация пакетов по номеру виртуального канала, а также сокращение адресной части пакета, а значит, и избыточности заголовка. К недостаткам следует отнести невозможность распараллеливания потока данных между двумя абонентами по параллельным путям, а также неэффективность установления виртуального пути для кратковременных потоков данных.

Сети X.25 относятся к одной из наиболее старых и отработанных технологий глобальных сетей. Трехуровневый стек протоколов сетей X.25 хорошо зарекомендовал себя на ненадежных зашумленных линях связи, исправляя ошибки и управляя потоком данных на канальном и пакетном уровнях.

Сети Frame Relay работают на основе постоянных виртуальных каналов, а служба коммутируемых виртуальных каналов стала применяться на практике только недавно. Сети Frame Relay создавались специально для передачи пульсирующего компьютерного трафика, поэтому при резервировании пропускной способности указывается средняя скорость передачи (CIR) и согласованный объем пульсаций (Bc).

Технология ATM является дальнейшим развитием идей предварительного резервирования пропускной способности виртуального канала, реализованных в технологии Frame Relay. Технология ATM поддерживает основные типы трафика, существующие у абонентов разного типа: трафик CBR, характерный для телефонных сетей и сетей передачи изображения, VBR, характерный для компьютерных сетей, а также для передачи компрессированных голоса и изображения.

Для каждого типа трафика пользователь может заказать у сети значения нескольких параметров качества обслуживания — максимальную битовую скорость (PCR), среднюю битовую скорость (SCR), максимальную величину пульсаций (MBS), а также необходимость контроля временных соотношений между передатчиком и приемником, что важно для трафика, чувствительного к задержкам.

Технология ATM сама не определяет новые стандарты для физического уровня, а пользуется существующими. Основным стандартом для ATM является физический уровень каналов технологий SONET/SDH и PDH.

Вопросы и задания

1. С помощью каких параметров можно описать виртуальный канал?
2. Что нужно делать, если линия связи (физический канал), по которой проходит виртуальный канал, отказала?
3. Перечислите все основные этапы установления виртуального канала.

4. Может ли сеть X.25 работать без устройств PAD?
5. Что может произойти с трафиком, обслуживаемым с максимальными усилиями, если приоритетный трафик на входном интерфейсе Frame Relay не будет ограничен по средней интенсивности?
6. Каким образом пользователь может подключиться к встроенному устройству PAD через телефонную сеть, если он работает за терминалом, который не поддерживает процедуры вызова абонента через телефонную сеть автоматически?
7. Если у вашего предприятия появилась необходимость соединить многочисленные сети филиалов с центральной сетью и между собой, но в распоряжении имеются только выделенные аналоговые линии связи с установленными синхронными модемами 19,2 Кбит/с, то какую технологию вы выберете, X.25, Frame Relay или ATM? Обоснуйте факторы, которые повлияют на ваше решение.
8. Какую функцию алгоритма ведра маркеров не поддерживает алгоритм дырявого ведра?
9. Перечислите 5 классов трафика в соответствии с классификацией ATM.
10. Какую категорию услуг целесообразно выбрать для передачи голоса через сеть ATM?
11. Сколько виртуальных каналов должно быть установлено в каждом направлении между каждой парой коммутаторов ATM, если необходимо передавать три класса трафика с разными уровнями качества обслуживания?
12. Для какой из категории услуг сеть ATM явно управляет потоком данных? Почему для других категорий услуг управление потоком данных не используется?
13. Вы хотите вручную настроить постоянный виртуальный канал в двух корпоративных сетях ATM, соединенных публичной сетью ATM. Вы не хотите, чтобы ваши номера VCI зависели от номеров виртуальных каналов, используемых администратором в публичной сети ATM. Какой вид коммутации вы закажете у поставщика услуг публичной сети ATM?
14. Вы соединили две локальные сети удаленным мостом, работающим через постоянный виртуальный канал в сети Frame Relay. Сеанс протокола NetBEUI между компьютерами разных сетей часто разрывается, в то же время в том случае, когда компьютеры принадлежат одной локальной сети, их взаимодействие протекает без проблем. В чем может быть причина такой ситуации?
15. Сравните количество кадров, которое порождает обмен двумя TCP-сообщениями (посылка данных и получение квитанции) между двумя конечными хостами, соединенными одним промежуточным коммутатором для случаев, когда этот коммутатор является коммутатором X.25 и когда этот коммутатор является коммутатором Frame Relay?
16. В каком случае процент дошедших кадров через сеть Frame Relay до конечного узла будет выше: когда услуга заказана на основании параметров CIR, Bc и Be или когда услуга заказана на основании только параметров CIR и Bc (подразумевается, что значения параметров CIR и Bc в обоих случаях совпадают) при условии, что сеть Frame Relay недогружена, а узел-источник отправляет данные со скоростью, часто значительно превышающей CIR?

17. Пусть коммутатор Frame Relay и IP-маршрутизатор реализованы на основе одной и той же архитектуры и процессорах одного и того быстродействия. Будет ли коммутатор Frame Relay обладать большей производительностью, чем IP-маршрутизатор? Обоснуйте свое мнение.
18. Решите задачу инжиниринга трафика для сети ATM, изображенной на рис. 21.12. Вы должны обеспечить наиболее равномерную загрузку всех ресурсов сети для предложенной нагрузки, приведенной на рис. 21.13.

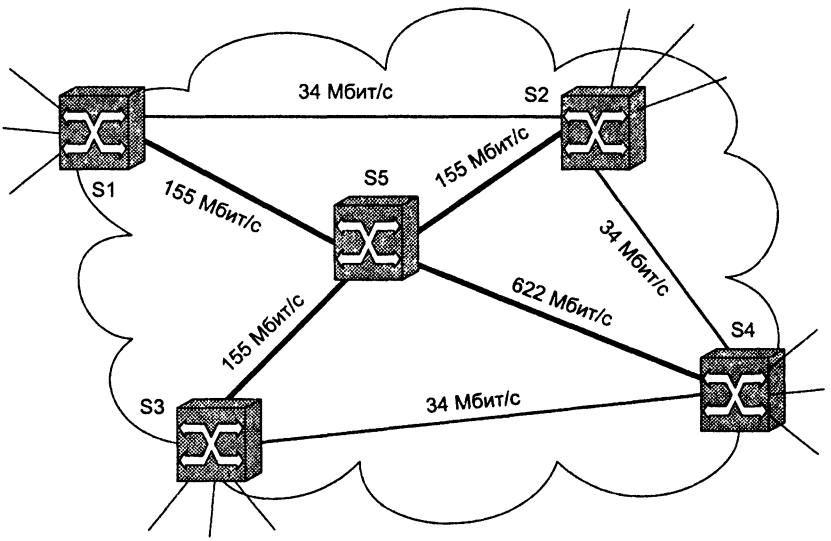


Рис. 21.12. Сеть ATM

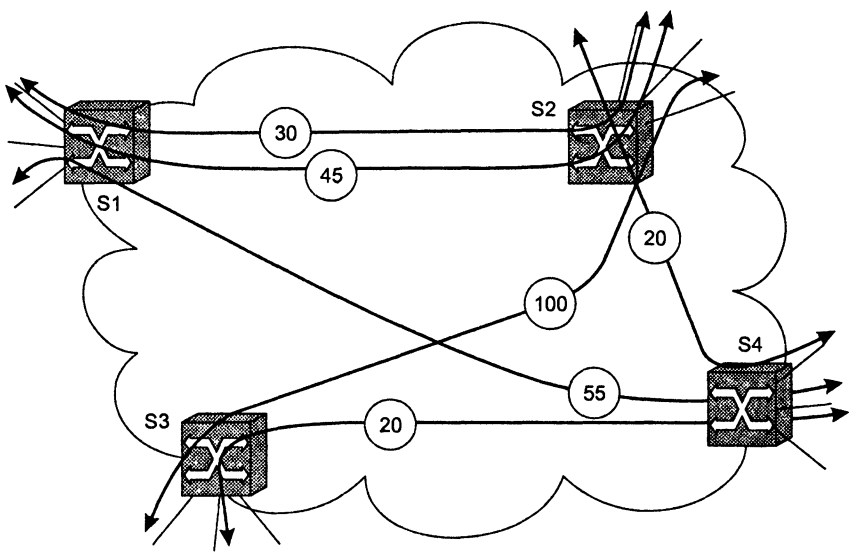


Рис. 21.13. Предложенная нагрузка

ГЛАВА 22 Технология IP в глобальных сетях

Революционный рост популярности Интернета привел к тому, что сегодня практически каждая глобальная сеть должна быть способна передавать трафик протокола IP. Это означает, что почти все современные глобальные сети являются составными IP-сетями, а отличия между ними заключаются в лежащих под уровнем IP технологиях.

Наиболее простой структурой обладают глобальные IP-сети, в которых IP-маршрутизаторы соединяются между собой непосредственно двухточечными линиями связи. В этом случае IP-маршрутизаторы объединяют вырожденные компонентные сети, которые не являются сетями в обычном понимании этого термина, так как состоят всего из двух узлов — интерфейсов соседних маршрутизаторов. Такие сети названы в книге «чистыми» IP-сетями, так как под уровнем IP-маршрутизаторов не существует уровня коммутаторов, и все задачи продвижения пакетов IP-маршрутизаторы решают самостоятельно.

В середине 90-х годов наиболее популярной структурой глобальной IP-сети стала многоуровневая структура, в которой под уровнем IP в качестве компонентных сетей используются сети ATM и FR. Применение на двух уровнях сетей с коммутацией пакетов, причем использующих разные принципы работы (дейтаграммный и виртуальных каналов) делают глобальную IP-сеть сложной и дорогой. Однако эти недостатки перевешиваются возможностями передачи мультимедийной информации и применения методов QoS и инжиниринга трафика для оптимизации загрузки ресурсов сети.

Новым словом в области интеграции IP с технологиями виртуальных каналов является технология MPLS. Эта технология занимает промежуточное место между уровнем IP и уровнем таких технологий, как ATM, FR или Ethernet, интегрируя их в единую эффективную технологию.

Глава завершается описанием систем сетевого управления, базирующихся на протоколе SNMP, который широко используется для управления не только IP-маршрутизаторами (для чего этот протокол был создан), но и телекоммуникационными устройствами разных типов, от SDH- или DWDM-мультиплексоров до телефонных коммутаторов.

«Чистые» глобальные IP-сети

Ключевые слова: «чистая» IP-сеть, пакетная сеть, работающая поверх SONET, протокол HDLC, асинхронный сбалансированный режим, кадр-команда и кадр-ответ, бит-стаффинг, протокол PPP, протокол управления линией связи, протокол управления сетью, многоканальный протокол PPP, протокол аутентификации по паролю, протокол аутентификации по квитированию вызова, порт с разделением каналов, ненумерованный интерфейс.

Структура глобальной IP-сети

Для предоставления качественных и разнообразных услуг большинство крупных глобальных сетей. особенно сетей коммерческих операторов связи, строится сегодня по четырехуровневой схеме (рис. 22.1).

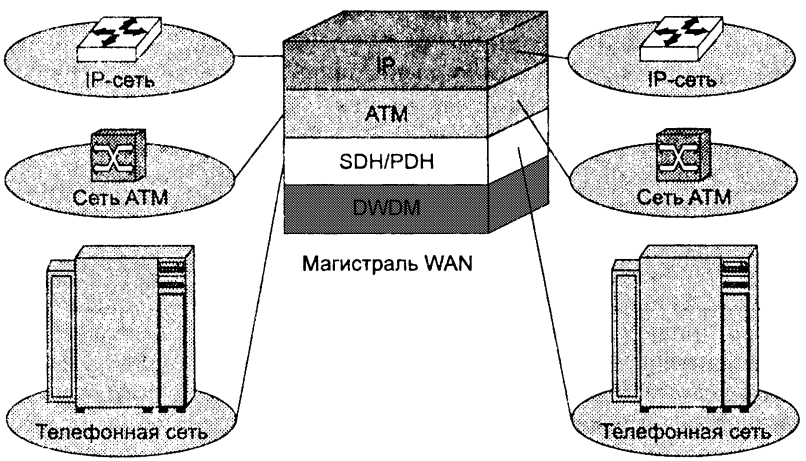


Рис. 22.1. Четырехуровневая структура современной глобальной сети

Два нижних уровня — это уровни *первичной сети*. На самом нижнем уровне первичной сети может работать наиболее скоростная на сегодняшний день технология DWDM, образующая спектральные каналы со скоростями 10 Гбит/с и выше. На следующем уровне, поверх DWDM, может применяться технология SDH (с сетью доступа PDH), с помощью которой пропускная способность спектральных каналов делится на более мелкие TDM-подканалы, связывающие интерфейсы коммутаторов пакетной сети (или телефонных коммутаторов).

На основе первичной сети оператор сети может достаточно быстро организовать постоянный цифровой канал между точками подключения оборудования следующего уровня — *наложенной сети* — пакетной или телефонной.

Верхний уровень в приведенной на рисунке модели глобальной сети образован IP-сетью.

ПРИМЕЧАНИЕ

Оба уровня первичной сети выполняют функции физического уровня модели OSI, поэтому не следует непосредственно сопоставлять уровни, показанные на рис. 22.1, с уровнями этой модели.

На рисунке показан наиболее масштабируемый на сегодня вариант построения первичной сети, включающий уровни DWDM и SDH. Такое построение сегодня характерно пока только для наиболее крупных территориальных сетей, покрывающих страны и континенты. Во многих менее масштабных магистралях уровень DWDM отсутствует, технология SDH тоже применяется не всегда — вместо нее может работать менее скоростная и отказоустойчивая, но более экономичная технология PDH.

В более простом случае первичная сеть вообще отсутствует, и под слоем IP может располагаться сеть ATM или Frame Relay, коммутаторы которой соединяются непосредственно кабельными или беспроводными линиями связи. Последнее решение, хотя и требует меньших начальных затрат, страдает от недостатка гибкости — чтобы подключить новое устройство, необходимо физически прокладывать новую линию связи, в то время как наличие разветвленной первичной сети дает возможность установить новый канал в сети путем перепрограммирования матрицы коммутации мультиплексоров и кросс-коннекторов DWDM/SDH.

Основным назначением уровня ATM на модели, изображенной на рис. 22.1, является создание инфраструктуры *постоянных виртуальных каналов* с *гарантированным качеством обслуживания*, соединяющих интерфейсы IP-маршрутизаторов. Для каждого класса IP-трафика в сети ATM образуется отдельный виртуальный канал, обеспечивающий требуемые для трафика параметры QoS — среднюю скорость, величину пульсаций, уровень задержек, уровень потерь. Применение ATM под уровнем IP позволяет не только обеспечить для пользовательского трафика необходимое качество обслуживания, но и дает возможность оператору решить задачу инжиниринга трафика предоставив сбалансированную загрузку всех линий связи первичной сети.

Уровень IP, освобожденный в представленной модели от проблем обеспечения параметров QoS, выполняет свои классические функции — образует составную сеть и предоставляет IP-услуги конечным пользователям, передающим по глобальной сети свой IP-трафик транзитом или взаимодействующим по IP с Интернетом.

Несмотря на сложность многослойной структуры, подобные сети получили большое распространение и для крупных операторов комплексных услуг являются на сегодня фактическим стандартом глобальной сети, с помощью которой можно оказывать комплексные услуги — IP, ATM, классической телефонии, а также услуги по предоставлению цифровых каналов в аренду. Пользователи услуг, отличных от IP, взаимодействуют непосредственно с тем уровнем сети оператора, который им нужен, то есть с ATM, телефонной сетью, SDH или DWDM.

Однако долгое время IP-сети не имели такой сложной многослойной структуры. Классическая IP-сеть состояла из маршрутизаторов, непосредственно соединенных каналами связи. Эти сети не поддерживали QoS, так как трафик приложе-

ний 80-х годов не был чувствительным к задержкам. После появления многослойных глобальных IP-сетей возникла потребность различать эти два вида сетей, поэтому для классических IP-сетей мы будем использовать термин «чистая» IP-сеть.

«Чистая» IP-сеть отличается от многослойной тем, что под уровнем IP нет другой сети с коммутацией пакетов, такой как ATM или Frame Relay, и IP-маршрутизаторы связываются между собой выделенными каналами (физическими или соединениями PDH/SDH/DWDM).

Структура «чистой» IP-сети представлена на рис. 22.2.

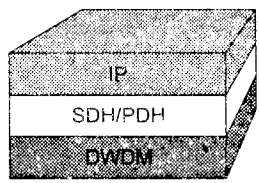


Рис. 22.2. Структура «чистой» IP-сети

В такой сети цифровые каналы по-прежнему образуются инфраструктурой двух нижних уровней, а этими каналами непосредственно пользуются интерфейсы IP-маршрутизаторов без какого-либо промежуточного уровня. В том случае, когда IP-маршрутизатор использует каналы, образованные в сети SDH/SONET, вариант IP-сети получил название **пакетной сети, работающей поверх SONET** (Packet Over SONET, POS).

«Чистая» IP-сеть может успешно применяться для передачи чувствительного к задержкам трафика современных приложений в двух случаях:

- если IP-сеть работает в режиме низкой нагрузки, поэтому сервисы всех типов не страдают от эффекта очередей, так что сеть не требует применения методов поддержки параметров QoS;
- если слой IP обеспечивает поддержку параметров QoS собственными средствами за счет применения механизмов IntServ или DiffServ.

Для того чтобы маршрутизаторы в модели «чистой» IP-сети могли использовать цифровые каналы, на этих каналах должен работать какой-либо протокол канального уровня. Существует несколько протоколов канального уровня, специально разработанных для двухточечных соединений глобальных сетей. В эти протоколы встроены процедуры, полезные при работе в глобальных сетях:

- *управление потоком данных*;
- *взаимная аутентификация удаленных устройств*, часто необходимая для защиты сети от «ложного» маршрутизатора, перехватывающего и перенаправляющего трафик для его прослушивания;
- *согласование параметров обмена данными на канальном и сетевом уровнях* — при удаленном взаимодействии, когда два устройства расположены в разных

городах, перед началом обмена часто необходимо автоматически согласовывать такие параметры, например, как MTU.

Из набора существующих двухточечных протоколов протокол IP сегодня использует два: HDLC и PPP. Существует также устаревший протокол SLIP (Serial Line Internet Protocol — межсетевой протокол для последовательного канала), который долгое время был основным протоколом удаленного доступа индивидуальных клиентов к IP-сети через телефонную сеть. Однако сегодня он полностью вытеснен протоколом PPP.

Помимо уже упомянутых протоколов, в глобальных сетях на выделенных каналах IP-маршрутизаторы нередко используют какой-либо из высокоскоростных вариантов Ethernet: Fast Ethernet, Gigabit Ethernet или 10G Ethernet. Все варианты Ethernet не поддерживают перечисленных выше процедур, полезных для глобальных сетей, но чашу весов в данном случае перевешивает популярность этой технологии в локальных сетях.

Протоколы семейства HDLC

Протокол HDLC (High-level Data Link Control — высокоуровневое управление линией связи) представляет целое семейство протоколов, образующих канальный уровень для следующих сетей и устройств:

- LAP-B — сетей X.25,
- LAP-D — сетей ISDN,
- LAP-M — асинхронно-синхронных модемов,
- LAP-F — сетей Frame Relay.

Первое, что мы отметим по поводу протокола HDLC, — его сложность. Он может работать в нескольких, весьма отличающихся друг от друга режимах, поддерживает не только двухточечные соединения, но и соединения с одним источником и несколькими приемниками, он также предусматривает различные функциональные роли взаимодействующих станций. Читатели, особо интересующиеся данным протоколом, всегда смогут найти его достаточно полное описание в доступных и широко известных книгах, таких, например, как [1 и 46], а мы в этой книге ограничимся рассмотрением функций протокола HDLC для двухточечных соединений, так как только такие соединения используются «чистыми» IP-сетями.

HDLC поддерживает три режима логического соединения, отличающиеся ролями взаимодействующих устройств. Мы рассмотрим один из них — **асинхронный сбалансированный режим** (Asynchronous Balance Mode, ABM) по той же причине — именно этот режим используют IP-маршрутизаторы. В режиме ABM оба устройства равноправны и обмениваются кадрами, которые делятся на кадры-**команды** и кадры-**ответы**.

Формат HDLC-кадра содержит следующие поля (рис. 22.3).

- *Открывающий и закрывающий флаги*, представляющие собой коды 01111110, обрамляют HDLC-кадр, позволяя приемнику определить начало и конец кадра.

Благодаря этим флагам в HDLC-кадре отсутствует поле длины кадра. Но возникает новая задача: как отличить байты данных, имеющие значение 01111110, от байта флага. На помощь приходит техника **бит-стаффинга**. Бит-стаффинг работает только во время передачи поля данных кадра. Если передатчик обнаруживает, что передано подряд пять единиц, то он автоматически вставляет дополнительный ноль в последовательность передаваемых битов (даже если после этих пяти единиц и так идет ноль). Поэтому последовательность 01111110 никогда не появится в поле данных кадра. Аналогичная схема работает в приемнике и выполняет обратную функцию. Когда после пяти единиц обнаруживается ноль, он автоматически удаляется из поля данных кадра.

- *Поле данных* предназначено для передачи по сети пакетов протоколов вышележащих уровней — сетевых протоколов IP, IPX, AppleTalk, DECnet, в редких случаях — прикладных протоколов, когда те вкладывают свои сообщения непосредственно в кадры канального уровня. Поле данных может отсутствовать в управляющих кадрах и некоторых ненумерованных кадрах.
- *Адрес* (1 или 2 байта) выполняет свою обычную функцию идентификации одного из нескольких возможных устройств только в конфигурациях «с одним источником и несколькими приемниками». В двухточечной конфигурации адрес HDLC используется для обозначения направления передачи кадра — из сети к устройству пользователя или наоборот. Очевидно, что эта функция адреса имеет смысл только при передаче кадра через UNI.
- *Поле управления* занимает 1 или 2 байта. Его структура зависит от *типа* передаваемого кадра. Тип кадра определяется первыми битами поля управления: 0 — *информационный*, 01 — *управляющий*, 11 — *ненумерованный* тип. Бит P/F входит в структуру поля управления кадров всех типов, он по-разному используется в кадрах-командах и в кадрах-ответах. Например, станция-приемник при получении от станции-передатчика кадра-команды с установленным битом P немедленно должна ответить управляющим кадром-ответом, установив бит F.

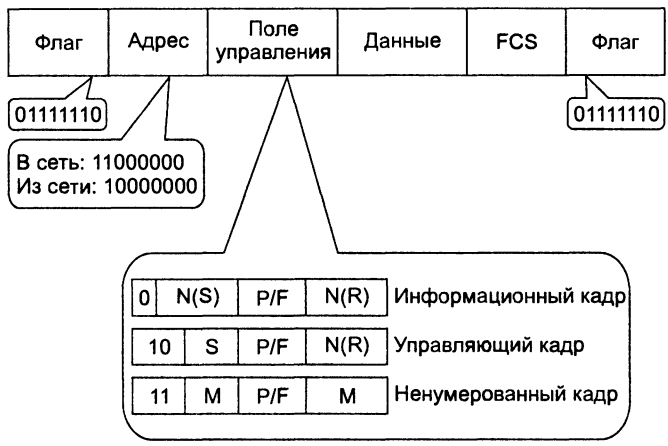


Рис. 22.3. HDLC-кадр

Остановимся подробнее на структуре и назначении разных типов кадров.

- *Ненумерованные кадры* предназначены для установления и разрыва логического соединения, а также информирования об ошибках. Поле М ненумерованных кадров содержит коды, определяющие тип команд, которыми пользуются два узла на этапе установления соединения, например:
  - SABME (Set Asynchronous Balanced Mode Extended — установить асинхронный сбалансированный расширенный режим) — эта команда является запросом на установление соединения (расширенный режим означает использование двухбайтовых полей управления для кадров остальных двух типов);
  - UA (Unnumbered Acknowledgment — ненумерованная положительная квитанция) — подтверждение установления или разрыва соединения;
  - REST (Resetting connection — сброс соединения) — запрос на разрыв соединения.
- *Управляющие кадры* предназначены для передачи команд и ответов в контексте установленного логического соединения, в том числе для передачи запросов на повторную передачу искаженных информационных блоков:
  - REJ (Reject — отказ) — часто используется как отрицательная квитанция приемника;
  - RNR (Receiver Not Ready — приемник не готов) — может использоваться для замедления потока кадров, поступающих на приемник;
  - RR (Receiver Ready — приемник готов) — часто используется как положительная квитанция.
- *Информационные кадры* предназначены для передачи данных пользователя. В процессе передачи информационных блоков осуществляется их нумерация в соответствии с алгоритмом скользящего окна.

После установления соединения данные и положительные квитанции начинают передаваться в информационных кадрах. Логический канал HDLC является *дуплексным*, так что информационные кадры, а значит, и положительные квитанции могут передаваться в обоих направлениях. Если же потока информационных кадров в обратном направлении нет или же нужно передать отрицательную квитанцию, то используются управляющие кадры.

При работе HDLC для обеспечения надежности передачи используется *скользящее окно* размером в 7 кадров (при размере поля управления 1 байт) или 127 кадров (при размере поля управления 2 байта). Для поддержания алгоритма окна в информационных кадрах станции-отправителя отводится два поля:

- N(S) — номер отправляемого кадра;
- N(R) — номер кадра, который станция ожидает получить от своего партнера по диалогу.

Предположим для определенности, что станция А отправила станции В информационный кадр с некоторыми значениями $N_A(S)$ и $N_A(R)$. Если в ответ на этот кадр приходит кадр от станции В, в котором номер посланного этой станцией кадра $N_B(S)$ совпадает с номером ожидаемого станцией А кадра $N_A(R)$, то пере-

дача считается корректной. Если станция A принимает кадр-ответ, в котором номер отправленного кадра $N_B(S)$ неравен номеру ожидаемого $N_A(R)$, то станция A этот кадр отбрасывает и посылает отрицательную квитанцию REJ (отказ) с номером $N_A(R)$. Приняв отрицательную квитанцию, станция B обязана повторить передачу кадра с номером $N_A(R)$, а также *всех* кадров с большими номерами, которые она уже успела отослать, пользуясь механизмом скользящего окна, к примеру, 120 кадров.

Когда поток данных от приемника к передатчику отсутствует, управляющая команда RR с установленным полем N(R) используется как положительная квитанция. Если механизм окна не справляется с регулировкой потока кадров, то применяется управляющая команда RNR, которая требует от передатчика полной приостановки передачи до получения команды RR.

Протокол PPP

Протокол PPP (Point-to-Point Protocol) является стандартным протоколом Интернета. Протокол PPP так же, как и HDLC, представляет собой целое семейство протоколов, в которое, в частности, входят:

- **протокол управления линией связи** (Link Control Protocol, LCP);
- **протокол управления сетью** (Network Control Protocol, NCP);
- **многоканальный протокол PPP** (Multi Link PPP, MLPPP);
- **протокол аутентификации по паролю** (Password Authentication Protocol, PAP);
- **протокол аутентификации по квитированию вызова** (Challenge Handshake Authentication Protocol, CHAP).

ПРИМЕЧАНИЕ

При разработке протокола PPP за основу был взят формат HDLC-кадров и дополнен несколькими полями. Эти дополнительные поля протокола PPP вложены в поле данных HDLC-кадра. Позже были разработаны стандарты, описывающие вложение PPP-кадра в кадры Frame Relay и других протоколов глобальных сетей. Хотя PPP и работает с HDLC-кадром, он не поддерживает, подобно протоколу HDLC, процедуры надежной передачи кадров и управления их потоком.

Особенностью протокола PPP, отличающей его от других протоколов канального уровня, является *сложная переговорная процедура* принятия параметров соединения. Стороны обмениваются различными параметрами, такими как качество линии, размер кадров, тип протокола аутентификации и тип инкапсулируемых протоколов сетевого уровня.

В корпоративной сети конечные системы часто отличаются размерами буферов для временного хранения пакетов, ограничениями на размер пакета, списком поддерживаемых протоколов сетевого уровня. Физическая линия, связывающая конечные устройства, может варьироваться от низкоскоростной аналоговой линии до высокоскоростной цифровой линии с различными уровнями качества обслуживания.

Протокол, в соответствии с которым принимаются параметры соединения, называется *протоколом управления линией связи* LCP. Чтобы справиться со всеми возможными ситуациями, в протоколе PPP имеется набор стандартных установок, действующих по умолчанию и учитывающих все стандартные конфигурации. При установлении соединения два взаимодействующих устройства для нахождения взаимопонимания пытаются сначала использовать эти установки. Каждый конечный узел описывает свои возможности и требования. Затем на основании этой информации принимаются параметры соединения, устраивающие обе стороны. Переговорная процедура протоколов может и не завершиться соглашением о каком-нибудь параметре. Если, например, один узел предлагает в качестве MTU значение 1000 байт, а другой отвергает это предложение и в свою очередь предлагает значение 1500 байт, которое отвергается первым узлом, то по истечении тайм-аута переговорная процедура может закончиться безрезультатно.

Одним из важных параметров PPP-соединения является *режим аутентификации*. Для целей аутентификации PPP предлагает по умолчанию *протокол аутентификации по паролю* (PAP), передающий пароль по линии связи в открытом виде, или *протокол аутентификации по квитированию вызова* (CHAP), не передающий пароль по линии связи и поэтому обеспечивающий более высокий уровень безопасности сети. Пользователям также разрешается добавлять новые алгоритмы аутентификации. Кроме того, пользователи могут влиять на выбор алгоритмов компрессии заголовка и данных.

Многопротокольная поддержка — способность протокола PPP поддерживать несколько протоколов сетевого уровня — обусловила распространение PPP как стандарта де-факто. Внутри одного PPP-соединения могут передаваться потоки данных различных сетевых протоколов, включая IP, Novell IPX, AppleTalk, DECnet, XNS, Banyan VINES и OSI, а также данные протоколов канального уровня локальной сети.

Каждый протокол сетевого уровня конфигурируется отдельно с помощью соответствующего *протокола управления сетью* (NCP). Под конфигурированием понимается, во-первых, констатация того факта, что данный протокол будет использоваться в текущем сеансе PPP, а во-вторых, переговорное согласование некоторых параметров протокола. Больше всего параметров устанавливается для протокола IP, включая IP-адреса взаимодействующих узлов, IP-адреса DNS-серверов, признак компрессии заголовка IP-пакета и т. д. Для каждого протокола конфигурирования протокола верхнего уровня, помимо общего названия NCP, используется особое название, построенное путем добавления аббревиатуры CP (Control Protocol) к имени конфигурируемого протокола, например, для IP — это протокол IPCP, для IPX — IPXCP и т. п.

Расширяемость протокола. Под этим свойством PPP понимается как возможность включения новых протоколов в стек PPP, так и возможность использования собственных протоколов пользователей вместо рекомендуемых в PPP по умолчанию. Это позволяет наилучшим образом настроить PPP для каждой конкретной ситуации.

Одной из привлекательных свойств протокола PPP является его способность использования нескольких физических линий связи для образования одного ло-

гического канала, то есть агрегирование каналов (см. также об агрегировании каналов в главе 16). Эту возможность реализует *многоканальный протокол MLPPP*.

Использование выделенных линий IP-маршрутизаторами

Схема использования выделенной линии маршрутизатором показана на рис. 22.4. Для соединения порта маршрутизатора с выделенной линией необходимо использовать устройство DCE соответствующего типа. Это устройство требуется для согласования физического интерфейса маршрутизатора с интерфейсом физического уровня, используемого выделенной линией, например V.35 с T1.

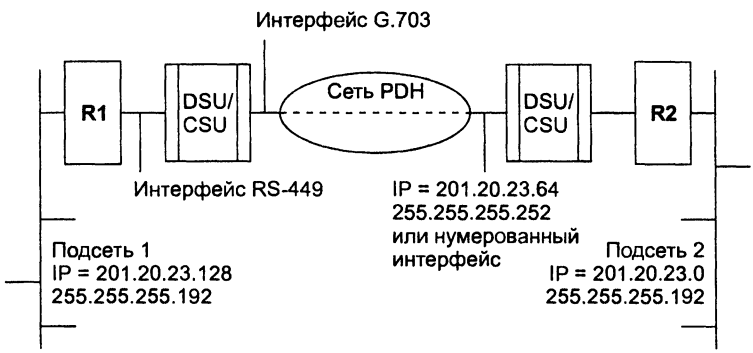


Рис. 22.4. Соединение IP-сетей с помощью выделенной линии

Если выделенная линия является аналоговой, то устройством DCE будет модем, а если цифровой — то аппаратура DSU/CSU.

Порт маршрутизатора может включать встроенное устройство DCE. Например, маршрутизатор, рассчитанный на работу с каналом SDH, обычно имеет встроенный порт с интерфейсом SDH определенной скорости STM-N.

Встроенные порты PDH/SDH могут как поддерживать, так и не поддерживать внутреннюю структуру кадров этих технологий. В том случае, когда порт различает подкадры, из которых состоит кадр, например отдельные тайм-слоты кадра E1 или отдельные виртуальные контейнеры VC-12 (2 Мбит/с), входящие в кадр STM-1, и порт может использовать их как отдельные физические подканалы, то говорят, что это **порт с разделением каналов**. Каждому такому каналу присваивается отдельный IP-адрес. В противном случае порт целиком рассматривается как один физический канал с одним IP-адресом.

В качестве примера на рис. 22.4 выбрано соединение двух маршрутизаторов через цифровой канал E1, установленный в сети PDH. Маршрутизатор использует для подключения к каналу устройство DSU/CSU с внутренним интерфейсом RS-449 и внешним интерфейсом G.703, который определен в качестве интерфейса доступа к каналам PDH.

Маршрутизаторы после подключения к выделенной линии и локальной сети необходимо конфигурировать. Выделенный канал является отдельной IP-подсетью,

как и локальные подсети 1 и 2, которые он соединяет. Этой подсети можно также дать некоторый IP-адрес из диапазона адресов, которым распоряжается администратор составной сети. В приведенном примере выделенному каналу присвоен адрес подсети 201.20.23.64, состоящей из 2-х узлов, что определяется маской 255.255.255.252.

Интерфейсам маршрутизаторов, связанных выделенной линией, можно и не присваивать IP-адрес — такой интерфейс маршрутизатора называется **ненумерованным**. Действительно, отсылая пакеты протокола маршрутизации (RIP или OSPF) по выделенному каналу, маршрутизаторы непременно их получат. Протокол ARP на выделенном канале не используется, так как аппаратные адреса на выделенном канале не имеют практического смысла.

Функционирование IP-сети поверх сети ATM/FR

Ключевые слова: многослойная сеть IP/ATM, оверлейная (наложенная) сеть, QoS, виртуальный канал, топология виртуальных каналов, ATM-коммутаторы, логический интерфейс, подинтерфейс, конфигурирование интерфейса, классы трафика, протокол Q.2931.

Взаимодействие слоев IP и ATM

При построении IP-сети поверх сети ATM/FR между слоем каналов и слоем IP работает сеть ATM или Frame Relay (FR). Так как скорости, на которых работает сеть Frame Relay, как правило, не превышают 2 Мбит/с[1], а уровень задержек и их вариаций не входит в число параметров QoS, поддерживаемых данной технологией, то чаще всего в качестве промежуточного слоя магистрали применяется технология ATM.

Взаимодействие слоя IP со слоем ATM иллюстрирует рис. 22.5.

В сети ATM проложено шесть постоянных виртуальных каналов, соединяющих порты IP-маршрутизаторов. Каждый порт маршрутизатора должен поддерживать технологию ATM в качестве конечного узла. После того как виртуальные каналы установлены, маршрутизаторы могут пользоваться ими как физическими, посылая данные порту соседнего (по отношению к виртуальному каналу) маршрутизатора.

В сети ATM образуется сеть виртуальных каналов с собственной топологией. Топология виртуальных каналов, соответствующая сети, представленной на рис. 22.5, показана на рис. 22.6. Сеть ATM прозрачна для IP-маршрутизаторов,

[1] Последние версии стандартов Frame Relay подняли предел скорости до 622 Мбит/с.

они ничего не знают о физических связях между портами ATM-коммутаторов. IP-сеть является наложенной (оверлейной) по отношению к сети ATM.

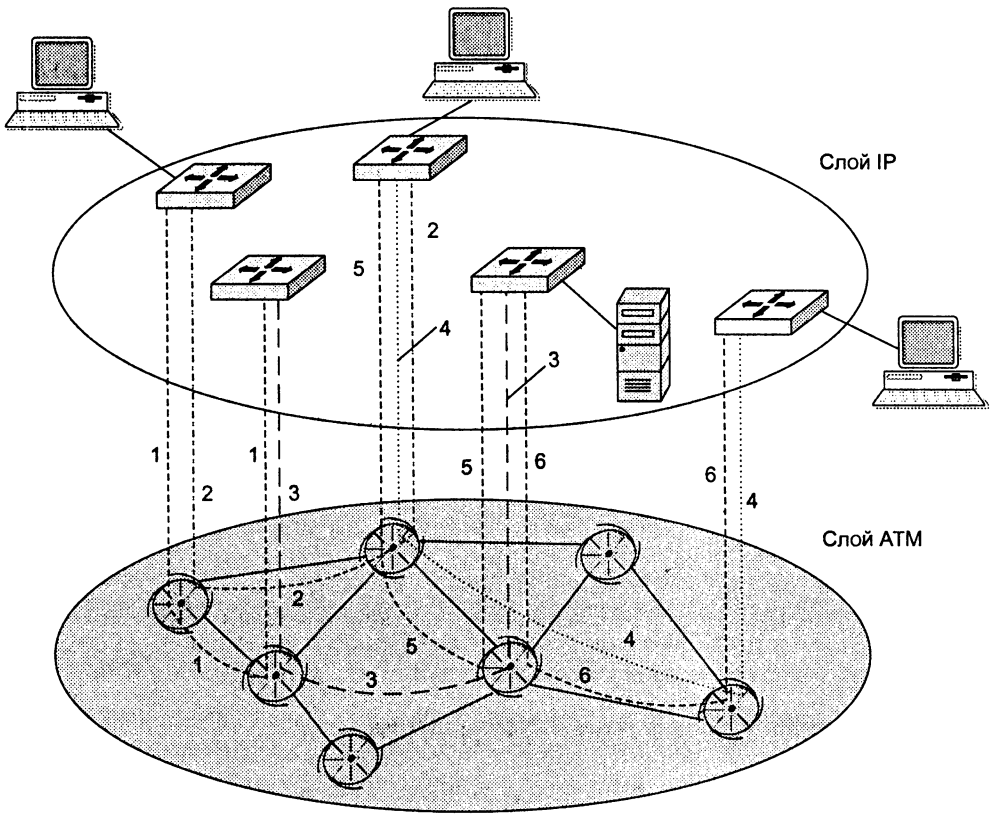


Рис. 22.5. Взаимодействие слоев IP и ATM

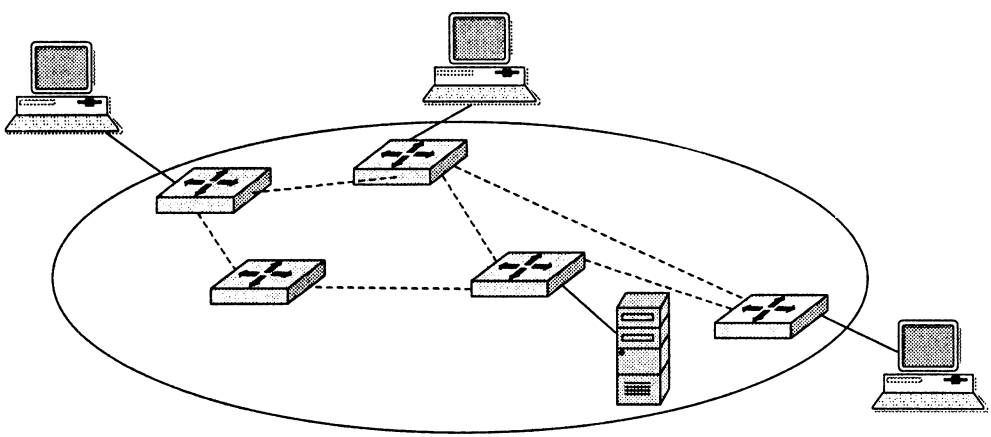


Рис. 22.6. Топология связей между маршрутизаторами

Конфигурирование интерфейсов маршрутизаторов

Для того чтобы протокол IP мог корректно работать, ему необходимо знать соответствие между IP-адресами соседей и адресами виртуальных каналов ATM, с помощью которых достижим соответствующий IP-адрес, то есть уметь отображать сетевые адреса на аппаратные, роль которых в данном случае играют адреса виртуальных каналов ATM. Другими словами, протоколу IP необходим некий вариант протокола ARP. Поскольку сеть ATM не поддерживает широковещательных запросов, таблица соответствия адресов не может быть создана автоматически. Администратор IP-сети должен вручную выполнить конфигурирование каждого интерфейса маршрутизатора, задав таблицу соответствия для всех номеров виртуальных каналов, исходящих и входящих в этот интерфейс. При этом физический интерфейс может быть представлен в виде набора логических интерфейсов (или *подинтерфейсов*), имеющих IP-адреса.

Например, в маршрутизаторах компании Cisco Systems конфигурирование логического интерфейса, соответствующего виртуальному каналу с адресом VPI/VCI, равным 0/36, выглядит следующим образом:

```
pvc 0/36
protocol ip 10.2 1.1
```

После выполнения этих команд маршрутизатор будет знать, что в случае необходимости пересылки пакета по адресу 10.2.1.1 ему нужно будет разбить пакет на последовательность ATM-ячеек (с помощью функции SAR интерфейса ATM) и отправить их все по постоянному виртуальному каналу с адресом 0/36.

Если многослойная сеть IP/ATM должна передавать трафик различных классов с соблюдением параметров QoS для каждого класса, то соседние маршрутизаторы должны быть связаны несколькими виртуальными каналами, по одному для каждого класса. Маршрутизатору должна быть задана политика классификации пакетов, позволяющая отнести передаваемый пакет к определенному классу. Пакеты каждого класса направляются на соответствующий виртуальный канал, который обеспечивает трафику требуемые параметры QoS. Однако предварительно необходимо провести инжиниринг трафика для сети ATM, определив оптимальные пути прохождения трафика и соответствующим образом проложив виртуальные каналы. Результатом такой работы будет соблюдение требований к средним скоростям потоков, а коэффициент загрузки каждого интерфейса ATM-коммутаторов не превысит определенной пороговой величины, гарантирующей каждому классу трафика приемлемый уровень задержек.

Оверлейная IP-сеть может также использовать режим коммутируемых виртуальных каналов (SVC) для передачи IP-трафика. Этот режим подходит для неустойчивых потоков, которые существуют в течение небольших периодов времени. Создавать для таких потоков инфраструктуру постоянных виртуальных каналов невыгодно, так как большую часть времени они будут простаивать. Для того чтобы маршрутизаторы могли использовать режим SVC, необходимо задать отображение IP-адресов, но не на номера виртуальных каналов, а на ATM-адреса конечных точек сети ATM, то есть ATM-адреса интерфейсов маршрутизатора.

Эта функция разрешения адресов, как и в предыдущем случае, выполняется администратором вручную. Один из вариантов задания такого отображения для маршрутизаторов Cisco имеет следующий вид:

```
Map-list a
ip 10.1 0 3 atm-nsap 33.3333.33.333333 3333.3333.3333.3333.3333.3333.3333.33
```

Если задано такое соответствие адресов, маршрутизатор для отправки пакета по IP-адресу 10.1.0.3 предварительно устанавливает с помощью протокола Q.2931 коммутируемый виртуальный канал SVC с ATM-адресом 33.3333.33.333333.3333.3333.3333.3333.3333.3333.3333.33, а затем, автоматически получив от этого протокола адрес VPI/VCI, отправляет по нему ячейки, на которые разбит исходный пакет. Интерфейс соседнего маршрутизатора, получив все ячейки, объединяет их в исходный пакет и передает наверх протоколу IP.

Если по коммутируемому виртуальному каналу нужно передавать трафик с некоторыми требуемыми параметрами QoS, то эти параметры передаются протоколу Q.2931, который выбирает маршрут для виртуального канала с их учетом.

Функционирование IP-сети поверх сети ATM очень популярно у операторов связи, которые предоставляют услуги с заключением соглашения об уровне обслуживания (SLA).

Многопротокольная коммутация с помощью меток

Ключевые слова: многопротокольная коммутация с помощью меток, коммутирующий по меткам маршрутизатор, IP-коммутация, коммутация на основе тегов, таблица продвижения, протокол распределения меток, путь коммутации по меткам, пограничный коммутирующий по меткам маршрутизатор, стек меток, технологии MPLS IGP, MPLS TE и MPLS VPN, ускоренная MPLS-коммутация, класс эквивалентного продвижения, строгий и свободный TE-туннели, протокол RSVP TE.

Технология **многопротокольной коммутации с помощью меток** (MultiProtocol Label Switching, MPLS) считается сегодня многими специалистами одной из самых перспективных транспортных технологий. Эта технология объединяет технику виртуальных каналов с функциональностью стека TCP/IP.

Объединение происходит за счет того, что одно и то же сетевое устройство, называемое **коммутирующим по меткам маршрутизатором** (Label Switch Router, LSR), выполняет функции как IP-маршрутизатора, так и коммутатора виртуальных каналов. Причем это не механическое объединение двух устройств, а тесная интеграция, когда функции каждого устройства дополняют друг друга и используются совместно.

Многопротокольность технологии MPLS состоит в том, что она может использовать протоколы маршрутизации не только стека TCP/IP, но и любого другого стека, например IPX/SPX. В этом случае вместо протоколов маршрутизации

RIP IP, OSPF и IS-IS будут применяться протоколы RIP IPX или NLSP, а общая архитектура LSR останется такой же.

Совмещение коммутации и маршрутизации в одном устройстве

Впервые идея объединения маршрутизации и коммутации в одном устройстве была реализована в середине 90-х годов компанией *Ipsilon*, которая начала выпускать комбинированные устройства IP/ATM. В этих устройствах была реализована новая технология **IP-коммутации** (IP switching), которая решала проблему неэффективной передачи кратковременных потоков данных с помощью описанных ранее коммутируемых виртуальных каналов. Для того чтобы пакеты кратковременных потоков передавались в сети ATM-коммутаторов без длительной предварительной процедуры установления виртуального канала, компания Ipsilon предложила встроить во все ATM-коммутаторы блоки IP-маршрутизации. Эти блоки строили таблицы маршрутизации с помощью стандартных для стека TCP/IP протоколов RIP, OSPF и IS-IS.

Передача IP-пакета, принадлежащего кратковременному потоку, осуществлялась по сети Ipsilon следующим образом. Пакет поступал из узла-отправителя на комбинированное устройство IP/ATM, которое разбивало этот пакет на ATM-ячейки. Каждая ячейка затем в соответствии с технологией IP-коммутации передавалась от одного устройства IP/ATM к другому, а затем к адресату по маршруту, определяемому обычными таблицами маршрутизации IP, хранящимися в этих устройствах.

При этом стандартное для технологии ATM виртуальное соединение между устройствами IP/ATM не устанавливалось, за счет чего передача кратковременных IP-потоков существенно ускорялась. Долговременные потоки передавались устройствами IP/ATM традиционным для ATM способом — с помощью виртуальных каналов. Так как топология сети является одной и той же как для протоколов IP, так и для протоколов ATM, это позволяет использовать один и тот же протокол маршрутизации для обеих частей комбинированного устройства.

Для реализации своей технологии компания Ipsilon встроила в устройства IP/ATM фирменные протоколы, ответственные за распознавание длительности потоков данных и установление виртуальных каналов для долговременных потоков. Эти протоколы были оформлены в виде проектов стандартов Интернета, но стандартами Интернета не стали.

Технология IP-коммутации была разработана для сетей операторов связи. Эти сети принимают на границе с другими сетями IP-трафик и ускоренно передают его через свою магистраль. Важным обстоятельством здесь является то, что одни ISP могут применять эту технологию *независимо* от других ISP, оставаясь для внешнего мира операторами обычной IP-сети.

Технология IP-коммутации была сразу замечена операторами связи и стала достаточно популярной. Инициативу Ipsilon развила компания *Cisco Systems*, создав собственную **технологию коммутации на основе тегов** (tag switching), которая явилась значительным шагом вперед на пути объединения протоколов IP с тех-

никой виртуальных соединений, однако она, так же как и IP-коммутация, не стала стандартной технологией.

На основе этих фирменных технологий рабочая группа IETF, состоящая из специалистов различных компаний, создала в конце 90-х годов технологию MPLS.

LSR и таблица продвижения данных

Главный принцип технологий-предшественниц в MPLS был сохранен.

