Network Working Group D. Perkins Request for Comments: 1547
Carnegie Mellon University Category: Informational December 1993

Requirements for an Internet Standard Point-to-Point Protocol

Status of this Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Abstract

This document discusses the evaluation criteria for an Internet Standard Data Link Layer protocol to be used with point-to-point links. Although many industry standard protocols and ad hoc protocols already exist for the data link layer, none are both complete and sufficiently versatile to be accepted as an Internet Standard. In preparation to designing such a protocol, the features necessary to qualify a point-to-point protocol as an Internet Standard are discussed in detail. An analysis of the strengths and weaknesses of several existing protocols on the basis of these requirements demonstrates the failure of each to address key issues.

Historical Note: This was the design requirements document dated June 1989, which was followed for RFC-1134 through the present. It is now published for completeness and future guidance.

Table of Contents

1. Introduction

The Internet has seen explosive growth in the number of hosts supporting IP [1]. The vast majority of these hosts are connected to Local Area Networks (LANs) of various types, Ethernet being the most common. Most of the other hosts are connected through Wide Area Networks (WANs), such as X.25 style Public Data Networks (PDNs).

In the past, relatively few of these hosts were connected with simple point-to-point links. Yet, point-to-point serial links are among the oldest methods of data communications, and almost every host supports point-to-point connections. For example, asynchronous RS-232 interfaces are essentially ubiquitous.

One reason for the small number of point-to-point IP links was the lack of a single established encapsulation protocol. There were plenty of non-standard (and at least one de facto standard) encapsulation protocols available, but there was not one which was agreed upon as an Internet Standard.

A number of protocols have been proposed to the Internet community, but no consensus was reached as to which protocol should be adopted as a standard. The reason may be that these proposals often addressed specific problems rather than providing general purpose service.

For example, one of the most successful protocols to-date was Rick Adam's SLIP protocol for BSD UNIX [9]. SLIP provides only the most rudimentary support for sending IP datagrams over asynchronous serial lines, and ignores issues such as the use of protocols other than IP and the use of synchronous links.

This document proposes a set of requirements for an Internet Standard point-to-point protocol (ISPPP). Its purpose is not to propose any one design for the standard; any solutions outlined in the text are intended only as examples, and do not preclude other implementations.

The document is divided into four major sections. The first section defines a number of technical terms used in this document. The second section lists the proposed requirements and details some

issues that are ignored by other protocols. The third section attempts to clarify a number of non-requirements. The fourth section analyzes existing protocols in light of the proposed requirements and discusses the failure of each to address key issues.

1.1 Definitions of Terms

This section defines many of the terms which will be used in further sections of this document. The terms "layer" and "level" are used extensively and refer to protocol layers as defined by the International Organization For Standardization's Reference Model (ISORM) standard. In particular, the terms Physical Layer, Data Link Layer and Network Layer refer to layers one, two and three respectively of the ISORM. A "higher layer" refers to one with a numerically larger layer number.

datagram

The unit of transmission in the network layer (such as IP). A datagram may be encapsulated in one or more packets (q.v.) passed to the data link layer.

data link layer

Layer two in the ISO reference model. Defines how bits transmitted and received by the physical layer are recognized as bytes and frames. May also define procedures for error detection and correction, sequencing and flow control.

fragment

The result of fragmentation. Fragmentation at the network layer breaks large datagrams into multiple parts less than or equal to the size of the packets passed to the data link layer. Fragmentation at the data link layer breaks large packets into multiple frames.

frame

The unit of transmission at the data link layer. A frame may include a header and/or a trailer along with some number of units of data.

framing protocol

A protocol at the data link level for marking the beginning and end of a frame transmitted across a link.

internet

An interconnected system of networks tied together by a common "internet protocol" providing a common and consistent network address structure.

Internet

Specifically refers to the IP Internet.

Internet Standard Point-to-Point Protocol (ISPPP)

A point-to-point protocol which is declared an official Internet Standard. This protocol does not yet exist, but its proposed characteristics are presented in this paper.

Maximum Transmission Unit (MTU)

The maximum allowable length for a packet (q.v.) transmitted over a point-to-point link without incurring network layer fragmentation.

network layer

Layer three in the ISO reference model. Responsible for routing packets (q.v) between physical networks.

octet

A unit of transmission consisting of 8 bits. On most machines an octet is the same as a byte or a character, but this need not be true.

packet

The unit of transmission passed across the interface between the network layer and the data link layer. A packet is usually mapped to a frame (q.v); the exception is when data link layer fragmentation is being performed.

physical layer

The first layer in the ISO reference model. Describes electrical, mechanical and timing characteristics of a link.

point-to-point protocol (ppp)

A data link layer protocol for the transmission of packets (q.v.)

over a point-to-point link. In the following discussion, the acronym "ppp" refers to any generic point-to-point protocol.

serial line IP (slip)

Often incorrectly used as a synonym for "point-to-point protocol", "slip" specifically refers to any protocol for the transmission of IP datagrams over a serial point-to-point line.