> Протоколы маршрутизации используются для определения топологии сети, а для продвижения данных внутри границ сети одного поставщика услуг применяется техника виртуальных каналов.

Принцип объединения протоколов различных технологий иллюстрируют рис. 22.7 и 22.8. На первом из них показана упрощенная архитектура стандартного IP-маршрутизатора, на втором — архитектура комбинированного устройства LSR, поддерживающего технологию MPLS.

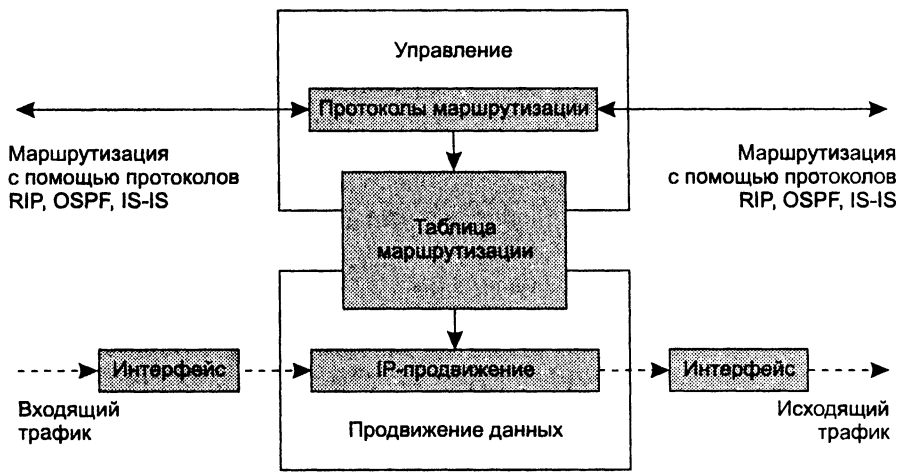


Рис. 22.7. Архитектура IP-маршрутизатора

Так как устройство LSR выполняет все функции IP-маршрутизатора, оно содержит все блоки последнего, а для поддержки функций MPLS в LSR включен ряд дополнительных блоков, относящихся как к управлению, так и к продвижению данных.

В качестве примера можно указать на *блок продвижения по меткам*, который передает IP-пакет не на основе IP-адреса назначения, а на основе поля метки. При принятии решения о выборе следующего хопа блок продвижения по меткам использует *таблицу коммутации*, которая в стандарте MPLS носит название **таблицы продвижения**. Таблица продвижения в технологии MPLS похожа на ана-

логичные таблицы других технологий, основанных на технике виртуальных каналов (табл. 22.1).

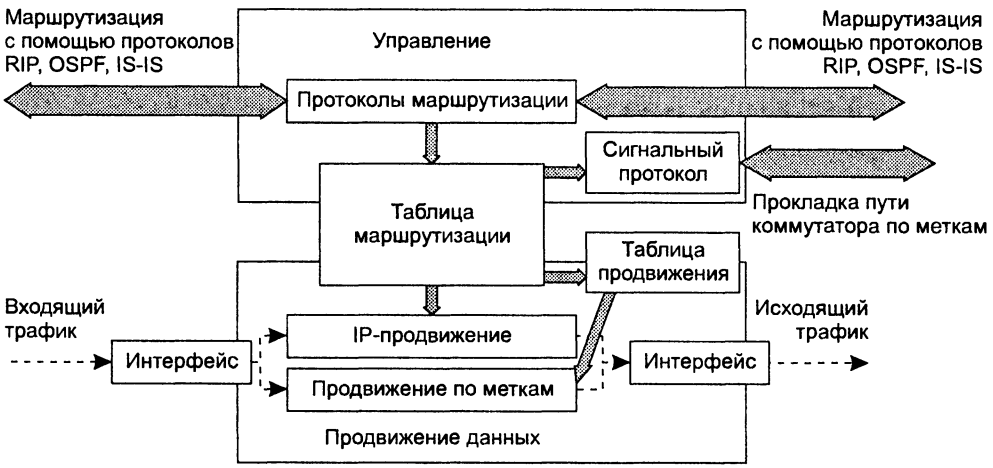


Рис. 22.8. Архитектура LSR

Таблица 22.1. Пример таблицы продвижения в технологии MPLS

Входной интерфейс	Метка	Следующий хоп	Действия
S0	245	S1	256
S0	27	S2	45
...	...	...	...

Внимательный читатель заметил, наверно, небольшое отличие данной таблицы от той обобщенной таблицы коммутации, представленной на рис. 21.1. Действительно, вместо поля выходного интерфейса здесь поле следующего хопа, а вместо поля выходной метки — поле действий. В большинстве случаев обработки MPLS-кадров эти поля используются точно таким же образом, как соответствующие им поля обобщенной таблицы коммутации. То есть значение поля следующего хопа является значением интерфейса, на который нужно передать кадр, а значение поля действий — новым значением метки. Однако в некоторых случаях эти поля используются в других целях, о чем будет сказано позже.

Эти таблицы для каждого устройства LSR формируются *сигнальным протоколом*, который в MPLS имеет название **протокол распределения меток** (Label Distribution Protocol, LDP). Он функционально аналогичен сигнальным протоколам технологий ATM и Frame Relay.

Формируя таблицы продвижения на LSR, протокол LDP уже знакомым нам образом прокладывает через сеть виртуальные маршруты, которые в технологии MPLS называют **путями коммутации по меткам** (Label Switching Path, LSP).

Пути коммутации по меткам

На рис. 22.9 показана MPLS-сеть, взаимодействующая с несколькими IP-сетями.

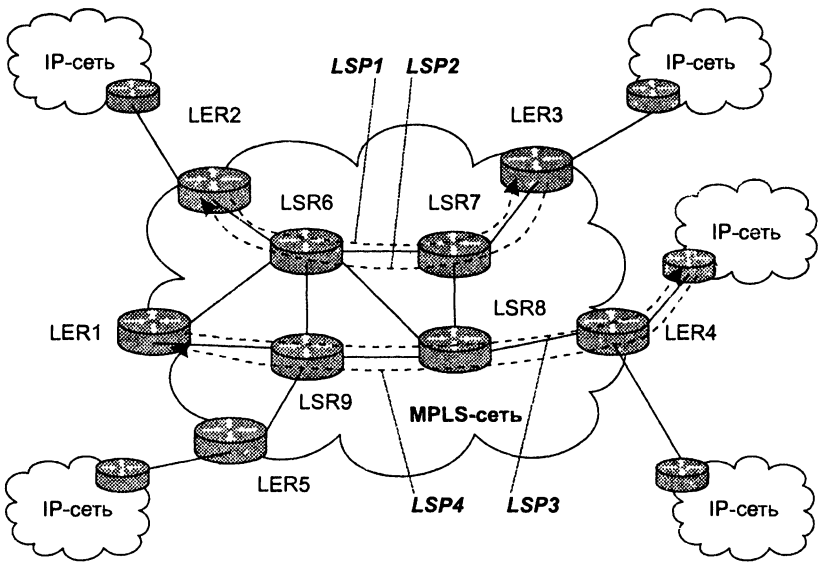


Рис. 22.9. MPLS-сеть

Пограничные устройства LSR в технологии MPLS имеют специальное название — **пограничные коммутирующие по меткам маршрутизаторы** (Label switch Edge Routers, LER).

Устройство LER, являясь функционально более сложным, принимает трафик от других сетей в форме стандартных IP-пакетов, а затем добавляет к нему метку и направляет вдоль соответствующего пути к выходному устройству LER через несколько промежуточных устройств LSR. При этом пакет продвигается не на основе IP-адреса назначения, а на основе метки.

Как и в других технологиях, использующих технику виртуальных каналов, метка имеет локальное значение в пределах каждого устройства LER и LSR, то есть при передаче пакета с входного интерфейса на выходной выполняется смена значения метки.

Пути LSP прокладываются в MPLS не при появлении долговременного потока данных, как в технологии IP-коммутации, а *предварительно* в соответствии с топологией межсетевых соединений.

LSP представляет собой *однонаправленный* виртуальный канал, поэтому для передачи трафика между двумя устройствами LER нужно установить, по крайней мере, два пути коммутации по меткам — по одному в каждом направлении. На рис. 22.9 показаны две пары путей коммутации по меткам, соединяющие устройства LER1 и LER3, а также LER1 и LER4. Очевидно, что для обеспечения связи

между всеми сетями этого недостаточно. Устройства LER с помощью путей коммутации по меткам должны образовывать полносвязную структуру, которая имеет место в реальных MPLS-сетях и на рисунке не показана только в виду громоздкости ее графического представления.

Выходное устройство LER, выполнив удаление метки, передает пакет в следующую сеть уже в стандартной IP-форме. Таким образом, технология MPLS остается прозрачной для остальных IP-сетей.

Обычно в MPLS-сетях используется усовершенствованный по сравнению с описанным алгоритм обработки пакетов. Усовершенствование заключается в том, что удаление метки выполняет не последнее на пути устройство, а *предпоследнее*. Действительно, после того как предпоследнее устройство определит на основе значения метки следующий хоп, метка в MPLS-кадре уже не нужна, так как последнее устройство, то есть выходное устройство LER, будет продвигать пакет на основе значения IP-адреса. Это небольшое изменение алгоритма продвижения кадра позволяет сэкономить одну операцию над MPLS-кадром. В противном случае последнее вдоль пути устройство должно было бы удалить метку, а уже затем выполнить просмотр таблицы IP-маршрутизации.

Заголовок MPLS и технологии канального уровня

Заголовок MPLS состоит из нескольких полей (рис. 22.10).

- *Метка* (20 бит) используется для выбора соответствующего пути коммутации по меткам.
- *Время жизни* (TTL). Это поле, занимающее 8 бит, дублирует аналогичное поле IP-пакета. Это необходимо для того, чтобы устройства LSR могли отбрасывать «заблудившиеся» пакеты только на основании информации, содержащейся в заголовке MPLS, не обращаясь к заголовку IP.
- *Класс услуги* (Class of Service, CoS). Поле CoS, занимающее 3 бита, первоначально было зарезервировано для развития технологии, но в последнее время используется в основном для указания класса трафика, требующего определенного показателя QoS.
- *Признак дна стека меток* — S (1 бит).

Концепцию стека меток мы рассмотрим в следующем разделе, а пока для пояснения механизма взаимодействия MPLS с технологиями канального уровня рассмотрим ситуацию, когда заголовок MPLS включает только одну метку.

Как видно из рисунка, технология MPLS поддерживает несколько типов кадров: PPP, Ethernet, Frame Relay и ATM. Это не означает, что под слоем MPLS работает какая-либо из перечисленных технологий, например Ethernet. Это означает только то, что в технологии MPLS используются форматы кадров этих технологий для помещения в них пакета сетевого уровня, которым почти всегда сегодня является IP-пакет.

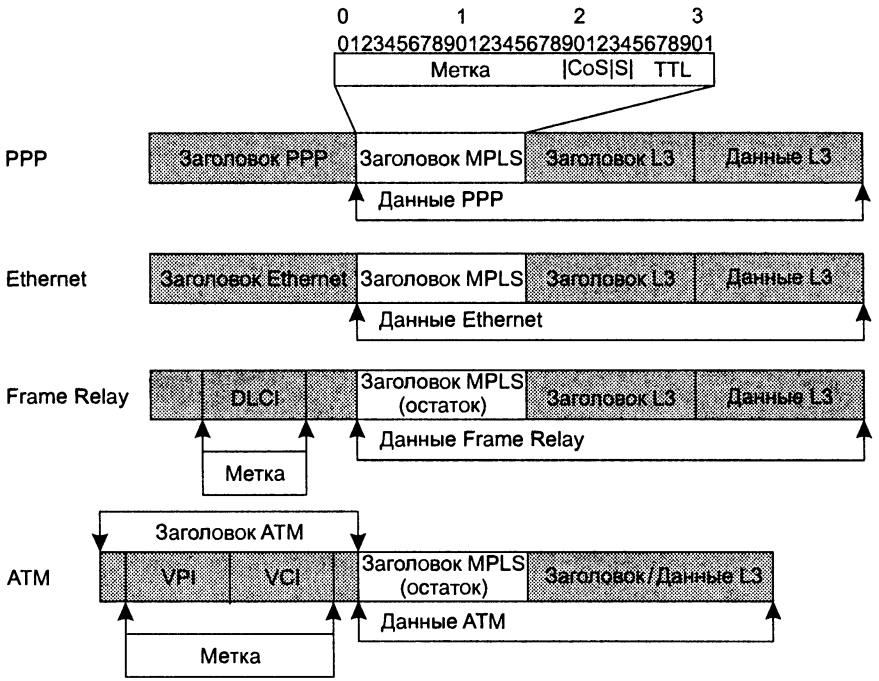


Рис. 22.10. Форматы заголовков нескольких разновидностей технологии MPLS

Продвижение кадра в MPLS-сети происходит на основе метки MPLS и техники LSP, а не на основе адресной информации и техники той технологии, формат кадра которой MPLS использует. Таким образом, если в MPLS применяется кадр Ethernet, то MAC-адреса источника и приемника хотя и присутствуют в соответствующих полях кадра Ethernet, но для продвижения кадров не задействуются[1].

В кадрах PPP, Ethernet и Frame Relay заголовок MPLS помещается между оригинальным заголовком и заголовком пакета 3-го уровня. С ячейками ATM технология MPLS поступает по-другому: она пользуется имеющимися полями VPI/VCI в заголовках этих ячеек для меток виртуальных соединений. Поля VPI/VCI используются только для хранения поля метки, остальная часть заголовка MPLS с полями CoS, S и TTL размещается в поле данных ATM-ячеек и не используется при передаче ячеек ATM-коммутаторами, поддерживающими MPLS.

Далее для определенности при рассмотрении примеров мы будем подразумевать, что используется формат кадров MPLS/PPP.

---

[1] Исключением является вариант технологии MPLS/Ethernet на разделяемой среде. Однако такой вариант может встретиться только в локальной сети, а технология MPLS разработана для глобальных сетей, поэтому мы не будем рассматривать этот редкий случай.

Стек меток

Наличие **стека меток** является одним из оригинальных свойств MPLS. Концепция стека меток является развитием концепции двухуровневой адресации виртуальных путей с помощью меток VPI/VCI, принятой в ATM.

Стек меток позволяет создавать систему агрегированных путей LSP с любым количеством уровней иерархии. Для поддержки этой функции MPLS-кадр, который перемещается вдоль иерархически организованного пути, должен включать столько заголовков MPLS, сколько уровней иерархии имеет путь. Напомним, что заголовок MPLS каждого уровня имеет собственный набор полей: метка, CoS, TTL и S. Последовательность заголовков организована как стек, так что всегда имеется метка, находящаяся на вершине стека, и метка, находящаяся на дне стека, при этом последняя сопровождается признаком S = 1. Над метками выполняются следующие операции, задаваемые в поле действий таблицы продвижения:

- *Push* — поместить метку в стек. В случае пустого стека эта операция означает простое присвоение метки пакету. Если же в стеке уже имеются метки, в результате этой операции новая метка сдвигает «старые» в глубь стека, сама оказываясь на вершине.
- *Swap* — заменить текущую метку новой.
- *Pop* — выталкивание (удаление) верхней метки, в результате чего все остальные метки стека поднимаются на один уровень.

Продвижение MPLS-кадра всегда происходит на основе метки, находящейся в данный момент на вершине стека. Рассмотрим сначала продвижение MPLS-кадра по *одноуровневому* пути в MPLS-сети, показанной на рис. 22.11.

Сеть состоит из трех MPLS-доменов. На рисунке показан путь LSP1, проложенный в домене 1, и путь LSP2 в домене 2. LSP1 соединяет устройства LER1 и LER2, проходя через устройства LSR1, LSR2 и LSR3. Пусть начальной меткой пути LSP1 является метка 256, которая была присвоена пакету пограничным устройством LER1. На основании этой метки пакет поступает на устройство LSR1, которое по своей таблице продвижения определяет новое значение метки пакета (272) и переправляет его на вход LSR2. Устройство LSR2, действуя аналогично, присваивает пакету новое значение метки (132) и передает его на вход LSR3. Устройство LSR3, будучи предпоследним устройством в пути LSP1, выполняет операцию Pop и удаляет метку из стека. Устройство LER2 продвигает пакет уже на основании IP-адреса.

На рисунке также показан путь LSP2 в домене 2. Он соединяет устройства LER3 и LER4, проходя через устройства LSR4, LSR5 и LSR6, и определяется последовательностью меток 188, 112, 101.

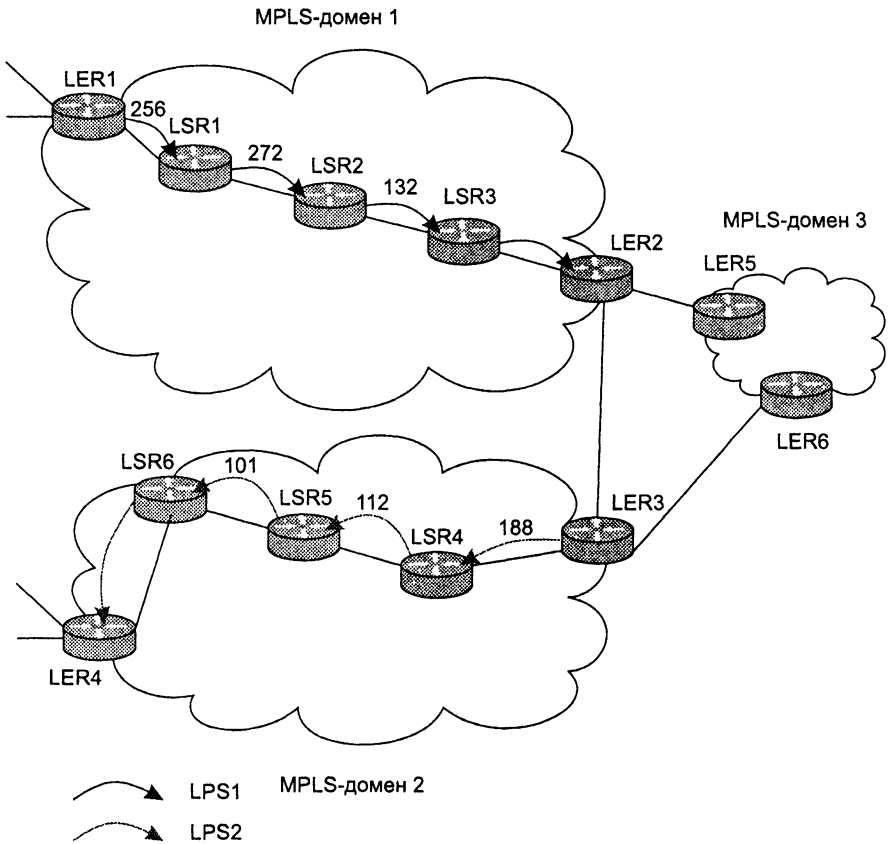


Рис. 22.11. Пути LSP1 и LSP2, проложенные в доменах 1 и 2 MPLS-сети

Для того чтобы IP-пакеты могли передаваться на основе техники MPLS не только внутри каждого домена, но и между доменами (например, между устройствами LER1 и LER4), существует два принципиально разных решения.

- Первое решение состоит в том, что между LER1 и LER4 устанавливается один *одноуровневый* путь коммутации по меткам, соединяющий пути LSP1 и LSP2 (которые в этом случае становятся одним путем). Это простое, на первый взгляд, решение плохо работает в том случае, когда MPLS-домены принадлежат разным поставщикам услуг, не позволяя им действовать независимо друг от друга.
- Вторым более перспективным решением является применение *многоуровневого* подхода к соединению двух MPLS-доменов, принадлежащих, возможно, разным поставщикам услуг.

В данном примере в соответствии со вторым подходом был создан путь коммутации по меткам второго уровня (LSP3), соединяющий устройства LER1 и LER4. Этот путь определяет последовательность хопов *между доменами*, а не между внутренними устройствами LSR каждого домена. Так, LSP3 состоит из хопов LER1 — LER2 — LER3 — LSR4. В этом отношении многоуровневый подход MPLS концептуально очень близок подходу протокола BGP, определяющего путь между автономными системами.

Рассмотри более детально, как работает технология MPLS в случае путей коммутации по меткам двух уровней (рис. 22.12).

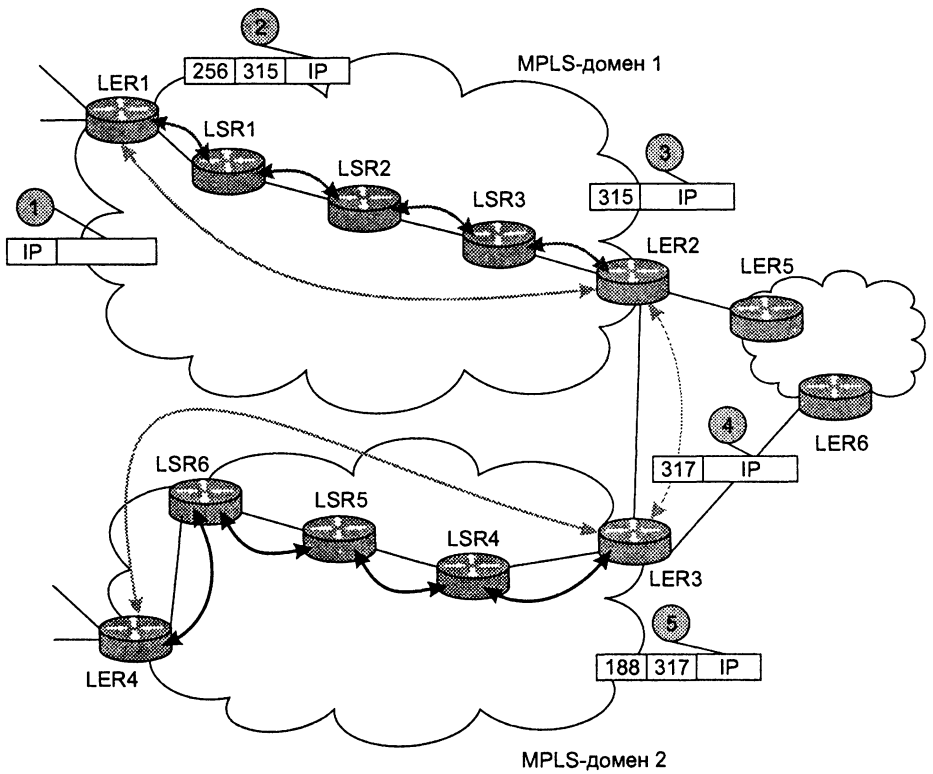


Рис. 22.12. Использование стека меток иерархией путей

В устройстве LER1 начинается два пути — LSP1 и LSP3, что обеспечивается соответствующей записью в таблице продвижения устройства LER1 (табл. 22.2).

IP-пакеты, поступающие на интерфейс S0 устройства LER1, продвигаются на его выходной интерфейс S1, где для них создается заголовок MPLS, включающий метку 315 верхнего уровня (LSP3), которая на этот момент является верхушкой стека меток. Затем эта метка проталкивается на дно стека (действие Push), а верхней становится метка 256, относящаяся к LSP1.

Таблица 22.2. Запись в таблице продвижения LER1

Входной интерфейс	Метка	Следующий хоп	Действия
...	...	...	...
S0	—	S1	315 Push 256
...	...	...	...

Далее MPLS-кадр с меткой 256 поступает на выходной интерфейс S1 пограничного устройства LER1 и передается на вход LSR1. Устройство LSR1 обрабатывает кадр в соответствии со своей таблицей продвижения (табл. 22.3). Метка 256, находящаяся на вершине стека, заменяется меткой 272. (Заметьте, что метка 315, находящаяся ниже в стеке, игнорируется устройством LSR1.)

Таблица 22.3. Запись в таблице продвижения LSR1

Входной интерфейс	Метка	Следующий хоп	Действия
...	...	...	...
S0	256	S1	272
...	...	...	...

Аналогичные действия выполняет устройство LSR2, которое заменяет метку меткой 132 и отправляет кадр следующему по пути устройству LSR3 (табл. 22.4).

Таблица 22.4. Запись в таблице продвижения LSR3

Входной интерфейс	Метка	Следующий хоп	Действия
...	...	...	...
S0	132	S1	Pop
...	...	...	...

Работа устройства LSR3 несколько отличается от работы устройств LSR1 и LSR2, так как оно является *предпоследним* устройством LSR для пути LSP1. В соответствии с записью в табл. 22.4 устройство LSR3 выполняет выталкивание (Pop) из стека метки 132, относящейся к пути LSP1. В результате верхней меткой стека становится метка 315, принадлежащая пути LSP3. Операция выталкивания метки предпоследним устройством называется Hop Popping (PHP).

Устройство LER2 продвигает поступивший на его входной интерфейс S0 кадр на основе своей записи таблицы продвижения (табл. 22.5). Устройство LER2 сначала заменяет метку 315 пути LSP3 значением 317, затем проталкивает ее на дно

стека и помещает на вершину стека метку 188, которая является меткой пути LSP2, внутреннего для домена 2. Перемещение кадра вдоль пути LSP2 происходит аналогичным образом.

Таблица 22.5. Запись в таблице продвижения LER2

Входной интерфейс	Метка	Следующий хоп	Действия
...	...	...	...
S0	315	S1	317 Push 188
...	...	...	...

Описанная модель двухуровневого пути легко может быть расширена для любого количества уровней.

Области применения технологии MPLS

Мы рассмотрели основные принципы, лежащие в основе технологии MPLS. В настоящее время существует несколько областей практического применения MPLS, в которых эти принципы дополняются специфическими механизмами и протоколами, необходимыми для достижения необходимой функциональности. Ниже перечислены области применения MPLS, получившие наибольшее распространение.

- **MPLS IGP.** В данном случае технология MPLS применяется только *для ускорения продвижения пакетов* сетевого уровня, следующих вдоль маршрутов, выбираемых стандартными внутренними шлюзовыми протоколами (IGP), которые и дали название этой области применения MPLS.
- **MPLS TE.** В этом случае пути коммутации по меткам выбираются *для решения задач инжиниринга трафика* (TE) на основе модифицированных протоколов маршрутизации. Техника MPLS TE не только позволяет обеспечить рациональную и сбалансированную загрузку всех ресурсов сети поставщика услуг, но и создает хорошую основу для предоставления транспортных услуг с гарантированными параметрами QoS.
- **MPLS VPN.** Эта область применения позволяет поставщику *предоставлять услуги виртуальных частных сетей* (VPN) на основе разграничения трафика без обязательного шифрования информации.

В данной главе мы рассмотрим первые две области применения MPLS, а сети MPLS VPN будут рассмотрены в главе 24, в которой изучаются и другие варианты организации виртуальных частных сетей. Заметим, что все три «ипостаси» технологии MPLS могут сосуществовать в одной сети, что позволяет предоставлять пользователю комбинированные услуги.

Технология MPLS IGP

Главной целью технологии MPLS IGP является ускорение продвижения пакетов через сеть поставщика услуг за счет замены маршрутизации коммутацией. Поэтому данная область применения называется также **ускоренной MPLS-коммутацией**.

При использовании технологии MPLS IGP пути коммутации по меткам прокладываются в соответствии с существующей топологией IP-сетей и не зависят от интенсивности трафика между этими сетями. Это свойство иллюстрирует рис. 22.13.

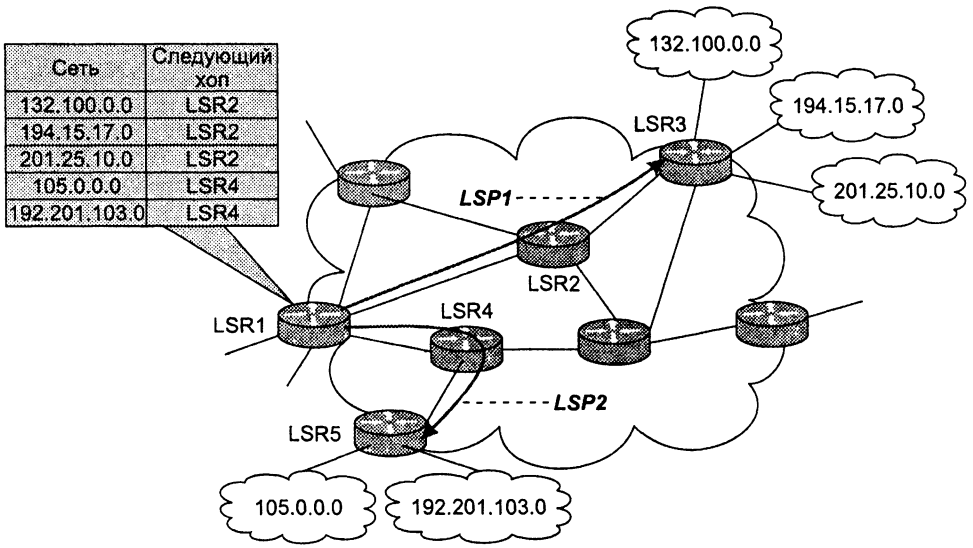


Рис. 22.13. Прокладка пути коммутации по меткам с помощью протокола LDP

Все устройства LSR поддерживают сигнальный *протокол распределения меток* (LDP). Кроме того, каждое устройство LSR должно поддерживать один из стандартных протоколов IGP, например RIP, IS-IS или OSPF.

В результате работы протоколов маршрутизации или же после ручной модификации администратором сети в таблице маршрутизации устройства LSR может появиться запись о новой сети назначения, для которой в сети поставщика услуг еще не проложен путь коммутации по меткам. В этом случае данное устройство автоматически инициирует процедуру прокладки нового пути. Для прокладки нового пути коммутации по меткам устройства LSR используют обычный алгоритм продвижения пакета в IP-сети на основе таблиц маршрутизации.

Пусть, например, устройство LSR1 обнаруживает, что в его таблице маршрутизации появилась новая запись о сети назначения 132.100.0.0, и в качестве следующего хопа указано устройство LSR2. В то же время виртуальный путь к этой сети не проложен, так как в таблице продвижения отсутствует соответствующая запись.

LSR1 становится инициатором прокладки пути к сети 132.100.0.0 и посылает LDP-запрос устройству LSR2. В этом запросе указывается IP-адрес сети назначения (132.100.0.0), к которой нужно проложить новый путь. Этот запрос принимается и обрабатывается устройством LSR2 на основе информации, имеющейся в его таблицах маршрутизации и продвижения. Если маршрутизатор LSR2 находит, что у него также нет проложенного пути к сети 132.100.0.0, он передает LDP-запрос следующему устройству LSR, адрес которого указан в его таблице маршрутизации как следующий хоп для сети 132.100.0.0. В примере, показанном на рис. 22.13, таким устройством является LSR3, на котором путь коммутации по меткам должен закончиться, так как следующий хоп ведет за пределы сети поставщика услуг.

ПРИМЕЧАНИЕ

Возникает вопрос: как устройство LSR3 узнает о том, что является последним в сети поставщика услуг на пути к сети 132.100.0.0? Дело в том, что LDP является протоколом, ориентированным на соединение, и при установлении логического LDP-соединения возможно применение автоматической аутентификации устройств, так что LDP-сеансы устанавливаются только между устройствами одного поставщика услуг, который задает для всех принадлежащих его сети устройств LSR соответствующую информацию для взаимной аутентификации.

Устройство LSR3, обнаружив, что для пути к сети 132.100.0.0 оно является пограничным, назначает для прокладываемого пути метку, еще не занятую его входным интерфейсом S0, и сообщает об этой метке устройству LSR2 в LDP-объявлении. В свою очередь, LSR2 назначает неиспользуемую его интерфейсом S0 метку и сообщает об этом в LDP-объявлении устройству LSR1. После этого новый путь коммутации по меткам, ведущий от LSR1 к сети 132.100.0.0, считается проложенным, и вдоль него пакеты начинают передаваться уже на основе меток и таблиц продвижения, а не IP-адресов и таблиц маршрутизации.

Было бы нерационально прокладывать отдельный путь для каждой сети назначения каждого маршрутизатора. Поэтому устройства LSR стараются строить агрегированные пути коммутации по меткам и передавать вдоль них пакеты, следующие к некоторому набору сетей. Так, LSR1 передает по пути LSP1 пакеты, следующие не только к сети 132.100.0.0, но и к сетям 194.15.17.0 и 201.25.10.0, так как пути к этим сетям совпадают в пределах MPLS-сети поставщика услуг.

Для передачи пакетов узлам сетей 105.0.0.0 и 192.201.103.0 у устройства LSR1 имеется другой путь, а именно LSP2. С помощью протокола LDP можно агрегировать также пути, которые совпадают не для всей последовательности устройств LSR от входного пограничного устройства до выходного, а имеют только часть общих устройств LSR. Все адреса сетей назначения, имеющие один и тот же следующий хоп, образуют так называемый **класс эквивалентного продвижения** (Forwarding Equivalence Class, FEC) для данного устройства LSR.

Протокол MPLS IGP *ускоряет* продвижение пакетов за счет сокращения просматриваемых таблиц, так как обычно таблица маршрутизации содержит гораздо больше записей, чем таблица продвижения. Особенно ощутима эта разница для крупных магистралей, где маршрутизаторы могут оперировать с таблицами маршрутизации размером в несколько десятков тысяч записей. Другим факто-

ром, влияющим на ускорения продвижения пакетов, является отсутствие этапов замены кадров канального уровня каждым маршрутизатором, что характерно для технологии IP.

Технология MPLS TE

Технология MPLS TE служит для прокладки в сети путей коммутации по меткам, *обеспечивающих гарантированную среднюю пропускную способность* в соответствии с принципами инжиниринга трафика, описанными в главе 7. В этом заключается основное отличие технологии MPLS TE от технологии MPLS IGP, которая обеспечивает прокладку путей коммутации по меткам, исходя из известной топологии составной сети, а трафик при этом игнорируется.

Кроме того, в отличие от MPLS IGP в технологии MPLS TE пути коммутации по меткам, называемые здесь **TE-туннелями**, не прокладываются автоматически. TE-туннели прокладываются только по инициативе администратора сети, и в этом отношении TE-туннели подобны каналам PVC в технологиях ATM и Frame Relay.

MPLS TE поддерживает туннели двух типов:

- **строгий TE-туннель** определяет все промежуточные узлы между двумя пограничным устройствами;
- **свободный TE-туннель** определяет только часть промежуточных узлов от одного пограничного устройства до другого, а остальные промежуточные узлы выбираются устройством LSR самостоятельно.

На рис. 22.14 показаны оба типа туннелей.

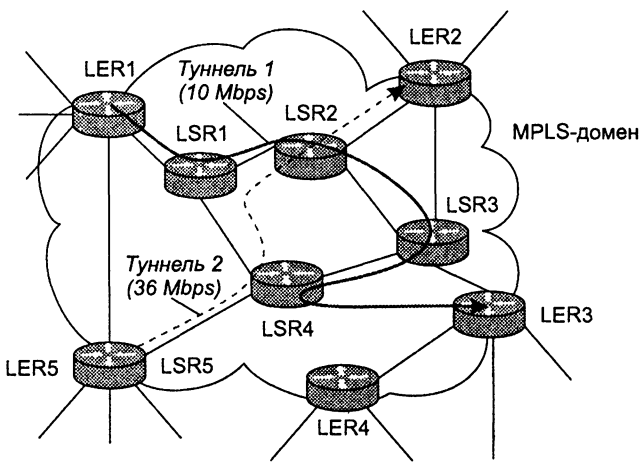


Рис. 22.14. Два типа TE-туннелей в технологии MPLS

Туннель 1 является примером строгого туннеля, при его задании администратор указал как начальный и конечный узлы туннеля, так и все промежуточные узлы, то есть последовательность IP-адресов для устройств LER1, LSR1, LSR2, LSR3, LER3. Таким образом, администратор сам решил задачу инжиниринга трафика,

выбрав путь, вдоль которого существует достаточная неиспользуемая пропускная способность. Администратор при установлении туннеля 1 задает не только последовательность адресов, но и требуемую пропускную способность. Несмотря на то что выбор пути происходит в автономном режиме, все устройства сети вдоль туннеля 1 проверяют, действительно ли они обладают запрошенной неиспользуемой пропускной способностью, и только в случае положительного ответа туннель устанавливается.

При установке туннеля 2 (свободного) администратор задает только начальный и конечный узлы туннеля, то есть устройства LER5 и LER2. Промежуточные устройства LSR4 и LSR2 находятся автоматически начальным узлом туннеля 2, то есть устройством LER5, а затем с помощью сигнального протокола устройство LER5 сообщает этим и конечному устройствам о необходимости установления туннеля.

Независимо от типа туннеля он всегда обладает таким параметром, как резервируемая пропускная способность. В нашем примере туннель 1 резервирует для трафика 10 Мбит/с, а туннель 2 — 36 Мбит/с. Эти значения определяются администратором, и технология MPLS TE никак не влияет на их выбор, она только отрабатывает запрошенное резервирование. Чаще всего администратор оценивает резервируемую для туннеля пропускную способность на основании измерений трафика в сети, тенденций изменения трафика, а также собственной интуиции. Некоторые реализации MPLS TE позволяют затем автоматически скорректировать величину зарезервированной пропускной способности на основании автоматических измерений реальной интенсивности трафика, проходящего через туннель.

Однако само по себе установление в MPLS-сети TE-туннеля еще не означает передачи по нему трафика. Оно означает только то, что в сети действительно существует возможность передачи трафика по туннелю со средней скоростью, не превышающей зарезервированное значение. Для того чтобы данные были переданы по туннелю, администратору предстоит еще одна ручная процедура: задание для начального устройства туннеля условий, определяющих, какие именно пакеты должны передаваться по туннелю. Условия могут быть весьма разнообразными; в качестве признаков агрегированного потока, который должен передаваться по туннелю, могут выступать все традиционные признаки: IP-адрес назначения и источника, тип протокола, номера TCP- и UDP-портов, номер интерфейса входящего трафика, значения приоритета в протоколах DSCP и IP и т. д.

Таким образом, устройство LER должно сначала провести *классификацию трафика*, затем выполнить *профилирование*, удостоверившись, что средняя скорость потока не превышает зарезервированную, и наконец, начать *маркировать* пакеты, используя начальную метку TE-туннеля, чтобы передавать трафик через сеть с помощью техники MPLS. В этом случае расчеты, выполненные на этапе выбора пути для туннеля, дадут нужный результат — баланс ресурсов сети при соблюдении средней скорости для каждого потока.

Однако мы еще не рассмотрели специфический набор протоколов, которые устройства LER и LSR сети используют для прокладки свободных туннелей или проверки работоспособности созданных администратором строгих туннелей.

Для выбора и проверки путей через туннели в технологии MPLS TE используются расширения протоколов маршрутизации, работающих на основе алгоритма состояния связей. Сегодня такие расширения стандартизованы для протоколов OSPF и IS-IS. Для решения задачи TE в протоколы OSPF и IS-IS включены новые типы объявлений для распространения по сети информации о номинальной и незарезервированной (доступной для TE-потоков) величинах пропускной способности каждой связи. Таким образом, ребра результирующего графа сети, создаваемого в топологической базе каждого устройства LER или LSR, будут маркированы этими двумя дополнительными параметрами. Располагая таким графом, а также параметрами потоков, для которых нужно определить TE-пути, устройство LER может найти рациональное решение, удовлетворяющее одному из сформулированных в главе 7 ограничений на использование ресурсов сети. Чаще всего решение ищется по наиболее простому критерию, который состоит в минимизации максимального значения коэффициента использования вдоль выбранного пути, то есть критерием оптимизации пути является значение min $\{K_{max\,i}\}$ для всех возможных путей.

В общем случае администратору необходимо проложить несколько туннелей для различных агрегированных потоков. Для упрощения задачи оптимизации выбор путей для этих туннелей обычно осуществляется по очереди, причем администратор определяет очередность на основе своей интуиции. Очевидно, что поиск TE-путей по очереди снижает качество решения — при одновременном рассмотрении всех потоков в принципе можно добиться более рациональной загрузки ресурсов.

Пример

В примере, показанном на рис. 22.15, ограничением является максимально допустимое значение коэффициента использования ресурсов, равное 0,65. В варианте 1 решение было найдено при очередности рассмотрения потоков 1, 2, 3. Для первого потока был выбран путь A-B-C, так как в этом случае он, с одной стороны, удовлетворяет ограничению (все ресурсы вдоль пути — каналы A-B, A-C и соответствующие интерфейсы маршрутизаторов оказываются загруженными на 50/155 = 0,32), а с другой — обладает минимальной метрикой (65 + 65 = 130). Для второго потока также был выбран путь A-B-C, так как и в этом случае ограничение удовлетворяется — результирующий коэффициент использования оказывается равным 50 + 40/155 = 0,58. Третий поток направляется по пути A-D-E-C и загружает ресурсы каналов A-D, D-E и E-C на 0,3. Решение 1 можно назвать удовлетворительным, так как коэффициент использования любого ресурса в сети не превышает 0,58.

Однако существует лучший способ, представленный в варианте 2. Здесь по верхнему пути A-B-C были направлены потоки 2 и 3, а поток 1 — по нижнему пути A-D-E-C. Ресурсы верхнего пути оказываются загруженными на 0,45, а нижнего — на 0,5, то есть налицо более равномерная загрузка ресурсов, а максимальный коэффициент использования всех ресурсов сети не превышает 0,5. Этот вариант может быть получен при одновременном рассмотрении всех трех потоков с учетом ограничения min (max Ki) или же при рассмотрении потоков по очереди в последовательности 2, 3, 1.

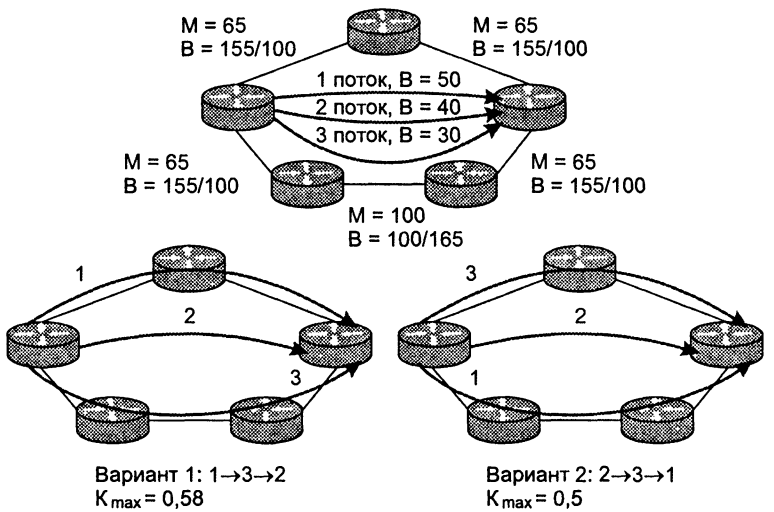


Рис. 22.15. Зависимость качества решения задачи TE от очередности выбора туннелей

Тем не менее в производимом сегодня оборудовании применяется вариант технологии MPLS TE с последовательным рассмотрением потоков. Он проще в реализации и ближе к стандартным для протоколов OSPF и IS-IS процедурам нахождения кратчайшего пути для одной сети назначения (в отсутствие ограничений найденное решение для набора кратчайших путей не зависит от последовательности учета сетей, для которых производился поиск). Кроме того, при изменении ситуации — появлении новых потоков или изменении интенсивности существующих — найти путь удается только для одного потока.

Возможен также способ, при котором внешняя по отношению к сети вычислительная система, работающая в автономном режиме, определяет оптимальное решение для набора потоков. Это может быть достаточно сложная система, которая включает подсистему имитационного моделирования, способную учесть не только средние интенсивности потоков, но и их пульсации, и оценить не только загрузку ресурсов, но и результирующие параметры QoS — задержки, потери и т. п. После нахождения оптимального решения его можно модифицировать уже в оперативном режиме поочередного поиска путей.

В технологии MPLS TE информация о найденном рациональном пути используется полностью — то есть запоминаются IP-адреса источника, всех транзитных маршрутизаторов и конечного узла. Вспомним, такая маршрутизация называется *маршрутизацией от источника*. Поэтому достаточно, чтобы поиском путей занимались только пограничные устройства сети (LER), а промежуточные устройства (LSR) лишь поставляли им информацию о текущем состоянии резервирования пропускной способности каналов.

После нахождения пути независимо от того, найден он был устройством LER или администратором, его необходимо зафиксировать. Для этого в MPLS TE используется расширение уже рассмотренного нами протокола резервирования ресурсов (RSVP), который часто в этом случае называют протоколом **RSVP TE**.

Сообщения RSVP TE передаются от одного устройства LSR другому в соответствии с данными о найденных IP-адресах маршрута. При установлении нового пути в сигнальном сообщении наряду с последовательностью адресов пути указывается также и резервируемая пропускная способность. Каждое устройство LSR, получив такое сообщение, вычитает запрашиваемую пропускную способность из пула свободной пропускной способности соответствующего интерфейса, а затем объявляет остаток в сообщениях протокола маршрутизации, например CSPF.

В заключение рассмотрим вопрос о соотношении технологий MPLS TE и QoS. Как видно из описания, основной целью MPLS TE является использование возможностей MPLS для достижения внутренней цели поставщика услуг, а именно достижения сбалансированной загрузки всех ресурсов своей сети. Однако при этом также создается основа для предоставления транспортных услуг с гарантированными параметрами QoS, так как трафик по TE-туннелям передается при соблюдении некоторого максимального уровня коэффициента использования ресурсов. Как мы знаем из материала главы 7, коэффициент использования ресурсов оказывает решающее влияние на процесс образования очереди, так что потоки, передаваемые по TE-туннелям, передаются с некоторым гарантированным уровнем QoS.

Для того чтобы обеспечить разные параметры QoS для разных классов трафика, поставщику услуг необходимо для каждого класса трафика установить в сети отдельную систему туннелей. При этом для чувствительного к задержкам классам трафика нужно выполнить резервирование таким образом, чтобы максимальный коэффициент использования ресурсов туннеля находился в диапазоне 0,2–0,3, иначе задержки пакетов и их вариации выйдут за допустимые пределы.

Сетевое управление в IP-сетях

Ключевые слова: система управления сетью, интегрированная система управления, управление конфигурацией сети и именованием, обработка ошибок, анализ производительности и надежности, управление безопасностью, учет работы сети, система управления системой, идентификатор общности.

Любая сложная вычислительная сеть требует дополнительных специальных средств управления помимо имеющихся в стандартных сетевых операционных системах. Это связано с большим количеством разнообразного коммуникационного оборудования, работа которого критически важна для выполнения сетью своих основных функций. Распределенный характер крупной сети делает невозможным поддержание ее работы без централизованной системы управления, которая бы в автоматическом режиме собирала информацию о состоянии каждого концентратора, коммутатора, мультиплексора и маршрутизатора и предоставляла эту информацию оператору сети.

Одной из первых систем сетевого управления, получившей широкое распространение, был программный продукт SunNet Manager, выпущенный в 1989 году

компанией SunSoft. Система SunNet Manager была ориентирована на управление коммуникационным оборудованием и контроль трафика сети. Именно эти функции имеют чаще всего в виду, когда говорят о **системе управления сетью** (Network Management System, NMS).

Обычно система управления работает в *автоматизированном* режиме, выполняя наиболее простые действия по управлению сетью автоматически, а сложные решения предоставляя принимать человеку на основе подготовленной системой информации.

Сами системы управления представляют собой сложные программно-аппаратные комплексы, поэтому существует граница целесообразности их применения. В небольшой сети можно применять отдельные программы управления наиболее сложными устройствами, например коммутатором, поддерживающим технику VLAN. Обычно каждое устройство, которое требует достаточно сложного конфигурирования, производитель сопровождает автономной программой конфигурирования и управления. Однако при росте сети может возникнуть проблема объединения разрозненных программ управления устройствами в единую систему управления, и для решения этой проблемы придется, возможно, отказаться от этих программ и заменить их **интегрированной системой управления**.

Функциональные группы задач управления

Независимо от объекта управления, желательно, чтобы система управления выполняла ряд функций, которые определены международными стандартами, обобщающими опыт применения систем управления в различных областях. Существуют рекомендации ITU-T X.700 и близкий к ним стандарт ISO 7498-4, которые делят задачи системы управления на пять функциональных групп.

- **Управление конфигурацией сети и именованием** (Configuration Management) заключается в конфигурировании параметров как элементов сети (Network Element, NE), так и сети в целом. Для элементов сети, таких как маршрутизаторы, мультиплексоры и т. п., путем конфигурирования определяются сетевые адреса, идентификаторы (имена), географическое положение и пр. Для сети в целом управление конфигурацией обычно начинается с построения карты сети, то есть отображении реальных связей между элементами сети и изменений этих связей при образовании новых физических или логических каналов, изменении таблиц коммутации и маршрутизации.
- **Обработка ошибок** включает выявление, определение и устранение последствий сбоев и отказов в работе сети.
- **Анализ производительности и надежности** связан с оценкой на основе накопленной статистической информации таких параметров, как время реакции системы, пропускная способность реального или виртуального канала связи между двумя конечными абонентами сети, интенсивность трафика в отдельных сегментах и каналах сети, вероятность искажения данных при их передаче через сеть, а также коэффициент готовности сети или ее определенной транспортной службы. Результаты анализа производительности и надежности позволяют контролировать *соглашение об уровне обслуживания* (SLA), заклю-

чаемое между пользователем сети и ее администраторами (или компанией, продающей услуги). Без средств анализа производительности и надежности поставщик услуг публичной сети или отдел информационных технологий предприятия не сможет ни проконтролировать, ни тем более обеспечить нужный уровень обслуживания для конечных пользователей сети.

- **Управление безопасностью** подразумевает контроль доступа к ресурсам сети (данным и оборудованию) и сохранение целостности данных при их хранении и передаче через сеть. Базовыми элементами управления безопасностью являются процедуры аутентификации пользователей, назначение и проверка прав доступа к ресурсам сети, распределение и поддержка ключей шифрования, управления полномочиями и т. п. Часто функции этой группы не включаются в системы управления сетями, а либо реализуются в виде специальных продуктов (например, систем аутентификации и авторизации Kerberos, различных защитных экранов, систем шифрования данных), либо входят в состав операционных систем и системных приложений.
- **Учет работы сети** включает регистрацию времени использования различных ресурсов сети (устройств, каналов и транспортных служб) и ведение биллинговых операций (плата за ресурсы). Ввиду специфического характера оплаты услуг у различных поставщиков и различными формами соглашения об уровне обслуживания, эта группа функций обычно не включается в коммерческие системы и платформы управления типа HP OpenView, а реализуется в заказных системах, разрабатываемых для конкретного заказчика.

Хотя модель управления OSI не делает различий между управляемыми объектами — каналами, сегментами локальных сетей, коммутаторами и маршрутизаторами, модемами и мультиплексорами, аппаратным и программным обеспечением компьютеров, на практике деление систем управления по типам управляемых объектов широко распространено.

Ставшими классическими системы управления сетями, например SunNet Manager, HP OpenView или Cabletron Spectrum, управляют только *коммуникационными* объектами корпоративных сетей, такими как маршрутизаторы и коммутаторы.

В тех случаях, когда управляемыми объектами являются *компьютеры*, их системное и прикладное программное обеспечение, для системы управления часто используют особое название — **система управления системой** (System Management System, SMS).

SMS обычно автоматически собирает информацию об установленных в сети компьютерах и создает записи в специальной БД об аппаратных и программных ресурсах. SMS может централизованно устанавливать и администрировать приложения, которые запускаются с файловых серверов, а также удаленно измерять наиболее важные параметры компьютера, операционной системы, СУБД (например, коэффициент использования процессора или физической памяти, интенсивность страничных прерываний и др.). SMS может давать администратору возможность брать на себя удаленное управление компьютером в режиме эмуляции графического интерфейса популярных операционных систем. Примерами систем управления системами являются Microsoft System Management Server (SMS), CA Unicenter, HP Operationscenter. На практике уже несколько лет так-

же заметна отчетливая тенденция интеграции систем управления сетями и системами в единые интегрированные продукты управления сетями.

Архитектуры систем управления сетями

Основным элементом любой системы управления сетью лежит схема взаимодействия «менеджер — агент — управляемый объект» (рис. 22.16). На основе этой схемы могут быть построены системы практически любой сложности с большим количеством агентов, менеджеров и ресурсов разного типа.

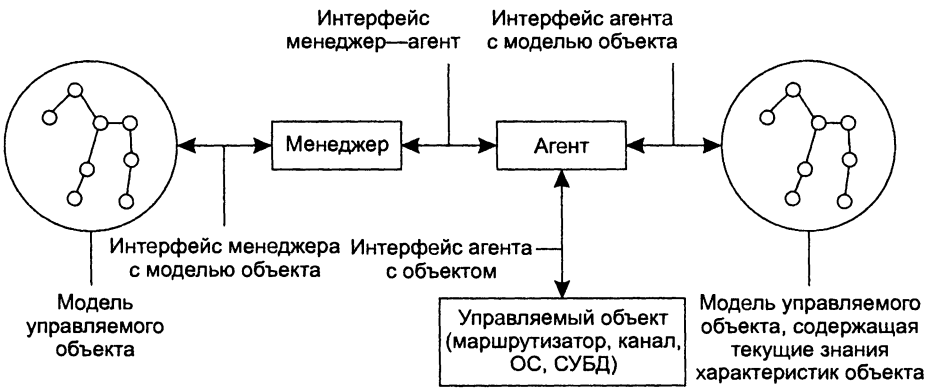


Рис. 22.16. Взаимодействие агента, менеджера и управляемого объекта

Чтобы можно было автоматизировать управление объектами сети, создается некоторая *модель управляемого объекта*, называемая *базой данных* управляющей информации (Management Information Base, MIB). MIB отражает только те характеристики объекта, которые нужны для его контроля. Например, модель маршрутизатора обычно включает такие характеристики, как количество портов, их тип, таблицу маршрутизации, количество кадров и пакетов протоколов канального, сетевого и транспортного уровней, прошедших через эти порты. Менеджер и агент работают с одной и той же моделью управляемого объекта, однако в использовании этой модели агентом и менеджером имеются существенные различия.

Агент наполняет MIB управляемого объекта текущими значениями его характеристик, а *менеджер* извлекает из MIB данные, на основании которых он узнает, какие характеристики он может запросить у агента и какими параметрами объекта можно управлять. Таким образом, агент является посредником между управляемым объектом и менеджером. Агент поставляет менеджеру только те данные, которые предусматриваются MIB.

Менеджер и агент взаимодействуют по стандартному протоколу. Этот протокол позволяет менеджеру запрашивать значения параметров, хранящихся в MIB, а также передавать агенту информацию, на основе которой тот должен управлять объектом. Обычно менеджер работает на отдельном компьютере, взаимодействуя с несколькими агентами.

Агенты могут встраиваться в управляемое оборудование, а могут и работать на отдельном компьютере, связанном с управляемым оборудованием. Для получения требуемых данных об объекте, а также для выдачи на него управляющих воздействий агент должен иметь возможность взаимодействовать с ним. Однако многообразие типов управляемых объектов не позволяет стандартизовать способ взаимодействия агента с объектом. Эта задача решается разработчиками при встраивании агентов в коммуникационное оборудование или в операционную систему. Агент может снабжаться специальными датчиками для получения информации, например датчиками релейных контактов или датчиками температуры. Агенты могут отличаться разным уровнем интеллекта — они могут обладать как самым минимальным интеллектом, необходимым для подсчета проходящих через оборудование кадров и пакетов, так и весьма высоким, достаточным для выполнения самостоятельных действий по выполнению последовательности управляющих команд в аварийных ситуациях, построению временны́х зависимостей, фильтрации аварийных сообщений и т. п.

Различают *внутриполосное управление*, когда управляющие сигналы идут по тому же каналу, по которому передаются пользовательские данные, и *внеполосное управление*, то есть осуществляемое вне канала, по которому передаются пользовательские данные. Например, если сообщения протокола, по которому менеджер взаимодействует с агентом, встроенным в маршрутизатор, передаются по той же сети, что и пользовательские данные, то это будет внутриполосным управлением. Если же менеджер контролирует коммутатор первичной сети, работающий по технологии FDM, с помощью отдельной сети X.25, к которой подключен агент, то это будет внеполосным управлением. Передача данных управления по тому же каналу, на котором работает сеть, более экономична, так как не требует создания отдельной инфраструктуры передачи управляющих данных. Однако внеполосное управление надежнее, так как соответствующее оборудование может выполнять свои функции тогда, когда те или иные сетевые элементы выходят из строя и основные каналы передачи данных оказываются недоступными.

Схема «менеджер — агент — управляемый объект» позволяет строить достаточно сложные в структурном отношении распределенные системы управления (рис. 22.17).

Как показано на рисунке, каждый агент управляет определенным элементом сети, параметры которого помещает в соответствующую базу MIB. Менеджеры извлекают данные из баз MIB своих агентов, обрабатывают их и хранят в собственных базах данных. Операторы, работающие за рабочими станциями, могут соединиться с любым из менеджеров и с помощью графического интерфейса просмотреть данные об управляемой сети, а также выдать менеджеру некоторые директивы по управлению сетью или ее элементами.

Наличие нескольких менеджеров позволяет распределить между ними нагрузку по обработке данных управления, обеспечивая масштабируемость системы. Как правило, используются два типа связей между менеджерами — одноранговая (рис. 22.18) и иерархическая (рис. 22.19).

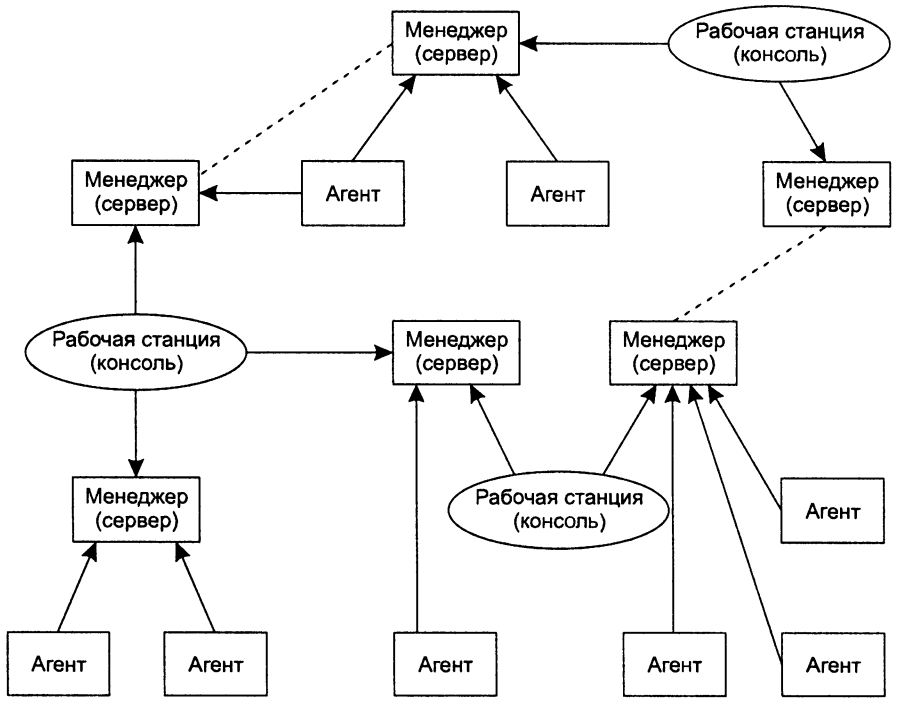


Рис. 22.17. Распределенная система управления на основе нескольких менеджеров и рабочих станций

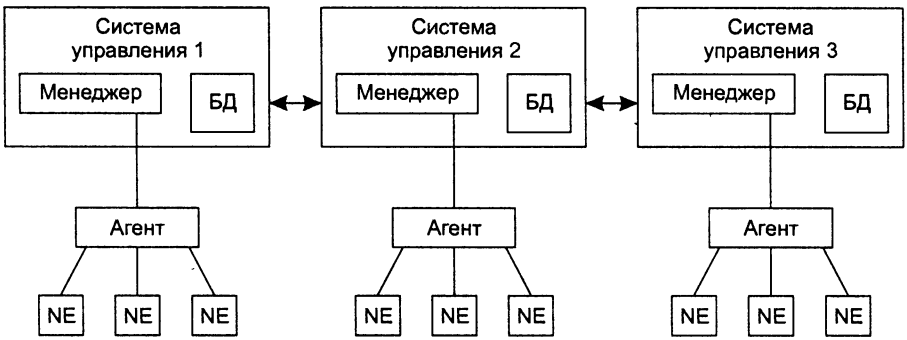


Рис. 22.18. Одноранговые связи между менеджерами

В случае *одноранговых* связей каждый менеджер управляет своей частью сети на основе информации, получаемой от нижележащих агентов. Центральный менеджер отсутствует. Координация работы менеджеров достигается за счет обмена информацией между базами данных менеджеров. Одноранговое построение системы управления сегодня считается неэффективным и устаревшим.

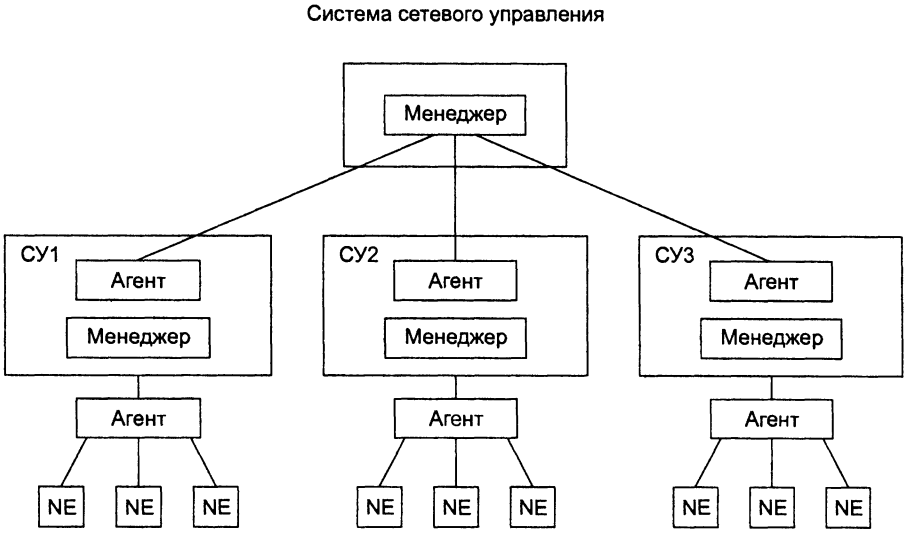


Рис. 22.19. Иерархические связи между менеджерами

Значительно более гибким является *иерархическое* построение связей между менеджерами. Каждый менеджер нижнего уровня выполняет также функции агента для менеджера верхнего уровня. Такой агент работает уже с укрупненной моделью (MIB) своей части сети, в которой собирается именно та информация, которая нужна менеджеру верхнего уровня для управления сетью в целом. Обычно для разработки моделей сети на разных уровнях проектирование начинают с верхнего уровня, на котором определяется состав информации, требуемой от менеджеров-агентов более низкого уровня. Такой подход сокращает объемы информации, циркулирующей между уровнями системы управления.

Модель «менеджер — агент — управляемый объект» лежит в основе таких популярных стандартов управления, как стандарты Интернета на основе протокола SNMP и стандартов управления ISO/OSI на основе протокола CMIP (Common Management Information Protocol).

Стандарты систем управления на основе протокола SNMP

Нет ничего более постоянного, чем временное. Протокол SNMP (Simple Management Network Protocol — простой протокол сетевого администрирования) может служить еще одним подтверждением этой азбучной истины. Разработанный как временное и очень простое решение для IP-сетей, он настолько понравился разработчикам оборудования и сетевым администраторам, что на долгие годы стал протоколом № 1 в системах управления. И это несмотря на то, что уже давно существует гораздо более мощный (и, соответственно, сложный) протокол CMIP, к тому же являющийся международным стандартом ITU-T.

Однако когда появилась вторая версия протокола (SNMPv2), она не была поддержана производителями сетевого оборудования и распространения не получила. Разработчики стандартов из IETF стараются переломить ситуацию, предложив спецификацию третьей версии (SNMPv3). Существенные улучшения протокола, обеспечивающие гибкое администрирование агентов систем управления и защиту управляющей информации, обратная совместимость с системами на основе базовой версии SNMPv1, а также открытая архитектура позволяют авторам SNMPv3 надеяться на успешное практическое воплощение своего детища.

SNMP — это протокол прикладного уровня, разработанный для стека TCP/IP, хотя имеются его реализации и для других стеков, например IPX/SPX. Протокол SNMP используется для получения от сетевых устройств информации об их статусе, производительности и других характеристиках, которые хранятся в MIB. Простота SNMP во многом определяется простотой баз данных MIB SNMP, особенно их первых версий MIB-I и MIB-II.

Ниже перечислены элементы, которые стандартизуются в системах управления, построенных на основе протокола SNMP.

- Протокол взаимодействия агента и менеджера (собственно протокол SNMP).
- Язык описания моделей MIB и SNMP-сообщений — язык абстрактной синтаксической нотации ASN.1 (стандарт ISO 8824:1987, рекомендации ITU-T X.208). Стандарты определяют структуру базы данных MIB, в том числе набор типов ее объектов, их имена и допустимые операции над ними (например, чтение).
- Несколько конкретных моделей MIB (MIB-I, MIB-II, RMON, RMON 2), имена объектов которых регистрируются в дереве стандартов ISO. Древовидная структура MIB содержит обязательные (стандартные) поддеревья, а также в ней могут находиться частные поддеревья, позволяющие изготовителю интеллектуальных устройств управлять какими-либо специфическими функциями устройства на основе специфических объектов MIB.

Все остальное отдается «на откуп» разработчику системы управления.

SNMP — это протокол типа «запрос-ответ», то есть на каждый запрос, поступивший от менеджера, агент должен передать ответ. Особенностью протокола является его чрезвычайная простота — он включает в себя всего несколько команд.

- Команда **Get-request** используется менеджером для получения от агента значения какого-либо объекта по его имени.
- Команда **GetNext-request** используется менеджером для извлечения значения следующего объекта (без указания его имени) при последовательном просмотре таблицы объектов.
- С помощью команды **Get-response** SNMP-агент передает менеджеру ответ на команду **Get-request** или **GetNext-request**.
- Команда **Set** позволяет менеджеру изменять значения какого-либо объекта. С помощью команды **Set** и происходит собственно управление устройством. Агент должен «понимать» смысл значений объекта, который используется для управления устройством, и на основании этих значений выполнять реальное управляющее воздействие — отключить порт, приписать порт определенной

линии VLAN и т. п. Команда Set пригодна также для задания условия, при выполнении которого SNMP-агент должен послать менеджеру соответствующее сообщение. Может быть определена реакция на такие события, как инициализация агента, рестарт агента, обрыв связи, восстановление связи, неверная аутентификация и потеря ближайшего маршрутизатора. Если происходит любое из этих событий, то агент инициализирует прерывание.

- Команда Trap используется агентом для сообщения менеджеру о возникновении особой ситуации.

Версия SNMPv2 добавляет к этому набору команду GetBulk, которая позволяет менеджеру получить несколько переменных за один запрос.

Структура SNMP MIB

На сегодня существует несколько стандартов на базы данных управляющей информации для протокола SNMP. Основными являются стандарты MIB-I и MIB-II, а также версия базы данных для удаленного управления RMON (Remote Monitoring) MIB. Кроме того, существуют стандарты для специальных MIB-устройств конкретного типа (например, MIB для концентраторов или MIB для модемов), а также частные базы данных MIB конкретных фирм-производителей оборудования.

Первоначальная спецификация MIB-I определяла только операции чтения значений переменных. Операции изменения или установки значений объекта являются частью спецификаций MIB-II.

Версия MIB-I (RFC 1156) определяет 114 объектов, которые подразделяются на 8 групп.

- *System* — общие данные об устройстве (например, идентификатор поставщика, время последней инициализации системы).
- *Interfaces* — параметры сетевых интерфейсов устройства (например, их количество, типы, скорости обмена, максимальный размер пакета).
- *Address Translation Table* — описание соответствия между сетевыми и физическими адресами (например, по протоколу ARP).
- *Internet Protocol* — данные, относящиеся к протоколу IP (адреса IP-шлюзов, хостов, статистика о IP-пакетах).
- *ICMP* — данные, относящиеся к протоколу обмена управляющими сообщениями ICMP.
- *TCP* — данные, относящиеся к протоколу TCP (например, о TCP-соединениях).
- *UDP* — данные, относящиеся к протоколу UDP (число переданных, принятых и ошибочных UPD-дейтаграмм).
- *EGP* — данные, относящиеся к протоколу EGP, используемому в Интернете (число принятых с ошибками и без ошибок сообщений).

Из этого перечня групп переменных видно, что стандарт MIB-I разрабатывался с жесткой ориентацией на управление маршрутизаторами, поддерживающими протоколы стека TCP/IP.

В версии MIB-II (RFC 1213), принятой в 1992 году, был существенно (до 185) расширен набор стандартных объектов, а число групп увеличилось до 10.

На рис. 22.20 приведен пример древовидной структуры базы объектов MIB-II. На нем показаны две из 10 возможных групп объектов — System (имена объектов начинаются с префикса Sys) и Interfaces (префикс if).

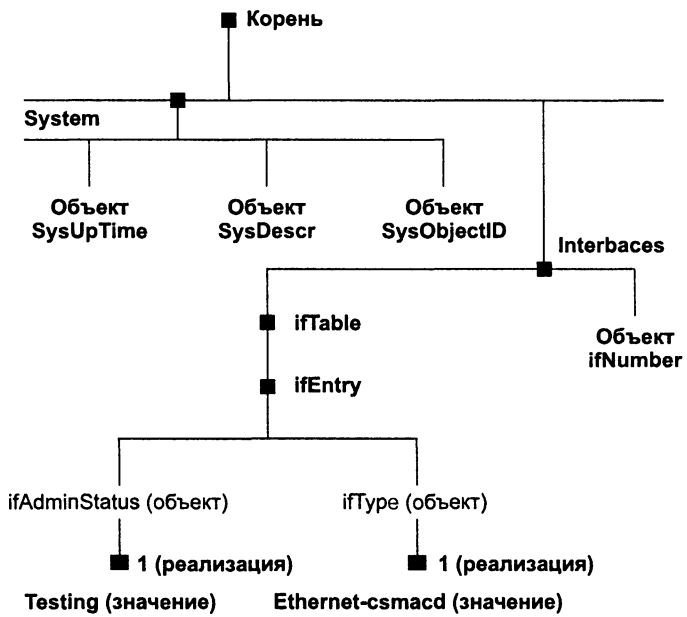


Рис. 22.20. Стандартное дерево MIB-II (фрагмент)

Объект SysUpTime содержит значение продолжительности времени работы системы с момента последней перезагрузки, объект SysObjectID — идентификатор устройства (например, маршрутизатора).

Объект ifNumber определяет количество сетевых интерфейсов устройства, а объект ifEntry является вершиной поддерева, описывающего один из конкретных интерфейсов устройства. Входящие в это поддерево объекты ifType и ifAdminStatus определяют соответственно тип и состояние одного из интерфейсов, в данном случае интерфейса Ethernet.

Ниже перечислены объекты, описывающие конкретные интерфейсы устройства.

- ifType — тип протокола, который поддерживает интерфейс. Этот объект принимает значения всех стандартных протоколов канального уровня, например rfc877-x25, ethernet-csmacd, iso88023-csmacd, iso88024-tokenBus, iso88025-tokenRing и т. д.

- ifMtu — максимальный размер пакета сетевого уровня, который можно послать через этот интерфейс.
- ifSpeed — пропускная способность интерфейса в битах в секунду (100 для Fast Ethernet).
- ifPhysAddress — физический адрес порта, для Fast Ethernet им будет MAC-адрес.
- ifAdminStatus — желаемый статус порта:
  - up — готов передавать пакеты;
  - down — не готов передавать пакеты;
  - testing — находится в тестовом режиме.
- ifOperStatus — фактический текущий статус порта, имеет те же значения, что и ifAdminStatus.
- ifInOctets — общее количество байтов, принятое данным портом, включая служебные, с момента последней инициализации SNMP-агента.
- ifInUcastPkts — количество пакетов с индивидуальным адресом интерфейса, доставленных протоколу верхнего уровня.
- ifInNUcastPkts — количество пакетов с широковещательным или групповым адресом интерфейса, доставленных протоколу верхнего уровня.
- ifInDiscards — количество пакетов, которые были приняты интерфейсом, оказались корректными, но не были доставлены протоколу верхнего уровня, скорее всего из-за переполнения буфера пакетов или же по иной причине.
- ifInErrors — количество пришедших пакетов, которые не были переданы протоколу верхнего уровня из-за обнаружения в них ошибок.

Помимо объектов, описывающих статистику по входным пакетам, имеются аналогичные объекты, но относящиеся к выходным пакетам.

Как видно из описания объектов MIB-II, эта база данных не дает детальной статистики по характерным ошибкам кадров Ethernet, кроме того, она не отражает изменение характеристик во времени, что часто интересует сетевого администратора. Эти ограничения были впоследствии сняты новым стандартом на MIB — RMON MIB, который специально ориентирован на сбор детальной статистики по протоколу Ethernet. Возможности RMON MIB включают также построение временны́х зависимостей значений параметров.

Для именования переменных базы MIB и однозначного определения их форматов используется дополнительная спецификация, называемая SMI (Structure of Management Information — структура управляющей информации). Например, спецификация SMI включает в качестве стандартного имя IpAddress и определяет его формат как строку из 4 байт. Другой пример — имя Counter, для которого определен формат в виде целого числа в диапазоне от 0 до $2^{32} - 1$.

Имена переменных MIB могут быть записаны как в символьном, так и в числовом форматах. Символьный формат используется для представления переменных в текстовых документах и на экране дисплея, а числовой — в сообщениях

протокола SNMP. Например, символьному имени SysDescr соответствует числовое имя 1.3.6.1.2.1.1.1.

Составное числовое имя объекта базы данных MIB протокола SNMP соответствует полному имени этого объекта в дереве регистрации объектов стандартизации ISO. Разработчики протокола SNMP не стали использовать традиционный для стандартов Интернета способ фиксации числовых параметров протокола в специальном документе RFC. Вместо этого они зарегистрировали объекты баз данных MIB протокола SNMP во всемирном дереве регистрации стандартов ISO (рис. 22.21).

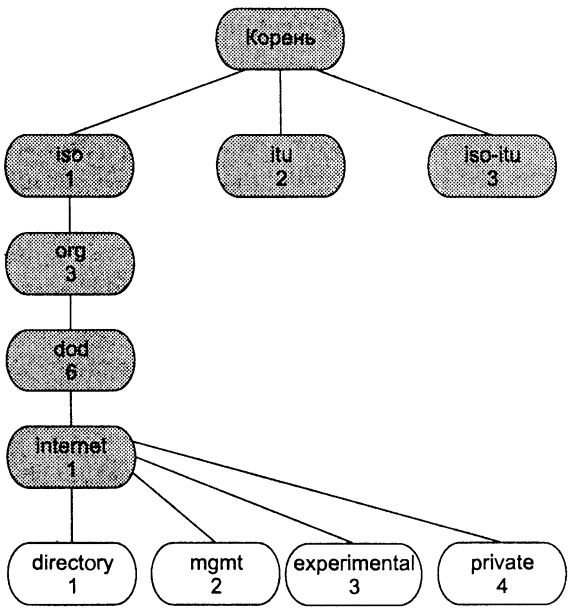


Рис. 22.21. Пространство имен объектов ISO

Как и в любых сложных системах, пространство имен объектов ISO имеет древовидную иерархическую структуру, причем на рисунке показана только его верхняя часть. От корня этого дерева отходят три ветви, соответствующие стандартам, контролируемым ISO, ITU и совместно ISO-ITU. В свою очередь, организация ISO создала ветвь для стандартов, создаваемых национальными и международными организациями (ветвь org). Стандарты Интернета создавались под эгидой Министерства обороны (Department of Defense, DoD) США, поэтому стандарты MIB попали в поддерево dod-internet, а далее, естественно, в группу стандартов управления сетью — ветвь mgmt. Объекты любых стандартов, создаваемых под эгидой ISO, однозначно идентифицируются составными символьными именами, начинающимися от корня этого дерева. В сообщениях протоколов используются не символьные имена, а однозначно соответствующие им составные числовые имена. Каждая ветвь дерева имен объектов нумеруется в дереве целыми числами слева направо, начиная с единицы, и эти числа и заменяют сим-

вольные имена. Поэтому полному символьному имени объекта MIB iso.org.dod.internet.mgmt.mib соответствует полное числовое имя — 1.3.6.1.2.1.

Группа объектов private (4) зарезервирована за стандартами, создаваемыми частными компаниями, например Cisco, Hewlett-Packard и т. п. Это же дерево регистрации используется для именования классов объектов CMIP и TMN.

Соответственно, каждая группа объектов MIB-I и MIB-II также имеет кроме кратких символьных имен, приведенных выше, полные символьные имена и соответствующие им числовые имена.

Формат SNMP-сообщений

Протокол SNMP обслуживает передачу данных между агентами и менеджерами. SNMP использует дейтаграммный транспортный протокол UDP, не обеспечивающий надежной доставки сообщений. Протокол, организующий надежную передачу дейтаграмм на основе соединений TCP, весьма загружает управляемые устройства, которые на момент разработки протокола SNMP были не очень мощные, поэтому от услуг протокола TCP решили отказаться.

SNMP-сообщения, в отличие от сообщений многих других коммуникационных протоколов, не имеют заголовков с фиксированными полями. SNMP-сообщение состоит из произвольного количества полей, и каждое поле предваряется описателем его типа и размера.

Любое SNMP-сообщение состоит из трех основных частей: *версии протокола*, *идентификатора общности* и *области данных*.

Идентификатор общности (community string) используется для группирования устройств, управляемых определенным менеджером. Идентификатор общности является аналогом пароля, так как для того, чтобы устройства могли взаимодействовать по протоколу SNMP, они должны иметь одно и то же значение этого идентификатора (по умолчанию часто используется строка «public»).

В области данных, собственно, и содержатся описанные выше команды протокола, имена объектов и их значения. Область данных состоит из одного или более блоков PDU, каждый из которых может относиться к одному из пяти различных типов PDU, соответствующих пяти командам протокола SNMP: GetRequest-PDU, GetNextRequest-PDU, GetResponse-PDU, SetRequest-PDU, Trap-PDU. И наконец, для каждого типа PDU имеется определение его формата. Например, формат блока GetRequest-PDU включает следующие поля:

- идентификатор запроса;
- статус ошибки (есть или нет);
- индекс ошибки (тип ошибки, если она есть);
- список имен объектов SNMP MIB, включенных в запрос.

На рис. 22.22 показано сообщение протокола SNMP, которое представляет собой запрос о значении объекта SysDescr (числовое имя 1.3.6.1.2.1.1.1).

30	29	02	01	00			
SEQUENCE	len = 41	INTEGER	len = 1	vers = 0			
04	06	70	75	62	6C	69	63
string	len = 6	p	u	b	l	i	c
A0	1C	02	04	05	AE	56	02
getreq	len = 28	INTEGER	len = 4	--------	requested ID	--------	--------
02	01	00	02	01	00		
INTEGER	len = 1	status	INTEGER	len = 1	error	index	
30	0E	30	0C	06	08		
SEQUENCE	len = 14	SEQUENCE	len = 12	objectid	len = 8		
2B	06	01	02	01	01	01	00
1,3	6	1	2	1	1	1	0
05	00						
null	len = 0						

Рис. 22.22. Пример сообщения протокола SNMP

Как видно из рисунка, сообщение начинается с кода 30 (все коды шестнадцатеричные), который соответствует ключевому слову SEQUENCE (последовательность) и говорит о том, что сообщение состоит из последовательности полей. Длина последовательности указывается в следующем байте (41 байт). Далее следует поле, которое представляет собой целое число (integer) длиной 1 байт — это версия (vers) протокола SNMP (в данном случае 0, то есть SNMPv1, а 1 означала бы SNMPv2). Поле идентификатора общности community имеет тип string (строка символов) длиной в 6 байт со значением public. Остальную часть сообщения составляет блок данных GetRequest-PDU. То, что это операция Get-request, говорит код A0, а общая длина этого блока данных равна 28 байт. В соответствии со структурой блока данных Getrequest-PDU далее идет поле идентификатора запроса (он определен как целое 4-байтовое число и имеет значение 05 AE 56 02). Затем в блоке следует два однобайтовых целых числа статуса и индекса ошибки, которые в запросе установлены в 0. И наконец, завершает сообщение список имен объектов, значения которых запрашиваются данной командой. Этот список в примере состоит из одной переменной с именем 1.3.6.1.2.1.1.1.0, которое соответствует символьному имени SysDescr. Признак null (значение 05) говорит о том, что достигнут конец сообщения.

Спецификация RMON базы данных MIB

Добавлением к функциональным возможностям SNMP является спецификация RMON, которая обеспечивает удаленное взаимодействие с базой MIB. До появления RMON протокол SNMP не мог использоваться удаленным образом, он допускал только локальное администрирование устройств. База RMON MIB обла-

дает улучшенным набором свойств для удаленного администрирования, так как содержит агрегированную информацию об устройстве, не требующую передачи по сети больших объемов данных. Объекты RMON MIB включают дополнительные счетчики ошибок в пакетах, более гибкие средства анализа трендов и статистики, более мощные средства фильтрации для захвата и анализа отдельных пакетов, а также более сложные условия установления сигналов предупреждения. Интеллектуальность агентов RMON MIB выше, чем агентов MIB-I или MIB-II, что позволяет им выполнять значительную часть работы по обработке информации об устройстве, которую раньше выполняли менеджеры. Эти агенты могут располагаться внутри различных коммуникационных устройств или выполняться в виде отдельных программных модулей на универсальных персональных компьютерах и ноутбуках.

Объекту RMON присвоен номер 16 в наборе объектов MIB, а сам объект RMON объединяет 10 групп объектов (десятую группу составляют специальные объекты протокола Token Ring).

- Statistics (1) — текущие накопленные статистические данные о характеристиках пакетов, количестве коллизий и т. п.
- History (2) — статистические данные, сохраненные через определенные промежутки времени для последующего анализа тенденций их изменений.
- Alarms (3) — пороговые значения статистических показателей, при превышении которых агент RMON посылает сообщение менеджеру.
- Hosts (4) — данные о хостах сети, в том числе их MAC-адресах.
- Host TopN (5) — таблица наиболее загруженных хостов сети.
- Traffic Matrix (6) — статистика об интенсивности трафика между каждой парой хостов сети, упорядоченная в виде матрицы.
- Filter (7) — условия фильтрации пакетов.
- Packet Capture (8) — условия захвата пакетов.
- Event (9) — условия регистрации и генерации событий.

Данные группы пронумерованы в указанном порядке, поэтому, например, группа Hosts имеет числовое имя 1.3.6.1.2.1.16.4.

Всего стандарт RMON MIB определяет около 200 объектов в 10 группах, зафиксированных в двух документах — RFC 1271 для сетей Ethernet и RFC 1513 для сетей Token Ring.

Отличительной чертой стандарта RMON MIB является его независимость от протокола сетевого уровня (в отличие от стандартов MIB-I и MIB-II, ориентированных на протоколы TCP/IP). Поэтому он удобен для гетерогенных сред, использующих различные протоколы сетевого уровня.

Рассмотрим более подробно группу Statistics, которая определяет, какую информацию о кадрах (называемых в стандарте пакетами) Ethernet может предоставить агент RMON. Группа History основана на объектах группы Statistics, так как ее объекты просто позволяют строить временные ряды для объектов группы Statistics.

В группу Statistics входят наряду с некоторыми другими следующие объекты:

- etherStatsDropEvents — общее число событий, при которых пакеты были проигнорированы агентом из-за недостатка его ресурсов (сами пакеты при этом не обязательно были потеряны интерфейсом);
- etherStatsOctets — общее число байтов (включая ошибочные пакеты), принятых из сети (исключая преамбулу и включая байты контрольной суммы);
- etherStatsPkts — общее число полученных пакетов (включая ошибочные);
- etherStatsBroadcastPkts — общее число хороших пакетов, которые были посланы по широковещательному адресу;
- etherStatsMulticastPkts — общее число хороших пакетов, полученных по групповому адресу;
- etherStatsCRCAlignErrors — общее число полученных пакетов, которые имели длину (исключая преамбулу) в диапазоне между 64 и 1518 байт, не содержали целое число байтов или имели неверную контрольную сумму;
- etherStatsUndersizePkts — общее число пакетов, которые имели длину меньше, чем 64 байт, но были правильно сформированы;
- etherStatsOversizePkts — общее число полученных пакетов, которые имели длину больше, чем 1518 байт, но были тем не менее правильно сформированы;
- etherStatsFragments — общее число полученных пакетов, которые не состояли из целого числа байтов или имели неверную контрольную сумму и имели к тому же длину, меньшую 64 байт;
- etherStatsJabbers — общее число полученных пакетов, которые не состояли из целого числа байтов или имели неверную контрольную сумму и имели к тому же длину, большую 1518 байт;
- etherStatsCollisions — наилучшая оценка числа коллизий на данном сегменте Ethernet;
- etherStatsPkts64Octets — общее количество полученных пакетов (включая плохие) размером 64 байт;
- etherStatsPkts65to127Octets — общее количество полученных пакетов (включая плохие) размером от 65 до 127 байт;
- etherStatsPkts128to255Octets — общее количество полученных пакетов (включая плохие) размером от 128 до 255 байт;
- etherStatsPkts256to511Octets — общее количество полученных пакетов (включая плохие) размером от 256 до 511 байт;
- etherStatsPkts512to1023Octets — общее количество полученных пакетов (включая плохие) размером от 512 до 1023 байт;
- etherStatsPkts1024to1518Octets — общее количество полученных пакетов (включая плохие) размером от 1024 до 1518 байт.

Как видно из описания объектов, с помощью агента RMON, встроенного в повторитель или другое коммуникационное устройство, можно провести очень детальный анализ работы сегмента Ethernet или Fast Ethernet. Сначала можно по-

лучить данные о встречающихся в сегменте типах ошибок в кадрах, а затем целесообразно собрать с помощью группы History зависимости интенсивности этих ошибок от времени (в том числе привязав их ко времени). После анализа временны́х зависимостей часто уже можно сделать некоторые предварительные выводы об источнике ошибочных кадров и на этом основании сформулировать более тонкие условия захвата кадров со специфическими признаками (задав условия в группе Filter), соответствующими выдвинутой версии. После этого можно провести еще более детальный анализ путем изучения захваченных из объектов группы Packet Capture кадров.

Позже был принят стандарт RMON 2, который распространяет идеи интеллектуальной базы RMON MIB на протоколы верхних уровней, выполняя часть работы анализаторов протоколов.

Недостатки протокола SNMP

Протокол SNMP служит основой многих систем администрирования, хотя имеет несколько принципиальных недостатков.

- *Отсутствие средств взаимной аутентификации агентов и менеджеров.* Единственным средством, которое можно было бы отнести к средствам аутентификации, является так называемая строка общности в сообщениях. Эта строка передается по сети в открытой форме в SNMP-сообщении и служит основой для объединения агентов и менеджеров, так что агент взаимодействует только с теми менеджерами, у которых та же строка общности, что и строка, хранящаяся в памяти агента. Это, безусловно, не способ аутентификации, а способ структурирования агентов и менеджеров. Версия SNMP v.2 должна была ликвидировать этот недостаток, но в результате разногласий между разработчиками стандарта новые средства аутентификации хотя и появились в этой версии, но как необязательные.
- *Работа через ненадежный протокол UDP* (а именно так работает подавляющее большинство реализаций агентов SNMP) приводит к потерям аварийных сообщений от агентов к менеджерам, что может привести к некачественному администрированию. Исправление ситуации путем перехода на надежный транспортный протокол с установлением соединения чревато потерей связи с огромным количеством встроенных агентов SNMP, имеющихся в установленном в сетях оборудовании. (Протокол CMIP изначально работает поверх надежного транспорта стека OSI и этим недостатком не страдает.)

Разработчики платформ администрирования стараются преодолеть эти недостатки. Например, в системе HP OV Telecom DM TMN, являющейся платформой для разработки многоуровневых систем администрирования в соответствии со стандартами ISO, работает новая реализация SNMP, организующая надежный обмен сообщениями между агентами и менеджерами за счет самостоятельной организации повторных передач SNMP-сообщений при их потере.

Выводы

Большинство современных глобальных сетей являются составными IP-сетями, а отличия между ними заключаются в лежащих под уровнем IP технологиях.

Крупные глобальные сети часто строятся по четырехуровневой схеме, где два нижних уровня — это уровни первичной сети, образуемые технологиями DWDM и SDH. На основе первичной сети оператор сети строит каналы наложенной (оверлейной) сети — пакетной или телефонной. IP-сеть образует верхний уровень.

В более простом случае первичная сеть для образования постоянных каналов вообще отсутствует, и под слоем IP может располагаться сеть ATM или Frame Relay, коммутаторы которой соединяются непосредственно кабельными или беспроводными линиями связи.

«Чистая» IP-сеть отличается от многослойной тем, что под уровнем IP нет другой сети с коммутацией пакетов, такой как ATM или Frame Relay, и IP-маршрутизаторы связываются между собой выделенными каналами (физическими или соединениями PDH/SDH/DWDM).

Из набора существующих двухточечных протоколов протокол IP сегодня использует два: HDLC и PPP. Каждый из них представляет целое семейство протоколов, работающих на канальном уровне. При работе HDLC для обеспечения надежности передачи используется скользящее окно. Логический канал HDLC является дуплексным, так что информационные кадры, а значит, и положительные квитанции могут передаваться в обоих направлениях.

Особенностью протокола PPP, отличающей его от других протоколов канального уровня, является сложная переговорная процедура принятия параметров соединения. Одним из важных параметров PPP-соединения является режим аутентификации. Для целей аутентификации PPP предлагает по умолчанию протокол аутентификации по паролю PAP или протокол аутентификации по квитированию вызова CHAP.

Технология MPLS считается сегодня многими специалистами одной из самых перспективных транспортных технологий. Главный принцип MPLS: протоколы маршрутизации используются для определения топологии сети, а для продвижения данных внутри границ сети одного поставщика услуг применяется техника виртуальных каналов.

Объединение техники виртуальных каналов с функциональностью стека TCP/IP происходит за счет того, что одно и то же сетевое устройство называемое коммутирующим по меткам маршрутизатором (LSR), выполняет функции как IP-маршрутизатора, так и коммутатора виртуальных каналов.

Основные области применения MPLS: MPLS IGP, MPLS TE и MPLS VPN. MPLS IGP применяется для ускорения продвижения пакетов сетевого уровня, MPLS TE — для решения задач инжиниринга трафика, MPLS VPN позволяет поставщику предоставлять услуги VPN на основе разграничения трафика без обязательного шифрования информации.

Централизованная система управления сетью (NMS) в автоматическом режиме осуществляет контроль трафика в сети, собирает информацию о состоянии каждого коммуникационного устройства, предоставляет эту информацию администратору.

В тех случаях, когда управляемыми объектами являются компьютеры, их системное и прикладное программное обеспечение, для системы управления сетью часто используют особое название — система управления системой (SMS).

В основе любой системы управления сетью лежит схема взаимодействия «менеджер — агент — управляемый объект». Состояние управляемого объекта отображается в БД управляющей информации (MIB). Агент наполняет MIB управляемого объекта текущими значениями его характеристик, а менеджер извлекает из MIB информацию, анализирует ее и выдает команды агенту выполнить то или иное управляющее воздействие.

Вопросы и задания

1. Какие причины привели к созданию нескольких моделей IP WAN («чистые» IP-сети, IP поверх ATM, IP поверх Frame Relay и IP поверх MPLS)?
2. Корректно ли утверждение, что IP-сети поверх ATM или IP-сети поверх Frame Relay состоят из двух уровней сетей с коммутацией пакетов, а IP-сети поверх MPLS состоят только из одного уровня сети с коммутацией пакетов? Обоснуйте свой ответ.
3. Какие функции выполняет уровень ATM в модели IP поверх ATM?
4. Сравните главные свойства HDLC и PPP.
5. В чем состоят цели процедуры установления соединения в протоколе HDLC и протоколе PPP?
6. Какой механизм использует протокол HDLC для восстановления утерянных или искаженных кадров?
7. Почему протокол PPP называется многопротокольным?
8. В чем необходимость включения функции взаимной аутентификации в протокол PPP?
9. Перечислите основные этапы конфигурирования маршрутизатора при использовании выделенной линии.
10. Какие новые идеи были реализованы в технологии IP-коммутации?
11. Какие концепции технологии IP-коммутации были сохранены в MPLS в неизменном виде, а какие были модифицированы?
12. Перечислите функциональные модули IP-маршрутизатора, которые используются в LSR.
13. Какие новые возможности дает использование стека меток MPLS?
14. Предположим, что LSR использует формат кадров Ethernet. Означает ли это, что LSR продвигает кадры на основе таблицы продвижения, полученной в соответствии со стандартом IEEE 802.1D?
15. Каким образом можно установить путь LSP, проходящий через несколько доменов MPLS?
16. В чем состоит отличие между MPLS IGP и MPLS TE?
17. Что является аналогом туннелей MPLS TE в технологиях ATM и Frame Relay?
18. Можно ли в сети, поддерживающей MPLS, передавать часть трафика с помощью обычного IP-продвижения?
19. Перечислите группы функций системы управления сетью в соответствии со стандартом X.700.
20. Есть ли отличия между системами управления сетями и системами управления системами? Если да, то в чем они состоят?
21. Какие функции системы управления сетью выполняются агентами и какие — менеджерами?

22. Перечислите стандартные типы MIB.
23. Какие типы имен используются протоколом SNMP для именования переменных MIB?
24. Когда используется команда Trap?
25. Предположим, что вы являетесь архитектором глобальной IP-сети. Какие вопросы вы должны задать заказчику, чтобы понять, какой тип многоуровневой модели («чистые» IP-сети, IP поверх ATM, IP поверх Frame Relay и IP поверх MPLS) вам целесообразно применить?
26. Измерения показали, что уровень битовых ошибок в линии связи равен 10^{-4}. Какой тип протокола вы выберете для этой линии — HDLC или PPP?
27. Сформируйте таблицу продвижения для устройства LSR1, изображенного на рис. 22.13.
28. Какие исходные данные необходимы проектировщику, чтобы решить проблемы инжиниринга трафика в сети, показанной на рис. 22.14? Предложите свой вариант таких данных и решите эту задачу.

ГЛАВА 23 Удаленный доступ

Термин *удаленный доступ* (remote access) часто используют в том случае, когда речь идет о доступе пользователя домашнего компьютера к Интернету или сети предприятия, которая находится от него на значительном расстоянии, так что использование глобальных связей обязательно. В последнее время под удаленным доступом стали понимать не только доступ изолированных компьютеров, но и домашних сетей, объединяющих несколько компьютеров членов семьи. Такими же небольшими сетями располагают малые офисы предприятий, насчитывающие 2–3 сотрудника.

Организация удаленного доступа является одной из наиболее острых проблем компьютерных сетей в настоящее время. Она получила название «проблемы последней мили», где под последней милей подразумевается расстояние от точки присутствия (Point Of Presence, POP) оператора связи до помещений клиентов. Сложность этой проблемы определяется несколькими факторами. С одной стороны, современным пользователям необходим высокоскоростной доступ, обеспечивающий качественную передачу трафика любого типа — данных, голоса, видео. Для этого нужны скорости несколько мегабит или, по крайней мере, несколько сотен килобит в секунду. С другой стороны, подавляющее большинство домов в больших и малых городах и особенно в сельской местности по-прежнему соединены с точками присутствия операторов связи абонентскими окончаниями телефонной сети, которые ограничивают пользователя скоростями в десятки килобит в секунду.

Кардинальная перестройка кабельной инфраструктуры доступа вряд ли возможна в ближайшее время, слишком масштабна эта задача из-за огромного количества зданий и домов, географически рассеянных по огромной территории. И хотя в некоторых странах в последнее время стали прокладывать к домам высокоскоростные оптические линии, таких стран не так уж много, да и этот процесс затронул пока только большие города и крупные здания с множеством потенциальных пользователей.

Долгое время наиболее распространенной технологией доступа был коммутируемый доступ, когда пользователь устанавливал коммутируемое соединение с корпоративной сетью или Интернетом через телефонную сеть с помощью модема. Такой способ обладает существенным недостатком — скорость доступа ограничена несколькими десятками килобит в секунду из-за фиксированной узкой полосы пропускания примерно в 3,4 кГц, выделяемой каждому абоненту телефонной сети (вспомните технику мультиплексирования, применяемую в телефонных сетях и описанную в главе 9). Такие скорости сегодня устраивают все меньше и меньше пользователей.

Для организации скоростного удаленного доступа сегодня привлекаются различные технологии, в которых используется только существующая инфраструктура абонентских окончаний — телефонные сети или сети кабельного телевидения. После достижения POP поставщика услуг по такому окончанию компьютерные данные уже не следуют по телефонной сети или сети кабельного телевидения, а ответвляются с помощью специального оборудования в сеть пере-

дачи данных. Это позволяет преодолеть ограничения на полосу пропускания, отводимую абоненту в телефонной сети или сети кабельного телевидения, и повысить скорость доступа. Наиболее популярными технологиями такого типа являются технология ADSL, использующая телефонные абонентские окончания, и кабельные модемы, работающие поверх сети кабельного телевидения. Эти технологии обеспечивают скорость от нескольких сотен килобит до нескольких мегабит в секунду.

Применяются также различные беспроводные технологии доступа, обеспечивающие как фиксированный, так и мобильный доступ. Набор применяемых беспроводных технологий очень широк, в него входят и беспроводные сети Ethernet (802.11), различные фирменные технологии, передача данных по сети мобильной телефонии, а также технологии фиксированного доступа, например нового стандарта 802.16.

В этой главе мы рассмотрим основные схемы и наиболее популярные технологии удаленного доступа.

Схемы удаленного доступа

Ключевые слова: незащищенное соединение, распределитель, коммутируемый (телефонный) модем, устройство ADSL, кабельный модем, сервер удаленного доступа, концентратор удаленного доступа, мультиплексор доступа, терминальная система, общедоступный домен Интернета, режим удаленного узла, режим удаленного управления, режим терминального доступа, протокол telnet.

Рисунок 23.1 иллюстрирует разнообразный и пестрый мир удаленного доступа.

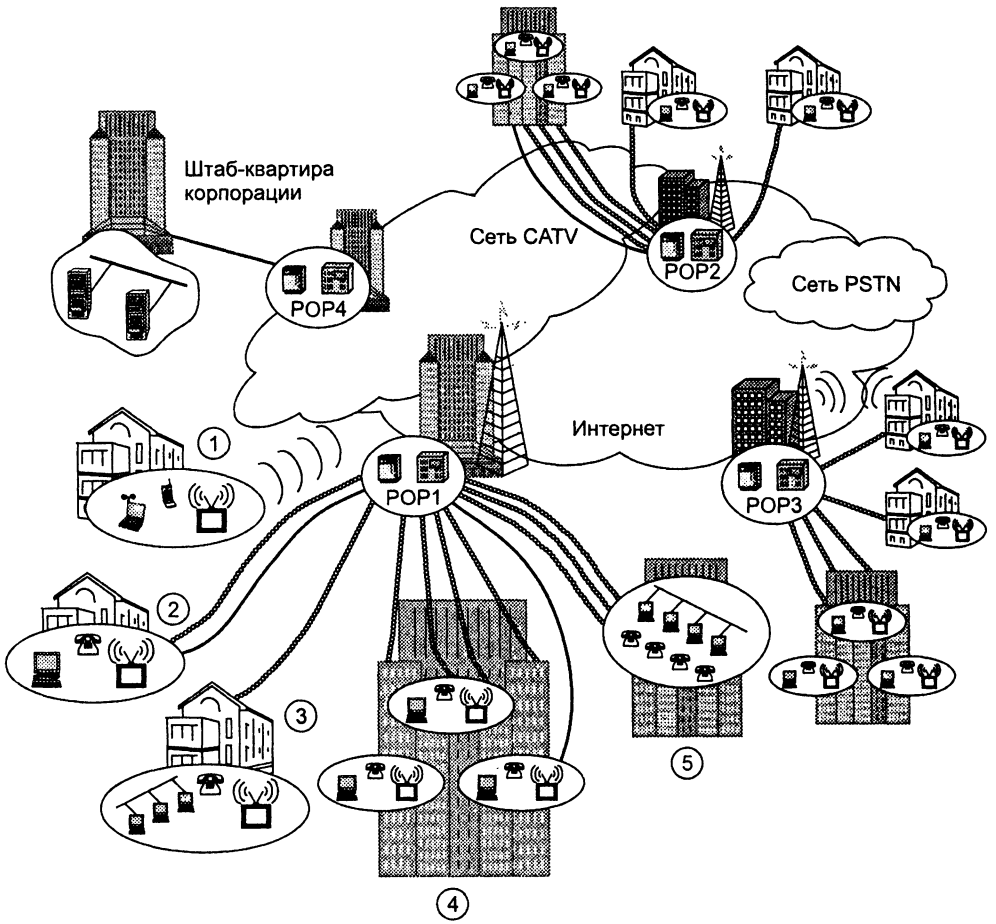


Рис. 23.1. Клиенты удаленного доступа

Мы видим здесь клиентов различных типов, отличающихся используемым оборудованием и требованиями к параметрам доступа. Кроме того, помещения клиентов могут быть соединены с ближайшей точкой доступа оператора связи (то есть с ближайшим центральным офисом, если пользоваться терминологией опе-

раторов телефонной сети) различными способами — с помощью аналогового или цифрового окончания телефонной сети, телевизионного кабеля, беспроводной связи. Наконец, сам оператор связи может иметь различную специализацию, то есть быть либо поставщиком телефонных услуг, либо поставщиком услуг Интернета, либо оператором кабельного телевидения или же быть универсальным оператором, предоставляющим весь спектр услуг и обладающим собственными сетями всех типов.

Типы клиентов и абонентских окончаний

Рассмотрим каждый элемент схемы доступа, показанный на рис. 23.1, более подробно.

Клиенты 1 и 2 являются наиболее типичными пользователями, так как каждый из них имеет только один компьютер, которому необходимо обеспечить доступ к удаленной компьютерной сети. Помимо компьютера эти клиенты пользуются телефоном и телевизором, поэтому абонентские окончания этих устройств можно использовать для организации доступа компьютера к сети передачи данных.

Клиент 2 пользуется двумя кабельными абонентскими окончаниями: традиционным аналоговым телефонным на основе витой пары и коаксиальным телевизионным кабелем кабельного телевидения. Эти абонентские окончания обладают существенно различными характеристиками. Так, витая пара при расстоянии 1–2 км между помещением клиента и POP поставщика услуг обычно имеет полосу пропускания примерно несколько мегагерц, в то время как коаксиальный кабель обеспечивает полосу пропускания в несколько десятков мегагерц.

У клиента 1 отсутствуют проводные абонентские окончания, так как он пользуется мобильным телефоном, кроме того, он не является клиентом кабельного телевидения, принимая телевизионный сигнал только по воздуху.

Таким образом, для организации удаленного доступа для клиента 2 поставщик услуг может использовать либо существующее телефонное абонентское окончание, либо телевизионный кабель. Для клиента 1 такой возможности нет, поэтому поставщик услуг должен предоставить ему беспроводную связь или же проложить новый кабель между его домом и ближайшей точкой присутствия.

Отличительной особенностью клиентов 1 и 2 является несимметричный характер трафика, так как домашние пользователи в основном загружают информацию на свой компьютер в процессе путешествий по Интернету. Ответом на такие потребности являются асимметричные технологии, такие как ADSL.

Клиент 3 отличается от двух предыдущих тем, что имеет несколько компьютеров, объединенных в локальную сеть. Таким клиентом может быть как частное лицо, так и небольшой офис предприятия. Удаленный доступ для локальной сети отличается повышенными требованиями к пропускной способности. Кроме того, трафик может иметь симметричный характер, если домашняя сеть включает сервер, поставляющий информацию пользователям Интернета или сотрудникам других офисов предприятия. Так как клиент 3 не имеет кабельного окончания сети CATV (cable TV), то ему можно обеспечить доступ только по телефонному окончанию. Клиент 3 может организовать свою IP-сеть различными способами.

Он может попросить у поставщика услуг пул IP-адресов, так чтобы каждый его компьютер имел отдельный публичный постоянный IP-адрес. Это наиболее гибкий вариант для клиента, так как в этом случае каждый его компьютер может быть полноправным узлом Интернета и исполнять роль не только клиентской машины, но и сервера с зарегистрированным доменным именем. Очевидно, что в этом случае локальная сеть клиента должна иметь пограничный маршрутизатор, через который осуществлять связь с сетью поставщика услуг. Другой вариант организации IP-сети может быть основан на использовании техники NAT, описанной в главе 20.

Клиенты 4 являются жителями многоквартирного дома, который соединен с POP многочисленными витыми парами телефонных абонентских окончаний (по одной для каждой квартиры), а также кабелем CATV. Использование одного кабеля CATV для большого количества клиентов создает дополнительные проблемы для организации доступа, так как кабель в этом случае является разделяемой средой. Применение телефонных абонентских окончаний для удаленного доступа жителей многоквартирного дома ничем не отличается от подключения отдельного абонента (клиента 2). И хотя большая часть жильцов дома использует обычные аналоговые телефонные окончания, в доме есть также несколько квартир, жильцы которых являются абонентами сети ISDN, окончания которой являются цифровыми (при том, что они, так же как и аналоговые телефонные окончания, работают на витой паре). Хотя сеть ISDN была разработана как универсальная, то есть предоставляющая наряду с сервисами телефонии и сервисы передачи данных, на практике она используется как обычная телефонная сеть.

Клиенты 5 также являются жильцами многоквартирного дома, но в этом доме поставщик услуг развернул локальную сеть. К этой локальной сети подключаются компьютеры тех жильцов дома, которые решили стать абонентами данного поставщика услуг. Такой вариант эффективен для поставщика услуг при достаточно большом количестве абонентов в доме. Локальная сеть многоквартирного дома требует более высоких скоростей доступа, чем отдельные компьютеры или домашние сети индивидуальных клиентов, поэтому поставщик услуг должен использовать абонентское окончание с широкой полосой пропускания — для этой цели может быть использован существующий кабель CATV, специально проложенный коаксиальный кабель Ethernet или также заново проложенный оптический кабель.

Поставщик услуг удаленного доступа может обслуживать клиентов всех типов или же специализироваться на каком-то определенном типе клиентов, например жителях частных или многоквартирных домов, работниках небольших офисов. Универсальный поставщик услуг доступа должен поддерживать любые варианты организации «последней мили», что усложняет его оборудование и применяемые технологии доступа.

В любом случае, для передачи данных по какому-либо абонентскому окончанию поставщик услуг должен обеспечить для этого окончания передачу компьютерных данных и совместить эту передачу с передачей информации, для которой это окончание было спроектировано, например с аналоговой телефонной информацией или с сигналом кабельного телевидения. Затем на основе этих средств

физического уровня поставщик услуг должен предоставить клиенту тот или иной вариант сервиса доступа.

Еще одной проблемой, которую должен решить оператор доступа, является организация доступа клиентов, которые физически подключены к абонентским окончаниям *других* поставщиков услуг связи. Так, пусть на рисунке POP1 и POP2 принадлежат поставщику A, а POP3 — поставщику B. Для того чтобы поставщик A мог предоставлять услуги доступа к сети передачи данных клиентам, подключенным к POP3, у него должно быть заключено соответствующее соглашение с поставщиком B. Это соглашение может регламентировать различные способы взаимодействия поставщиков услуг, которые мы уже обсуждали в главе 5. Например, поставщик услуг A может арендовать у поставщика услуг B те абонентские окончания, которыми пользуются его клиенты, с тем чтобы затем передавать получаемые по ним данные в свою сеть и направлять их далее в соответствии с потребностями клиентов. В другом случае абонентские окончания могут оставаться в распоряжении поставщика услуг B, который должен отделять поступающие компьютерные данные от телефонной или телевизионной информации и направлять в сеть поставщика услуг A. Очевидно, что между сетями передачи данных поставщиков услуг A и B должно быть обеспечено взаимодействие.

Наиболее простой вариант доступа в Интернет предоставляет клиенту **незащищенное соединение** с серверами корпоративной сети, что грозит плохими последствиями. Во-первых, конфиденциальные данные, передаваемые по Интернету, могут быть перехвачены или искажены. Во-вторых, при таком способе администратору корпоративной сети трудно ограничить доступ к своей сети несанкционированных пользователей, так как заранее неизвестны IP-адреса легальных пользователей (сотрудников предприятия). Поэтому предприятия предпочитают пользоваться защищенным доступом, основанном на технологии виртуальных частных сетей (VPN). Эта технология рассматривается подробно в следующей главе.

Мультиплексирование информации на абонентском окончании

Как мы видим на рис. 23.1, большинство домов и многоквартирных зданий связаны с POP либо телефонными абонентскими окончаниями, либо абонентскими окончаниями кабельного телевидения.

Поэтому для обеспечения клиентов *тремя* основными на сегодня видами доступа (к телефонной сети, телевизионной сети и сети передачи данных) необходимо реализовать одновременную передачу информации разного типа по одной линии связи. Например, совместить передачу данных с передачей голоса и по телефонному окончанию или же совместить передачу данных с передачей телевизионного сигнала по коаксиальному кабелю.

В идеале желательно использовать единственное абонентское окончание, способное передавать информацию всех трех типов. К сожалению, витая пара на эту роль не подходит, так как ее полоса пропускания на расстояниях в несколько

километров не превышает 1 МГц. Этого явно недостаточно для одновременной передачи голоса, компьютерных данных со скоростями в несколько мегабит в секунду и цветного телевизионного изображения.

Поэтому на роль консолидирующего абонентского окончания могут претендовать только коаксиальный кабель сети CATV и широкополосные беспроводные линии связи. Естественно, мы имеем в виду уже существующие и широко распространенные типы абонентских окончаний. Если же говорить о прокладке нового кабеля, что актуально в основном для новых крупных зданий, то к этому списку нужно добавить оптический кабель.

Почти во всех технологиях доступа, которые мы будем рассматривать в следующих разделах, используется мультиплексирование каких-либо двух или всех трех упомянутых выше типов информации на абонентском окончании. Так, в линии ADSL аналоговые телефонные окончания используются для мультиплексирования голоса и компьютерных данных, кабельные модемы совмещают передачу телевизионного изображения и компьютерных данных по коаксиальному кабелю. Существуют также различные технологии беспроводного доступа, которые обеспечивают передачу телевизионного сигнала и компьютерных данных, а иногда и телефонии в одном абонентском окончании. Исключением является только наиболее старая технология доступа, а именно коммутируемый доступ, при котором аналоговое абонентское окончание может использоваться телефоном или модемом компьютера только попеременно.

Схема организации доступа с помощью универсального абонентского окончания показана на рис. 23.2.

Наиболее часто для мультиплексирования информации в абонентском окончании применяется техника FDM. Каждому из трех типов информации выделяется определенная полоса частот, ширина которой соответствует потребностям абонента. Для телефонного соединения выделяется полоса 4 кГц, соответствующая стандартной полосе абонента аналоговых телефонных сетей. Компьютерным данным нужна более широкая полоса, при асимметричном доступе для преобладающего нисходящего (входящего) трафика нужно выделить полосу, как минимум в несколько сотен килогерц, а лучше — в несколько мегагерц. Менее интенсивный восходящий (выходящий) трафик требует полосы в несколько десятков килогерц. В кабельном телевидение традиционно используются полосы по 6 МГц для каждого абонента, но при этом передается только нисходящий трафик.

Для того чтобы реализовать выбранную схему FDM, в помещении клиента и точках присутствия устанавливаются **распределители**, которые выполняют операции мультиплексирования и демультиплексирования сигналов. Распределитель чаще всего представляет собой пассивный фильтр, который выделяет нужные диапазоны частот и передает каждый диапазон на отдельный выход. К выходу распределителя подключаются терминальные устройства абонента — телефон, телевизор и компьютер. Так как компьютер использует дискретные сигналы для обмена данными, то для него требуется дополнительное устройство, которое будет преобразовывать дискретные сигналы в аналоговые сигналы необходимого диапазона частот.

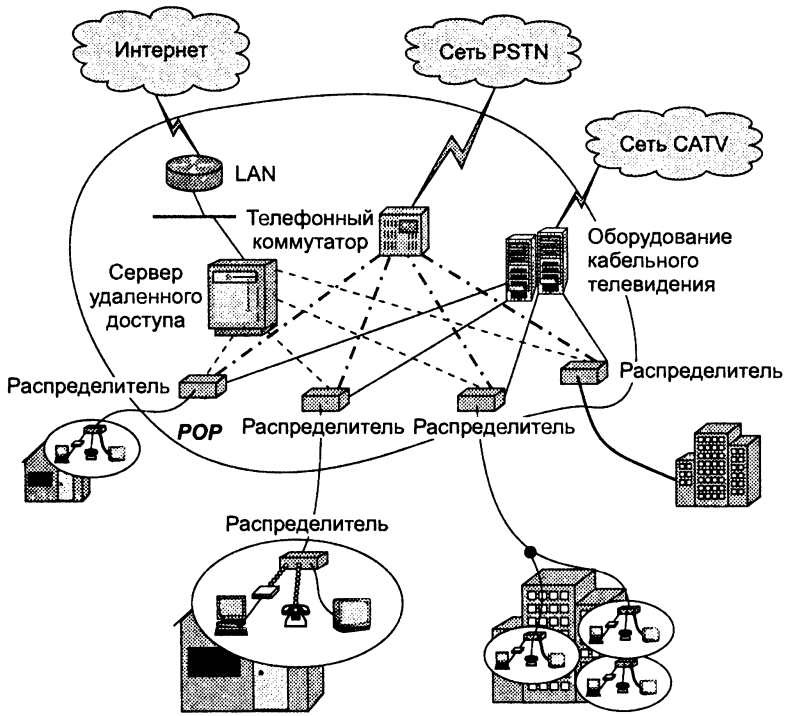


Рис. 23.2. Мультиплексирование трех типов информации в абонентских окончаниях

Большинство пользователей привыкли иметь дело с **коммутируемыми (телефонными) модемами**, которые работают со стандартной полосой 4 кГц аналоговых телефонных сетей. Телефонные модемы не разделяют эту полосу с другими устройствами, целиком используя ее для передачи компьютерных данных. Очевидно, что распределитель в этом случае не нужен.

Существуют также **устройства ADSL** и **кабельные модемы**; первые работают на абонентских окончаниях телефонных сетей, а вторые — на кабелях CATV. В этих случаях распределитель необходим, так как по этим окончаниям вместе с компьютерными данными передается и основная для этих окончаний информация, телефонная или телевизионная.

В POP поставщика услуг каждое абонентское окончание также подключено к распределителю, который выполняет аналогичные операции мультиплексирования и демультиплексирования на другом конце кабеля. В результате телефонная информация поступает с телефонных выходов распределителя на телефонный коммутатор поставщика услуг, который передает ее в телефонную сеть. Телевизионные сигналы от соответствующих выходов распределителя собираются на оборудовании CATV, которое может быть связано с сетью CATV этого поставщика услуг.

И, наконец, компьютерные данные поступают на устройство, концентрирующее компьютерный трафик и передающее его в локальную сеть поставщика услуг.

Это устройство называют по-разному, на рисунке использовано одно из популярных названий — **сервер удаленного доступа** (Remote Access Server, RAS). Можно встретить и другие названия, например, **концентратор удаленного доступа** (Remote Access Concentrator, RAC), **мультиплексор доступа** или **терминальная система**. Будем для определенности называть здесь такое устройство сервером удаленного доступа. Оно содержит обычно большое количество модемов, которые выполняют обратные операции по отношению к модемам пользователей, то есть модулируют нисходящий трафик и демодулируют восходящий. Помимо модемов RAS включает маршрутизатор, который собирает трафик от модемов и передает его в локальную сеть POP. Из этой локальной сети трафик передается обычным способом в Интернет или в определенную корпоративную сеть.

Мы рассмотрели обобщенную схему доступа, которая в зависимости от выбранного типа абонентского окончания и типа модема порождает различные технологии доступа. Нужно подчеркнуть, что в терминах модели OSI все они являются технологиями физического уровня, так как создают поток битов между компьютером клиента и локальной сетью поставщика услуг. Для работы протокола IP над этим физическим уровнем должен использоваться один из протоколов канального уровня. Сегодня наиболее часто при удаленном доступе применяется PPP, который поддерживает такие важные функции, как назначение IP-адреса клиентскому компьютеру, а также аутентификацию пользователя.

Режим удаленного узла

Наиболее распространенной услугой сегодня является предоставление **доступа к общедоступному домену Интернета**. При этом подразумевается, что поставщик услуг обеспечивает маршрутизацию IP-трафика между компьютером и любым сайтом Интернета, имеющим публичный адрес (или же имеющим частный адрес и использующим технику NAT для публичного доступа). Когда клиент располагает одним компьютером, для предоставления такой услуги поставщик услуг обычно использует режим удаленного узла.

Режим удаленного узла позволяет компьютеру клиента стать узлом удаленной локальной сети, что означает для его пользователя возможность получения всего спектра услуг обычного пользователя узла, физически расположенного в локальной сети.

Для этого поставщик услуг резервирует для своих клиентов удаленного доступа пул IP-адресов из диапазона адресов одной из своих подсетей. Для тех клиентов, которые не нуждаются в постоянном доступе к Интернету, услуга предоставляется как коммутируемая, и IP-адрес им назначается динамически и только на время подключения клиента. Режим удаленного узла позволяет экономить адреса подсетей, так как в стандартном режиме IP-маршрутизатор должен назначить каждому своему порту адрес отдельной подсети, что для одного узла, из которого состоят сети многих клиентов, явно избыточно. Для тех же клиентов, которым требуется постоянное соединение, адрес может назначаться как на постоянной основе, так и динамически, на время активности клиента.

Для обеспечения режима удаленного узла RAS поставщика услуг поддерживает *протокол Proxy-ARP*, рассмотренный в главе 17. Эта особенность отличает сервер удаленного доступа от обычного IP-маршрутизатора (рис. 23.3).

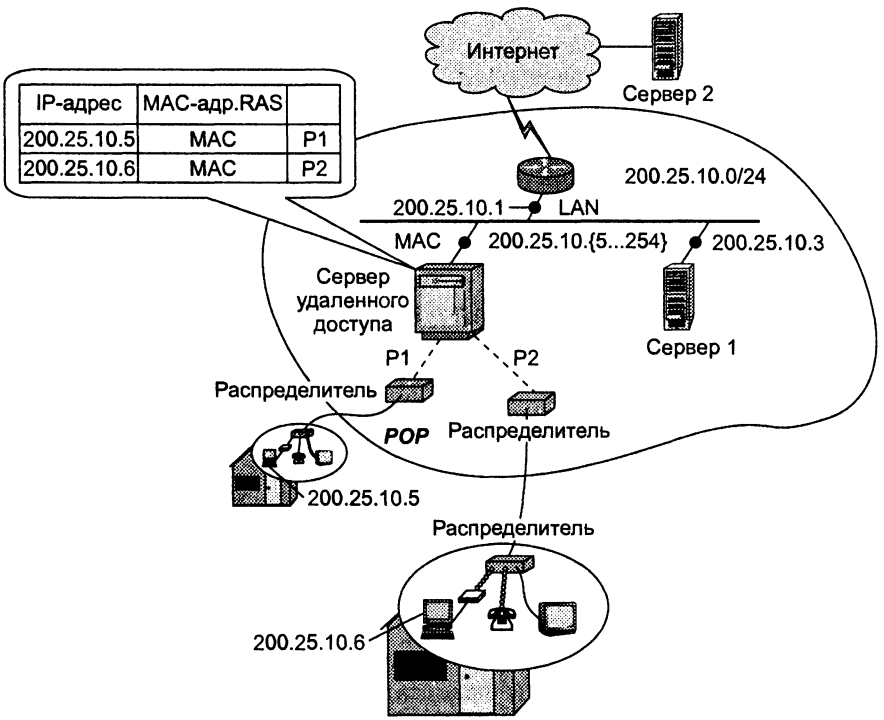


Рис. 23.3. Использование протокола Proxy-ARP при организации удаленного доступа

Для удаленных узлов в локальной сети поставщика услуг, имеющей адрес 200.25.10.0/24, выделен пул адресов от 200.25.10.5 до 200.25.10.254. Если клиент пользуется коммутируемым сервисом, то при его соединении с сетью поставщика услуг (например, по протоколу PPP) ему *временно* назначается адрес из этого пула. Так, компьютеру первого клиента был назначен адрес 200.25.10.5, а компьютеру второго клиента — адрес 200.25.10.6. При подключении к сети этих удаленных узлов сервер удаленного доступа заносит в специальную таблицу, являющуюся аналогом ARP-таблицы, следующие записи:

200.25.10.5 — MAC — P1

200.25.10.6 — MAC — P2

Здесь MAC обозначает адрес внутреннего интерфейса сервера удаленного доступа, а P1 и P2 — номера портов, к которым подключены клиенты удаленного доступа.

Если, например, сервер 2 (см. рис. 23.3), подключенный к сети одного из поставщиков услуг, посылает пакет компьютеру первого клиента, то маршрутизатор поставщика услуг считает, что пакет направлен к одному из узлов, принадлежа-

щих непосредственно присоединенной подсети 200.25.10.0/24. Поэтому маршрутизатор посылает ARP-запрос, содержащий адрес 200.25.10.5. На этот запрос отвечает не компьютер первого клиента, а RAS, сообщая в ARP-ответе маршрутизатору *собственный* MAC-адрес. После этого маршрутизатор направляет IP-пакет, упакованный в кадр Ethernet с MAC-адресом RAS. RAS извлекает IP-пакет из пришедшего кадра Ethernet и по IP-адресу определяет в таблице номер порта, на который ему нужно направить пакет. В данном случае это порт P1. RAS инкапсулирует пакет в кадр PPP, используемый для работы на абонентском окончании, соединяющем RAS с компьютером первого клиента.

В том случае, когда у клиента имеется своя локальная сеть, узлы которой имеют зарегистрированные публичные IP-адреса, RAS работает как обычный маршрутизатор, и такой режим уже не называется режимом удаленного узла.

Режим удаленного управления и протокол Telnet

Режим удаленного управления, называемый также режимом **терминального доступа**, предполагает, что пользователь превращает свой компьютер в виртуальный терминал другого компьютера, к которому он получает удаленный доступ.

В период становления компьютерных сетей, то есть в 70-е годы, поддержка такого режима была одой из главных функций сети. Устройства PAD сетей X.25 существовали именно для того, чтобы обеспечить удаленный доступ к мэйнфреймам для пользователей, находившихся в других городах и работавших за простыми алфавитно-цифровыми терминалами.

Режим удаленного управления обеспечивается специальным протоколом прикладного уровня, работающим поверх протоколов, реализующих транспортное соединение удаленного узла с компьютерной сетью. Существует большое количество протоколов удаленного управления, как стандартных, так и фирменных. Для IP-сетей наиболее старым протоколом этого типа является telnet (RFC 854).

Протокол **telnet**, который работает в архитектуре «клиент-сервер», обеспечивает эмуляцию алфавитно-цифрового терминала, ограничивая пользователя режимом командной строки.

При нажатии клавиши соответствующий код перехватывается клиентом telnet, помещается в TCP-сообщение и отправляется через сеть узлу, которым пользователь хочет управлять. При поступлении на узел назначения код нажатой клавиши извлекается из TCP-сообщения сервером telnet и передается операционной системе (ОС) узла. ОС рассматривает сеанс telnet как один из сеансов локального пользователя. Если ОС реагирует на нажатие клавиши выводом очередного символа на экран, то для сеанса удаленного пользователя этот символ также упаковывается в TCP-сообщение и по сети отправляется удаленному узлу. Клиент telnet извлекает символ и отображает его в окне своего терминала, эмулируя терминал удаленного узла.

Протокол telnet был реализован в среде Unix и наряду с электронной почтой и FTP-доступам к архивам файлов был популярным сервисом Интернета. Сегодня этот протокол редко используется в публичном домене Интернета, так как

никто не хочет предоставлять посторонним лицам возможность управлять собственным компьютером. Хотя для защиты от несанкционированного доступа в технологии telnet применяются пароли, они передаются через сеть в виде обычного текста, поэтому могут быть легко перехвачены и использованы. Поэтому telnet преимущественно используется в пределах одной локальной сети, где возможностей для перехвата пароля гораздо меньше. Сегодня основной областью применения telnet является управление не компьютерами, а коммуникационными устройствами — маршрутизаторами, коммутаторами и хабами. Таким образом, он уже скорее не пользовательский протокол, а протокол администрирования, то есть альтернатива SNMP.

Тем не менее отличие между протоколами telnet и SNMP принципиальное. Telnet предусматривает обязательное участие человека в процессе администрирования, так как, по сути, он только передает команды, которые вводит администратор при конфигурировании или мониторинге маршрутизатора или другого коммуникационного устройства. Протокол SNMP, наоборот, рассчитан на автоматические процедуры мониторинга и управления, хотя и не исключает возможности участия администратора в этом процессе. Для устранения опасности, создаваемой передачей паролей в открытом виде через сеть, коммуникационные устройства усиливают степень своей защиты. Обычно применяется многоуровневая схема доступа, когда открытый пароль дает возможность только чтения базовых характеристик конфигурации устройства, а доступ к средствам изменения конфигурации требует другого пароля, который уже не передается в открытом виде.

Удаленное управление также возможно и в графическом режиме. Для Unix стандартом де-факто является система X Window, являющаяся разработкой Массачусетсского технологического института (Massachusetts Institute of Technology, MIT). Для Windows существует ряд фирменных протоколов управления, например, VNC (Virtual Network Computing), Microsoft Terminal Server или протоколы компании WinFrame.

Удаленное управление имеет свои достоинства и недостатки. Для пользователя часто удобно использовать более мощный компьютер, установленный в сети предприятия, а не свой домашний. Кроме того, получив терминальный доступ, он может запустить на удаленном компьютере любое приложение, а не только задействовать сервис WWW или FTP. Еще одно преимущество заключается в том, что пользователь фактически получает все права пользователя внутренней сети предприятия, в то время как в режиме удаленного узла его права обычно ограничены администратором.

Удаленное управление также очень экономично потребляет пропускную способность сети, особенно при эмуляции режима командной строки. Действительно, в этом случае по сети передаются только коды клавиш и экранные символы, а не файлы или страницы веб-документов.

Недостаток удаленного управления состоит в его опасности для сети предприятия при несанкционированном доступе. Кроме того, администратору трудно контролировать потребление ресурсов компьютера, находящегося под удаленным управлением.

Коммутируемый аналоговый доступ

Ключевые слова: аналоговый телефон, импульсный набор, тоновый набор, сигнальная система 7, двухточечный протокол туннелирования, стандарты V.34, V.34+, V.90 и V.92, протокол коррекции ошибок и стандарт V.42, протокол доступа к линии связи для модемов, стандарты сжатия данных V.42bis и MNP-5, протокол сжатия синхронных потоков данных.

Основная идея коммутируемого доступа состоит в том, чтобы использовать имеющуюся сеть PSTN для организации коммутируемого соединения между компьютером домашнего пользователя и сервером удаленного доступа, установленным на границе телефонной и компьютерной сетей. Компьютер пользователя подключается к телефонной сети с помощью коммутируемого модема, который поддерживает стандартные процедуры набора номера и имитирует работу телефонного аппарата для установления соединения с RAS. Коммутируемый доступ может быть аналоговым или цифровым, в зависимости от типа абонентского окончания сети. В этом разделе мы рассмотрим доступ через аналоговые окончания, а в следующем — через цифровые.

Принцип работы телефонной сети

Первые телефонные сети были полностью аналоговыми, так как в них абонентское устройство (телефонный аппарат) преобразовывало звуковые колебания, являющиеся аналоговыми сигналами, в колебания электрического тока (также аналоговые сигналы). Коммутаторы телефонной сети тоже передавали пользовательскую информацию в аналоговой форме, перенося эти сигналы в другую область частотного спектра с помощью методов частотного уплотнения (FDM), описанных в главе 9.

Сегодня в телефонных сетях голос между коммутаторами все чаще передается в цифровой форме по каналам PDH/SDH с помощью технологии TDM. Однако абонентские окончания остаются в основном аналоговыми, что позволяет пользоваться теми же сравнительно простыми и недорогими аналоговыми телефонными аппаратами, что и раньше.

Типичная структура телефонной сети представлена на рис. 23.4. Сеть образована некоторым количеством телефонных коммутаторов, которые соединены между собой цифровыми или, в редких случаях, аналоговыми каналами. Топология связей между телефонными коммутаторами в общем случае носит произвольный характер, хотя часто используется многоуровневая иерархия, когда несколько коммутаторов нижнего уровня подключаются к коммутатору более высокого уровня и т. п.

К коммутаторам нижнего уровня с помощью абонентских окончаний, которые представляют собой медные пары, подключаются телефонные аппараты абонентов. Обычно длина абонентского окончания не превышает одного-двух километров, однако иногда оператор вынужден использовать и более протяженные окончания, до 5–6 км, если имеется несколько удаленных абонентов, для которых строительство отдельной точки присутствия экономически неоправданно.

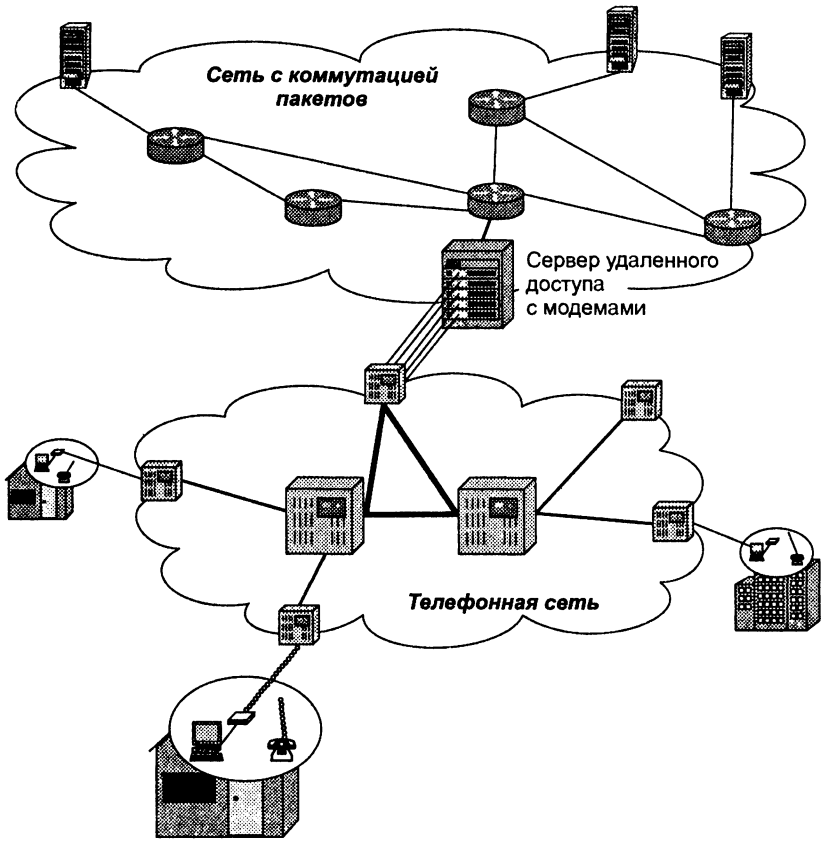


Рис. 23.4. Доступ через телефонную сеть с аналоговыми окончаниями

Телефонная сеть, как и любая сеть с коммутацией каналов, требует обязательной процедуры предварительного установления соединения между абонентскими устройствами, в случае успеха которой в сети устанавливается канал между этими абонентами, через который они могут вести разговор. Процедура установления соединения реализуется с помощью *сигнального протокола*. Напомним, что в аналоговых телефонных сетях каждому абонентскому соединению выделяется полоса пропускания шириной в 4 кГц. Из этой полосы 3,1 кГц предназначается для передачи собственно голоса, а оставшиеся 900 Гц служат для передачи сигнальной информации между аналоговыми коммутаторами, а также в качестве защитной полосы частот между каналами, выделенными различным пользователям.

Существует большое количество различных сигнальных протоколов, разработанных за долгие годы существования телефонных сетей. Они делятся на два класса: сигнальные протоколы UNI работают между телефоном пользователя и первым коммутатором сети, а сигнальные протоколы NNI — между коммутаторами сети. Так как модем подключается к телефонной сети в качестве абонентского устройства, то он должен поддерживать только протокол UNI.

Аналоговый телефон — достаточно примитивное устройство, поэтому поддерживаемый им сигнальный протокол также предельно прост. Процедура вызова абонента обычно представляет собой последовательность замыканий и размыканий электрической цепи, образуемой проводами абонентского окончания. В ответ на первое замыкание телефонный коммутатор подает на абонентскую цепь некоторое напряжение, которое воспроизводится в виде постоянного гудка динамика телефонной трубки. Человек активно участвует в процедуре вызова, набирая в ответ на гудок цифры вызываемого номера.

Существует два способа передачи номера в сеть. При **импульсном наборе** каждая цифра передается соответствующим числом последовательных импульсов размыкания-замыкания частотой 10 или 20 Гц.

При **тоновом наборе** (Dual Tone Multi Frequency, DTMF) для кодирования цифр и символов используется комбинация сигналов двух групп: низкочастотной (697, 770, 852 и 941 Гц) и высокочастотной (1209, 1336, 1477 и 1633 Гц).

Сочетания этих частот дают 16 комбинаций, которые используются для кодирования, как показано в табл. 23.1.

Таблица 23.1. Кодирование цифр и символов при тоновом наборе

1209 Гц	1336 Гц	1477 Гц	1633 Гц	
1	2	3	A	**697 Гц**
4	5	6	B	**770 Гц**
7	8	9	C	**852 Гц**
*	0	#	D	**941 Гц**

Частота 1633 Гц является расширением стандарта DTMF, с помощью которого кодируются дополнительные символы A, B, C и D, отсутствующие на стандартной клавиатуре телефонов, но используемые модемами и некоторыми приложениями.

Тоновый набор выполняется с частотой 10 Гц сигналами длительностью в 50 мс с паузами также в 50 мс.

Так как одна цифра номера при импульсном наборе передается несколькими импульсами, а при тоновом наборе — одним сигналом, то скорость тонового набора в несколько раз выше, чем импульсного.

После приема такого условного «сообщения» от телефонного аппарата первый коммутатор телефонной сети маршрутизирует сообщение дальше. Если этот коммутатор является цифровым, то он преобразует поступающий от абонента аналоговый сигнал в цифровую форму.

Чтобы добиться развитой логики обработки вызовов, современные телефонные коммутаторы используют протоколы **сигнальной системы 7** (Signaling System 7, SS7), в которых применяется техника коммутации пакетов. Эти протоколы построены в соответствии с моделью OSI, покрывая уровни от физического до

прикладного. И хотя мы еще не раз будем упоминать SS7, подробное рассмотрение этих протоколов выходит за рамки темы данной книги, их описание можно найти в учебниках, посвященных телефонии, например [26].

Нужно подчеркнуть, что пользовательские данные по-прежнему передаются в телефонных сетях с помощью техники коммутации каналов, а техника коммутации пакетов используется сигнальными протоколами только для установления соединения. Наряду с протоколами SS7 в телефонной сети может задействоваться также большое количество более старых сигнальных протоколов, в том числе аналоговых.

Удаленный доступ через телефонную сеть

Для того чтобы получить доступ в Интернет или корпоративную сеть через телефонную сеть, модем пользователя должен выполнить вызов по одному из номеров, присвоенному модемам, находящимся на сервере удаленного доступа. После установления соединения между модемами в телефонной сети образуется канал с полосой пропускания около 4 кГц. Точное значение имеющейся в распоряжении модемов полосы зависит от типа телефонных коммутаторов на пути от модема пользователя до модема RAS и от поддерживаемых ими сигнальных протоколов. В любом случае эта полоса не превышает 4 кГц, что принципиально ограничивает скорость передачи данных модемом.

Наивысшим достижением современных модемов на канале тональной частоты является достижение скорости в 33,6 Кбит/с, если на пути следования информации приходилось выполнять *аналого-цифровое преобразование*, и 56 Кбит/с, если преобразование было *цифро-аналоговым*. Такая асимметрия связана с тем, что аналого-цифровое преобразование вносит существенно более значительные искажения в передаваемые дискретные данные, чем цифро-аналоговое.

Очевидно, что такие скорости нельзя назвать приемлемыми для большинства современных приложений, которые широко используют графику и другие мультимедийные формы представления данных.

Модемы RAS обычно устанавливаются в точке присутствия поставщика услуг, при этом, естественно, совсем не обязательно, чтобы это был тот же самый поставщик услуг, который обеспечивает доступ данному удаленному пользователю. В 80-е годы и в первой половине 90-х, когда Интернет еще не был столь популярен, многие крупные корпорации самостоятельно обеспечивали удаленный доступ для своих сотрудников. В этом случае сервер удаленного доступа устанавливался в ближайшей к локальной сети штаб-квартиры корпорации точке присутствия или же в помещении самой штаб-квартиры. Сотрудники корпорации, работающие дома или же находящиеся в командировке, присоединяли свои модемы к локальному поставщику услуг и звонили на модем сервера удаленного доступа корпорации. Иногда это был и международный звонок, если сотрудник находился в командировке в другой стране. Компьютерный трафик проходил основную часть пути по телефонной сети, и стоимость такого доступа зависела от расстояния, что характерно для телефонных сетей.

Сегодня Интернет позволяет использовать телефонную сеть гораздо более экономичным способом. Она нужна теперь не для соединения с RAS предприятия, а для соединения с RAS поставщика услуг Интернета. Если же целью пользователя является доступ не в Интернет, а в корпоративную сеть, то он задействует Интернет как промежуточную сеть, которая ведет к корпоративной сети (также подключенной к Интернету). Поскольку плата за доступ в Интернет не зависит от расстояния до узла назначения, удаленный доступ к ресурсам корпорации стал сегодня намного дешевле даже с учетом оплаты за локальный телефонный звонок и доступ в Интернет. Правда, при такой двухступенчатой схеме доступа пользователю приходится выполнять аутентификацию дважды — при доступе к RAS поставщика услуг и при доступе к RAS предприятия. Существуют протоколы, которые исключают подобное дублирование, например **двухточечный протокол туннелирования** (Point-to-Point Tunneling Protocol, PPTP). При работе PPTP сервер удаленного доступа поставщика услуг передает транзитом запрос пользователя серверу аутентификации предприятия и, в случае положительного ответа, соединяет пользователя через Интернет с корпоративной сетью.

RAS может подключаться к телефонному коммутатору как с помощью аналоговых окончаний, так и с помощью цифровых. Мощные серверы удаленного доступа, оснащенные несколькими десятками модемов, обычно подключаются с помощью цифровых окончаний, используя линии связи T1/E1. В этом случае при передаче информации из сети передачи данных к пользователю аналого-цифровое преобразование не выполняется, поэтому скорость передачи данных в этом направлении (нисходящем) может достигать 56 Кбит/с. Однако это можно осуществить только в том случае, когда все телефонные коммутаторы вдоль пути к пользователю будут цифровыми. В том же случае, когда хотя бы один телефонный коммутатор является аналоговым, максимальная скорость обмена и в нисходящем направлении, как и в исходящем (в направлении от пользователя к сети), будет ограничена значением 33,6 Кбит/с.

Модемы

Хотя коммутируемый модем предоставляет компьютеру услуги физического уровня, сам он представляет собой устройство, в котором реализованы функции двух нижних уровней модели OSI — физического и канального. Канальный уровень нужен модему для того, чтобы выявлять и исправлять ошибки, появляющиеся из-за искажений битов при передаче через телефонную сеть. Вероятность битовой ошибки в этом случае довольно высока, поэтому функция исправления ошибок является очень важной для модема. Для протокола, которой работает поверх модемного соединения между удаленным компьютером и RAS, канальный протокол модема незаметен, его работа проявляется только в том, что интенсивность битовых ошибок (BER) снижается до приемлемого уровня. Так как в качестве канального протокола между компьютером и RAS сегодня в основном используется протокол PPP, который не занимается восстановлением искаженных и потерянных кадров, то способность модема исправлять ошибки является очень полезной.

Протоколы и стандарты модемов определены в рекомендациях ITU-T серии V и делятся на три группы:

- стандарты, определяющие скорость передачи данных и метод кодирования;
- стандарты исправления ошибок;
- стандарты сжатия данных.

Стандарты метода кодирования и скорости передачи данных. Модемы являются одними из наиболее старых и заслуженных устройств передачи данных; в процессе своего развития они прошли долгий путь, прежде чем научились работать на скоростях до 56 Кбит/с.

Первые модемы работали со скоростью 300 бит/с и исправлять ошибки не умели. Эти модемы функционировали в асинхронном режиме, означающем, что каждый байт передаваемой компьютером информации передавался асинхронно по отношению к другим байтам, для чего он сопровождался стартовыми и стоповыми символами, отличающимися от символов данных. Асинхронный режим упрощает устройство модема и повышает надежность передачи данных, но существенно снижает скорость передачи информации, так как каждый байт дополняется одним или двумя избыточными старт-стопными символами.

Современные модемы могут работать как в асинхронном, так и синхронном режимах.

Переломным моментом в истории модемов стало принятие **стандарта V.34**, который повысил максимальную скорость передачи данных в два раза, с 14 до 28 Кбит/с по сравнению со своим предшественником — стандартом V.32. Особенностью стандарта V.34 являются *процедуры динамической адаптации* к изменениям характеристик канала во время обмена информацией. В V.34 определено 10 согласительных процедур, по которым модемы после тестирования линии выбирают свои основные параметры: несущую полосу и полосу пропускания, фильтры передатчика и др. Адаптация осуществляется в ходе сеанса связи — без прекращения и без разрыва установленного соединения. Возможность такого адаптивного поведения была обусловлена развитием техники интегральных схем и микропроцессоров. Первоначальное соединение модемов проводится по стандарту V.21 на минимальной скорости 300 бит/с, что позволяет работать на самых плохих линиях. Затем модемы продолжают переговорный процесс до тех пор, пока не будет достигнута максимально возможная в данных условиях производительность. Применение адаптивных процедур сразу позволило поднять скорость передачи данных более чем в два раза по сравнению с предыдущим стандартом — V.32 bis.

Принципы адаптивной настройки к параметрам линии были развиты в **стандарте V.34+**. Стандарт V.34+ позволил несколько повысить скорость передачи данных за счет усовершенствования метода кодирования. Один передаваемый кодовый символ несет в новом стандарте в среднем не 8,4 бита, как в протоколе V.34, а 9,8. При максимальной скорости передачи кодовых символов в 3429 бод (это ограничение преодолеть нельзя, так как оно определяется полосой пропускания канала тональной частоты) усовершенствованный метод кодирования дает скорость передачи данных в 33,6 Кбит/с (3429 × 9,8 = 33 604).

Протоколы V.34 и V.34+ позволяют работать на 2-проводной выделенной линии в дуплексном режиме. Дуплексный режим передачи в стандартах V.34, V.34+ обеспечивается не путем частотного разделения канала, а одновременной передачей данных в обоих направлениях. Принимаемый сигнал определяется вычитанием с помощью процессоров DSP передаваемого сигнала из общего сигнала в канале. Для этой операции используются также процедуры эхо-подавления, так как передаваемый сигнал, отражаясь от ближнего и дальнего концов канала, вносит искажения в общий сигнал.

ПРИМЕЧАНИЕ

Заметьте, что метод передачи данных, описанный в проекте стандарта 802.3ab, определяющего работу технологии Gigabit Ethernet на витой паре категории 5, взял многое из стандартов V.32–V.34+.

Стандарт V.90 описывает технологию, направленную на обеспечение недорогого и быстрого способа доступа пользователей к сетям поставщиков услуг. Этот стандарт описывает асимметричный обмен данными: со скоростью до 56 Кбит/с из сети и со скоростью до 33,6 Кбит/с в сеть. Стандарт совместим со стандартом V.34+. Именно этот стандарт имелся в виду в предыдущем разделе, когда мы говорили о возможности нисходящей передачи данных со скоростью 56 Кбит/с при условии, что вдоль всего пути не встретится ни одного аналого-цифрового преобразователя.

В **стандарте V.92** учитывается возможность принятия модемом второго вызова во время соединения. В таких случаях современные станции передают на телефонный аппарат специальные двойные тоновые сигналы, так что абонент может распознать эту ситуацию и, нажав на аппарате кнопку Flash, переключиться на второе соединение, переведя первое соединение в режим удержания. Модемы предыдущих стандартов в таких случаях просто разрывают соединение, что не всегда удобно для абонента — может быть в этот момент он заканчивает загружать из Интернета большой файл, и вся его работа пропадает.

Типовая структура соединения двух компьютеров или локальных сетей через маршрутизатор с помощью аналоговых окончаний приведена на рис. 23.5.

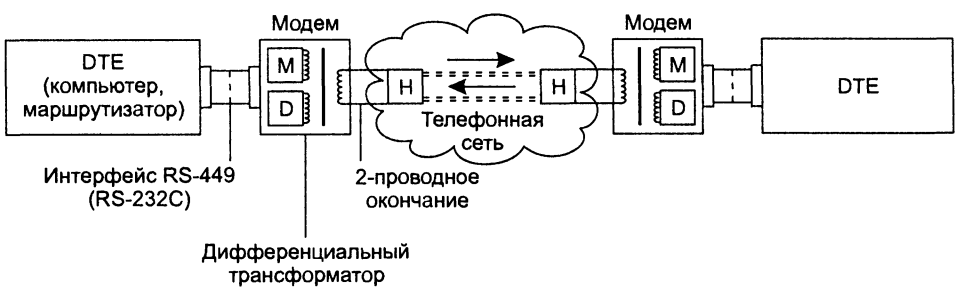


Рис. 23.5. Соединение компьютеров с помощью коммутируемых модемов

Коррекция ошибок. Для модемов, работающих с DTE по асинхронному интерфейсу, комитет CCITT разработал **протокол коррекции ошибок V.42**. До его принятия

в модемах, работающих по асинхронному интерфейсу, коррекция ошибок обычно выполнялась по фирменным протоколам Microcom. Эта компания реализовала в своих модемах несколько разных процедур коррекции ошибок, назвав их сетевыми протоколами Microcom (Microcom Networking Protocol, MNP) классов 2–4.

В стандарте V.42 основным является другой протокол — **протокол доступа к линии связи для модемов** (Link Access Protocol for Modems, LAP-M). Однако стандарт V.42 поддерживает и процедуры MNP 2–4, поэтому модемы, соответствующие рекомендации V.42, позволяют устанавливать связь без ошибок с любым модемом, поддерживающим этот стандарт, а также с любым MNP-совместимым модемом. Протокол LAP-M принадлежит описанному в главе 22 семейству HDLC и в основном работает так же, как и другие протоколы этого семейства, — с установлением соединения, кадрированием данных, нумерацией кадров и восстановлением кадров с поддержкой метода скользящего окна. Основное отличие от других протоколов этого семейства — более развитые переговорные процедуры, для которых в протоколе LAP-M предусмотрены дополнительные типы кадров — XID и BREAK.

С помощью кадров взаимной идентификации (Exchange Identification, XID) модемы при установлении соединения могут договориться о некоторых параметрах протокола, например о максимальном размере поля данных кадра, о величине тайм-аута при ожидании квитанции, о размере окна и т. п. Эта процедура напоминает переговорные процедуры протокола PPP. Команда BREAK служит для уведомления модема-напарника о том, что поток данных временно приостанавливается. При асинхронном интерфейсе с DTE такая ситуация может возникнуть. Команда BREAK посылается в ненумерованном кадре, она не влияет на нумерацию потока кадров сеанса связи. После возобновления поступления данных модем продолжает работать так, как если бы паузы в передаче не было.

Сжатие данных. Почти все современные модемы при работе по асинхронному интерфейсу поддерживают **стандарты сжатия данных CCITT V.42bis** и **MNP-5** (обычно с коэффициентом 1:4, некоторые модели — до 1:8). Сжатие данных увеличивает пропускную способность линии связи. Передающий модем автоматически сжимает данные, а принимающий их восстанавливает. Модем, поддерживающий протокол сжатия, всегда пытается установить связь со сжатием данных, но если второй модем этот протокол не поддерживает, то и первый модем переходит на обычную связь без сжатия.

При работе модемов по синхронному интерфейсу наиболее популярным является **протокол сжатия синхронных потоков данных** (Synchronous Data Compression, SDC) компании Motorola.

Коммутируемый доступ через сеть ISDN

Ключевые слова: цифровая сеть с интегрированным обслуживанием, цифровое абонентское окончание, терминальное оборудование, сетевое окончание, каналы типа B, D и H, начальный интерфейс, основной интерфейс, контрольная точка, терминальное оборудование 1 и 2, терминальный адаптер, устройства сетевого окончания 1 и 2.

Назначение и структура ISDN

Целью создания технологии ISDN (Integrated Services Digital Network — **цифровая сеть с интегрированным обслуживанием**) было построение всемирной сети, которая должна была прийти на смену телефонной сети и, будучи такой же доступной и распространенной, предоставлять миллионам своих пользователей разнообразные услуги, как телефонные, так и передачи данных. Передача телевизионных программ по ISDN не предполагалась, поэтому было решено ограничиться пропускной способностью абонентского окончания для массовых пользователей в 128 Кбит/с.

Если бы цель разработчиков ISDN была достигнута в полной мере, то проблема доступа домашних пользователей к Интернету и корпоративным сетям была бы полностью решена. Однако по многим причинам внедрение ISDN происходило очень медленно — процесс, который начался в 80-е годы, растянулся больше чем на десять лет, так что к моменту появления в домах пользователей некоторые услуги ISDN просто морально устарели. Так, скорость доступа 128 Кбит/с сегодня уже не является достаточной для всех пользователей. Существует, правда, другой интерфейс ISDN, который обеспечивает скорость доступа до 2 Мбит/с, но он достаточно дорог для массового пользователя и его обычно используют только предприятия для подключения своих сетей.

Хотя сеть ISDN и не стала той новой публичной сетью, на роль которой она претендовала, ее услуги сегодня являются достаточно доступными. Далее мы рассмотрим структуру этой сети и ее возможности для организации удаленного доступа.

Архитектура сети ISDN предусматривает несколько видов услуг (рис. 23.6):

- некоммутируемые средства (выделенные цифровые каналы);
- коммутируемая телефонная сеть общего пользования;
- сеть передачи данных с коммутацией каналов;
- сеть передачи данных с коммутацией пакетов;
- сеть передачи данных с трансляцией кадров (режим сети Frame Relay);
- средства контроля и управления работой сети.

Как видно из приведенного списка, транспортные службы сетей ISDN действительно покрывают очень широкий спектр услуг, включая популярные услуги сети Frame Relay. Стандарты ISDN описывают также ряд услуг прикладного уровня: факсимильную связь на скорости 64 Кбит/с, телексную связь на скорости 9600 бит/с, видеотекс на скорости 9600 бит/с и некоторые другие

Все услуги основаны на передаче информации в цифровой форме. Интерфейс пользователя также является цифровым, то есть все его абонентские устройства (телефон, компьютер, факс) должны передавать в сеть цифровые данные. Организация **цифрового абонентского окончания** (Digital Subscriber Line, DSL) стала одним из серьезных препятствий на пути распространения ISDN, так как требовала модернизации миллионов абонентских окончаний.

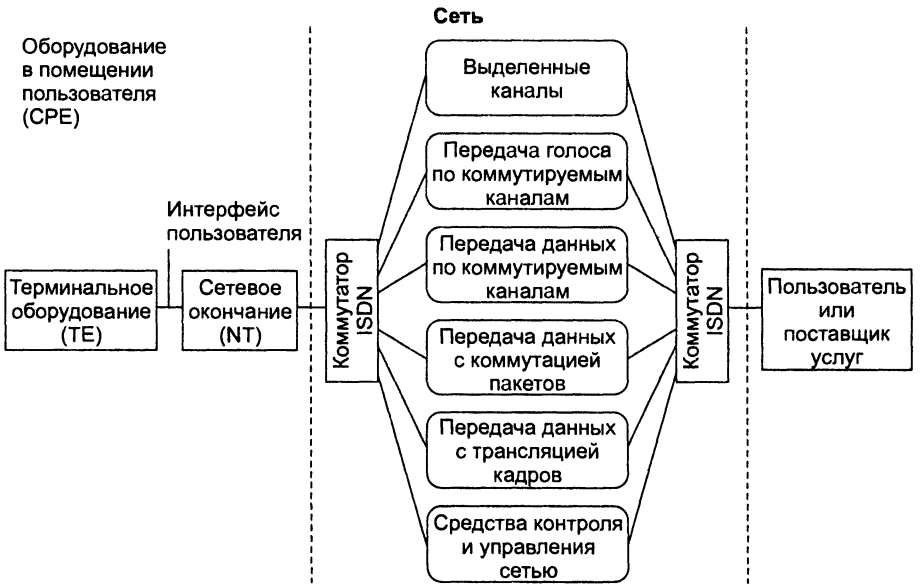


Рис. 23.6. Услуги сети ISDN

На практике не все сети ISDN поддерживают все стандартные службы. Служба Frame Relay, хотя и была разработана в рамках сети ISDN, реализуется, как правило, с помощью отдельной сети коммутаторов кадров, не пересекающейся с сетью коммутаторов ISDN.

Базовой скоростью сети ISDN является скорость канала DS-0, то есть 64 Кбит/с. Эта скорость ориентируется на самый простой метод кодирования голоса — PCM, хотя дифференциальное кодирование и позволяет передавать голос с тем же качеством на скорости 32 или 16 Кбит/с.

Одной из оригинальных идей, положенных в основу ISDN, является совместное использование принципов коммутации каналов и пакетов. Однако сеть с коммутацией пакетов, работающая в составе ISDN, выполняет только служебные функции — с помощью этой сети передаются сообщения сигнального протокола. А вот основная информация, то есть сам голос, по-прежнему передается с помощью сети с коммутацией каналов. В таком разделении функций есть вполне понятная логика — сообщения о вызове абонентов образуют пульсирующий трафик, поэтому его эффективнее передавать по сети с коммутацией пакетов.

Интерфейсы BRI и PRI

Одним из основных принципов ISDN является предоставление пользователю стандартного интерфейса, с помощью которого пользователь может запрашивать у сети разнообразные услуги. Этот интерфейс образуется между двумя типами оборудования, устанавливаемого в помещении пользователя (Customer Premises Equipment, CPE):

- **терминальным оборудованием** (Terminal Equipment, TE) пользователя (компьютер с соответствующим адаптером, маршрутизатор, телефонный аппарат);
- **сетевым окончанием** (Network Termination, NT), которое представляет собой устройство, завершающее линию связи с ближайшим коммутатором ISDN.

Пользовательский интерфейс основан на каналах трех типов: B, D и H.

Каналы типа B обеспечивают передачу пользовательских данных (оцифрованного голоса, компьютерных данных или смеси голоса и данных) и с более низкими скоростями, чем 64 Кбит/с. Разделение данных выполняется с помощью техники TDM. Разделением канала B на подканалы в этом случае должно заниматься пользовательское оборудование, сеть ISDN всегда коммутирует целые каналы типа B. Каналы типа B могут соединять пользователей с помощью техники коммутации каналов друг с другом, а также образовывать так называемые полупостоянные соединения, которые эквиваленты соединениям выделенных каналов обычной телефонной сети. Канал типа B может также подключать пользователя к коммутатору сети X.25.

Канал типа D является каналом доступа к служебной сети с коммутацией пакетов, передающей сигнальную информацию со скоростью 16 или 64 Кбит/с. Передача адресной информации, на основе которой осуществляется коммутация каналов типа B в коммутаторах сети, является основной функцией канала D. Другой его функцией является поддержание услуг низкоскоростной сети с коммутацией пакетов для пользовательских данных. Обычно эта услуга выполняется сетью в то время, когда каналы типа D свободны от выполнения основной функции.

Каналы типа H предоставляют пользователям возможности высокоскоростной передачи данных со скоростью 384 Кбит/с (H0), 1536 Кбит/с (H11) или 1920 Кбит/с (H12). На них могут работать службы высокоскоростной передачи факсов, видеоинформации, качественного воспроизведения звука.

Пользовательский интерфейс ISDN представляет собой набор каналов определенного типа и с определенными скоростями. Сеть ISDN поддерживает два вида пользовательского интерфейса — начальной (Basic Rate Interface, BRI) и основной (Primay Rate Interface, PRI) скоростей передачи данных.

Начальный интерфейс ISDN предоставляет пользователю два канала по 64 Кбит/с для передачи данных (каналы типа B) и один канал с пропускной способностью 16 Кбит/с для передачи управляющей информации (канал типа D). Все каналы работают в дуплексном режиме. В результате суммарная скорость интерфейса BRI для пользовательских данных составляет 144 Кбит/с по каждому направлению, а с учетом служебной информации — 192 Кбит/с. Различные каналы пользовательского интерфейса разделяют один и тот же физический двухпроводный кабель по технологии TDM, то есть являются логическими, а не физическими каналами. Данные по интерфейсу BRI передаются кадрами, состоящими из 48 бит. Каждый кадр содержит по 2 байта каждого из двух каналов B, а также 4 бита канала D. Передача кадра длится 250 мс, что обеспечивает скорость передачи данных 64 Кбит/с для каналов B и 16 Кбит/с — для канала D. Кроме битов данных кадр содержит служебные биты для синхронизации кадров, а также обеспечения

нулевой постоянной составляющей электрического сигнала. Интерфейс BRI может поддерживать не только схему 2B + D, но и B + D и просто D.

Начальный интерфейс стандартизован в рекомендации I.430.

Основной интерфейс ISDN предназначен для пользователей с повышенными требованиями к пропускной способности сети. Интерфейс PRI поддерживает либо схему 30B + D, либо схему 23B + D. В обеих схемах канал D обеспечивает скорость 64 Кбит/с. Первый вариант предназначен для Европы, второй — для Северной Америки и Японии. Ввиду большой популярности скорости цифровых каналов 2,048 Мбит/с в Европе и скорости 1,544 Мбит/с в остальных регионах привести стандарт на интерфейс PRI к общему варианту не удалось.

Возможны варианты интерфейса PRI с меньшим количеством каналов типа B, например 20B + D. Каналы типа B могут объединяться в один логический высокоскоростной канал с общей скоростью до 1920 Кбит/с. При установке у пользователя нескольких интерфейсов PRI все они могут иметь один канал типа D, при этом количество каналов B в том интерфейсе, который не имеет канала D, может увеличиваться до 24 или 31.

Основной интерфейс может быть также основан на каналах типа H. При этом общая пропускная способность интерфейса все равно не должна превышать 2,048 или 1,544 Мбит/с. Для каналов H0 возможны интерфейсы 3H0 + D для американского варианта и 5H0 + D для европейского. Для каналов H1 возможен интерфейс, состоящий только из одного канала H11 (1,536 Мбит/с) для американского варианта или одного канала H12 (1,920 Мбит/с) и одного канала D для европейского варианта. Кадры интерфейса PRI имеют структуру кадров DS-1 для каналов T1 или E1.

Основной интерфейс PRI стандартизован в рекомендации I.431.

ВНИМАНИЕ

Как каналы B, так и каналы D являются логическими каналами абонентского окончания, которое физически представляет собой одну витую пару. Каналы D и B образуются путем применения техники TDM к физической среде, образуемой этой витой парой.

Стек протоколов ISDN

В сети ISDN существует два стека протоколов: стек каналов типа D и стек каналов типа B (рис. 23.7).

Сеть каналов типа D внутри сети ISDN служит транспортной системой с коммутацией пакетов, применяемой для передачи сообщений сигнализации. Прообразом этой сети послужила технология сетей X.25. Для сети каналов D определены три уровня протоколов:

- физический протокол определяется стандартом I.430/431;
- канальный протокол LAP-D определяется стандартом Q.921;
- на сетевом уровне может использоваться протокол сигнализации Q.931, с помощью которого выполняется маршрутизация вызова абонента службы с коммутацией каналов.

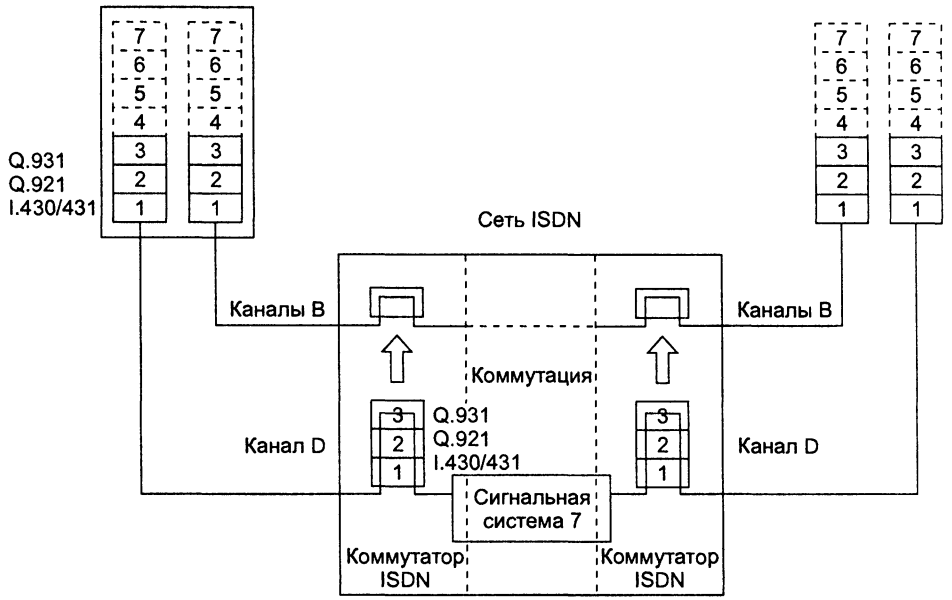


Рис. 23.7. Структура сети ISDN

Каналы типа B образуют сеть с коммутацией каналов, которая передает данные абонентов, то есть оцифрованный голос. В терминах модели OSI на каналах типа B в коммутаторах сети ISDN определен только протокол физического уровня — протокол I.430/431. Коммутация каналов типа B происходит по указаниям, полученным по каналу D. Когда кадры протокола Q.931 маршрутизируются коммутатором, при этом происходит одновременная коммутация очередной части составного канала от исходного абонента к конечному.

Протокол LAP-D принадлежит к уже не раз упомянутому нами семейству HDLC. Протокол LAP-D обладает всеми «родовыми чертами» этого семейства, но имеет и некоторые особенности. Адрес кадра LAP-D состоит из двух байтов — один байт определяет код службы, которой пересылаются вложенные в кадр пакеты, а второй требуется для адресации одного из терминалов, если у пользователя к абонентскому окончанию подключено несколько терминалов. Терминальное устройство ISDN может поддерживать разные услуги — установления соединения по протоколу Q.931, коммутации пакетов X.25, мониторинга сети и т. п. Протокол LAP-D обеспечивает два режима работы: с установлением соединения и без установления соединения. Последний режим используется, например, для мониторинга сети.

Протокол Q.931 является сигнальным протоколом ISDN для участка пользователь-сеть, то есть протоколом типа UNI. Он переносит в своих пакетах ISDN-адрес вызываемого абонента, на основании которого и происходит настройка коммутаторов на поддержку составного канала типа B. Процедуру установления соединения по протоколу Q.931 иллюстрирует рис. 23.8.

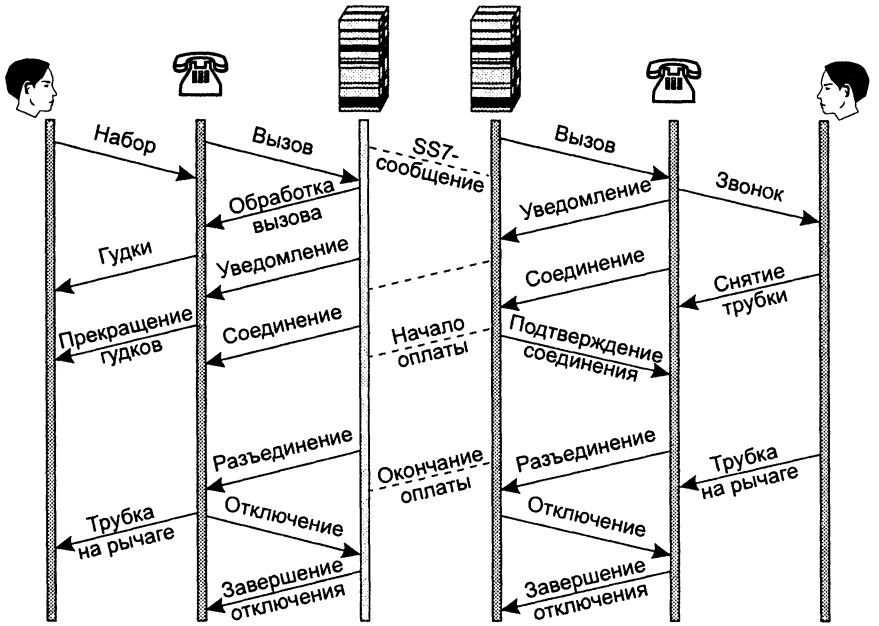


Рис. 23.8. Базовая процедура установления соединения в ISDN по протоколу Q.931

После того как пользователь снял трубку и набрал номер вызываемого абонента, телефонный аппарат ISDN формирует пакет «Вызов» («Set up») и отправляет его по каналу D коммутатору ISDN, к которому он подключен. Этот коммутатор отвечает аппарату абонента пакетом «Обработка вызова», с приходом которого аппарат начинает генерировать длинные гудки. Одновременно коммутатор запоминает факт запроса на установление соединения и передает принятое сообщение следующему коммутатору, адрес которого он находит по таблице, аналогичной таблице маршрутизации маршрутизаторов пакетных сетей. При этом сообщение протокола Q.931 транслируется в сообщение «Начальный адрес» («Initial Address Message, IAM) протокола SS7 аналогичного назначения (на рисунке SS7-сообщения не детализированы). Проходя через сеть, SS7-сообщения устанавливают в промежуточных коммутаторах состояние готовности к установлению соединения. Выходной коммутатор сети, к которому подключен аппарат вызываемого абонента, преобразует SS7-сообщение «Начальный адрес» в сообщение «Вызов» протокола Q.931, на основании которого телефонный аппарат начинает звонить. Если абонент снимает трубку, то его аппарат генерирует сообщение «Соединение» («Connect»), которое в обратном порядке проходит через все промежуточные коммутаторы (преобразованное, естественно, в соответствующее SS7-сообщение). При этом обратном проходе коммутаторы устанавливают состояние соединения, коммутируя соответствующим образом каналы типа B.

Любое абонентское устройство ISDN должно поддерживать Q.931, так что телефон ISDN намного сложнее своего аналогового коллеги. Как видно из рисунка, внутри сети сообщения Q.931 транслируются в сообщения протокола SS7, который является протоколом взаимодействия коммутатор-коммутатор (NNI), а затем снова преобразуются в сообщения Q.931 на абонентском окончании.

Использование ISDN для передачи данных

Несмотря на значительные отличия от аналоговых телефонных сетей, сети ISDN сегодня используются в основном так же, как аналоговые телефонные сети, то есть как сети с коммутацией каналов, но только более скоростные: интерфейс BRI дает возможность установить дуплексный режим обмена со скоростью 128 Кбит/с (логическое объединение двух каналов типа B), а интерфейс PRI — 2,048 Мбит/с. Кроме того, качество цифровых каналов гораздо выше, чем аналоговых. Это значит, что процент искаженных кадров будет гораздо ниже, а полезная скорость обмена данными существенно выше.

Обычно интерфейс BRI используется в коммуникационном оборудовании для подключения отдельных компьютеров или небольших локальных сетей домашних пользователей, а интерфейс PRI — для подключения сети средних размеров с помощью маршрутизатора. Схема удаленного доступа через ISDN показана на рис. 23.9.

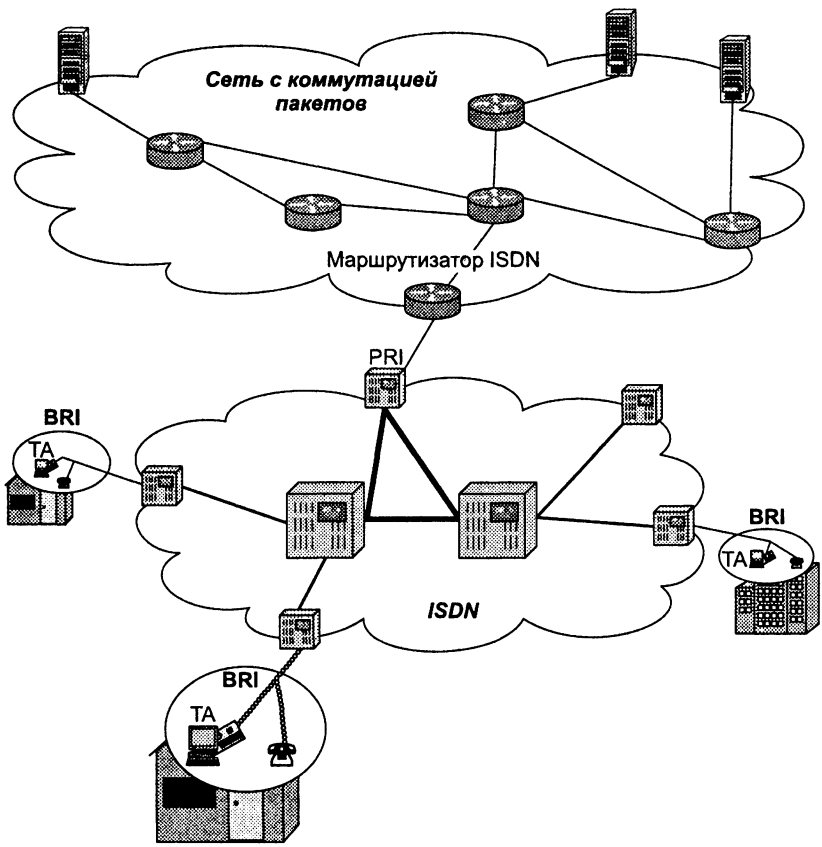


Рис. 23.9. Удаленный доступ с использованием ISDN

Подключение пользовательского оборудования к сети ISDN осуществляется в соответствии со схемой, разработанной ITU-T (рис. 23.10). Оборудование делит-

ся на функциональные группы, и в зависимости от группы различают несколько **контрольных точек** соединения разных групп оборудования между собой.

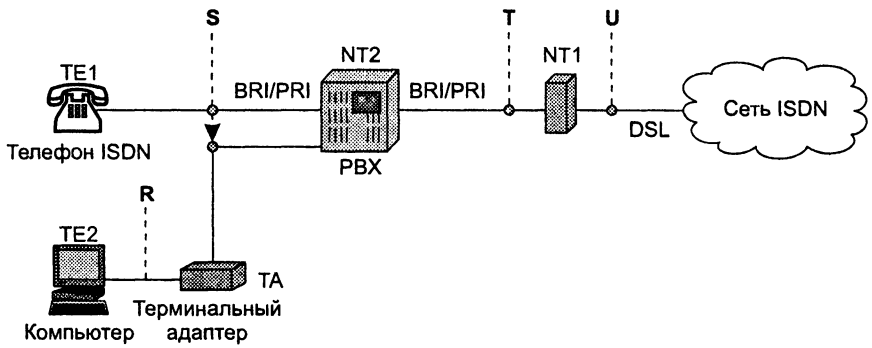


Рис. 23.10. Подключение пользовательского оборудования ISDN

Терминальным оборудованием 1 (TE1) может быть цифровой телефон или факс-аппарат. *Контрольная точка S* соответствует точке подключения отдельного терминального устройства к устройству сетевого окончания (устройство типа NT1) или концентратору пользовательских интерфейсов (устройство типа NT2). TE1 по определению поддерживает один из пользовательских интерфейсов ISDN: BRI или PRI.

Если терминальное оборудование TE1 пользователя подключено через интерфейс BRI, то цифровое абонентское окончание выполняется по 2-проводной схеме (как и обычное окончание аналоговой телефонной сети). Для кодирования данных на участке DSL до точки подключения к сети ISDN (*контрольная точка U*) используется потенциальный код 2B1Q. Дуплексный режим DSL образован путем одновременной передачи сигналов по одной витой паре в обоих направлениях с эхо-подавлением и вычитанием своего сигнала из суммарного. Максимальная длина абонентского окончания для этого варианта составляет 5,5 км.

При использовании терминальным оборудованием TE1 интерфейса PRI цифровое абонентское окончание должно представлять собой канал T1 или E1, то есть 4-проводную линию с максимальной длиной около 1800 м. Соответственно, на участке DSL до точки U используется код HDB3 (Европа) или B8ZS (Америка).

Терминальное оборудование 2 (TE2) в отличие от TE1 не поддерживает интерфейсы BRI и PRI. Таким оборудованием может быть компьютер, маршрутизатор с последовательными интерфейсами, не относящимися к ISDN, например RS-232C, X.21 или V.35. Для подключения подобного оборудования к сети ISDN необходимо использовать терминальный адаптер. **Терминальный адаптер** (Terminal Adaptor, TA) согласует интерфейс TE2 с интерфейсом PRI или BRI. Для компьютеров терминальные адаптеры выпускаются в формате сетевых адаптеров. *Контрольная точка R* соответствует точке подключения терминального оборудования TE2 к адаптеру TA. Тип абонентского окончания не зависит от того, работает терминальное оборудование через TA или непосредственно.

Устройства сетевого окончания 2 (NT2) представляют собой устройства канального или сетевого уровня, которые выполняют функции концентрации пользовательских интерфейсов и их мультиплексирования. Например, к этому типу оборудования относятся: офисная АТС, коммутирующая несколько интерфейсов BRI, маршрутизатор, работающий в режиме коммутации пакетов (например, по каналу D), простой мультиплексор TDM, который мультиплексирует несколько низкоскоростных каналов в один канал типа B. Точка подключения оборудования типа NT2 к абонентскому сетевому окончанию (устройству NT1) называется *контрольной точкой T*. Поскольку наличие данного типа оборудования не является обязательным (в отличие от NT1), то контрольные точки S и T объединяются и обозначаются как *контрольная точка S/T*. Физически интерфейс в точке S/T представляет собой 4-проводную линию. Для интерфейса BRI в качестве метода кодирования выбран биполярный метод AMI, причем логическая единица кодируется нулевым потенциалом, а логический ноль — чередованием потенциалов противоположной полярности. Для интерфейса PRI используются другие коды — те же, что и для интерфейсов T1 и E1, то есть соответственно B8ZS и HDB3.

Устройства сетевого окончания 1 (NT1) — это устройства физического уровня, которые согласуют интерфейс BPR или PRI с цифровым абонентским окончанием (DSL), соединяющим пользовательское оборудование с сетью ISDN. Фактически NT1 представляет собой устройство типа CSU, которое согласует методы кодирования, количество используемых линий и параметры электрических сигналов. *Контрольная точка U* соответствует точке подключения устройства NT1 к сети.

ПРИМЕЧАНИЕ

Устройство NT1 может принадлежать оператору сети (хотя всегда устанавливается в помещении пользователя), а может принадлежать и пользователю. В Европе принято считать устройство NT1 частью сетевого оборудования, поэтому пользовательское оборудование (например, маршрутизатор с интерфейсом ISDN) выпускается без встроенного устройства NT1. В Северной Америке принято считать устройство NT1 принадлежностью пользовательского оборудования, поэтому пользовательское оборудование часто выпускается со встроенным устройством NT1.

Таким образом, для удаленного доступа необходимо оснастить компьютеры пользователей терминальными адаптерами, а в POP установить маршрутизатор, имеющий один или несколько интерфейсов PRI. В этом случае максимальная скорость доступа для отдельного пользователя будет равна скорости передачи двух каналов типа B, то есть 128 Кбит/с. Драйверы терминальных адаптеров ISDN умеют объединять два отдельных физических канала типа B в один логический канал. Для этого используется расширение протокола PPP — многоканальный протокол PPP (RFC 1990).

Если пользователь удаленного доступа согласен ограничиться скоростью 64 Кбит/с, он может использовать второй канал типа B своего интерфейса BRI для параллельной работы телефона ISDN, что невозможно сделать при использовании аналогового коммутируемого модема.

Технологии xDSL

Ключевые слова: асимметричное цифровое абонентское окончание, симметричное цифровое абонентское окончание, цифровое абонентское окончание с адаптируемой скоростью передачи, сверхбыстрое цифровое абонентское окончание, мультиплексор доступа к цифровому абонентскому окончанию.

В середине 90-х годов появилась альтернатива цифровому абонентскому окончанию ISDN. Эта альтернатива представляет собой семейство технологий под общим названием **xDSL**, которое включает технологии:

- **асимметричного цифрового абонентского окончания** (Asymmetric Digital Subscriber Line, ADSL), которую в коммерческих предложениях операторов связи часто называют широкополосным доступом;
- **симметричного цифрового абонентского окончания** (Symmetric Digital Subscriber Line, SDSL);
- **цифрового абонентского окончания с адаптируемой скоростью передачи** (Rate Adaptive Digital Subscriber Line, RADSL);
- **сверхбыстрого цифрового абонентского окончания** (Very high-speed Digital Subscriber Line, VDSL).

Мы рассмотрим основные принципы доступа на основе технологий xDSL на примере ADSL как наиболее распространенной технологии, так как именно она была разработана для самой массовой категории пользователей, которым нужен доступ из дома в Интернет или через Интернет к корпоративной сети.

Для доступа через ADSL, так же как и для аналогового коммутируемого доступа, нужны телефонные абонентские окончания и модемы. Однако принципиальным отличием доступа через ADSL от коммутируемого доступа является то, что ADSL-модемы работают только на абонентском окончании, в то время как коммутируемые модемы используют возможности телефонной сети, устанавливая в ней соединение, проходящее через несколько транзитных коммутаторов.

Поэтому если традиционные телефонные модемы (например, V.34, V.90) должны обеспечивать передачу данных на канале с полосой пропускания в 3100 Гц, то ADSL-модемы получают в свое распоряжение полосу порядка 1 МГц — эта величина зависит от длины кабеля, проложенного между помещением пользователя и POP, и сечения проводов этого кабеля.

Схема доступа через ADSL показана на рис. 23.11. Эта схема близка к общей схеме использования универсального абонентского окончания (см. рис. 23.2), за исключением того, что при доступе через ADSL игнорируется наличие телевизоров у пользователей, доступ для телефонов и компьютеров является совместным.

ADSL-модемы, подключаемые к обоим концам короткой линии между абонентом и POP, образуют три канала: высокоскоростной нисходящий канал передачи данных из сети в компьютер, менее скоростной восходящий канал передачи данных из компьютера в сеть и канал телефонной связи, по которому передаются обычные телефонные разговоры. Передача данных в канале от сети к абоненту

происходит со скоростью от 1,5 до 6 Мбит/с, а в канале от абонента к сети — со скоростью от 16 Кбит/с до 1 Мбит/с; для телефона оставлена традиционная полоса в 4 кГц (рис. 23.12).

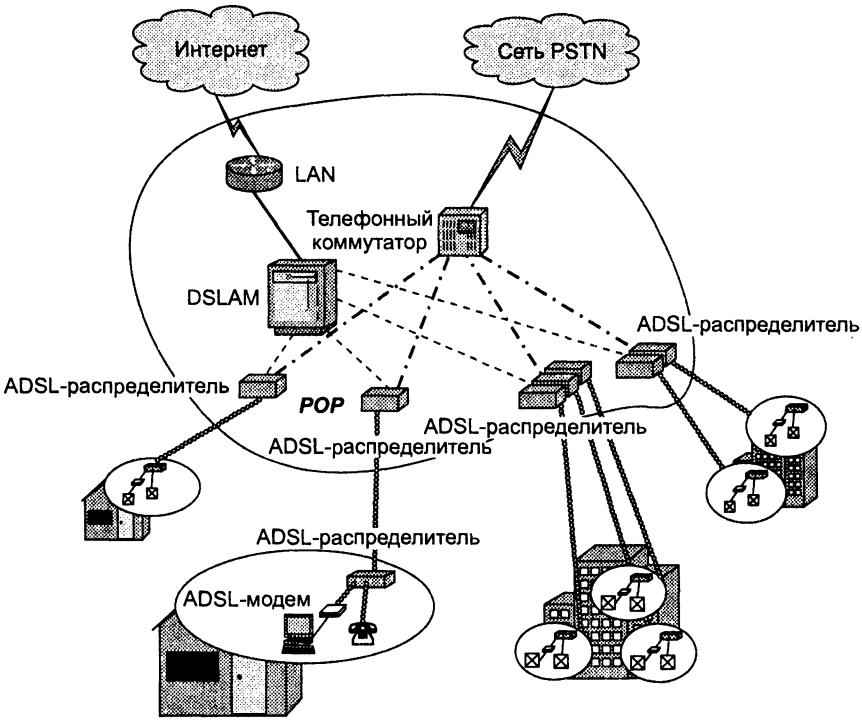


Рис. 23.11. Отличия условий работы ADSL-модемов от обычных модемов

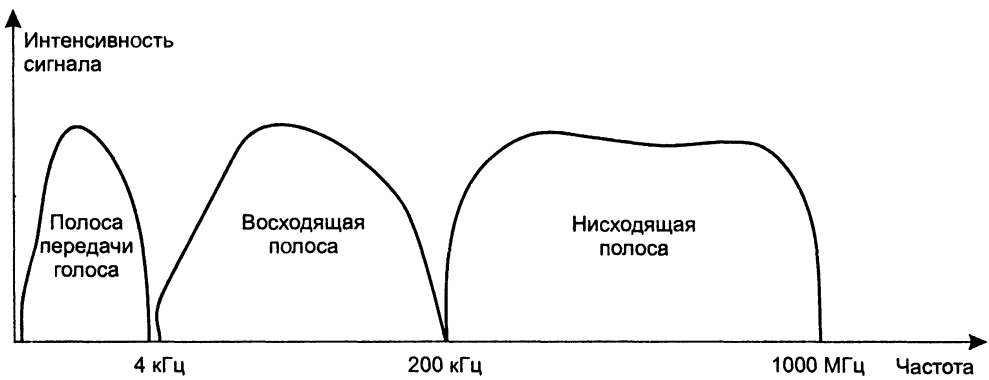


Рис. 23.12. Распределение полосы пропускания абонентского окончания между каналами ADSL

Для асимметрии нисходящей и восходящей скоростей полоса пропускания абонентского окончания делится между каналами также асимметрично. На рис. 23.12

показано оценочное распределение полосы между каналами. Примерные значения объясняются тем, что точные значения никогда заранее не известны, так как зависят от длины абонентского окончания, сечения провода и качества витой пары в целом. Кроме того, распределение полосы между каналами зависит не только от возможностей технологии и модема, но и от желания поставщика услуг. Как правило, ADSL-модемы допускают некоторую настройку, позволяющую изменять распределение полосы и скорость передачи данных в каждом направлении.

В помещении клиента устанавливается распределитель, который выполняет разделение частот между ADSL-модемом и обычным аналоговым телефоном, обеспечивая их совместное сосуществование.

В POP устанавливается так называемый **мультиплексор доступа к цифровому абонентскому окончанию** (Digital Subscriber Line Access Multiplexer, DSLAM). Он принимает компьютерные данные, отделенные распределителями на дальнем конце абонентских окончаний от голосовых сигналов. DSLAM-мультиплексор должен иметь столько ADSL-модемов, сколько пользователей удаленного доступа обслуживает поставщик услуг с помощью телефонных абонентских окончаний.

После преобразования модулированных сигналов в дискретную форму DSLAM отправляет данные на IP-маршрутизатор, который также обычно находится в помещении POP. Далее данные поступают в магистраль передачи данных поставщика услуг и доставляются в соответствии с IP-адресами назначения — на публичный сайт Интернета или в корпоративную сеть пользователя. Отделенные распределителем голосовые сигналы передаются на телефонный коммутатор, который обрабатывает их так, как если бы абонентское окончание пользователя было непосредственно к нему подключено.

Широкое распространение технологий xDSL должно сопровождаться некоторой перестройкой работы поставщиков услуг Интернета и поставщиков услуг телефонных сетей, так как их оборудование должно теперь работать совместно. Возможен также вариант, когда альтернативный оператор связи берет оптом в аренду большое количество абонентских окончаний у традиционного местного оператора или же арендует некоторое количество модемов в DSLAM.

Стандарт G.992.1 описывает работу трансиверов ADSL-модемов. Технология ADSL поддерживает несколько вариантов кодирования информации (DMT, CAP и 2B1Q). Достижения технологий xDSL во многом определяются достижениями техники кодирования, которая за счет применения процессоров DSP смогла повысить скорость передачи данных при одновременном увеличении расстояния между модемом и оборудованием DSLAM.

Скорости каналов ADSL существенно зависят от качества физической линии и расстояния между модемом и оборудованием DSLAM. Чем больше это расстояние, тем ниже скорости. Обычно модем позволяет изменять скорость передачи данных, поэтому при установке его на определенное абонентское окончание можно подобрать оптимальный режим работы, обеспечивающий при удовлетворительном качестве передачи максимально возможную скорость.

Высокие скорости ADSL-модемов создают новую проблему для поставщиков услуг, а именно проблему дефицита пропускной способности. Действительно,

если каждый абонент доступа через ADSL будет загружать данные из Интернета с максимальной скоростью, например 1 Мбит/с, то при 100 абонентах поставщику услуг потребовался бы канал с пропускной способностью 100 Мбит/с, то есть Fast Ethernet, а если разрешить пользователям работать со скоростью 6 Мбит/с, то уже нужен канал ATM 622 Мбит/с или Gigabit Ethernet. Для обеспечения необходимой скорости многие устройства DSLAM имеют встроенный коммутатор ATM или Gigabit Ethernet. Технология ATM привлекает разработчиков DSLAM не только своей высокой скоростью, но и тем, что она ориентирована на соединение. При использовании сети ATM на канальном уровне компьютер пользователя перед передачей данных должен обязательно установить соединение с сетью поставщика услуг. Это дает возможность контролировать доступ пользователей и учитывать время использования и объем переданных данных, если оплата за услугу учитывает эти параметры.

Технология SDSL позволяет на одной паре абонентского окончания организовать два симметричных канала передачи данных. Канал тональной частоты в этом случае не предусматривается. Обычно скорости каналов в восходящем и нисходящем направлениях составляют по 2 Мбит/с, но, как и у технологии ADSL, эта скорость зависит от качества линии и расстояния до оборудования DSLAM. Технология SDSL разработана в расчете на небольшие офисы, локальные сети которых содержат собственные источники информации, например веб-сайты или серверы баз данных. Поэтому характер трафика здесь ожидается скорее симметричный, так как доступ через SDSL потребуется не только к внешним сетям из локальных сетей, но и к таким источникам информации извне.

Широкое применение доступа через xDSL наносит еще один удар технологии ISDN. При применении этого типа абонентских окончаний пользователь получает еще и интегрированное обслуживание двух сетей — телефонной и компьютерной. Но для пользователя наличие двух сетей оказывается незаметным, для него только ясно, что он может одновременно пользоваться обычным телефоном и подключенным к Интернету компьютером. Скорость же компьютерного доступа при этом превосходит возможности интерфейса PRI сети ISDN при существенно более низкой стоимости, определяемой низкой стоимостью инфраструктуры IP-сетей.

Доступ через сети CATV

Ключевые слова: модемная терминальная станция, кабельное телевидение, коаксиальные абонентские окончания, кабельный модем, восходящий и нисходящий частотные каналы, разделение канала во времени, арбитр, тайм-слот.

Кабельное телевидение является одной из телекоммуникационных услуг, для которой была создана собственная разветвленная инфраструктура абонентских окончаний. Хотя кабельное телевидение и уступает по распространенности телефонной сети, тем не менее количество коаксиальных абонентских окончаний, соединяющих дома и квартиры с точками присутствия поставщиков услуг, в некоторых странах стало приближаться к количеству абонентских телефонных окончаний.

Учитывая, что коаксиальный кабель обладает гораздо более широкой полосой пропускания (как минимум, 700–800 МГц), абонентское окончание CATV может вполне справиться с одновременной передачей телефонного, компьютерного и телевизионного трафиков.

Схема использования линий CATV в качестве универсальных окончаний для доступа в Интернет, телефонную сеть и сеть кабельного телевидения нами в общих чертах уже рассматривалась. Именно окончание CATV было выбрано в качестве примера на рис. 23.2. Теперь мы остановимся на некоторых деталях этого вида доступа.

Отличием абонентского окончания CATV является то, что к коаксиальному кабелю по схеме монтажного ИЛИ подключаются одновременно несколько абонентов (рис. 23.13). Это может быть несколько десятков домов или же сотен квартир многоквартирного дома. Поэтому абонентское окончание CATV представляет собой классическую разделяемую среду, которая используется, например, в сетях Ethernet на коаксиальном кабеле.

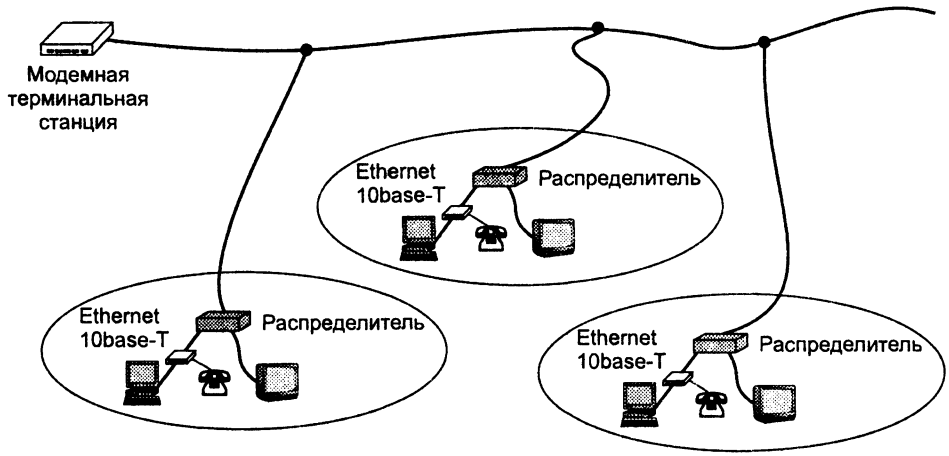


Рис. 23.13. Подключение кабельных модемов к окончанию CATV

В отсутствии кабельных модемов оборудование CATV используется для широковещательного распространения телевизионных программ до телевизионных приемников абонентов CATV из источника информации, расположенного в точке присутствия поставщика услуг. Для этого занимается диапазон частот от 50 до 550–868 МГц (точное значение зависит от национальной политики выделения частот). Каждой программе CATV выделяется в этом диапазоне полоса в 6 или 8 МГц, сигнал которой шифруется и может быть дешифрирован приемниками тех абонентов, которые подписались на прием определнной программы.

Для использования такого абонентского окончания в помещении каждого абонента высокоскоростного доступа устанавливается распределитель и кабельный модем, а в точке присутствия — головной модем, который еще называют **модемной терминальной станцией** (Cable Modem Termination Station, CMTS).

Для двунаправленной передачи компьютерных данных кабельные модемы пользователей и станция CMTS занимают неиспользуемые телевизионными программами частоты. Обычно это диапазон относительно низких частот от 5 до 50 МГц, расположенный ниже частот телевизионных программ, а также диапазон высоких частот выше 550 МГц.

Диапазон низких частот используется для менее скоростного восходящего канала, а диапазон высоких частот — для высокоскоростного нисходящего канала. Скорость передачи данных в восходящем направлении может доходить до 10 Мбит/с, а в нисходящем — до 30–40 Мбит/с. Модемы пользователей могут взаимодействовать только со станцией CMTS.

Так как восходящий и нисходящий каналы разделены по частотам, абонентское окончание CATV образует две разделяемые среды.

Для нисходящего канала CMTS является единственным передатчиком информации, поэтому здесь не возникает конкуренции за доступ к среде. Станция CMTS использует нисходящий канал для передачи по нему кадров данных всем абонентам, используя адресацию Ethernet и разделение канала во времени.

Восходящий канал задействуется в режиме множественного доступа всеми кабельными модемами, подключенными к данному абонентскому окончанию. В этой разделяемой среде CMTS играет роль *арбитра*. Каждый абонентский модем начинает передачу только после того, как получит разрешение на это от головного модема по прямому каналу. Для того чтобы один абонентский модем не занимал канал надолго, CMTS назначает каждому абонентскому модему тайм-слот ограниченного размера. Тайм-слоты распределяются только между активными модемами — это позволяет расходовать ограниченную пропускную способность максимально эффективно. Для вновь подключаемых абонентских модемов предназначены специальные тайм-слоты. При включении абонентский модем использует такой тайм-слот, чтобы оповестить CMTS о своем присутствии в сети. Далее он ожидает, когда ему будет выделен тайм-слот на равных основаниях с другими модемами.

Кабельный модем абонента может иметь разъем для подключения обычного телефона, для которого также выделяется полоса в 4 МГц в нижнем диапазоне частот. В этом случае абонент получает от одного поставщика услуг доступ трех типов — телефонный, компьютерный и телевизионный.

Беспроводной доступ

Ключевые слова: беспроводное абонентское окончание, многоканальная служба распределения, локальная служба распределения, стандарт IEEE 802.16.

Мы уже касались особенностей беспроводной передачи данных в предыдущих главах: в главе 9 были рассмотрены общие принципы беспроводной связи, а в главе 14 — технологии беспроводных локальных и персональных сетей. Беспроводная передача данных в последнее время широко используется также для организации доступа, особенно в тех случаях, когда поставщик услуг по какой-то причине не может обеспечить своим клиентам проводной доступ. Чаще всего это случается с альтернативными поставщиками услуг, которые не имеют в своем

распоряжении проводных абонентских окончаний к домам клиентов. Другим типичным примером является организация временного высокоскоростного доступа для определенного здания, например, при проведении конференции в помещении гостиницы, не оснащенном средствами проводного доступа необходимой пропускной способности.

Беспроводной доступ может быть как фиксированным, так и мобильным.

Фиксированный беспроводной доступ организуется для абонентов, компьютеры которых находятся в пределах ограниченной территории, чаще всего в пределах здания. В таком случае поставщик услуг может использовать направленную антенну и передатчик известной мощности, чтобы обеспечить устойчивый прием высокочастотных сигналов в такой узкой области покрытия, как здание. Если у поставщика услуг имеется достаточно большое количество абонентов фиксированного беспроводного доступа, то он обычно задействует несколько направленных антенн, чтобы покрыть все секторы, в которых находятся его абоненты.

Для беспроводного фиксированного доступа используется также термин **беспроводное абонентское окончание** (Wireless Local Loop, WLL). Этот термин хорошо отражает тот факт, что, несмотря на отсутствие кабелей, абоненты «привязаны» к определенной географической точке, как и в случае проводного абонентского окончания.

Существуют *узкополосные* и *широкополосные* беспроводные абонентские окончания. Первый тип не обеспечивает передачу телевизионного сигнала, а только сравнительно низкоскоростной компьютерный трафик (64–128 Кбит/с) и телефонный сигнал. Второй тип обычно основан на системах распространения телевизионного сигнала, поэтому работает с высокочастотными диапазонами и обеспечивает все три вида доступа, причем компьютерные данные передаются обычно со скоростями в несколько сотен килобит в секунду или несколько мегабит в секунду.

К системам последнего типа относятся **многоканальная служба распределения** (Multichannel Multipoint Distribution Service, MMDS) и **локальная служба распределения** (Local Multipoint Distribution Service, LMDS). MMDS работает в диапазоне 2,1 ГГц, а LMDS — 30 ГГц в Америке и 40 ГГц в Европе. Обе системы обеспечивают двунаправленную передачу сигналов для абонентов телевизионных, телефонных и компьютерных услуг. Так как система MMDS работает на существенно более низких частотах, чем LMDS, она обеспечивает гораздо более широкую область покрытия. Одна мачта с направленными антеннами MMDS обычно может обслуживать территорию радиусом в 50 км, в то время как радиус покрытия передатчиков LMDS обычно не превышает 5 км, а в городских условиях он может быть и того меньше. Зато LMDS может обеспечить для своих абонентов более высокие скорости доступа (до 155 Мбит/с).

Как в узкополосных, так и в широкополосных беспроводных абонентских окончаниях используются различные методы мультиплексирования сигналов для одновременной работы своих абонентов в одном секторе направленности антенны, а также для разделения телевизионного, телефонного и компьютерного трафиков. Обычно здесь применяется комбинация приемов FDM и TDM. Например, для каждого типа трафика может быть выделен определенный диапазон частот в соответствии с принципами частотного мультиплексирования. Затем, внутри диапазона частот, выделенного для компьютерного трафика, может применять-

ся асинхронное временно́е мультиплексирование с определенным алгоритмом доступа к общей среде, например с центральным арбитром. Для некоторых абонентов, которым необходима гарантированная полоса пропускания, может применяться синхронное временно́е мультиплексирование с образованием беспроводных каналов PDH/SDH.

К сожалению, технологии WLL до сих пор во многом являются фирменными, с несовместимым оборудованием доступа и центральными станциями. Для устранения этого недостатка сравнительно недавно был разработан стандарт **IEEE 802.16**, который определяет некоторые общие принципы использования частотного диапазона, методов мультиплексирования и предоставляемые услуги. Этот стандарт также предусматривает применение разнообразных методов мультиплексирования, как частотного, так и временно́го синхронного и асинхронного, чтобы учесть интересы различных производителей оборудования WLL и обеспечить максимальную гибкость таких систем.

Технология 802.11 также может использоваться для фиксированного беспроводного доступа. Однако она применяется в этом качестве не так часто, потому что ориентирована исключительно на компьютерный трафик и игнорирует особенности телефонного и телевизионного трафиков, а именно доступ с постоянной битовой скоростью. Метод доступа CDMA/CA, описываемый в 802.11, не может обеспечить требуемого уровня QoS для чувствительного к задержкам трафика. Тем не менее некоторые поставщики услуг применяют технологию 802.11 для фиксированного доступа в Интернет тех абонентов, которых удовлетворяет неопределенная пропускная способность. Эта технология также популярна для «кочевого» доступа в зонах временного пребывания абонентов, например в аэропортах или на железнодорожных вокзалах.

Беспроводной мобильный доступ в Интернет предоставляется сегодня в основном операторами мобильных телефонных сетей. Мобильная телефония второго поколения обеспечивает доступ в Интернет, используя в качестве транспорта с коммутацией пакетов протокол GPRS (General Packet Radio Service — служба пакетной радиосвязи общего назначения), который работает в сетях D-AMPS и GSM. Однако скорость такого доступа невысока (всего 2400–9800 Кбит/с). В мобильных сетях третьего поколения, которые только начинают разворачиваться, эта скорость должна существенно возрасти (до 2 Мбит/с).

Выводы

Термин «удаленный доступ» применяется в том случае, когда говорят о доступе домашних пользователей или сотрудников мелких филиалов предприятий к ресурсам Интернета или корпоративной сети.

Существуют различные категории клиентов удаленного доступа, отличающиеся используемыми абонентскими окончаниями, наличием или отсутствием домашней локальной сети, требованиями к скорости доступа и типом ресурсов, к которым требуется обеспечить доступ (ресурсы публичного домена Интернета или корпоративной сети).

Поставщик услуг обычно стремится сделать абонентское окончание универсальным, то есть способным передавать трафик трех основных терминальных устройств массового пользователя — телефона, телевизора и компьютера.

Базовым сервисом удаленного доступа является режим удаленного узла, когда компьютер пользователя становится узлом локальной сети поставщика услуг или своего предприятия.

Особым режимом удаленного доступа является удаленное управление, когда компьютер пользователя эмулирует терминал, подключенный к другому компьютеру. Удаленное управление позволяет получить пользователю полный контроль над другим компьютером и запускать на нем любые приложения. Это удобно для пользователя, но представляет большую потенциальную опасность для корпоративных ресурсов.

Наиболее старым видом удаленного доступа является коммутируемый доступ через аналоговые окончания PSTN. С помощью обычного модема компьютер устанавливает в телефонной сети соединение с сервером удаленного доступа, подключенного к сети с коммутацией пакетов.

Фиксированная полоса пропускания в 4 кГц, выделяемая пользователям телефонной сети, принципиально ограничивает скорость передачи обычных модемов. Модемы V.90 обеспечивают восходящую скорость до 33,6 Кбит/с и нисходящую скорость до 56 Кбит/с, но в последнем случае только тогда, когда все транзитные телефонные коммутаторы от клиента до сервера удаленного доступа являются цифровыми.

Технология ISDN была разработана для создания универсальной сети, оказывающей, в том числе, услуги компьютерного доступа. Однако сегодня ее скорость передачи (128 Кбит/с) считается слишком низкой для доступа массовых клиентов к мультимедийной информации.

Технология ADSL полностью использует полосу пропускания телефонного абонентского окончания, деля ее на три канала — дуплексный голосовой, восходящий (до 1 Мбит/с) и нисходящий (до 6 Мбит/с) компьютерные. Ограничение на полосу пропускания для абонента телефонной сети в 4 кГц не влияет на работу ADSL-модемов, так как компьютерные данные в ближайшей точке присутствия ответвляются в сеть с коммутацией пакетов.

Кабельные модемы работают на коаксиальном абонентском окончании CATV, которое является разделяемой средой для нескольких абонентов, подключенных к одному и тому же кабелю. Широкая полоса пропускания коаксиального кабеля обеспечивает восходящую скорость до 10 Мбит/с, а нисходящую — до 30–40 Мбит/с.

Для фиксированного беспроводного доступа используются множество фирменных технологий, обеспечивающих доставку пользователю телефонной, телевизионной и компьютерной информации. Для предоставления разнообразных услуг такой доступ требует сочетания частотного и временного мультиплексирования, а также коммутации каналов и пакетов.

Мобильный доступ пока существует в виде дополнительной низкоскоростной услуги по передаче данных через сотовые телефонные сети второго поколения. Стандарты сетей третьего поколения предусматривают более высокие скорости передачи данных, но их внедрение только начинается.

Вопросы и задания

1. Какие факторы определяют сложность удаленного доступа?
2. Каким образом может оказывать услуги доступа поставщик услуг, который не владеет кабельными абонентскими окончаниями?
3. Какие характеристики клиентов удаленного доступа нужно принимать во внимание при организации такого доступа?
4. Какое абонентское окончание можно назвать универсальным?
5. Удаленным узлом какой сети является компьютер пользователя удаленного доступа?

6. Чем отличаются режимы удаленного управления и удаленного узла?
7. Какой вид доступа используется при конфигурировании маршрутизаторов?
8. Почему скорости обычных (коммутируемых) модемов намного уступают скоростям ADSL-модемов и кабельных модемов?
9. Чем отличаются модемы от устройств DSU/CSU?
10. Вы убедились, что модем устойчиво работает на выделенном 2-проводном канале как в асинхронном, так и синхронном режимах. Какой режим вы предпочтете?
11. К устройству какого уровня в терминах модели OSI можно отнести обычный модем?
12. Какие функции выполняет устройство DSLAM?
13. Чем отличаются требования к локальной сети поставщика услуг, предоставляющего услуги коммутируемого доступа, от требований к локальной сети поставщика услуг доступа через ADSL?
14. Какой метод доступа к разделяемой среде используют кабельные модемы?
15. Можно ли использовать один коаксиальный кабель для организации доступа жильцов одного большого многоквартирного дома, насчитывающего 400 квартир?
16. Почему технология 802.11 редко используется для фиксированного беспроводного доступа?
17. В чем отличие технологий MMDS и LMDS?
18. Почему пользователя домашнего компьютера вряд ли устроит доступ по мобильной телефонной сети?
19. Вы купили модем V.90 и связываетесь по телефонной сети со своим знакомым, который также использует модем V.90. Вы уверены, что все телефонные коммутаторы на пути между вами и вашим знакомым работают в цифровом режиме. На какой скорости вы получите соединение со своим знакомым?
20. Какую услугу ISDN целесообразно использовать, если к сети ISDN подключены с помощью терминальных адаптеров два персональных компьютера и им нужно постоянно обмениваться данными со скоростью 2400 бит/с с пульсациями до 9600 бит/с, причем величины задержек пакетов не являются критичными?
21. Какую услугу ISDN целесообразно использовать, если к сети ISDN подключены с помощью маршрутизаторов две локальные сети, причем межсетевой трафик в течение длительного периода времени имеет интенсивность от 100 до 512 Кбит/с?
22. В каких случаях выгоднее использовать для удаленного доступа: сеть ISDN с интерфейсом B + D, выделенный цифровой канал 64 Кбит/с, постоянный виртуальный канал Frame Relay с согласованной скоростью передачи 64 Кбит/с?
23. Что необходимо изменить, если ADSL-модем работает на абонентском окончании с недопустимо высоким процентом ошибок?

ГЛАВА 24 Защита сетевого трафика

В этой, последней, главе книги мы рассмотрим некоторые наиболее популярные сервисы защиты передаваемой информации. Эти сервисы позволяют через публичную сеть, например Интернет, безопасно передавать информацию, обеспечивая ее аутентичность, целостность и конфиденциальность.

Наиболее простым средством для предоставления такого сервиса является технология защищенного канала, которая обеспечивает защиту трафика между двумя пользователями публичной сети, то есть в соответствии с двухточечной топологией. Такая защита осуществляется за счет комплекса средств, опирающихся на различные методы аутентификации пользователей и шифрования их трафика. В IP-сетях широко применяются две технологии защищенного канала — SSL и IPSec. Протокол SSL работает на уровне представления модели OSI, что делает его непрозрачным для приложений. Протокол IPSec является более универсальным средством, так как относится к сетевому уровню и полностью прозрачен для приложений, которые в случае использования IPSec не требуют модификации.

Более масштабным средством защиты трафика являются виртуальные частные сети (VPN). Подобная сеть является своего рода «сетью в сети», сервисом, создающим у пользователей иллюзию существования их частной сети внутри публичной сети. Одним из важнейших свойств такой «частной сети» является защищенность трафика от атак пользователей публичной сети. Сетям VPN доступна не только возможность имитации частной сети; они дают пользователю возможность задействовать собственное адресное пространство (например, частные IP-адреса, такие как адреса сети 10.0.0.0) и обеспечивать качество обслуживания, близкое к качеству выделенного канала.

Технологии, обеспечивающие безопасность данных и лежащие в основе VPN, делятся на два класса: технологии шифрования данных и технологии разделения трафика. Первый класс технологий опирается на технику защищенных каналов, применяя ее теперь уже в других масштабах, то есть объединяя не двух пользователей, а произвольное количество клиентских сетей. Типичным представителем этого класса VPN является технология IPSec VPN.

Второй класс технологий VPN опирается на технику постоянных виртуальных каналов (PVC), позволяющую надежно отделить трафик одного клиента от трафика остальных. В сетях VPN, работающих на основе разграничения трафика, шифрование не требуется, так как принцип функционирования постоянного виртуального канала исключает возможность атаки на него клиента, подключенного к другому постоянному виртуальному каналу. Виртуальные частные сети этого класса строятся на основе технологий ATM, Frame Relay и MPLS. Технологии ATM VPN и Frame Relay VPN являются не чем иным, как другим названием стандартных сервисов постоянных виртуальных каналов для этих технологий, рассмотренных нами в главе 21. В данной главе основное внимание уделяется новому типу виртуальных частных сетей этого класса, а именно — сетям MPLS VPN, которые позволяют предоставлять расширенный набор услуг VPN.

Сервис защищенного канала

Ключевые слова: сервис IPSec, защищенный протокол IP, защищенный канал, протокол SSL и его реализация TLS, слой защищенных сокетов, безопасность транспортного уровня, шифрование, симметричная схема шифрования, секретный ключ, вычислительно необратимая функция, хэш-функция, дайджест-функция, дайджест, цифровая подпись, асимметричный алгоритм шифрования, безопасная ассоциация, транспортный и туннельный режимы передачи, шлюз безопасности, значение проверки целостности, заголовок ESP.

Основное назначение сервиса **IPSec** (Internet Protocol Security — защищенный протокол IP) состоит в обеспечении безопасной передачи данных по IP-сетям. Применение протокола IPSec гарантирует целостность, аутентичность и конфиденциальность данных. Базовой технологией, на основе которой достигаются эти цели, является шифрование. Для протоколов такого назначения используется обобщенное название — **защищенный канал**. Термин «канал» подчеркивает тот факт, что защита данных обеспечивается на протяжении всего пути между двумя узлами сети (хостами или шлюзами)[1].

Иерархия сервисов защищенного канала

IPSec — это только одна из многих, хотя и очень популярная сегодня, технология безопасной передачи данных по публичной сети. Защищенный канал можно построить с помощью системных средств, реализованных на разных уровнях модели OSI (рис. 24.1).

Рис. 24.1. Протоколы, формирующие защищенный канал на разных уровнях модели OSI

[1] Свойства защищенного канала мы кратко обсуждали в главе 6.

Если защита данных осуществляется средствами верхних уровней (прикладного, представления или сеансового), то такой способ защиты не зависит от технологий транспортировки данных (IP или IPX, Ethernet или ATM), что можно считать несомненным достоинством. С другой стороны, приложения при этом становятся зависимыми от конкретного протокола защищенного канала, так как в них должны быть встроены явные вызовы функций этого протокола.

Защищенный канал, реализованный на самом высоком, прикладном, уровне защищает только вполне определенную сетевую службу, например файловую, гипертекстовую или почтовую. Так, протокол S/MIME защищает исключительно сообщения электронной почты. При таком подходе для каждой службы необходимо разрабатывать собственную защищенную версию протокола.

Популярный протокол **SSL** (Secure Socket Layer — слой защищенных сокетов) и его открытая реализация **TLS** (Transport Layer Security — безопасность транспортного уровня) работают на уровне представления. Ясно, что эти протоколы являются более универсальным средством защиты, чем протоколы безопасности прикладного уровня, так как ими могут воспользоваться *любые* приложения. Однако для этого приложения по-прежнему необходимо модифицировать, встраивая явные обращения к интерфейсам API данных протоколов безопасности.

Средства защищенного канала становятся прозрачными для приложений в тех случаях, когда они защищают кадры протоколов сетевого и канального уровней. Однако здесь мы сталкиваемся с другой проблемой — зависимостью сервиса защищенного канала от протокола нижнего уровня. Например, протокол PPTP (сам не являясь протоколом канального уровня) защищает кадры протокола PPP канального уровня, упаковывая их в IP-пакеты. При этом не имеет никакого значения, пакет какого протокола в свою очередь упакован в данном PPP-кадре: IP, IPX, SNA или NetBIOS. С одной стороны, это делает сервис PPTP достаточно универсальным, так как клиент сервиса защищенного канала может использовать любые протоколы в своей сети. С другой стороны, такая схема предъявляет жесткие требования к типу протокола канального уровня, используемому на участке доступа клиента к защищенному каналу — для протокола PPTP таким протоколом может быть *только PPP*. Хотя протокол PPP очень распространен в линиях доступа, сегодня конкуренцию ему составляют протоколы Gigabit/Fast Ethernet, которые все чаще работают не только в локальных, но и глобальных сетях.

Работающий на сетевом уровне протокол IPSec является компромиссным вариантом. С одной стороны, он прозрачен для приложений, с другой — может работать практически во всех сетях, так как основан на широко распространенном протоколе IP и использует любую технологию канального уровня (PPP, Ethernet, ATM и т. д.).

Распределение функций между протоколами IPSec

Протокол IPSec — называют в стандартах Интернета *системой*. Действительно, IPSec — это согласованный набор открытых стандартов, имеющий сегодня вполь-

не очерченное ядро, которое в то же время может быть достаточно просто дополнено новыми функциями и протоколами.

Ядро IPSec составляют три протокола:

- AH (Authentication Header — заголовок аутентификации) — гарантирует целостность и аутентичность данных;
- ESP (Encapsulating Security Payload — инкапсуляция зашифрованных данных) — шифрует передаваемые данные, обеспечивая конфиденциальность, может также поддерживать аутентификацию и целостность данных;
- IKE (Internet Key Exchange — обмен ключами Интернета) — решает вспомогательную задачу автоматического предоставления конечным точкам защищенного канала секретных ключей, необходимых для работы протоколов аутентификации и шифрования данных.

Как видно из краткого описания функций, возможности протоколов AH и ESP частично перекрываются (рис. 24.2). В то время как AH отвечает только за обеспечение целостности и аутентификации данных, ESP может шифровать данные и, кроме того, выполнять функции протокола AH (хотя, как увидим позднее, аутентификация и целостность обеспечиваются им в несколько урезанном виде). ESP может поддерживать функции шифрования и аутентификации/целостности в любых комбинациях, то есть либо всю группу функций, либо только аутентификацию/целостность, либо только шифрование.

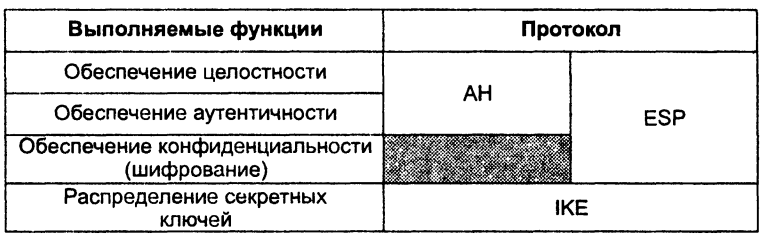


Рис. 24.2. Распределение функций между протоколами IPSec

Разделение функций защиты между протоколами AH и ESP вызвано применяемой во многих странах практикой ограничения экспорта и/или импорта средств, обеспечивающих конфиденциальность данных путем шифрования. Каждый из этих протоколов может использоваться как самостоятельно, так и одновременно с другим, так что в тех случаях, когда шифрование из-за действующих ограничений применять нельзя, систему можно поставлять только с протоколом AH. Естественно, подобная защита данных во многих случаях оказывается недостаточной. Принимающая сторона в этом случае получает лишь возможность проверить, что данные были отправлены именно тем узлом, от которого они ожидаются, и дошли в том виде, в котором были отправлены. Однако от несанкционированного просмотра данных на пути их следования по сети протокол AH защитить не может, так как не шифрует их. Для шифрования данных необходим протокол ESP.

Шифрование в протоколе IPSec

Для **шифрования** данных в протоколе IPSec может быть применен любой симметричный алгоритм шифрования. В **симметричных схемах шифрования** конфиденциальность основана на том, что отправитель и получатель обладают общим, известным только им, параметром функции шифрования. Этот параметр называется **секретным ключом**. Секретный ключ используется как для шифрования текста, так и для его дешифрирования.

На рис. 24.3 приведена классическая модель симметричной криптосистемы, теоретические основы которой впервые были изложены в 1949 году в работе Клода Шеннона (Claude Shannon). В данной модели есть три участника: отправитель, получатель, злоумышленник. Задача отправителя заключается в том, чтобы по общедоступному каналу передать некоторое сообщение в защищенном виде. Для этого он, используя ключ k, зашифровывает открытый текст X и передает зашифрованный текст Y. Задача получателя заключается в том, чтобы расшифровать текст Y и прочитать сообщение X. Предполагается, что отправитель имеет свой источник ключа. Сгенерированный ключ заранее по надежному каналу передается получателю.

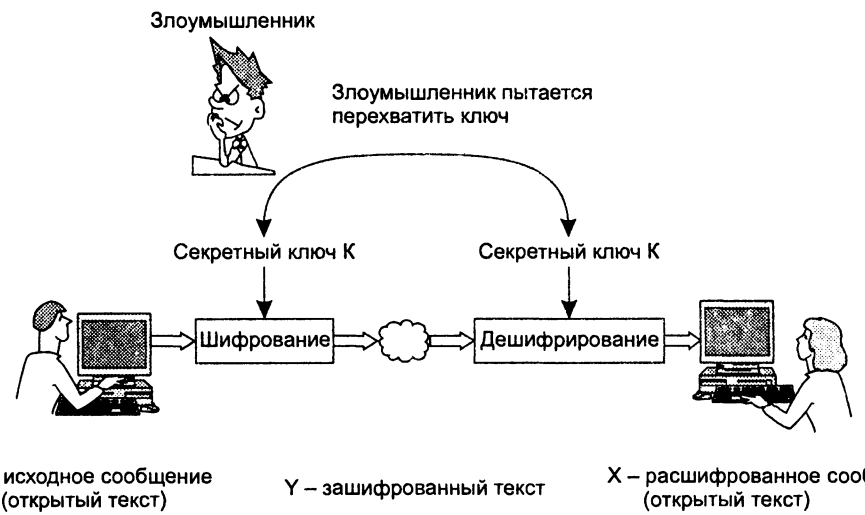


Рис. 24.3. Симметричная схема шифрования

ПРИМЕЧАНИЕ

Наиболее популярным симметричным алгоритмом шифрования данных в течение многих лет был алгоритм DES (Data Encryption Standard — стандарт шифрования данных), разработанный фирмой IBM в 1976 г. На смену DES в 2001 г. пришел новый более совершенный алгоритм AES (Advanced Encryption Standard — улучшенный стандарт шифрования), обладающий лучшим, чем DES, сочетанием показателей безопасности и быстродействия.

В основе обеспечения целостности и аутентификации данных также лежит один из приемов шифрования — шифрование с помощью **вычислительно необратимой**

функции (One-Way Function, OWF), частными случаями которой являются **хэш-функция** и **дайджест-функция**.

Значение, полученное в результате применения функций этого типа к данным, не позволяет и даже *не предполагает* восстановления исходных данных.

Зачем же нужны такие функции?

Поясним это на примере дайджест-функции. Однако прежде отметим особенность дайджест-функции: независимо от объема исходных данных, к которым применяется эта функция, результат, называемый **дайджестом**, состоит из фиксированного, как правило, небольшого числа байтов.

Пусть требуется гарантировать целостность данных при передаче их через ненадежную сеть. Для достижения этой цели на стороне отправителя вычисляется дайджест передаваемых данных. Дайджест передается по сети вместе с исходным сообщением (рис. 24.4). Получатель, зная, какая необратимая функция была применена для вычисления дайджеста, заново вычисляет его, используя полученное сообщение. Если значения полученного по сети и вычисленного локально дайджестов совпадают, значит, сообщение во время передачи не было подвергнуто никаким изменениям.

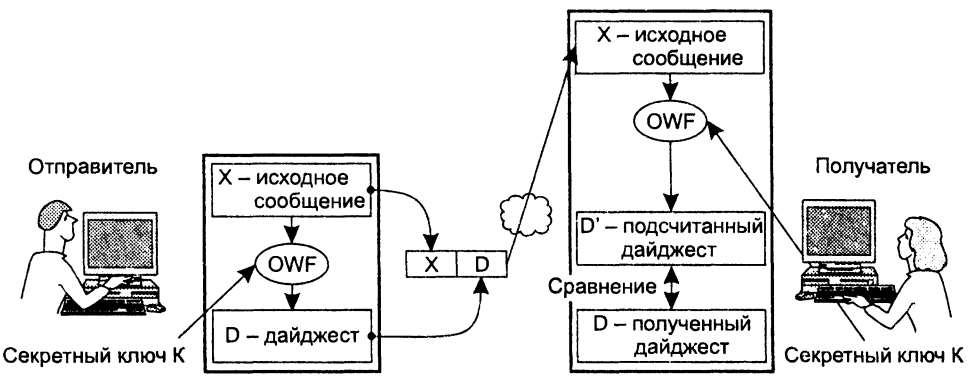


Рис. 24.4. Использование шифрования для обеспечения целостности и аутентичности данных

ПРИМЕЧАНИЕ

Наиболее популярными в системах безопасности в настоящее время является серия дайджест-функций MD2, MD4, MD5. Все они генерируют дайджесты фиксированной длины в 16 байт. Адаптированным вариантом MD4 является американский стандарт SHA, длина дайджеста в котором составляет 20 байт. Компания IBM поддерживает дайджест-функции MDC2 и MDC4.

Дайджест является своего рода контрольной суммой для исходного сообщения. Однако имеется и существенное отличие. Использование контрольной суммы — это средство проверки целостности передаваемых *по ненадежным линиям связи* сообщений, и оно не направлено на борьбу со *злонамеренными действиями*. В са-

мом деле, наличие контрольной суммы в передаваемом пакете не помешает злоумышленнику подменить исходное сообщение, добавив к нему новое значение контрольной суммы. В отличие от контрольной суммы при вычислении дайджеста используется секретный ключ. Если для получения дайджеста применялась необратимая функция с параметром (в качестве которого выступает секретный ключ), известным только отправителю и получателю, любая модификация исходного сообщения будет немедленно обнаружена.

Таким образом, вычислительно необратимая функция является средством решения сразу двух задач — контроля целостности и аутентичности данных. Такую схему передачи данных наряду с другими методами, позволяющими устанавливать подлинность автора сообщения, согласно терминологии ISO называют **цифровой подписью**. Основная область применения цифровой подписи — финансовые документы, сопровождающие электронные сделки, документы, фиксирующие международные договоренности и т. п.

Наиболее часто для построения схемы цифровой подписи используется **асимметричный алгоритм шифрования**. В основе этого алгоритма лежит концепция Диффи—Хеллмана (Diffie—Hellmann), заключающаяся в том, что каждый пользователь сети имеет свой секретный (закрытый) ключ, необходимый для формирования подписи в зашифрованном виде; все остальные пользователи используют для проверки подписи соответствующий этому секретному ключу открытый ключ.

Безопасная ассоциация

Для того чтобы протоколы AH и ESP могли выполнять свою работу по защите передаваемых данных, протокол IKE устанавливает между двумя конечными точками логическое соединение (рис. 24.5), которое в стандартах IPSec носит название **безопасной ассоциации** (Security Association, SA).

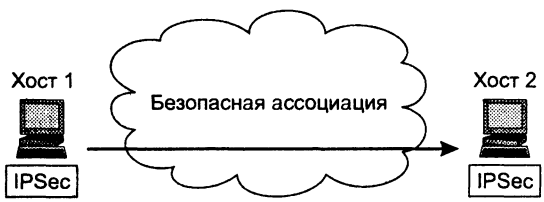


Рис. 24.5. Безопасная ассоциация

Стандарты IPSec позволяют конечным точкам защищенного канала использовать как одну ассоциацию SA для передачи трафика всех взаимодействующих через этот канал хостов, так и создавать для этой цели произвольное число ассоциаций SA, например, по одной на каждое соединение TCP. Это дает возможность выбирать нужную степень детализации защиты — от одной общей ассоциации для трафика множества конечных узлов до индивидуально настроенных ассоциаций для защиты каждого приложения.

Безопасная ассоциация в протоколе IPSec представляет собой однонаправленное (симплексное) логическое соединение, поэтому, если требуется обеспечить

безопасный двусторонний обмен данными, то необходимо установить две безопасные ассоциации. Эти ассоциации в общем случае могут иметь разные характеристики, например, в одну сторону при передаче запросов к базе данных достаточно только аутентификации, а для ответных данных, несущих ценную информацию, дополнительно нужно обеспечить конфиденциальность.

Установление безопасной ассоциации начинается со взаимной аутентификации сторон, потому что все меры безопасности теряют смысл, если данные передаются или принимаются не тем лицом или не от того лица. Выбираемые далее параметры SA определяют, какой из двух протоколов, AH или ESP, будет применяться для защиты данных, какие функции будет выполнять протокол (например, можно выполнять только аутентификацию и проверку целостности или, кроме того, еще и обеспечивать конфиденциальность). Очень важными параметрами безопасной ассоциации являются также секретные ключи, используемые в работе протоколов AH и ESP.

Протокол IPSec допускает как автоматическое, так и ручное установление безопасной ассоциации. При ручном способе администратор конфигурирует конечные узлы так, чтобы они поддерживали согласованные параметры ассоциации, включая и секретные ключи. При автоматической процедуре установления SA протоколы IKE, работающие по разные стороны канала, выбирают параметры в ходе переговорного процесса. Для каждой задачи, решаемой протоколами AH и ESP, предлагается несколько схем аутентификации и шифрования (рис. 24.6). Это делает протокол IPSec очень гибким средством. Заметим, что выбор дайджест-функции для решения задач целостности и аутентификации никак не влияет на выбор функции шифрования, обеспечивающей конфиденциальность данных.

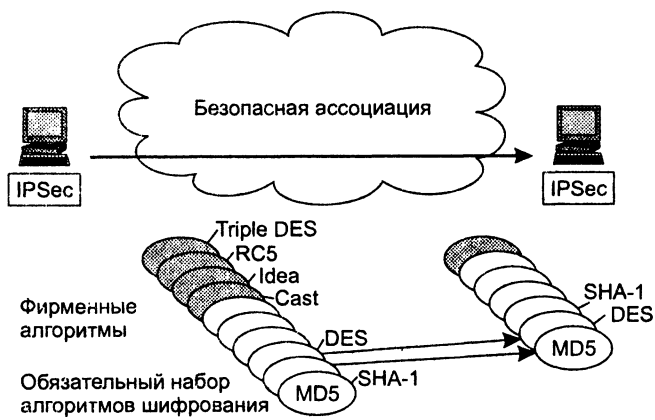


Рис. 24.6. Согласование параметров в протоколе ESP

Для обеспечения совместимости в стандартной версии IPsec определен некоторый обязательный «инструментальный» набор: в частности, для аутентификации данных всегда может быть использована одна из стандартных дайджест-функций MD5 либо SHA-1, а в число алгоритмов шифрования непременно вхо-

дит DES. При этом производители продуктов, включающих IPSec, вольны расширять протокол путем включения других алгоритмов аутентификации и шифрования, что они с успехом и делают. Например, многие реализации IPSec поддерживают популярный алгоритм шифрования Triple DES, а также сравнительно новые алгоритмы — Blowfish. Cast, CDMF, Idea, RC5.

Транспортный и туннельный режимы

Протоколы AH и ESP могут защищать данные в двух режимах: транспортном и туннельном. В **транспортном режиме** передача IP-пакета через сеть выполняется с помощью оригинального заголовка этого пакета, а в **туннельном режиме** исходный пакет помещается в новый IP-пакет, и передача данных по сети выполняется на основании заголовка нового IP-пакета.

Применение того или иного режима зависит от требований, предъявляемых к защите данных, а также от роли, которую играет в сети узел, завершающий защищенный канал. Так, узел может быть хостом (конечным узлом) или шлюзом (промежуточным узлом). Соответственно, имеются три схемы применения протокола IPSec:

- хост—хост;
- шлюз—шлюз;
- хост—шлюз.

В схеме хост—хост защищенный канал, или, что в данном контексте одно и то же, безопасная ассоциация, устанавливается между двумя конечными узлами сети (см. рис. 24.5). Тогда протокол IPSec работает на конечных узлах и защищает данные, передаваемые от хоста 1 к хосту 2. Для схемы хост—хост чаще всего используется транспортный режим защиты.

В соответствии со схемой шлюз—шлюз защищенный канал устанавливается между двумя промежуточными узлами, так называемыми **шлюзами безопасности** (Security Gateway, SG), на каждом из которых работает протокол IPSec (рис. 24.7). Защищенный обмен данными может происходить между любыми двумя конечными узлами, подключенными к сетям, которые расположены позади шлюзов безопасности. От конечных узлов поддержка протокола IPSec не требуется, они передают свой трафик в незащищенном виде через заслуживающие доверие внутренние сети предприятий. Трафик, направляемый в общедоступную сеть, проходит через шлюз безопасности, который и обеспечивает его защиту с помощью протокола IPSec. Шлюзы могут использовать только туннельный режим работы.

На рис. 24.7 пользователь компьютера с адресом IP1 посылает пакет по адресу IP2, используя туннельный режим протокола IPSec. Шлюз SG1 зашифровывает пакет целиком, вместе с заголовком, и снабжает его новым заголовком IP, в котором в качестве адреса отправителя указывает свой адрес — IP3, а в качестве адреса получателя — адрес IP4 шлюза SG2. Вся передача данных по составной IP-сети выполняется на основании заголовка внешнего пакета, а внутренний пакет становится при этом полем данных для внешнего пакета. На шлюзе SG2 протокол IPSec извлекает инкапсулированный пакет и расшифровывает его, приводя к исходному виду.

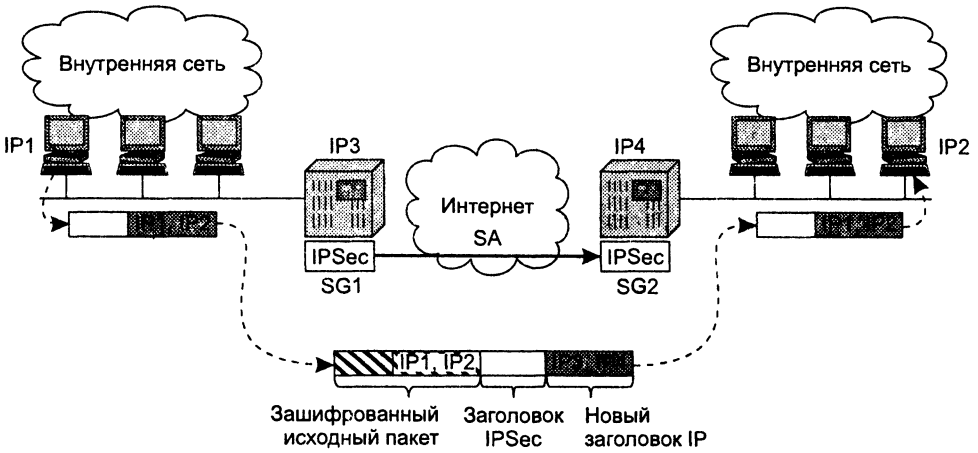


Рис. 24.7. Работа защищенного канала по схеме шлюз—шлюз в туннельном режиме

Схема хост—шлюз часто применяется при удаленном доступе. В этом случае защищенный канал организуется между удаленным хостом, на котором работает протокол IPSec, и шлюзом, защищающим трафик для всех хостов, входящих во внутреннюю сеть предприятия. Эту схему можно усложнить, создав параллельно еще один защищенный канал — между удаленным хостом и каким-либо хостом, принадлежащим внутренней сети, защищаемой шлюзом (рис. 24.8). Такое комбинированное использование двух безопасных ассоциаций позволяет надежно защитить трафик и во внутренней сети.

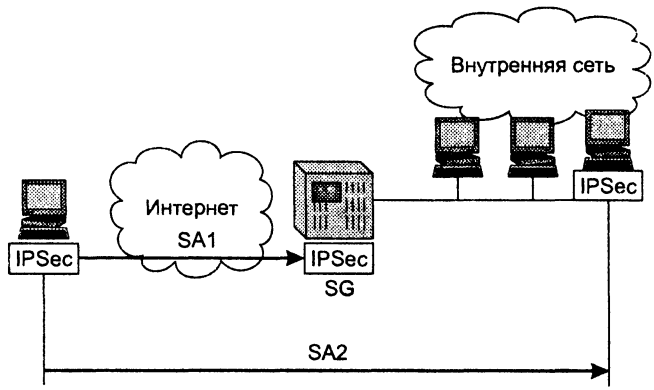


Рис. 24.8. Схема защищенного канала хост—шлюз

Протокол AH

Протокол AH позволяет приемной стороне убедиться, что:

- пакет был отправлен стороной, с которой установлена безопасная ассоциация;
- содержимое пакета не было искажено в процессе его передачи по сети;
- пакет не является дубликатом уже полученного пакета.

Две первые функции обязательны для протокола AH, а последняя выбирается при установлении ассоциации по желанию. Для выполнения этих функций протокол AH использует специальный заголовок (рис. 24.9).

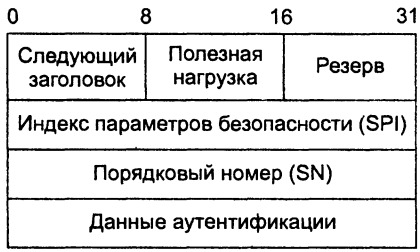


Рис. 24.9. Структура заголовка протокола AH

В поле *следующего заголовка* (next header) указывается код протокола более высокого уровня, то есть протокола, сообщение которого размещено в поле данных IP-пакета. Скорее всего, им будет один из протоколов транспортного уровня (TCP или UDP) или протокол ICMP, но может встретиться и протокол ESP, если он используется в комбинации с AH.

В поле *длины полезной нагрузки* (payload length) содержится длина заголовка AH.

Индекс параметров безопасности (Security Parameters Index, SPI) используется для связи пакета с предусмотренной для него безопасной ассоциацией. Немного позже мы обсудим его более подробно.

Поле *порядкового номера* (Sequence Number, SN) указывает на порядковый номер пакета и применяется для защиты от его ложного воспроизведения (когда третья сторона пытается повторно использовать перехваченные защищенные пакеты, отправленные реально аутентифицированным отправителем). Отправляющая сторона последовательно увеличивает значение этого поля в каждом новом пакете, передаваемом в рамках данной ассоциации, так что приход дубликата обнаружится принимающей стороной (если, конечно, в рамках ассоциации будет активизирована функция защиты от ложного воспроизведения). Однако в любом случае в функции протокола AH не входит восстановление утерянных и упорядочивание прибывающих пакетов — он просто отбрасывает пакет в том случае, когда обнаруживает, что аналогичный пакет уже был получен. Чтобы сократить требуемую для работы протокола буферную память, используется механизм скользящего окна — на предмет дублирования проверяются только те пакеты, чей номер находится в пределах окна. Окно обычно выбирается размером в 32 или 64 пакета.

Поле *данных аутентификации* (authentication data), которое содержит так называемое **значение проверки целостности** (Integrity Check Value, ICV), используется для аутентификации и проверки целостности пакета. Это значение, называемое также дайджестом, вычисляется с помощью одной из двух обязательно поддерживаемых протоколом AH вычислительно необратимых функций MD5 или SAH-1, но может использоваться и любая другая функция, о которой стороны договорились в ходе установления ассоциации. При вычислении дайджеста

пакета в качестве параметра функции OWF выступает симметричный секретный ключ, который был задан для данной ассоциации вручную или автоматически с помощью протокола IKE. Так как длина дайджеста зависит от выбранной функции, то это поле имеет в общем случае переменный размер.

Протокол AH старается охватить при вычислении дайджеста как можно большее число полей исходного IP-пакета, но некоторые из них в процессе передачи пакета по сети меняются непредсказуемым образом, поэтому не могут быть включены в аутентифицируемую часть пакета. Например, целостность значения поля времени жизни (TTL) в приемной точке канала оценить нельзя, так как оно уменьшается на единицу каждым промежуточным маршрутизатором и никак не может совпадать с исходным.

Местоположение заголовка AH в пакете зависит от того, в каком режиме — транспортном или туннельном — сконфигурирован защищенный канал. Результирующий пакет в транспортном режиме выглядит так, как показано на рис. 24.10.

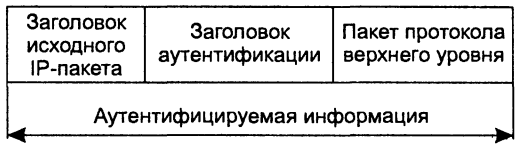


Рис. 24.10. Структура IP-пакета, обработанного протоколом AH в транспортном режиме

При использовании туннельного режима, когда шлюз IPSec принимает проходящий через него транзитом исходящий пакет и создает для него внешний IP-пакет, протокол AH защищает все поля исходного пакета, а также неизменяемые поля нового заголовка внешнего пакета (рис. 24.11).

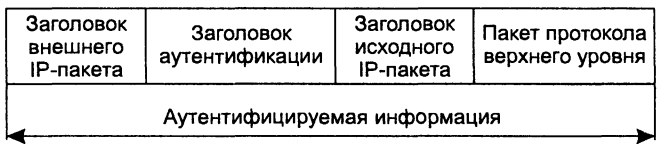


Рис. 24.11. Структура IP-пакета, обработанного протоколом AH в туннельном режиме

Протокол ESP

Протокол ESP решает две группы задач. К первой относятся задачи, аналогичные задачам протокола AH, — это обеспечение аутентификации и целостности данных на основе дайджеста, а ко второй — защита передаваемых данных путем их шифрования от несанкционированного просмотра.

Как видно на рис. 24.12, заголовок делится на две части, разделяемые полем данных. Первая часть, называемая собственно **заголовком ESP**, образуется двумя полями (SPI и SN), назначение которых аналогично одноименным полям протокола AH, и размещается перед полем данных. Остальные служебные поля протокола ESP, называемые концевиком ESP, расположены в конце пакета.

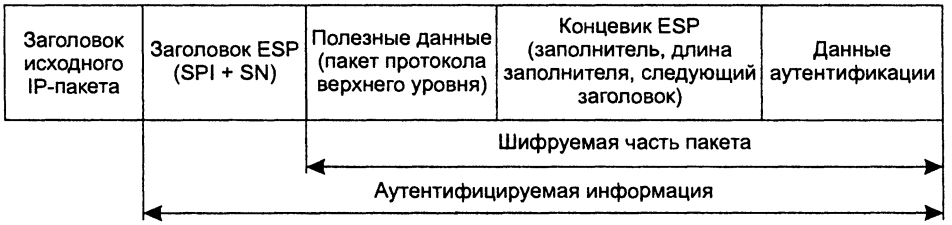


Рис. 24.12. Структура IP-пакета, обработанного протоколом ESP в транспортном режиме

Два поля концевика — *следующего заголовка* и *данных аутентификации* — также аналогичны полям заголовка AH. Поле данных аутентификации отсутствует, если при установлении безопасной ассоциации принято решение не использовать возможностей протокола ESP по обеспечению целостности. Помимо этих полей концевик содержит два дополнительных поля — *заполнителя* и *длины заполнителя*. Заполнитель может понадобиться в трех случаях. Во-первых, для нормальной работы некоторых алгоритмов шифрования необходимо, чтобы шифруемый текст содержал кратное число блоков определенного размера. Во-вторых, формат заголовка ESP требует, чтобы поле данных заканчивалось на границе четырех байтов. И наконец, заполнитель можно использовать, чтобы скрыть действительный размер пакета в целях обеспечения так называемой частичной конфиденциальности трафика. Правда, возможность маскировки ограничивается сравнительно небольшим объемом заполнителя — 255 байт, поскольку большой объем избыточных данных может снизить полезную пропускную способность канала связи.

На рис. 24.12 показано размещение полей заголовка ESP в *транспортном режиме*. В этом режиме ESP не шифрует заголовок IP-пакета, иначе маршрутизатор не сможет прочитать поля заголовка и корректно осуществить продвижение пакета между сетями. В число шифруемых полей не попадают также поля SPI и SN, которые должны передаваться в открытом виде для того, чтобы прибывший пакет можно было отнести к определенной ассоциации и защититься от ложного воспроизведения пакета.

В туннельном режиме заголовок исходного IP-пакета помещается после заголовка ESP и полностью попадает в число защищаемых полей, а заголовок внешнего IP-пакета протоколом ESP не защищается (рис. 24.13).

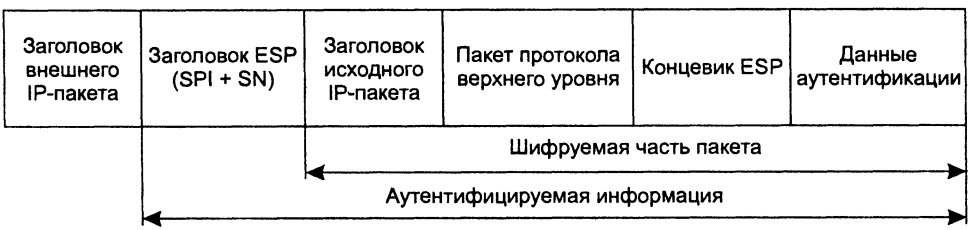


Рис. 24.13. Структура IP-пакета, обработанного протоколом ESP в туннельном режиме

Базы данных SAD И SPD

Итак, технология IPSec предлагает различные методы защиты трафика. Каким же образом протокол IPSec, работающий на хосте или шлюзе, определяет способ защиты, который он должен применить к трафику? Решение основано на использовании в каждом узле, поддерживающем IPSec, двух типов баз данных:

- безопасных ассоциаций (Security Associations Database, SAD);
- политики безопасности (Security Policy Database, SPD).

При установлении безопасной ассоциации, как и при любом другом логическом соединении, две стороны принимают ряд соглашений, регламентирующих процесс передачи потока данных между ними. Соглашения фиксируются в виде набора параметров. Для безопасной ассоциации такими параметрами являются, в частности, тип и режим работы протокола защиты (AH или ESP), методы шифрования, секретные ключи, значение текущего номера пакета в ассоциации и другая информация. Наборы текущих параметров, определяющих все активные ассоциации, хранятся на обоих оконечных узлах защищенного канала в виде баз данных безопасных ассоциаций (SAD). Каждый узел IPSec поддерживает две базы SAD — одну для исходящих ассоциаций, другую для входящих.

Другой тип базы данных — база данных политики безопасности (SPD) — определяет соответствие между IP-пакетами и установленными для них правилами обработки. Записи SPD состоят из полей двух типов — полей селектора пакета и полей политики защиты для пакета с данным значением селектора (рис. 24.14).

Селектор в SPD включает следующий набор признаков, на основании которых можно с большой степенью детализации выделить защищаемый поток:

- IP-адреса источника и приемника могут быть представлены как в виде отдельных адресов (индивидуальных, групповых или широковещательных), так и диапазонами адресов, заданными с помощью верхней и нижней границ либо с помощью маски;
- порты источника и приемника (то есть TCP- или UDP-портов);
- тип протокола транспортного уровня (TCP, UDP);
- имя пользователя в формате DNS или X.500;
- имя системы (хоста, шлюза безопасности и т. п.) в формате DNS или X.500.

Для каждого нового пакета, поступающего в защищенный канал, IPSec просматривает все записи в базе SPD и сравнивает значение селекторов этих записей с соответствующими полями IP-пакета. Если значение полей совпадает с каким-либо селектором, то над пакетом выполняются действия, определенные в поле политики безопасности данной записи. Политика предусматривает одну из следующих возможностей: передача пакета без изменения, отбрасывание, обработка средствами IPSec.

В последнем случае поле политики защиты должно содержать ссылку на запись в базе данных SAD, в которую помещен набор параметров безопасной ассоциации для данного пакета (на рис. 24.14 для исходящего пакета определена ассоциация SA3). На основании заданных параметров безопасной ассоциации к па-

кету применяется соответствующий протокол (на рисунке — ESP), функции шифрования и секретные ключи.

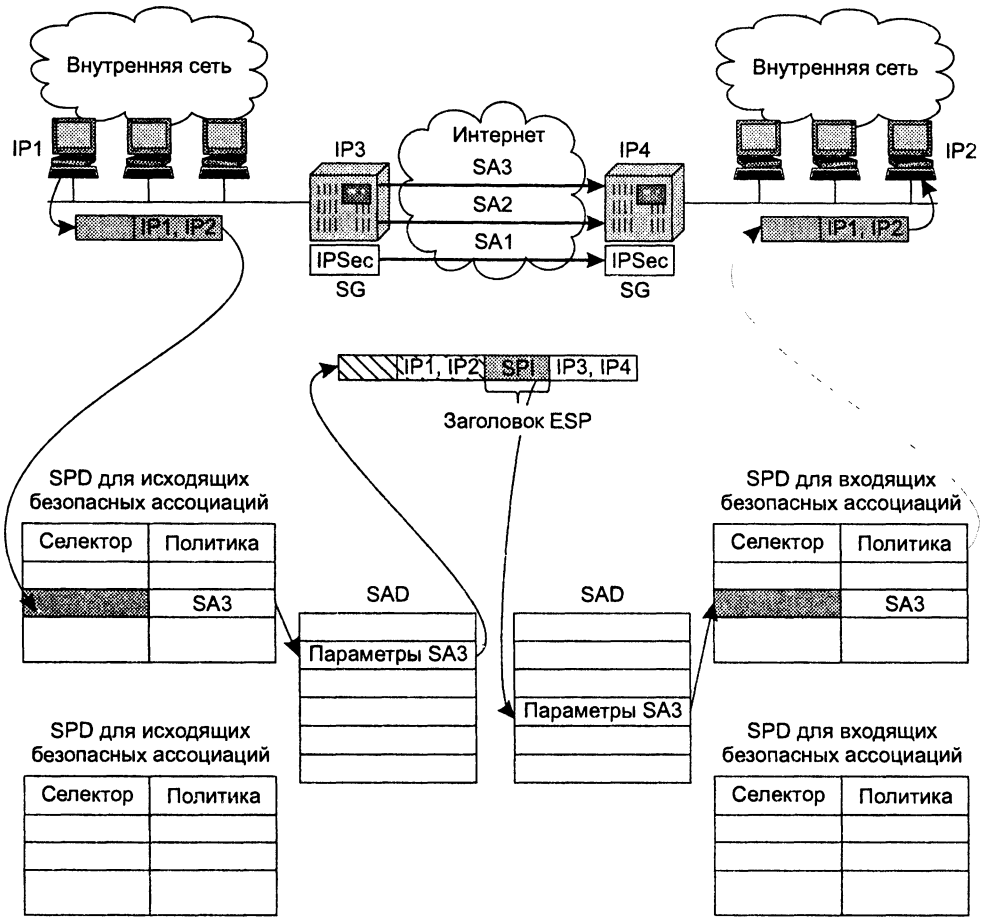


Рис. 24.14. Использование баз данных SPD и SAD

Если к исходящему пакету нужно применить некоторую политику защиты, но указатель записи SPD показывает, что в настоящее время нет активной безопасной ассоциации с требуемой политикой, то IPSec создает новую ассоциацию с помощью протокола IKE, помещая новые записи в базы данных SAD и SPD.

Базы данных политики безопасности создаются и администрируются либо пользователем (этот вариант больше подходит для хоста), либо системным администратором (вариант для шлюза), либо автоматически (приложением).

Выше мы рассмотрели, что установление связи между исходящим IP-пакетом и заданной для него безопасной ассоциацией происходит путем селекции. Однако остается другой вопрос: как *принимающий* узел IPSec определяет способ обработки прибывшего пакета, ведь при шифровании многие ключевые параметры

пакета, отраженные в селекторе, оказываются недоступными, а значит, невозможно определить соответствующую запись в базах данных SAD и SPD и, следовательно, тип процедуры, которую надо применить к поступившему пакету. Именно для решения этой проблемы в заголовках AH и ESP предусмотрено поле SPI. В это поле помещается указатель на строку базы данных SAD, в которой записаны параметры соответствующей безопасной ассоциации. Поле SPI заполняется протоколом AH или ESP во время обработки пакета в отправной точке защищенного канала. Когда пакет приходит в конечный узел защищенного канала, из его внешнего заголовка ESP или AH (на рисунке — из заголовка ESP) извлекается значение SPI, и дальнейшая обработка пакета выполняется с учетом всех параметров заданной этим указателем ассоциации.

Таким образом, для распознавания пакетов, относящихся к разным безопасным ассоциациям (SA), используются:

- на узле-отправителе — селектор;
- на узле-получателе — индекс параметров безопасности (SPI).

После дешифрирования пакета приемный узел IPSec проверяет его признаки (ставшие теперь доступными) на предмет совпадения с селектором записи SPD для входящего трафика, чтобы убедиться, что ошибки не произошло и выполняемая обработка пакета соответствует политике защиты, заданной администратором.

Использование баз SPD и SAD для защиты трафика позволяет достаточно гибко сочетать механизм безопасных ассоциаций, который предусматривает установление логического соединения, с дейтаграммным характером трафика протокола IP.

Сервис виртуальных частных сетей

Ключевые слова: виртуальная частная сеть, поддерживаемая клиентом и поддерживаемая поставщиком; виртуальная частная сеть на базе оборудования, установленного на территории потребителя, и на базе собственной инфраструктуры поставщика; интранет, или внутренняя сеть; экстранет, или внешняя сеть, сеть VPN уровней 2 и 3, сеть MPLS VPN, сеть VPN на основе шифрования.

Определение виртуальной частной сети

Из самого названия — **виртуальная частная сеть** (Virtual Private Network, VPN) — следует, что она каким-то образом воспроизводит свойства *реальной частной сети*. Без всяких натяжек назвать сеть *частной* можно только в том случае, если предприятие единолично владеет и управляет всей сетевой инфраструктурой — кабелями, кроссовым оборудованием, каналообразующей аппаратурой, коммутаторами, маршрутизаторами и другим коммуникационным оборудованием.

Главным отличием частной сети от общедоступной или совместно используемой несколькими предприятиями сети является ее *изолированность*.

Перечислим, в чем выражается эта изолированность.

- *Независимый выбор сетевых технологий.* Выбор ограничивается только возможностями производителей оборудования.
- *Независимая система адресации.* В частных сетях нет ограничений на выбор адресов — они могут быть любыми.
- *Предсказуемая производительность.* Собственные линии связи гарантируют заранее известную пропускную способность между узлами предприятия (для глобальных соединений) или коммуникационными устройствами (для локальных соединений).
- *Максимально возможная безопасность.* Отсутствие связей с внешним миром ограждает сеть от атак извне и существенно снижает вероятность «прослушивания» трафика по пути следования.

Однако частная сеть — решение крайне неэкономичное! Такие сети, особенно в национальном или международном масштабах, могут себе позволить только очень крупные и богатые предприятия. Роскошь создания частной сети — привилегия тех, кто имеет производственные предпосылки для создания собственной сетевой инфраструктуры. Например, нефтяные или газовые компании способны с относительно невысокими издержками прокладывать собственные технологические кабели связи вдоль трубопроводов. Частные сети были популярны в относительно далеком прошлом, когда общедоступные сети передачи данных были развиты очень слабо. Сегодня же их почти повсеместно вытеснили сети VPN, которые предлагают компромисс между качеством услуг и их стоимостью.

Технология VPN позволяет с помощью разделяемой несколькими предприятиями сетевой инфраструктуры реализовать сервисы, приближающиеся к сервисам частной сети по качеству (безопасность, доступность, предсказуемая пропускная способность, независимость в выборе адресов).

В зависимости от того, кто реализует сети VPN, они подразделяются на два вида.

- **Поддерживаемая клиентом виртуальная частная сеть** (Customer Provided VPN, CPVPN) отражает тот факт, что все тяготы поддержки сети VPN ложатся на плечи потребителя. Поставщик предоставляет только «простые» традиционные услуги общедоступной сети по объединению узлов клиента, а специалисты предприятия самостоятельно конфигурируют средства VPN и управляют ими.
- В случае **поддерживаемой поставщиком виртуальной частной сети** (Provider Provisioned VPN, PPVPN) поставщик услуг на основе собственной сети воспроизводит частную сеть для каждого своего клиента, изолируя и защищая ее от остальных. Такой способ организации VPN сравнительно нов и не столь широко распространен, как первый.

В последние год-два популярность сетей PPVPN растет — заботы по созданию и управлению VPN довольно обременительны и специфичны, поэтому многие предприятия предпочитают переложить их на плечи надежного поставщика. Реализация услуг VPN позволяет поставщику оказывать и ряд дополнительных услуг, включая контроль за работой клиентской сети, веб-хостинг и хостинг почтовых служб, хостинг специализированных приложений клиентов.

Помимо деления сетей VPN на CPVPN и PPVPN существует еще и другая классификация — в зависимости от места расположения устройств, выполняющих функции VPN. Виртуальная частная сеть может строиться:

- **на базе оборудования, установленного на территории потребителя** (Customer Premises Equipment based VPN, CPE-based VPN, или Customer Edge based VPN, CE-based VPN);
- **на базе собственной инфраструктуры поставщика** (Network-based VPN или Provider Edge based VPN, PE-based VPN).

В любом случае основную часть функций (или даже все) по поддержанию VPN выполняют пограничные устройства сети — либо потребителя, либо поставщика.

Сети, поддерживаемые поставщиком, могут строиться как на базе инфраструктуры поставщика, так и на базе оборудования, установленного на территории потребителя. Первый вариант наиболее понятен: поставщик управляет расположенным в его сети оборудованием. Во втором случае оборудование VPN расположено на территории клиента, но поставщик управляет им удаленно, что освобождает специалистов предприятия-клиента от достаточно сложных и специфических обязанностей.

Когда VPN поддерживается клиентом (CPVPN), оборудование всегда находится в его сети, то есть VPN строится на базе устройств клиента (CE-based).

Критерии оценки и сравнения VPN

Сеть VPN, как и любая *имитирующая система*[1], характеризуется, во-первых, тем, какие свойства объекта имитируются, во-вторых, степенью приближенности к оригиналу и, в-третьих, используемыми средствами имитации.

Рассмотрим, какие элементы частной сети являются предметом «виртуализации» в VPN.

Практически все сети VPN имитируют *собственные каналы* в сетевой инфраструктуре поставщика, предназначенной для обслуживания множества клиентов.

В том случае, когда имитируется инфраструктура каналов одного предприятия, услуги VPN называют также услугами **intranet** (*интранет*, или *внутренней сети*), а в том случае, когда к таким каналам добавляются также каналы, соединяющие предприятие с его предприятиями-партнерами, с которыми также необходимо обмениваться информацией в защищенном режиме, — услугами **extranet** (*экстранет*, или *внешней сети*).

[1] В данном случае VPN рассматривается как имитация частной сети предприятия.

ПРИМЕЧАНИЕ

Здесь возможна небольшая терминологическая путаница — имитация собственных каналов в среде сети оператора средствами временно́го мультиплексирования (телефонного, PDH или SDH) традиционно не считается виртуальной частной сетью. Сети, построенные на собственном сетевом оборудовании, но на арендованных физических каналах, обычно относят к действительно частным. Это связано с тем, что используемая в этих сетях синхронная технология TDM позволяет гарантированно изолировать информационные потоки пользователей друг от друга и обеспечить в разделяемой среде для каждого потока фиксированную пропускную способность, а также задержки и другие параметры QoS на уровне неразделяемой среды. Недаром такие каналы называют также выделенными — они предоставляются клиенту в единичное владение, и их ресурсами другие клиенты воспользоваться не смогут. Именно таким образом — не на собственных, а на арендованных каналах — работает подавляющее число частных сетей.

Термин «виртуальная частная сеть» применяется только тогда, когда «собственные» физические каналы имитируются средствами пакетных технологий: ATM, Frame Relay, X.25, IP или IP/MPLS. Качество связи между узлами клиентов в этом случае уже вполне ощутимо отличается от того, которое было бы при связи их действительно собственным физическим каналом. В частности, появляется неопределенность пропускной способности и других характеристик связи, поэтому определение «виртуальная» становится здесь уместным. При применении пакетных сетей для построения VPN клиентам предоставляются не только физические каналы, но и определенная технология канального уровня (например, ATM или Frame Relay), а при использовании IP — и сетевого.

Виртуальная частная сеть может имитировать не только физические каналы, но и более высокоуровневые свойства сети. Так, может быть спроектирована сеть VPN, способная поддерживать IP-трафик клиента с созданием эффекта изолированной IP-сети. В этом случае VPN производит некоторые дополнительные сетевые операции над клиентским трафиком — сбор разнообразной статистики, фильтрацию и экранирование взаимодействий между пользователями и подразделениями одного и того же предприятия (не нужно путать с экранированием от внешних пользователей — это основная функция VPN) и т. п.

Имитация сервисов прикладного уровня встречается в VPN гораздо реже, чем имитация собственно транспортных функций, но также возможна. Например, поставщик в состоянии поддерживать для клиента веб-сайты, почтовую систему или специализированные приложения управления предприятием.

Другим критерием, используемым при сравнении VPN, является *степень приближенности сервисов, предлагаемых VPN, к свойствам сервисов частной сети.*

Во-первых, важнейшим свойством сервисов частной сети является *безопасность.* Безопасность VPN подразумевает весь набор атрибутов защищенной сети — конфиденциальность, целостность и доступность информации при передаче через общедоступную сеть, а также защищенность внутренних ресурсов сетей потребителя и поставщика от внешних атак. Степень безопасности VPN варьируется в широких пределах, в зависимости от применяемых средств защиты — шифрования трафика, аутентификации пользователей и устройств, изоляции адресных пространств (например, на основе техники NAT), использования виртуальных каналов и двухточечных туннелей, затрудняющих подключение к ним несанк-

ционированных пользователей. Так как ни один способ защиты не дает абсолютных гарантий, то средства безопасности могут комбинироваться для создания эшелонированной обороны.

Во-вторых, желательно, чтобы сервисы VPN приближались к сервисам частной сети *по качеству обслуживания*. Качество транспортного обслуживания подразумевает, в первую очередь, гарантии пропускной способности для трафика клиента, к которым могут добавляться и другие параметры QoS — максимальные задержки и процент потерянных данных. В пакетных сетях пульсации трафика, переменные задержки и потери пакетов — неизбежное зло, поэтому степень приближения виртуальных каналов к каналам TDM всегда неполная и вероятностная (в среднем, но никаких гарантий для отдельно взятого пакета). Разные пакетные технологии отличаются различным уровнем поддержки параметров QoS. В ATM, например, механизмы качества обслуживания наиболее совершенны и отработаны, а в IP-сетях они только начинают внедряться. Поэтому далеко не каждая сеть VPN пытается воссоздать эти особенности частной сети. Считается, что безопасность — обязательное свойство VPN, а качество транспортного обслуживания — только желательное.

В-третьих, сеть VPN приближается к реальной частной сети, если она обеспечивает для клиента *независимость адресного пространства*. Это дает клиенту одновременно и удобство конфигурирования, и способ поддержания безопасности. Причем желательно, чтобы не только клиенты ничего не знали об адресных пространствах друг друга, но и магистраль поставщика имела собственное адресное пространство, неизвестное пользователям. В этом случае сеть поставщика услуг будет надежнее защищена от умышленных атак или неумышленных действий своих клиентов, а значит, более высоким будет качество предоставляемых услуг VPN.

Существенное влияние на свойства виртуальных частных сетей оказывают технологии, с помощью которых эти сети строятся. Все технологии VPN можно разделить на два класса в зависимости от того, каким образом они обеспечивают безопасность передачи данных:

- технологии разграничения трафика;
- технологии шифрования.

Сети VPN на основе разграничения трафика

В технологиях разграничения трафика используется техника постоянных виртуальных каналов, обеспечивающая надежную защиту трафика каждого клиента от намеренного или ненамеренного доступа к нему других клиентов публичной сети. К этому типу технологий относятся:

- ATM VPN;
- Frame Relay VPN;
- MPLS VPN.

Двухточечные виртуальные каналы этих технологий имитируют сервис выделенных каналов, проходя от пограничного устройства (Client Edge, CE) одного сайта клиента через поставщика к CE другого сайта клиента.

ВНИМАНИЕ

Под термином «сайт» здесь понимается территориально обособленный фрагмент сети клиента. Например, о корпоративной сети, в которой сеть центрального отделения связывается с тремя удаленными филиалами, можно сказать, что она состоит из четырех сайтов.

Защита данных достигается благодаря тому, что несанкционированный пользователь не может подключиться к постоянному виртуальному каналу, не изменив таблицы коммутации устройств поставщика услуг, а значит, ему не удастся провести атаку или прочитать данные. Свойство защищенности трафика является *естественным свойством* техники виртуальных каналов, поэтому сервисы ATM VPN и Frame Relay VPN являются на самом деле не чем иным, как обычными сервисами PVC сетей ATM или Frame Relay. Любой пользователь ATM или Frame Relay, использующий инфраструктуру PVC для связи своих локальных сетей, пользуется услугой VPN даже в том случае, когда он это явно не осознает. Это одно из «родовых» преимуществ техники виртуальных каналов по сравнению с дейтаграммной техникой, так как при применении последней без дополнительных средств VPN пользователь оказывается не защищенным от атак любого другого пользователя сети.

Так как в технологиях ATM и Frame Relay при передаче данных используется только два уровня стека протоколов, варианты VPN, построенные на их основе, называют также **сетями VPN уровня 2** (Layer 2 VPN, L2VPN). Наличие в технологиях ATM и Frame Relay механизмов поддержки параметров QoS позволяет ATM VPN и Frame Relay VPN достаточно хорошо приближаться к частным сетям на выделенных каналах.

Информация третьего уровня никогда не анализируется и не меняется в этих сетях — это одновременно и достоинство, и недостаток. Преимущество в том, что клиент может передавать по такому виртуальному каналу трафик любых протоколов, а не только IP. Кроме того, IP-адреса клиентов и поставщика услуг изолированы и независимы друг от друга — они могут выбираться произвольным образом, так как не используются при передаче трафика через магистраль поставщика. Никаких других знаний о сети поставщика услуг, помимо значений меток виртуальных каналов, клиенту не требуется. Недостаток этого подхода состоит в том, что поставщик не оперирует IP-трафиком клиента и, следовательно, не может оказывать дополнительные услуги, связанные с сервисами IP, а это сегодня очень перспективное направление бизнеса поставщиков услуг.

Главным недостатком сети L2VPN является ее достаточно высокая сложность и стоимость. При организации полносвязной топологии сайтов клиента зависимость операций конфигурирования от числа сайтов имеет квадратичный характер (рис. 24.15, *а*).

Действительно, для соединения N сайтов необходимо создать $N \times (N - 1)/2$ двунаправленных виртуальных каналов (или $N \times (N - 1)$ однонаправленных). В частности при значении N, равном 100, потребуется 5000 операций конфигурирования. И хотя они и выполняются с помощью автоматизированных систем администрирования, ручной труд и вероятность ошибки все равно сохраняются. При поддержке только услуг *интранет* общее количество конфигурируемых соединений прямо пропорционально количеству клиентов — и это хорошо!

Но оказание услуг *экстранет* ухудшает ситуацию, так как необходимо обеспечить связь сайтов разных клиентов. Масштабируемость сети ATM/FR VPN можно улучшить, если клиент откажется от полносвязной топологии и организует связи типа «звезда» через один или несколько выделенных транзитных сайтов (рис. 24.15, *б*). Конечно, производительность сети клиента при этом снизится, так как увеличится число транзитных передач информации. Однако экономия средств будет налицо — поставщики услуг взимают деньги за свои виртуальные каналы, как правило, «поштучно».

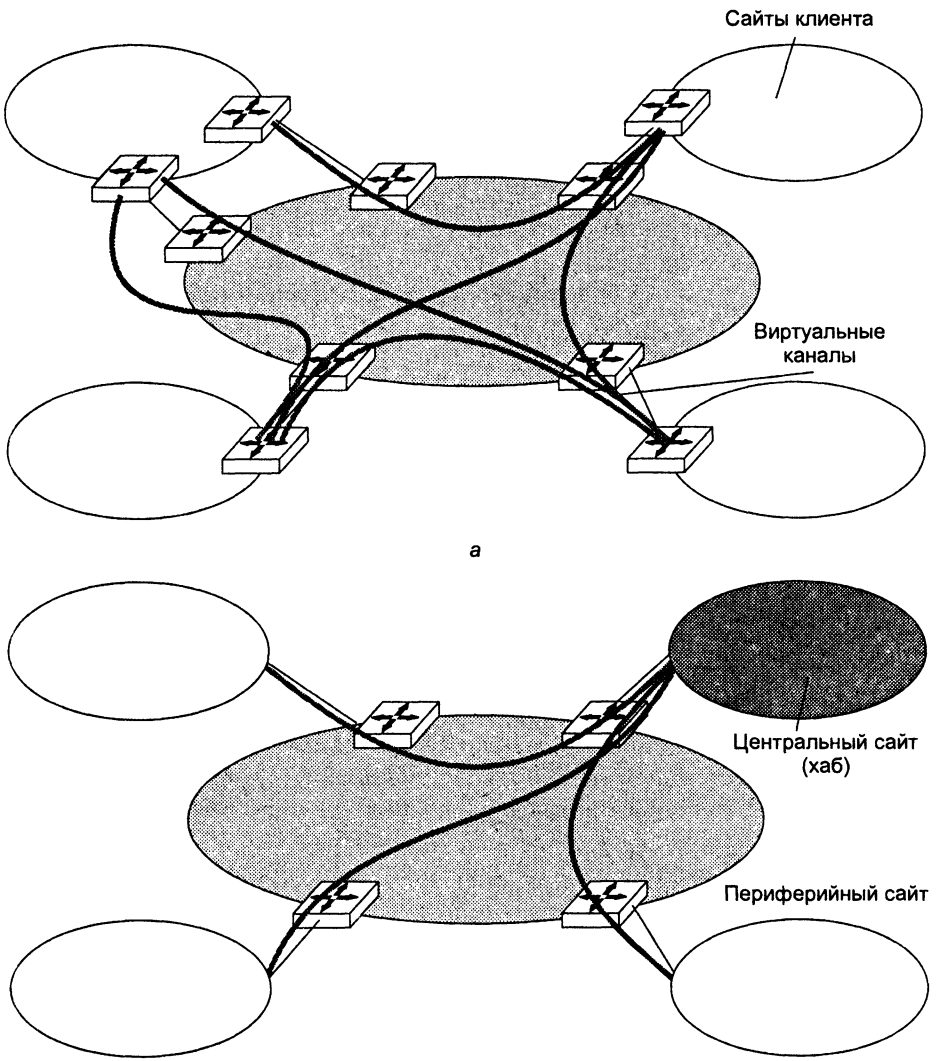


Рис. 24.15. Масштабируемость сети L2VPN: *а* — полносвязная топология; *б* — топология типа «звезда»

Клиенты сети ATM/FR VPN не могут нанести ущерб друг другу, а также атаковать IP-сеть поставщика. Сегодня поставщик услуг располагает IP-сетью, даже если он оказывает только услуги ATM/FR VPN, что очень маловероятно. Без IP-сети и ее сервисов администрирования он просто не сможет управлять своей сетью ATM/FR. IP-сеть является оверлейной (наложенной) по отношению к сетям ATM или FR, поэтому клиенты ATM/FR ничего не знают о ее структуре и о ее наличии (рис. 24.16).

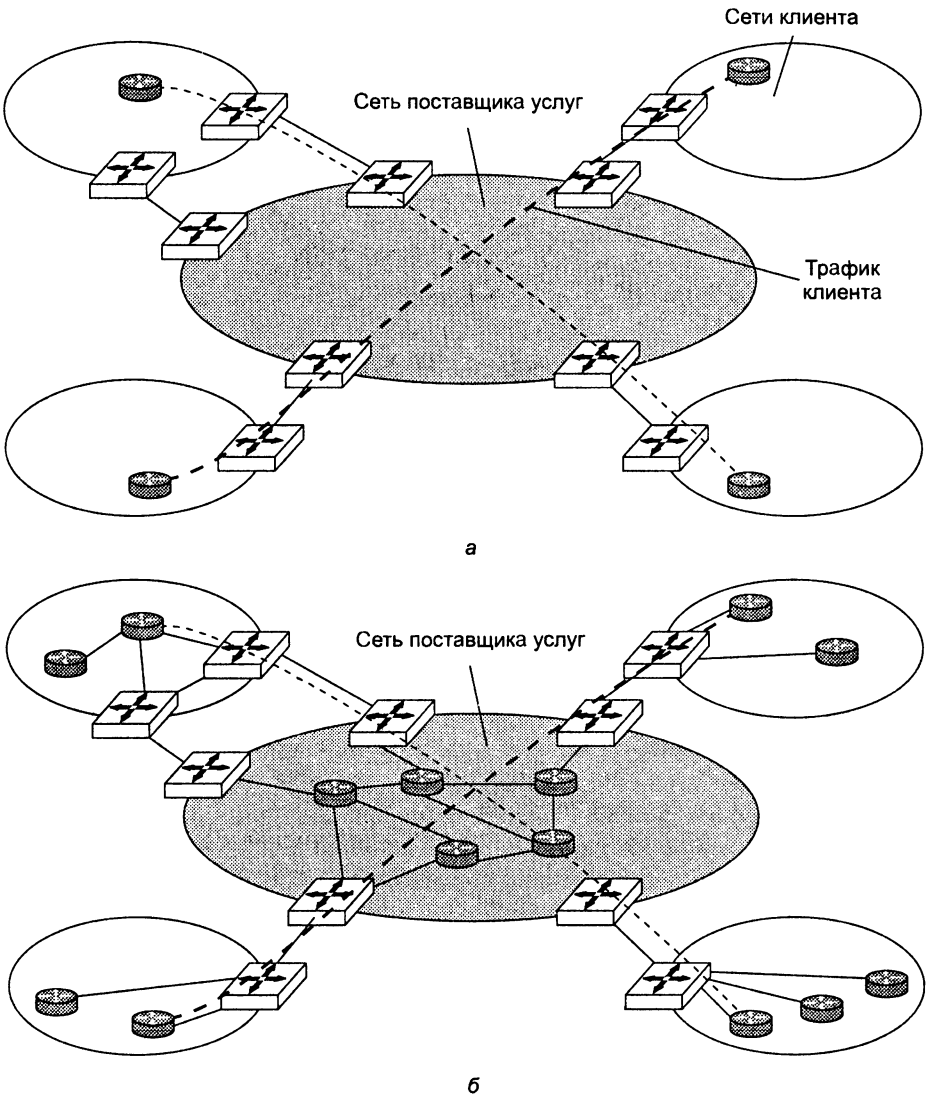


Рис. 24.16. Оверлейная и одноранговая модели VPN: *а* — в оверлейной модели внутренняя структура сети поставщика услуг скрыта от клиентов; *б* — в одноранговой модели сети поставщика и клиентов «знают» друг о друге

Сети MPLS VPN могут строиться как по схеме L2VPN, так и по другой схеме, использующей протоколы трех уровней. Такие сети называют **сетями VPN уровня 3** (Layer 3 VPN, L3VPN). В технологии L3VPN также применяется техника LSP для разграничения трафика клиентов внутри сети поставщика услуг, поддерживающей технологию MPLS.

Сеть L3VPN взаимодействует с сетями клиентов на основе IP-адресов, а L2VPN — на основе адресной информации второго уровня, например MAC-адресов или идентификаторов виртуальных каналов Frame Relay.

Использование сети MPLS VPN для поставщика услуг связано с существенным сокращением объема работ по конфигурированию VPN. Дело в том, что в технологии MPLS VPN объем работ пропорционален количеству сайтов клиента, а не квадрату этого количества, как в случае ATM/FR VPN. Другим преимуществом MPLS VPN является тесная интеграция VPN с другими приложениями MPLS, такими как инжиниринг трафика и поддержание параметров QoS. Ввиду особого интереса в сетевом мире к технологии MPLS VPN мы еще вернемся в этой главе к ее более подробному рассмотрению.

Сети VPN на основе шифрования

Еще одним классом технологий VPN являются сети **VPN на основе шифрования**. Они применяются в тех случаях, когда VPN строится в дейтаграммной сети, которая не может обеспечить разграничения трафика. Именно такой сетью является классическая IP-сеть.

Сегодня базовой технологией VPN на основе шифрования является технология IPSec, с помощью которой создается инфраструктура защищенных каналов, соединяющих сайты одного предприятия или нескольких предприятий-партнеров.

Стандарты IPSec обеспечивают высокую степень гибкости, позволяя выбирать нужный режим защиты (с шифрованием или только с обеспечением аутентичности и целостности данных), а также использовать различные алгоритмы аутентификации и шифрования. Режим инкапсуляции IPSec позволяет изолировать адресные пространства получателя (клиента) и поставщика услуг за счет применения двух IP-адресов — внешнего и внутреннего.

Чаще всего защищенные каналы нужны для сетей CPVPN, в которых клиент самостоятельно создает туннели IPSec через IP-сеть поставщика услуг. Причем от последнего требуется только предоставление стандартного сервиса по объединению сетей, а значит, доступны как услуги сети поставщика, так и услуги Интернета. Сложность конфигурирования сетей IPSec VPN высокая, поскольку туннели IPSec двухточечные, то есть при полносвязной топологии их количество пропорционально $N \times (N - 1)$. Необходимо учесть еще и непростую задачу поддержания инфраструктуры ключей.

Протокол IPSec может применяться и для создания сетей PPVPN — туннели в них также строятся на базе устройств клиента (CE-based), но эти устройства удаленно конфигурируются и администрируются поставщиком услуг.

Из всех свойств частной сети виртуальные частные сети на основе шифрования имитируют только ее защищенность и изолированность адресного пространства.

Пропускная способность каналов и другие параметры QoS этой технологией не поддерживаются, но если оператор предоставляет определенные параметры QoS (например, за счет дифференцированного обслуживания), это можно использовать при создании туннеля IPSec.

Технологии VPN на основе шифрования можно применять *совместно* с технологиями VPN на основе разделения трафика для повышения уровня защищенности виртуальных частных сетей. Технологии VPN на основе разделения трафика иногда критикуют за недостаточный уровень безопасности, считая, что отсутствие шифрования трафика позволяет персоналу поставщика услуг получить несанкционированный доступ к данным. Действительно, такая вероятность существует, поэтому клиент услуг VPN на основе разделения трафика, например MPLS VPN, может самостоятельно повысить защищенность своего трафика, применив, скажем, технику IPSec.

Технология MPLS VPN

Ключевые слова: сети MPLS L2VPN и MPLS L3VPN, пограничное устройство поставщика услуг, маршрутизатор магистральной сети поставщика, глобальная таблица маршрутизации, таблица маршрутизации и продвижения сети VPN, виртуальный маршрутизатор, адреса VPN-IPv4, различитель маршрутов, адрес сети назначения, адрес следующего маршрутизатора, метка виртуальной частной сети, расширенные атрибуты сообщества, маршрутная цель.

Сети MPLS VPN привлекают сегодня всеобщее внимание. Количество ведущих поставщиков услуг, предлагающих своим клиентам воспользоваться новым видом сервиса для экономичного построения своих внутренних и внешних сетей, постоянно растет, делая сети MPLS VPN доступными для пользователей все большего числа стран и регионов. От других технологий построения виртуальных частных сетей, таких как VPN на базе ATM/FR или IPSec, технологию MPLS VPN выгодно отличает хорошая масштабируемость, возможность автоматического конфигурирования и естественная интеграция с другими сервисами протокола IP, которые сегодня входят в обязательный набор любого успешного поставщика услуг, включая доступ в Интернет, Web и почтовые службы, хостинг.

Существует два варианта сетей MPLS VPN.

- В сетях **MPLS L3VPN** доставка трафика от клиента до пограничного устройства сети поставщика услуг осуществляется с помощью технологии IP (третий уровень).
- Сети **MPLS L2VPN** передают клиентский трафик в сеть поставщика услуг с помощью какой-либо технологии второго уровня, которой может быть Ethernet, Frame Relay или ATM.

В обоих случаях внутри сети поставщика услуг клиентский трафик передается с помощью технологии MPLS[1].

В данной книге рассматривается только MPLS L3VPN, как намного более зрелая технология, уже работающая во многих сетях поставщиков услуг. Несмотря на то что спецификация RFC 2547bis, которая определяет основные механизмы этой технологии, носит информационный статус, все реализации MPLS L3VPN производителями сетевого оборудования следуют этому документу, придавая ему статус фактического стандарта. В дальнейшем изложении мы для краткости будем опускать обозначение уровня L3, используя название MPLS VPN как синоним MPLS L3VPN.

Полная связность при абсолютной изолированности

Каждый клиент желает, чтобы поставщик услуг VPN связал между собой его сети, обеспечив абсолютную изолированность полученной единой сети от сетей других клиентов.

Эту задачу современному поставщику услуг приходится решать в противоречивых условиях доминирования технологии IP как универсального транспорта. Действительно, один из основных принципов работы составной IP-сети заключается в автоматическом связывании всех сетей в одно целое за счет распространения по сети маршрутной информации протоколами маршрутизации, такими как BGP, OSPF, IS-IS, RIP. С помощью подобного механизма на каждом маршрутизаторе сети автоматически создается таблица маршрутизации, в которой указываются пути следования пакетов к каждой из сетей, включенных в составную сеть (пути к отдельным сетям могут агрегироваться, но это не меняет сути).

Как же технология MPLS VPN разрешает парадокс обеспечения изолированности при сохранении связности? Достаточно элегантно — за счет автоматической фильтрации маршрутных объявлений и применения туннелей MPLS для передачи клиентского трафика по внутренней сети поставщика.

Для того чтобы изолировать сети друг от друга, достаточно поставить между ними заслон на пути распространения маршрутной информации. Для обмена маршрутной информацией в пределах сети узлы пользуются одним из внутренних протоколов маршрутизации (IGP), область действия которого ограничена автономной системой: RIP, OSPF или IS-IS. Если в таблице маршрутизации узла A нет записи о маршруте к узлу B (и отсутствует запись о маршруте по умолчанию), то говорят, что узел A не «видит» узла B.

В сети MPLS VPN подобный режим достигается за счет того, что маршрутные объявления, передаваемые сетью клиента, с помощью протокола BGP «перепрыгивают» через всю внутреннюю сеть поставщика услуг. После чего благодаря

[1] Сегодня уровень MPLS определить не так просто — терминология в этой области еще не устоялась. Но поскольку продвижение пакетов на основе локальных меток соответствует второму уровню, мы будем относить MPLS ко второму уровню.

особому конфигурированию с использованием многопротокольного расширения протокола BGP (MP-BGP) они попадают только в сети того же клиента. В результате маршрутизаторы разных клиентов не имеют маршрутной информации друг о друге и поэтому не могут обмениваться пакетами, то есть достигается желаемая изоляция (рис. 24.17).

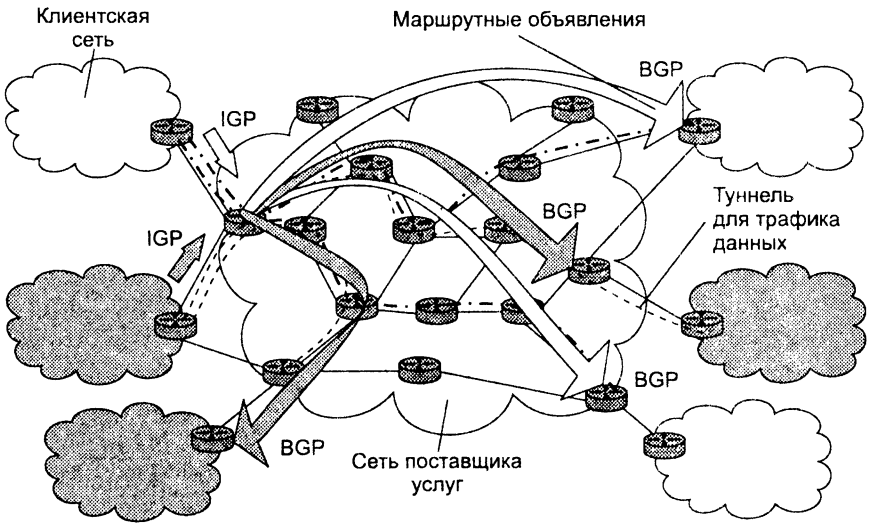


Рис. 24.17. Изоляция клиентских сетей с помощью туннелей

Еще одним следствием такого подхода является изолированность внутренней сети поставщика услуг от сетей клиентов — а это, в свою очередь, повышает надежность работы сети поставщика и ее масштабируемость (не нужно хранить таблицы большого размера с описанием сетей многочисленных клиентов на внутренних маршрутизаторах сети поставщика услуг).

Но как же все-таки связать территориально разнесенные сети клиента в единую виртуальную частную сеть, если внутренняя сеть поставщика услуг ничего о них не знает, во всяком случае, на уровне обычных таблиц маршрутизации? Для этого применяется достаточно традиционное средство — туннель между пограничными маршрутизаторами внутренней сети. Особенность рассматриваемой технологии состоит в применение туннеля MPLS (альтернативные решения могли бы основываться на туннелях IPSec или других туннелях класса «IP поверх IP»). Преимуществом туннелей MPLS VPN являются автоматический способ их прокладки и выгоды, получаемые за счет применения технологии MPLS как таковой — ускоренное продвижение пакетов по сети поставщика услуг и управление качеством обслуживания (QoS) для туннелей с инжинирингом трафика.

Для того чтобы описанные принципы построения MPLS VPN смогли найти воплощение в реальной сети, были разработаны несколько специфических механизмов и компонентов сети.

Компоненты сети MPLS VPN

В сети MPLS VPN легко выделить две области (рис. 24.18):

- IP-сети клиентов;
- магистральная сеть MPLS поставщика услуг.

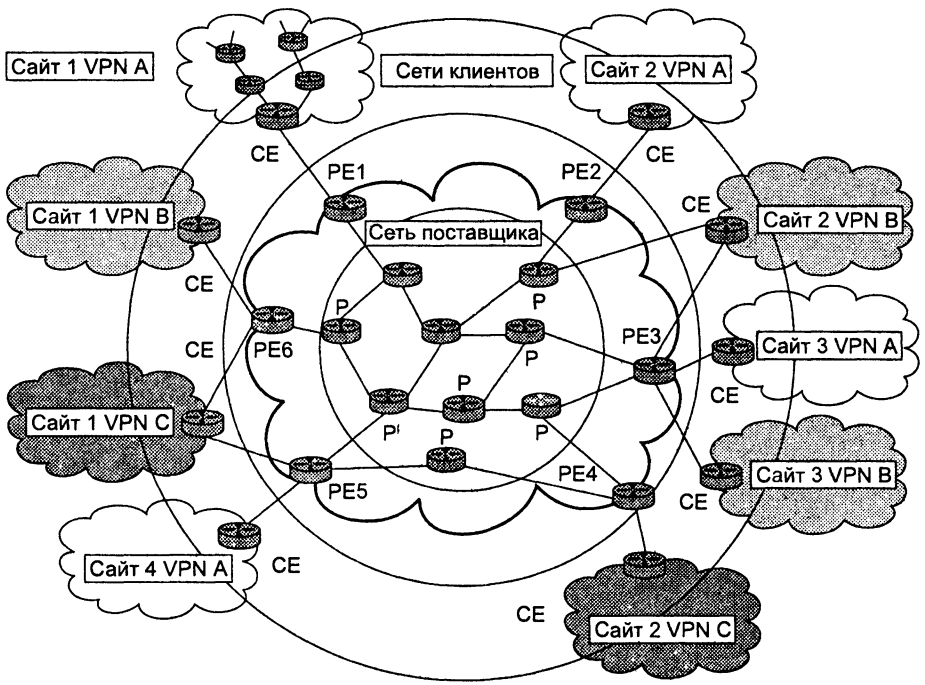


Рис. 24.18. Компоненты сети MPLS VPN

В общем случае у каждого клиента может быть несколько территориально обособленных IP-сетей (сайтов), каждая из которых, в свою очередь, может включать несколько подсетей, связанных маршрутизаторами. Принадлежащие одному клиенту сайты обмениваются IP-пакетами через сеть поставщика услуг и образуют виртуальную частную сеть этого клиента.

Маршрутизатор, с помощью которого сайт клиента подключается к магистрали поставщика, называется пограничным устройством клиента CE (Customer Edge router). Будучи компонентом сети клиента, маршрутизатор CE ничего не знает о существовании VPN. Он может быть соединен с магистральной сетью поставщика услуг несколькими каналами.

Магистральная сеть поставщика услуг является сетью MPLS, в которой IP-пакеты продвигаются на основе не IP-адресов, а локальных меток. Сеть MPLS состоит из коммутирующих по меткам маршрутизаторов (LSR), которые направляют трафик по предварительно проложенным путям коммутации по меткам (LSP) в соответствии со значениями меток.

В сети поставщика среди устройств LSR выделяют **пограничные маршрутизаторы** PE (Provider Edge router), к которым через маршрутизаторы CE подключаются сайты клиентов и **маршрутизаторы магистральной сети поставщика** P (Provider router P).

Маршрутизаторы CE и PE обычно связаны непосредственно физическим каналом, на котором работает какой-либо протокол канального уровня, например PPP, FR, ATM или Ethernet. Общение между CE и PE идет по стандартным протоколам стека TCP/IP. Поддержка MPLS нужна только для внутренних интерфейсов PE и всех интерфейсов P. Иногда полезно различать относительно направления продвижения трафика *входной и выходной* (*удаленный*) *маршрутизаторы PE.*

В магистральной сети поставщика только маршрутизаторы PE должны быть сконфигурированы для поддержки виртуальных частных сетей, только они «знают» о существующих сетях VPN.

Если рассматривать сеть с позиций VPN, то маршрутизаторы P непосредственно не взаимодействуют с маршрутизаторами CE, а просто обеспечивают туннели между входным и выходным маршрутизаторами PE.

Пограничные маршрутизаторы PE являются функционально более сложными, чем внутренние маршрутизаторы P сети поставщика услуг. На них возлагаются главные задачи по поддержке сетей VPN, а именно — задачи разграничения маршрутов и данных, поступающих от разных клиентов. Маршрутизаторы PE служат также оконечными точками путей LSP между сайтами заказчиков, и именно пограничный маршрутизатор поставщика услуг назначает метку IP-пакету для его транзита через внутреннюю сеть, образованную внутренними маршрутизаторами поставщика услуг.

Пути LSP могут быть проложены двумя способами: либо с применением технологии ускоренной маршрутизации (IGP) с помощью протокола LDP, либо на основе технологии инжиниринга трафика с помощью протоколов RSVP или CR-LDP. Прокладка LSP означает создание *таблиц коммутации по меткам* на всех пограничных и внутренних маршрутизаторах поставщика услуг, образующих данный путь (примеры таких таблиц можно найти в главе 22). В совокупности эти таблицы задают множество путей, образующих сети различных топологий для разных видов трафика клиентов.

Разграничение маршрутной информации

Для корректной работы VPN требуется, чтобы информация о маршрутах через магистральную сеть поставщика услуг не распространялась за ее пределы, а сведения о маршрутах в клиентских сайтах не становились известными за границами определенных сетей VPN.

Барьеры на пути распространения маршрутных объявлений могут устанавливаться соответствующим конфигурированием маршрутизаторов. Протокол маршрутизации должен быть оповещен о том, с каких интерфейсов и от кого он имеет право принимать объявления и на какие интерфейсы и кому их распространять.

Роль таких барьеров в сети MPLS VPN играют маршрутизаторы PE. Можно представить, что через маршрутизатор PE проходит невидимая граница между зоной клиентских сайтов и зоной ядра сети поставщика. По одну сторону располагаются интерфейсы, через которые PE взаимодействует с внутренними маршрутизаторами поставщика услуг, а по другую — интерфейсы, к которым подключаются сайты клиентов. С одной стороны, на PE поступают объявления о маршрутах магистральной сети, с другой стороны — объявления о маршрутах в сетях клиентов.

На рис. 24.19 показана схема разграничения маршрутной информации. На маршрутизаторе PE установлены несколько протокольных модулей IGP. Один из них сконфигурирован для приема и распространения маршрутных объявлений только с тех трех внутренних интерфейсов, которые связывают этот маршрутизатор PE с маршрутизаторами P. Два других модуля IGP обрабатывают маршрутную информацию от сайтов клиентов.

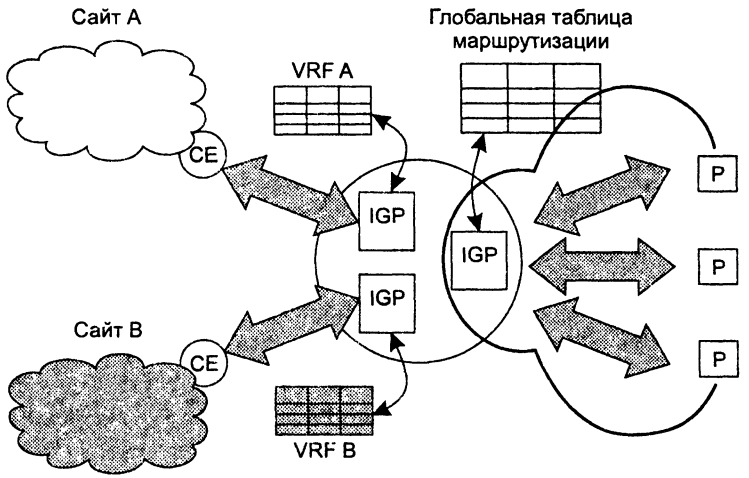


Рис. 24.19. Схема разграничения маршрутной информации

Аналогичным образом настроены и остальные устройства PE. Внутренние маршрутизаторы P принимают и обрабатывают маршрутную информацию протокола IGP, поступающую со всех интерфейсов. В результате на всех маршрутизаторах (и PE, и P) создается по таблице маршрутизации, где содержатся все маршруты в пределах *внутренней сети поставщика услуг*. Подчеркнем, что никакой информации о маршрутах к *сетям клиентов* в таблицах внутренних маршрутизаторов нет. Сети клиентов также ничего не «знают» о маршрутах в сети поставщика услуг.

На каждом из маршрутизаторов PE создается два типа таблиц маршрутизации:

- **глобальная таблица маршрутизации** создается на основе объявлений из *магистральной сети поставщика услуг*;
- **таблицы маршрутизации и продвижения сети VPN** (VPN Routing and Forwarding Instance, VRF) PE формирует на основе объявлений, поступающих *из сайтов клиентов*.

Сайты клиентов представляют собой обычные IP-сети, маршрутная информация в которых может передаваться и обрабатываться с помощью любого протокола маршрутизации класса IGP. Очевидно, что этот процесс никак не регламентируется поставщиком. Маршрутные объявления свободно распространяются между узлами в пределах каждого сайта до тех пор, пока не доходят до пограничных маршрутизаторов PE, служащих преградой для их дальнейшего распространения.

Разграничение маршрутов разных клиентов обеспечивается путем установки отдельного протокола маршрутизации на каждый интерфейс маршрутизатора PE, к которому подключен сайт клиента. Этот протокол принимает и передает клиентские маршрутные объявления только с одного определенного для него интерфейса, не пересылая их ни на внутренние интерфейсы, через которые пограничный маршрутизатор связан с внутренними маршрутизаторами, ни на интерфейсы, к которым подключены сайты других клиентов. В результате на маршрутизаторе PE создается несколько таблиц VRF.

Несколько упрощая, можно считать, что на каждом маршрутизаторе PE создается столько таблиц VRF, сколько сайтов к нему подключено. Фактически на маршрутизаторе PE организуется несколько **виртуальных маршрутизаторов**, каждый из которых работает со своей таблицей VRF.

Возможно и другое соотношение между сайтами и таблицами VRF. Например, если к некоторому пограничному маршрутизатору подключено несколько сайтов одной и той же сети VPN, то для них может быть создана одна общая таблица VRF. На рис. 24.19 показаны две таблицы VRF, одна из которых содержит описание маршрутов к узлам сайта A (VRF A), а другая — к узлам сайта B (VRF B).

Использование протокола MP-BGP для связывания сайтов

Чтобы связать территориально разнесенные сайты заказчика в единую сеть, необходимо, во-первых, создать для них общее пространство распространения маршрутной информации, во-вторых, проложить во внутренней сети пути, по которым принадлежащие разным сайтам узлы одной и той же сети VPN могли бы вести защищенный обмен данными.

Механизмом, с помощью которого сайты, принадлежащие к одной и той же сети VPN обмениваются маршрутной информацией, является уже упоминавшееся многопротокольное расширение для протокола BGP (MultiProtocol extensions for BGP, MP-BGP). Подробное описание этого протокола можно найти в спецификации RFC 2858. С его помощью пограничные маршрутизаторы организуют сеансы связи, в рамках которых обмениваются маршрутной информацией из своих таблиц VRF.

Особенность протокола BGP и его расширений заключается в том, что он получает и передает свои маршрутные объявления не всем непосредственно связанным с ним маршрутизаторам, как протоколы IGP, а только тем, которые указаны в его конфигурационных параметрах в качестве соседей. Причем соседями могут

быть «назначены» маршрутизаторы, находящиеся на расстоянии многих хопов. Маршрутизатор PE сконфигурирован так, что все получаемые от клиентских сайтов маршрутные объявления он с помощью MP-BGP пересылает только определенным в качестве соседей другим пограничным маршрутизаторам PE. Целенаправленное распространение маршрутов между маршрутизаторами PE обеспечивается надлежащим выбором атрибутов протокола MP-BGP[1].

Вопрос о том, кому отправлять маршрутные объявления, а кому нет, целиком зависит от топологии виртуальных частных сетей, поддерживаемых данным поставщиком услуг. Так, на рис. 24.18 маршрутизатор PE1 передает маршруты из таблицы VRF сайта 1, относящегося к сети VPN A, маршрутизаторам PE2, PE3, PE5, к которым подключены остальные сайты 2, 3 и 4 той же сети VPN A. Полученные маршруты заносятся в таблицы VRF соответствующих сайтов.

Итак, помимо маршрутов, поступающих от непосредственно подсоединенных к устройству PE сайтов, каждая таблица VRF дополняется маршрутами, получаемыми от других сайтов данной сети VPN по протоколу MP-BGP. Таким путем создаются таблицы, описывающие маршруты в рамках отдельной сети VPN.

Независимость адресных пространств

Если некоторое множество узлов никогда ни при каких условиях не получает маршрутную информацию от другого множества узлов, то адресация узлов в пределах каждого из этих множеств может выполняться независимо.

Ограничение области распространения маршрутной информации пределами отдельных сетей VPN изолирует адресные пространства каждой сети VPN, позволяя применять в ее пределах как публичные адреса Интернета, так и частные адреса, зарезервированные в соответствии со спецификацией RFC 1819.

Почему же в таком случае не сделать выбор адресов в пределах VPN совершенно произвольным и ограниченным только общими правилами адресации стека TCP/IP? Дело в том, что во многих случаях клиенты не хотят полной изоляции VPN: в частности, они нуждаются в выходе в Интернет. Независимое же (не согласованное с регламентирующими органами Интернета) назначение адресов узлам VPN может привести к совпадению внутренних адресов сайтов с уже использованными в Интернете публичными адресами, в результате чего связь с Интернетом станет невозможной. При применении зарезервированных частных адресов проблема связи клиентов VPN с внешним миром решается с помощью стандартной техники трансляции адресов (NAT)[2]. В любом случае должно соблюдаться требование уникальности адресов в пределах VPN.

Однако использование в разных сетях VPN одного и того же адресного пространства создает проблему для маршрутизаторов PE. Протокол BGP изначально был разработан в предположении, что все адреса, которыми он манипулирует, во-первых, относятся к семейству адресов IPv4 и, во-вторых, однозначно иден-

[1] Эти атрибуты описаны в документе «BGP Extended Communities Attribute», имеющем пока статус проекта стандарта Интернета.

[2] Подробнее об этом можно прочитать в главе 20.

тифицируют узлы сети, то есть являются глобально уникальными в пределах всей составной сети. Ориентация на глобальную уникальность адресов выражается в том, что, получив очередное маршрутное объявление, протокол BGP анализирует его, не обращая внимания на то, какой сети VPN принадлежит этот маршрут. Если на вход BGP поступают описания маршрутов к узлам разных сетей VPN, но с совпадающими адресами IPv4, то протокол BGP считает, что все они ведут к одному и тому же узлу, а, следовательно, как и предусмотрено в алгоритме его работы, он помещает в соответствующую таблицу VRF только один лучший (в соответствии с правилами выбора BGP) маршрут.

Эта проблема была решена в MPLS VPN применением вместо потенциально неоднозначных адресов IPv4 расширенных и однозначных адресов нового типа, а именно адресов **VPN-IPv4**, получаемых путем преобразования исходных адресов IPv4. Преобразование заключается в том, что ко всем адресам IPv4, составляющим адресное пространство той или иной сети VPN, добавляется префикс, называемый **различителем маршрутов** (Route Distinguisher, RD), который уникально идентифицирует эту сеть. В результате на маршрутизаторе PE все адреса, относящиеся к разным сетям VPN, обязательно будут отличаться друг от друга, даже если они имеют совпадающую часть — адрес IPv4.

Именно здесь оказалась полезной способность расширенного протокола MP-BGP переносить в маршрутных объявлениях адреса *разных типов*, в том числе IPv6, IPX, а главное, — VPN-IPv4. Адреса VPN-IPv4 используются только для маршрутов, которыми маршрутизаторы PE обмениваются по протоколу BGP. Прежде чем передать своему напарнику некоторый маршрут, входной маршрутизатор PE добавляет к его адресу назначения IPv4 префикс RD для данной VPN, тем самым преобразуя его в адрес VPN-IPv4.

Как уже было отмечено, префиксы RD должны гарантированно уникально идентифицировать VPN, чтобы избежать дублирования адресов. Упростить выбор RD, не создавая для этих целей дополнительных централизованных процедур (например, распределения RD органами Интернета подобно распределению адресов IPv4), предлагается за счет использования в качестве основы для RD заведомо уникальных чисел — либо номеров автономных систем, либо глобальных адресов интерфейсов PE со стороны магистральной сети поставщика.

RD имеет длину 8 байт и состоит из трех полей.

- Первое поле *типа* длиной 2 байт определяет тип и разрядность второго поля.
- Второе поле называется полем *администратора* и однозначно идентифицирует поставщика. Значение 0 в поле типа говорит о том, что в поле администратора указан IP-адрес интерфейса маршрутизатора PE, и длина данного поля составляет, естественно, 4 байт. Если же значение поля типа равно 1, то в качестве идентификатора поставщика выбрано значение номера его автономной системы, так что длина поля администратора составит уже 2 байт.
- Третье поле носит название поля *назначенного номера*, оно служит для обеспечения уникальности адресов VPN в пределах сети поставщика. Назначенные номера выбирает сам поставщик, это могут быть произвольные числа, главное, чтобы существовало однозначное соответствие между ними и VPN поставщика.

Рисунок 24.20 иллюстрирует сложный процесс обмена маршрутными объявлениями в сети MPLS VPN. Этот процесс включает преобразование адресов из формата IPv4 в формат VPN-IPv4, фильтрацию маршрутных объявлений (операции экспорта-импорта) и добавление к объявлениям меток VPN. Мы последовательно будем рассматривать эти вопросы, поэтому читатель должен быть готов, что не все на этом рисунке ему будет нужно и понятно с самого начала.

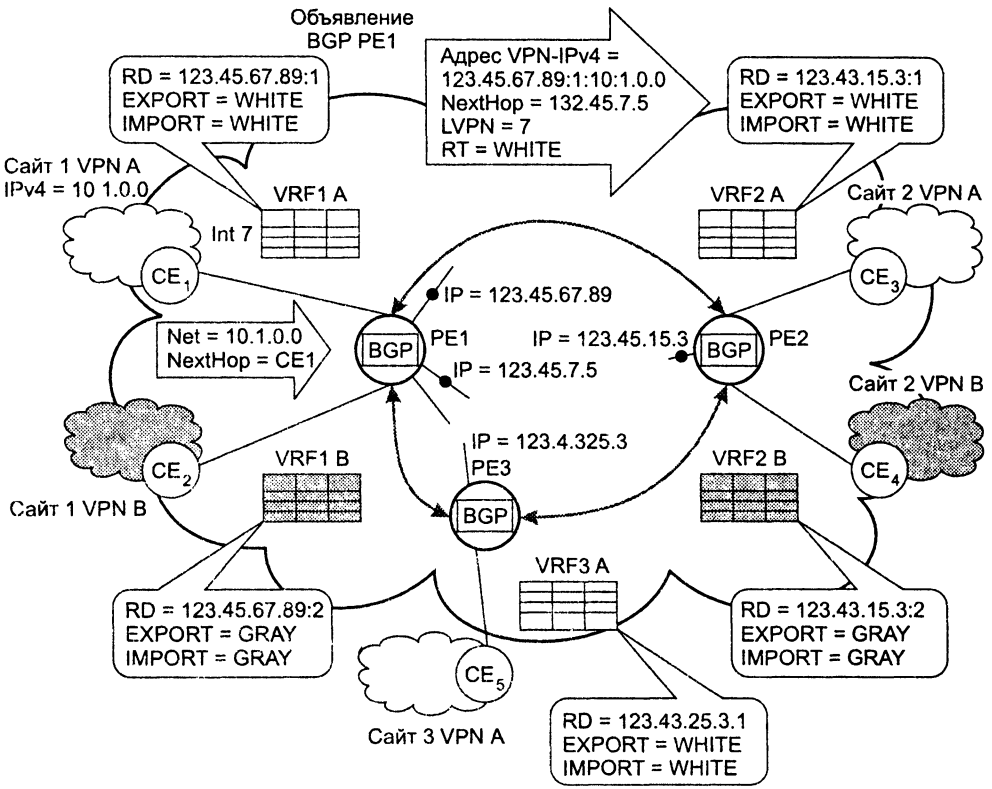


Рис. 24.20. Маршрутные объявления MP-BGP

Итак, на рис. 24.20 показан пример преобразования адресов формата IPv4 с целью обеспечения уникальности адресов в рамках всех сетей VPN одного поставщика услуг. При формировании RD для каждой из сетей VPN администратор сети сначала выбирает глобальный адрес одного из внешних интерфейсов маршрутизатора PE1 (на рисунке это адрес 123.45.67.89), затем добавляет к нему через двоеточие 1, получая значение RD, равное 123.45.67.89:1. Формат RD представлен в табл. 24.1.

Указанное значение RD администратор назначает для сети VPN A. При конфигурировании маршрутизаторов PE администратор указывает это значение для всех таблиц VRF, которые соответствуют сети VPN A. В частности, он задает это значение при создании VRF 1A, так что для протокола MP-BGP все адреса формата IPv4, которые находятся в таблице VRF1A, будут иметь RD 123.45.67:1,

в том числе все адреса с префиксом 10.1/16, которые PE1 получает от маршрутизатора CE1 сайта 1 в сети VPN A.

Таблица 24.1. Формат RD

Поле типа (2 байта)	Поле администратора (4 байта)	Поле назначенного номера (2 байта)
0	123.45.67.89	1

Аналогично, администратор выбирает для сетей VPN B значение RD равное 123.45.67.89:2, которое он указывает при конфигурировании VRF 1B на маршрутизаторе PE1. Это значение RD будет добавляться ко всем адресам IPv4, хранящимся в таблицах VRF 1B, при обработке их протоколом MP-BGP.

ПРИМЕЧАНИЕ

Все маршруты в таблицах VRF содержат адреса в формате IPv4.

Сформированные маршруты в формате VPN-IPv4 маршрутизатор PE1 передает по протоколу MP-BGP на маршрутизатор PE2, к которому подключен сайт 2 сети VPN B. Только благодаря добавлениям RD протоколы BGP, работающие на удаленных маршрутизаторах PE, различают маршруты с совпадающими адресами IPv4, относящимися к разным сетям VPN.

Документ RFC 2547bis не требует, чтобы все маршруты внутри одной сети VPN индексировались одним и тем же значением RD. Более того, один и тот же сайт, подключенный к разным интерфейсам одного маршрутизатора PE или к разным маршрутизаторам PE, может иметь различающиеся значения RD. Благодаря этому путь к одному и тому же узлу может описываться разными маршрутами, что дает возможность выбора того или иного маршрута для различных пакетов. Однако принципиально важно, чтобы значения RD разных сетей VPN не совпадали.

Генерация маршрутных объявлений MP-BGP

При получении от сайта клиента нового маршрута по протоколу класса IGP (RIP, OSPF или IS-IS) маршрутизатор PE заносит его в соответствующую таблицу VRF и распространяет дальше между другими сайтами данной сети VPN. Обмен маршрутной информацией между сайтами каждой отдельной сети VPN выполняется под управлением протокола MP-BGP. Маршрутное объявление MP-BGP имеет следующий, расширенный по сравнению с протоколом BGP набор атрибутов.

- **Адрес сети назначения в формате VPN-IPv4.**
- **Адрес следующего маршрутизатора** (NextHop). Протокол BGP указывает в данном случае адрес одного из внутренних (идущих к маршрутизаторам P) интерфейсов того маршрутизатора PE, на котором он работает.

- **Метка виртуальной частной сети** (VPN Label, LVPN) уникально определяет внешний интерфейс маршрутизатора PE и подключенный к нему сайт клиента, куда ведет объявляемый маршрут. Она назначается маршруту входным маршрутизатором PE при получении им локального маршрута от присоединенного маршрутизатора CE.
- **Расширенные атрибуты сообщества** (extended community attributes), один из которых — **маршрутная цель** (Route Target, RT) — является обязательным. Этот атрибут идентифицирует набор сайтов (VRF), входящих в данную сеть VPN, которым маршрутизатор PE должен посылать маршруты. Значение атрибута RT в объявлении о маршруте определяется *политикой экспорта* маршрутных объявлений[1], которая задается администратором при конфигурировании таблицы VRF, содержащей данный маршрут.

Пусть, например, маршрутизатор PE1 получает с сайта 1 сети VPN A по протоколу класса IGP следующее объявление о маршруте в формате IPv4 (см. рис. 24.20):

```
Net = 10.1.0.0
NextHop = CE1
```

На основании этого объявления в таблицу VRF 1A заносится соответствующая запись. Протокол BGP периодически просматривает таблицу VRF 1A и, обнаружив новую запись, генерирует объявление о маршруте, для чего выполняет следующие действия.

1. Добавляет к адресу сети назначения префикс RD (в данном случае он равен 123.45.67.89:1).
2. Переписывает значение поля NextHop, заменяя адрес внешнего интерфейса CE1 адресом внешнего интерфейса PE1, через который пролегает путь к адресу назначения (пусть в данном случае это будет 123.45.7.5).
3. Назначает метку LVPN, указывающую на таблицу VRF1A и интерфейс маршрутизатора PE1, к которому подключен сайт клиента, содержащий узел назначения (в данном случае значение метки равно 7, на рис. 24.20 этот интерфейс обозначен Int 7).
4. Задает атрибут RT (на рис. 24.20 значение атрибута RT условно обозначено как WHITE, что идентифицирует набор всех сайтов, входящих в сеть VPN A).

В результате получается такое маршрутное объявление:

```
VPN-IPv4: 123.45.67.89:1:10.1.0.0
Nexthop = 123.45.7.5
LVPN = 7
RT = WHITE
```

Это объявление протокол MP-BGP посылает всем своим соседям (на рисунке объявление помещено внутрь широкой стрелки).

Когда выходной маршрутизатор PE получает маршрут к сети VPN-IPv4, он делает обратное преобразование, отбрасывая префикс RD, и только потом помещает

[1] Политику экспорта и импорта маршрутных объявлений мы будем обсуждать в разделе «Механизм формирования топологии VPN».

маршрут в таблицу VRF2A и объявляет о нем связанному с ним маршрутизатору заказчика CE3 из данной сети VPN A. В результате в таблице VRF2A появляется новая запись:

```
Net = 10.1/16
Nexthop = 123.45.7.5 (BGP)
LVPN = 7
```

Перемещение пакета по сети MPLS VPN

Теперь, когда мы обсудили схему распространения маршрутной информации по сети MPLS VPN, давайте посмотрим, как перемещаются *данные* между узлами одной сети VPN.

Пусть, например, с сайта 2 сети VPN A узел с адресом 10.2.1.1/16 отправляет пакет узлу сайта 1 этой же сети VPN, имеющему адрес 10.1.0.3/16 (рис. 24.21). Стандартными транспортными средствами IP-пакет доставляется на пограничный маршрутизатор сайта CE3, в таблице которого для номера сети 10.1.0.0 в качестве следующего маршрутизатора указан маршрутизатор PE2. На маршрутизатор PE2 пакет поступает с интерфейса 2, поэтому для дальнейшего продвижения пакета он обращается к таблице VRF2A, связанной с данным интерфейсом.

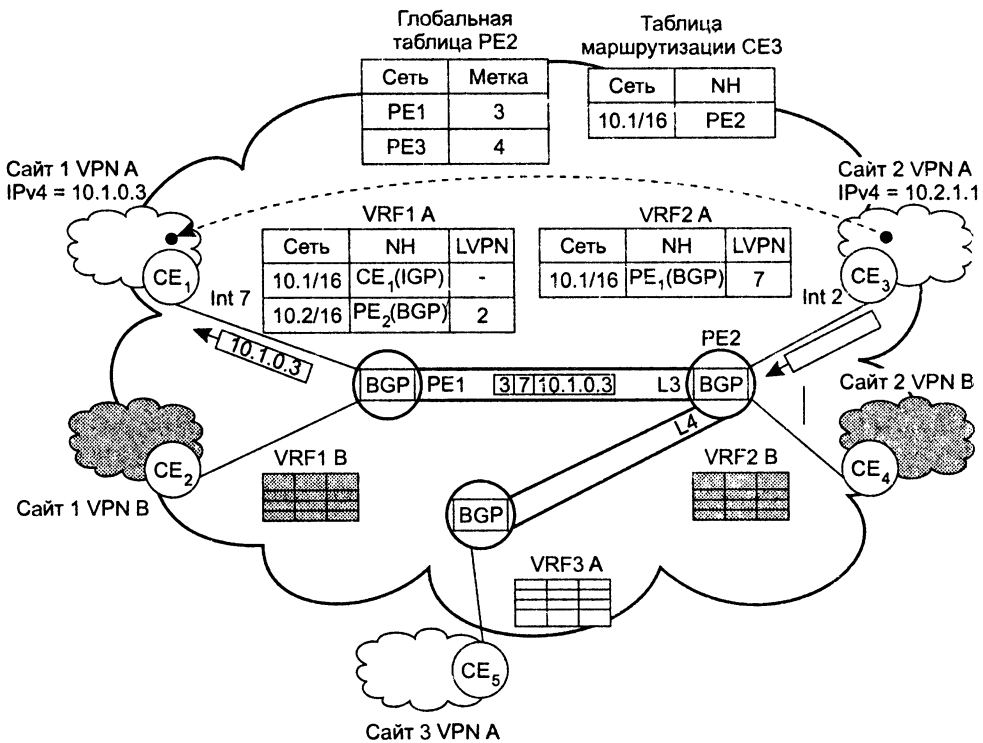


Рис. 24.21. Путешествие пакета по сети MPLS VPN

В таблице VRF2A адресу 10.1.0.0 соответствует запись протокола BGP, которая указывает, что следующим маршрутизатором (next hop) для пакета определен маршрутизатор PE1. Поле метки содержит значение LVPN = 7, определяющее интерфейс выходного маршрутизатора PE1. Это значение должно быть присвоено пакету для того, чтобы он попал в нужную сеть VPN. Здесь также указывается, что запись была сделана протоколом BGP, а не IGP. На этом основании маршрутизатор PE2 «понимает», что очередной маршрутизатор не является непосредственным соседом и путь к нему надо искать в глобальной таблице маршрутизации.

В глобальной таблице для адреса PE1 указывается начальное значение метки пути LSP, равное 3. Мы не будем останавливаться на способе прокладки пути между маршрутизаторами PE1 и PE2 — этот вопрос мы обсуждали при изучении технологии MPLS в главе 22.

В сетях MPLS VPN используются иерархические свойства путей MPLS, за счет чего пакет может быть снабжен несколькими метками, помещаемыми в стек. На входе во внутреннюю сеть поставщика, образуемую маршрутизаторами P, пакет будет снабжен двумя метками LVPN = 7 и L = 3. Метка LVPN интерпретируется как метка нижнего уровня — оставаясь на дне стека, она не используется, пока пакет путешествует по туннелю PE1-PE2. Продвижение пакета происходит на основании метки верхнего уровня L. Каждый раз, когда пакет проходит очередной маршрутизатор P вдоль туннеля, метка L анализируется и заменяется новым значением. И только после достижения конечной точки туннеля — маршрутизатора PE1 — из стека извлекается метка LVPN. В зависимости от ее значения пакет направляется на тот или иной выходной интерфейс маршрутизатора PE1 (на рис. 24.20 этот интерфейс обозначен Int 7).

Из таблицы VRF1A, связанной с данным интерфейсом и содержащей маршруты VPNA, извлекается запись о маршруте к узлу назначения, указывающая на CE1 в качестве следующего маршрутизатора. Заметим, что запись об этом маршруте была помещена в таблицу VRF1A протоколом IGP. Последний отрезок путешествия пакета от CE1 до узла 10.1.0.3 осуществляется традиционными средствами IP.

Механизм формирования топологии VPN

Политика экспорта/импорта маршрутов — мощный инструмент создания сетей VPN разных топологий.

При конфигурировании каждой таблицы VRF задаются два атрибута RT: один для определения политики экспорта, а другой для определения политики импорта маршрутов.

Маршрутные объявления MP-BGP всегда несут атрибут RT, говорящий об экспорте маршрута. Сравнение значений атрибутов RT в маршрутном объявлении и в параметрах VRF позволяет решить вопрос о принятии или отклонении предлагаемого маршрута. А это и означает формирование топологии сети. Рассмотрим этот механизм на примере.

Пусть изображенный на рис. 24.20 маршрутизатор PE2 получил объявление от PE1. Прежде чем сохранить информацию о маршруте, он проверяет значение

атрибута RT, содержащееся в объявлении, на совпадение с политикой импорта всех своих таблиц VRF, в данном случае VRF2A и VRF2B. Значение атрибута RT равно WHITE, поэтому маршрут добавляется (после преобразования в формат IPv4 путем удаления префикса RD) только в таблицу VRF2A, так как для нее определена политика импорта WHITE. Таблица VRF2B остается в неизменном виде, так как ее политика импорта говорит о том, что в нее должны помещаться только маршруты с атрибутом RT, равным GRAY.

Задание одного и того же значения для политики экспорта и импорта для всех таблиц VRF определенной сети VPN (именно этот случай для сети VPN A показан на рис. 24.20) приводит к полносвязной топологии — каждый сайт может посылать пакеты непосредственно тому сайту, в котором находится сеть назначения.

Существуют и другие варианты топологии VPN. Например, за счет конфигурирования политики экспорта/импорта можно реализовать такую популярную топологию, как «звезда», когда все сайты (spoke) общаются друг с другом через выделенный центральный сайт (hub).

Для достижения этого эффекта достаточно определить для VRF центрального сайта политику импорта как Import = spoke, а экспорта как Export = hub, а на таблицах VFR периферийных сайтов поступить наоборот, задав Import = hub, а Export = spoke (рис. 24.22).

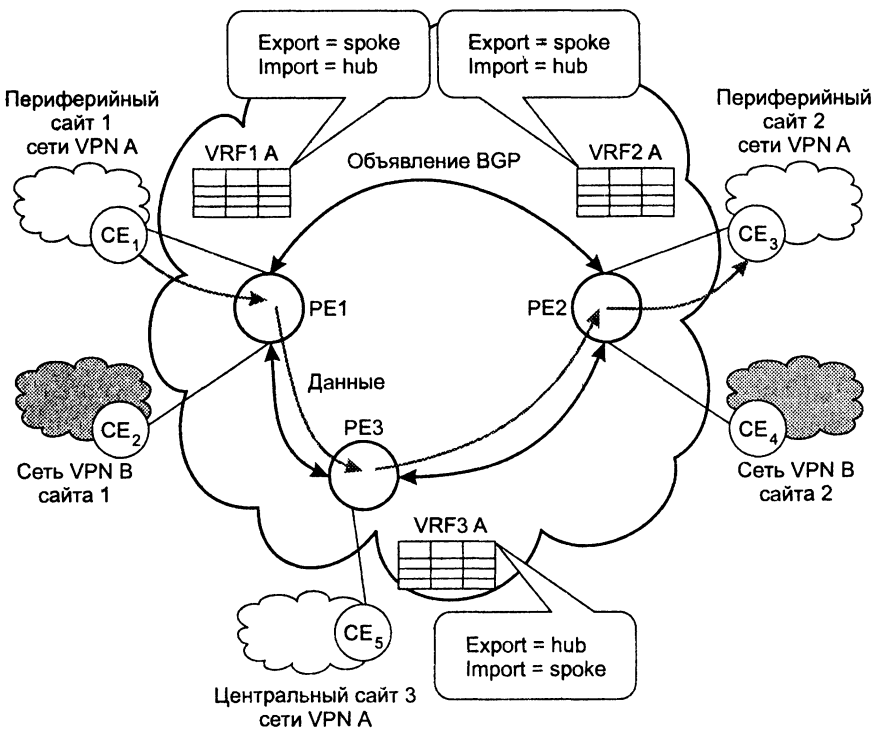


Рис. 24.22. Конфигурирование топологии звезда для сети VPN A

В результате таблицы VRF периферийных сайтов не будут принимать маршрутные объявления друг от друга, поскольку они передаются по сети протоколом MP-BGP с атрибутом RT = spoke, между тем как их политика импорта разрешает получать объявления с атрибутом RT = hub. Зато объявления VRF периферийных сайтов принимает таблица VRF центрального сайта, для которого как раз и определена политика импорта spoke. Этот сайт обобщает все объявления периферийных сайтов и отсылает их обратно, но уже с атрибутом RT = hub, что совпадает с политикой импорта периферийного сайта. Таким образом, в таблицах VRF каждого периферийного сайта появляются записи о сетях в других периферийных сайтах. А в качестве следующего транзитного узла будет указан адрес интерфейса PE, связанного с центральным сайтом, поскольку объявление пришло от него. Поэтому пакеты между периферийными сайтами будут проходить транзитом через пограничный маршрутизатор PE3, подключенный к центральному сайту.

Из описания механизмов MPLS VPN можно сделать вывод, что процесс конфигурирования новой или модификации существующей сети VPN достаточно сложен, но он хорошо формализуется и автоматизируется. Для исключения возможных ошибок конфигурирования, например приписывания сайту ошибочной политики импорта/экспорта маршрутных объявлений, что может привести к присоединению сайта к чужой сети VPN, некоторые производители разработали автоматизированные программные системы конфигурирования MPLS.

Степень защищенности

Повысить степень защищенности MPLS VPN можно с помощью традиционных средств, например, применяя средства аутентификации и шифрования протокола IPSec, устанавливаемые в сетях клиентов или в сети поставщика. Услуга MPLS VPN может легко интегрироваться с другими услугами IP, например с предоставлением доступа к Интернету пользователям VPN с защитой их сети средствами межсетевого экрана, установленного в сети поставщика. Поставщик также может предоставлять пользователям MPLS VPN услуги, базирующиеся на других возможностях MPLS, в частности гарантированное качество обслуживания на основе методов инжиниринга трафика MPLS. Что же касается сложностей ведения в маршрутизаторах поставщика услуг таблиц маршрутизации пользователей, на которые указывают некоторые специалисты, то они, на наш взгляд, несколько преувеличены, так как таблицы создаются автоматически с помощью стандартных протоколов маршрутизации и только на пограничных маршрутизаторах (PE). Механизм виртуального маршрутизатора полностью изолирует эти таблицы от глобальных таблиц маршрутизации поставщика услуг, что обеспечивает необходимые уровни надежности и масштабируемости решений MPLS VPN. Впрочем, реальное качество данной технологии покажет время и, скорее всего, достаточно скоро.

Технология MPLS VPN не обеспечивает безопасности за счет шифрования и аутентификации, как это делают технологии IPSec и PPTP, но допускает применение данных технологий как дополнительных мер защиты в случае необходимости.

Перед MPLS VPN не ставится задача поддержки качества обслуживания, но в случае необходимости поставщик может использовать методы дифференцированного обслуживания и инжиниринга трафика.

Выводы

Сервисы защиты передаваемой информации позволяют клиентам через публичную сеть, например Интернет, безопасно передавать информацию, обеспечивая ее аутентичность, целостность и конфиденциальность.

Сервисы защиты передаваемой информации можно построить с помощью системных средств, реализованных на разных уровнях модели OSI. К числу наиболее популярных сервисов подобного рода можно отнести протоколы защищенного канала SSL, IPSec, PPTP, а также сервисы VPN.

IPSec — это согласованный набор открытых стандартов, имеющий сегодня вполне очерченное ядро, включающее три протокола: протокол AH гарантирует целостность и аутентичность данных, протокол ESP шифрует передаваемые данные, обеспечивая их конфиденциальность, а также аутентификацию и целостность, протокол IKE устанавливает логическое соединение и распределяет секретные ключи.

При установлении однонаправленного логического соединения, называемого в IPSec безопасной ассоциацией, стороны принимают ряд соглашений, регламентирующих процесс передачи потока данных между ними: тип и режим работы протокола защиты (AH или ESP), методы шифрования, секретные ключи и др. Наборы текущих параметров, определяющих все активные ассоциации, хранятся на обоих оконечных узлах защищенного канала в виде баз данных SAD Каждая сторона поддерживает также другой тип базы данных — SPD, которая задает соответствие между признаками IP-пакетов и установленными для них правилами обработки.

Использование баз SPD и SAD для защиты трафика позволяет достаточно гибко сочетать механизм логических соединений с дейтаграммным характером трафика протокола IP.

Технология VPN позволяет с помощью разделяемой несколькими предприятиями сетевой инфраструктуры реализовать сервисы, приближающиеся к сервисам частной сети по качеству (безопасность, доступность, предсказуемая пропускная способность, независимость в выборе адресов).

Сеть VPN может быть реализована как самим предприятием (CPVPN), так и поставщиком услуг (PPVPN). Она может строиться на базе оборудования, установленного на территории как потребителя, так и поставщика услуг.

Технологии VPN можно разделить на два класса в зависимости от того, каким образом они обеспечивают безопасность передачи данных: технологии разграничения трафика (ATM VPN, Frame Relay VPN, MPLS VPN) и технологии на основе шифрования (IPSec VPN).

От других виртуальных частных сетей, таких как ATM/FR VPN или IPSec VPN, сеть MPLS VPN выгодно отличает хорошая масштабируемость, возможность автоматического конфигурирования и естественная интеграция с другими сервисами IP.

Механизмом, с помощью которого сайты, принадлежащие к одной и той же сети VPN, обмениваются маршрутной информацией, является многопротокольное расширение для протокола BGP (MP-BGP).

Политика экспорта/импорта маршрутных объявлений — мощный инструмент для создания сетей MPLS VPN разных топологий.

Вопросы и задания

1. Можно ли, не внося никаких изменений в прикладную программу, использовать протокол IPSec для безопасной передачи данных, генерируемых этой программой?
2. Зачем в системе IPSec наряду с протоколом AH, обеспечивающим целостность и аутентификацию, предусмотрен протокол ESP, также выполняющий эти функции?
3. Чтобы получатель имел возможность проверить целостность данных, многие протоколы помещают в пакет контрольную сумму. В IPSec для обеспечения целостности используется дайджест. Поясните, в чем разница этих двух подходов.
4. Предположим, на вашем компьютере работают три приложения, данные которых вы хотите передавать в зашифрованном виде своему партнеру, используя для этого протокол IPSec. Сколько безопасных ассоциаций необходимо создать для этого?
5. Сравните транспортный и туннельный режимы IPSec. Какой из них обеспечивает более высокую безопасность? Какой лучше масштабируется? Какой экономичней?
6. Приведите примеры того, как злоумышленник может воспользоваться информацией из заголовка IP.
7. Повышает ли безопасность передаваемых данных использование протокола AH в туннельном, а не в транспортном режиме?
8. Какой механизм защиты от дубликатов, навязываемых злоумышленниками, используется в IPSec?
9. Заполнитель является еще одним средством обеспечения конфиденциальности. Поясните почему.
10. Каким образом шлюз безопасности (SG) определяет, какой вид обработки необходимо применять к каждому из поступающих на него пакетов.
11. Какие свойства реальных частных сетей могут поддерживаться виртуальными частными сетями?
12. Предложите классификацию VPN.
13. Какие технологии в сетях VPN используются, чтобы обеспечить безопасность разграничения трафика?
14. В чем состоят достоинства и недостатки сети VPN уровня 3 по сравнению с сетью VPN уровня 2?
15. Какой недостаток вы считаете основным в сети IPSec VPN?
16. Опишите механизм, применяемый в технологии MPLS VPN для разграничения адресных пространств сетей разных клиентов.
17. Как формируется таблица VRF?

18. Как могло бы выглядеть маршрутное объявление протокола MP-BGP, работающего на устройстве PE2, полученное в результате трансляции следующего маршрутного объявления, поступившего от устройства CE3 (см. рис. 24.20):

```
Net = 10 2/16
NextHop = CE3
```

19. В сети MPLS VPN пакет снабжается двумя метками — внутренней LVPN и внешней L. Какую роль играет каждая из этих меток в продвижении пакета?

20. Вы изучили принципы работы сети MPLS VPN уровня 3 для случая, когда все сайты всех клиентов соединены с магистралью одного поставщика. Попробуйте развить эти принципы для более общего случая, когда магистраль поддерживается несколькими поставщиками (рис. 24.23).

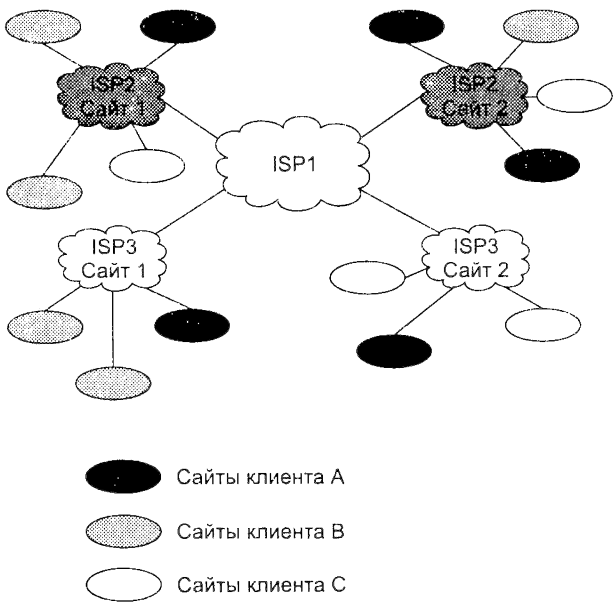


Рис. 24.23. Сеть VPN, построенная на основе сетей трех поставщиков

На этом рисунке сайты трех клиентов, A, B и C, присоединены к сетям поставщиков ISP2 и ISP3. Сети этих поставщиков, в свою очередь, соединяются друг с другом с помощью сети поставщика ISP1. Примените для решения задания иерархический подход, сделав ISP1 поставщиком верхнего уровня. В этом случае ISP2 и ISP3 будут исполнять для поставщика IPS1 роль клиентов из схемы MPLS VPN. Предложите возможную реализацию идеи иерархии поставщиков MPLS VPN, приняв в качестве основы возможности протокола BGP и идею стека меток MPLS.

21. Сравните количество необходимых виртуальных каналов и путей LSP, которые нужно проложить поставщику услуг VPN для двух случаев:

- поставщик использует сеть Frame Relay для предоставления услуг VPN;
- поставщик использует сеть IP/MPLS для предоставления услуг VPN.

У поставщика имеется 25 клиентов, сети каждого клиента состоят из 10 сайтов, соединенных с сетью поставщика услуг. Клиентам нужны услуги Интернета, то есть соединения между сайтами клиентов не предусматриваются.

22. В подразделе «Перемещение пакета по сети MPLS VPN» раздела «Технология MPLS VPN» приведен пример передачи пакета из узла 10.2.1.1/16 сайта 2 сети VPN A к узлу 10.1.0.3/16 сайта 1 той же сети VPN (см. рис. 24.21). Используя рис. 24.21, опишите передачу пакета в обратном направлении, то есть от узла 10.1.0.3/16 к узлу 10.2.1.1/16. Приведите возможное содержимое таблиц маршрутизации CE1 и PE1. Предложите собственные значения для недостающих данных и поместите на рисунок поясняющий текст.

Заключение. Заглядывая в будущее

Чем дальше в будущее мы хотим заглянуть, тем меньше у нас шансов обнаружить в этом будущем компьютерные сети в традиционном смысле этого термина, то есть сети, передающие только текст и числа. Главная тенденция для всех типов сетей — телефонных, компьютерных, телевизионных — конвергенция, поэтому уже сегодня компьютерные сети передают несвойственные им изначально типы трафика. Это, прежде всего, звук в разных видах: в форме интерактивного взаимодействия двух участников телефонного разговора; в форме вещания по запросу — передача песен или заранее записанных выступлений или интервью через Интернет; в форме голосовой почты. Передача изображения требует существенно более высокой пропускной способности и поэтому пока применяется гораздо в более скромных масштабах, однако даже при скорости доступа 64–128 Кбит/с можно просмотреть в реальном времени телепередачу в небольшом прямоугольном окошке на экране ПК.

Таким образом, телекоммуникационные сети будущего — это сети, одинаково хорошо передающие и пульсирующий трафик данных, и потоковый трафик звука и видео. Сети будущего унаследуют лучшие черты своих прародителей — телефонных и компьютерных сетей, а также сетей радио- и телевещания, но с использованием общей транспортной технологии, которая должна обеспечить передачу каждого типа трафика с требуемым для него качеством обслуживания (QoS). Такая технология должна, по общему мнению специалистов, основываться на технике коммутации пакетов и широко применять протокол-победитель IP, что роднит сети будущего с нынешними компьютерными сетями, но со значительными технологическими новациями.

В число таких усовершенствований, скорее всего, войдут терминальные устройства нового типа, которые будут сочетать функциональную мощь ПК с простотой в обращении телефона. Прообразом таких устройств сегодня являются органайзеры, персональные секретари и мобильные телефоны. Появление устройства, которое позволяет нажатием нескольких кнопок получить доступ к заранее заданным веб-страницам, организовать телефонный разговор, отправить электронное письмо с мультимедийными вложениями или заказать демонстрацию на экране нужного видеофильма (и получить доступ ко многим другим услугам,

которые сегодня пока еще только угадываются), придаст мощный импульс развитию телекоммуникаций.

Ответом на резкий рост потребности в сверхскоростном и качественном транспорте станет технология управляемых виртуальных путей на основе стандартов DWDM и GMPLS. Ядро новой публичной телекоммуникационной сети будет строиться на оптических кабелях с большим количеством волокон, что обеспечит мультитерабитную пропускную способность между узлами коммутации и создаст основу для передачи кажущихся сегодня немыслимыми объемов информации между абонентами сети. Для экономичности ядро должно поддерживать коммутацию только сверхскоростных потоков, таких как потоки данных определенной длины волны (DWDM-коммутация) или даже определенного волокна, не занимаясь более мелкими единицами коммутации. В результате технология SDH уступит свое место в ядре сети, сменив его на роль сети доступа к DWDM-коммутаторам. Еще одним революционным преобразованием станет управляемость ядра сети на основе технологии GMPLS, когда пути составных волокон, длин волн (и контейнеров SDH) создаются динамически с помощью единого сигнального протокола. Важно, что будет существовать и пользовательская версия этого протокола, то есть абонент ядра, например поставщик услуг, сможет пользоваться пропускной способностью гибко, в зависимости от текущих потребностей.

Низкая скорость доступа, особенно для массовых абонентов, является сегодня одним из основных препятствий на пути широкого внедрения новых мультимедийных услуг. Существует несколько путей решения этой проблемы — использование существующих медных абонентских окончаний, что наиболее подходит для массового индивидуального доступа; беспроводной доступ, как фиксированный, так и мобильный; прокладка оптических абонентских окончаний с использованием экономичной пассивной технологии PON. Для разделения пропускной способности каналов доступа будет применяться технология виртуальных соединений для микропотоков в форме ATM или IP/MPLS.

Несмотря на существенное повышение пропускной способности как ядра сети, так и сетей доступа, заторы трафика при одновременном повышении информационной емкости соединений все же возможны, поэтому для качественной передачи трафика в сетях будущего будут широко применяться методы поддержания показателей QoS. В ядре сети это будут методы, предоставляющие гарантии обслуживания крупным агрегированным потокам, несущим данные одного типа для большого количества абонентов, то есть методы, близкие к технологии DiffServ, начинающей находить применение в сетях операторов. В сетях доступа будут применяться методы поддержания показателей QoS для индивидуальных потоков, аналогичные тем, которые применяются в технологиях ATM и IntServ.

Изменяются и локальные сети. Вместо соединяющего компьютеры пассивного кабеля в них в большом количестве появилось разнообразное коммуникационное оборудование — коммутаторы, маршрутизаторы, шлюзы. Благодаря такому оборудованию стало возможным построение больших корпоративных сетей, насчитывающих тысячи компьютеров и имеющих сложную структуру. Возродился интерес к крупным компьютерам — в основном из-за того, что после спада эйфо-

рии по поводу легкости работы с персональными компьютерами выяснилось, что системы, состоящие из сотен серверов, обслуживать сложнее, чем несколько больших компьютеров. Поэтому на новом витке эволюционной спирали на предприятия стали возвращаться мэйнфреймы, но уже как полноправные сетевые узлы, поддерживающие технологию Ethernet или Token Ring, а также стек протоколов TCP/IP, ставший благодаря Интернету сетевым стандартом де-факто.

Вот только некоторые направления развития телекоммуникационных сетей, которые отчетливо видны уже сегодня.

Рекомендуемая и использованная литература

1. *Фред Халсалл.* Передача данных, сети компьютеров и взаимосвязь открытых систем. — М.: Радио и связь, 1995.
2. *Столлингс В.* Передача данных. — 4-е изд. СПб.: Питер, 2004.
3. *Столлингс В.* Современные компьютерные сети, 2-е изд. — СПб.: Питер, 2003.
4. *Куроуз Дж., Росс К.* Компьютерные сети, 4-е изд. — СПб.: Питер, 2004.
5. *Таненбаум Э.* Компьютерные сети, 4-е изд. — СПб.: Питер, 2002.
6. *Марк Спортак, Френк Паппас и др.* Компьютерные сети и сетевые технологии. — ТИД «ДС», 2002.
7. *Куинн Л., Рассел Р.* Fast Ethernet. — BHV-Киев, 1998.
8. *Эд Титтель, Стив Джеймс, Дэвид Пискителло, Лайза Пфайфер.* ISDN просто и доступно. — М.: Лори, 1999.
9. *Фейт Сидни.* TCP/IP. Архитектура, протоколы, реализация. — М.: Лори, 2000.
10. *Стивен Браун.* Виртуальные частные сети. — М.: Лори, 2001.
11. *Шринивас Вегешна.* Качество обслуживания в сетях IP. — Вильямс, 2003.
12. *Аннабел З. Додд.* Мир телекоммуникаций. Обзор технологий и отрасли. — М.: ЗАО «Олимп-Бизнес», 2002.
13. *Крейг Хант.* Персональные компьютеры в сетях TCP/IP/Перев. с англ. — BHV-Киев, 1997.
14. *Дженнингс Ф.* Практическая передача данных: Модемы, сети и протоколы/ Перев. с англ. — М.: Мир, 1989.
15. *Кеннеди Кларк, Кевин Гамильтон.* Принципы коммутации в локальных сетях. — Cisco, 2003.
16. *Дуглас Э. Камер.* Сети TCP/IP. Том 1. Принципы, протоколы и структура. — Вильямс, 2003.
17. *Оливер Ибе.* Сети и удаленный доступ. Протоколы, проблемы, решения. — ДМК Пресс, 2002.

18. *Стерн, Монти.* Сети предприятий на основе Windows NT для профессионалов/Перев. с англ. — СПб.: Питер, 1999.
19. *Блэк Ю.* Сети ЭВМ: протоколы стандарты, интерфейсы/Перев. с англ. — М.: Мир, 1990.
20. *Ричард Стивенс.* Протоколы TCP/IP. Практическое руководство. — Спб.: БХВ, 2003.
21. *Слепов Н. Н.* Синхронные цифровые сети SDH. — Эко-Трендз, 1998.
22. *Денисьев и Мирошников.* Средства связи для «последней мили». — Эко-Трендз, 1998.
23. *Щербо В. К.* Стандарты вычислительных сетей. Взаимосвязи сетей. Справочник. — М.: Кудиц-образ, 2000.
24. *Дилип Найк.* Стандарты и протоколы Интернета. — Channel Trading Ltd., 1999.
25. *Уолрэнд Дж.* Телекоммуникационные и компьютерные сети. Вводный курс. — М.: Постмаркет, 2001.
26. *Гольдштейн Б. С., Пинчук А. В., Суховицкий А. Л.* IP-телефония. — Радио и связь, 2001.
27. *Олифер В., Олифер Н.* MPLS на службе VPN. — LAN/Журнал сетевых решений, март 2002.
28. *Олифер В., Олифер Н.* IP-телефония в глобальном масштабе. — LAN/Журнал сетевых решений, сентябрь 2001.
29. *Олифер Н.* Агрегирование каналов в локальных сетях. — LAN/Журнал сетевых решений, февраль 2002.
30. *Олифер В., Олифер Н.* Виртуальные частные сети на основе MPLS. — LAN/Журнал сетевых решений, январь 2002.
31. *Олифер В., Петрусов Д.* Внедрение услуг IP-телефонии в сети оператора связи. — Аналитический и информационный журнал «Документальная электросвязь», № 8, январь 2002.
32. *Пятибратов и др.* Вычислительные системы, сети и телекоммуникации. — ФИС, 1998.
33. *Олифер Н.* Дифференцированная защита трафика средствами IPSec. — LAN/Журнал сетевых решений, август 2001.
34. *Олифер В., Олифер Н.* Искусство оптимизации трафика. — LAN/Журнал сетевых решений, декабрь 2001.
35. *Олифер Н.* Маршрутизатор как он есть. — LAN/Журнал сетевых решений, июль-август 2001.
36. *Олифер Н.* Маршрутизация в составных сетях. — LAN/Журнал сетевых решений, май 2001.
37. *Олифер В.* Направления развития средств безопасности предприятия. — Электроника, № 1, 2001.
38. *Олифер В. Г., Олифер Н. А.* Новые технологии и оборудование IP-сетей. — СПб.: БХВ-Санкт-Петербург, 2000.

39. *Олифер В., Олифер Н.* Протокол IP — на пути в будущее. — Мир Internet, № 7, 1997.
40. *Олифер Н.* Протоколы IPSec. — LAN/Журнал сетевых решений, март 2001.
41. *Олифер Н.* Резервирование соединений в локальных сетях. — LAN/Журнал сетевых решений, январь 2002.
42. *Олифер В. Г., Олифер Н. А.* Сетевые операционные системы. — СПб.: Питер, 2001.
43. TCP/IP Network Administration, 2/e, Craig Hunt, O`Reilly & Associates, 1998
44. Telecommunications for Managers, 3/e, Stanford H. Rowe, Prentice Hall, 1995
45. Data and Computer Communications, 5/e, William Stallings, Prentice Hall, 1997
46. Emerging Communications Technologies, 2/e, Uyless Black, Prentice Hall Professional, 1997.
47. Computer Networks: Protocols, Standards and Interface, 2nd Edition. Prentice-Hall, 1993.

Алфавитный указатель

B

C

D

E

F

G

H

I

M

N

O

P

Q

R

S

T

U

V

W

X

А

Б

В

Г

Д

Е

З

И

K

Л

M

Н

О

П

Р

C

T

У

Ф

X

Ц

Ч

Ш

Э

Я

Виктор Григорьевич Олифер, Наталья Алексеевна Олифер

Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов

3-е издание

Главный редактор	*Е. Строганова*
Заведующий редакцией	*А. Кривцов*
Руководитель проекта	*А. Жданов*
Литературный редактор	*А. Жданов*
Художник	*Е. Дьяченко*
Иллюстрации	*Г. Домрачева, С. Романов, М. Шендерова*
Корректор	*В. Листова*
Верстка	*Ю. Сергиенко*

Лицензия ИД № 05784 от 07.09.01.

Подписано к печати 21.06.05. Формат 70×100/16. Усл. п л. 69,66. Тираж 5000. Заказ 252

ООО «Питер Принт», 194044, Санкт-Петербург, пр. Б. Сампсониевский, 29а.

Налоговая льгота — общероссийский классификатор продукции ОК 005-93, том 2; 95 3005 — литература учебная.

Отпечатано с готовых диапозитивов в ОАО «Техническая книга»

190005, Санкт-Петербург, Измайловский пр., 29