SLIP

Although many proposed protocols are named "SLIP", this document will use SLIP (uppercase) to refer to Rick Adam's slip (q.v.) for BSD UNIX [9].

2. Required Features

In order for a point-to-point protocol to be accepted by the Internet community it must adequately address many requirements. This section itemizes and discusses the proposed requirements. Although the main emphasis of the discussion is on protocol architecture requirements, implementation requirements are sometimes discussed as well.

These particular requirements were chosen to assure that the ISPPP adequately serves the needs of its users. Some of these needs are universal and dictate clear requirements for the protocol; for example, a packet framing protocol is a fundamental necessity. Other needs are more specific and may even be conflicting. Connection liveness determination is very important on some links but can be very expensive on others. A standard protocol must address all of these needs; in particular, it must be able to resolve conflicts effectively.

Resolving these conflicts requires that a protocol feature have both enabled and disabled modes and that these modes must be compatible with each other. The enabled mode allows the protocol to solve problems in environments where they exist. The disabled mode allows problems to be ignored in environments where they do not exist. To assure interoperabilty, implementations are required to support both modes and allow the user (not necessarily human) to dynamically choose which is appropriate.

This is essentially the same solution used in the User Datagram Protocol (UDP) [2]. The UDP datagram checksum may be computed (enabled mode) or it may not (disabled mode). Compatibility is maintained by requiring the checksum to be transmitted as zero in disabled mode and ignored when received as zero in either mode. Implementations of UDP are generally encouraged to support both modes

but allow the application to choose modes.

2.1 Simplicity

The ISPPP must be simple. The Internet architecture very carefully places the most complexity in the transport layer (that is, TCP). The internetwork layer (IP) is a fairly simple, almost stateless protocol providing an unreliable datagram service. The data link layer need provide no more capability than the IP protocol; no error correction, sequencing or flow control is necessary. Including these would in most cases needlessly duplicate the capabilities of the transport layer, and might possibly decrease efficiency. This is not to say that these capabilities must never be included; there are some cases which may warrant them. For instance, very noisy links may be more efficiently handled using a more complex data link layer protocol such as CCITT's LAPB. Nevertheless, the watchword for a point-to-point protocol should be simplicity.

A simple design also decreases the incidence of programming errors, thereby increasing the likelihood of interoperability among different implementations. Since interoperability is a primary goal of standardization, this is another strong argument for simplicity.

2.2 Transparency

The ISPPP must be transparent to higher layers. The protocol must not place any constraints on transmitted data. All ISPPP data, including higher level headers as well as data, must be transported unmodified end-to-end. No restrictions are placed on how the ISPPP accomplishes this. For example, if the ISPPP uses a particular character for framing, it must also provide some way of disambiguating higher level data containing that character from a framing character (such as escaping or bit-stuffing). This is mainly an issue for the data link and physical layer protocols incorporated into the ISPPP.

2.3 Packet Framing

The ISPPP must be able to correctly and efficiently frame packets. A receiver must be able to locate correctly the beginning and end of each transmitted packet. Within each packet, the receiver must be able to identify the boundaries of each octet. Finally, within each octet, each bit must be located and identified. No restrictions other than those specified in this document are placed on the packet framing protocol.

2.4 Bandwidth Efficiency

The ISPPP must make efficient use of available bandwidth. At most, the ppp overhead may impose a few percent reduction in raw link bandwidth.

2.5 Protocol Processing Efficiency

The processing of the ISPPP headers must typically be very fast and efficient. The format for data packets should be very simple in the normal case, without complex field checking.

2.6 Protocol Multiplexing

The ISPPP must support multiplexing of many higher level protocols. Although the Internet community is interested mainly in IP, co- existence of other protocols is frequently required. IP networks must often support additional protocols such as AppleTalk, DECnet, IPX, and XNS. For point-to-point links to connect gateways on geographically separated Local Area Networks (LANs), the ISPPP must simultaneously support all protocols implemented on both the LANs and the gateways. This suggests that the ISPPP must include a protocol type field or other multiplexing scheme. Given the large number of protocols, the potential use of the protocol type field as a data compression aid, and the experimental nature of the Internet, eight bits of type field are not sufficient. Sixteen bits of type field are suggested, although twelve bits (4096 protocols) should suffice.

2.7 Multiple Physical and Data Link Layer Protocols

The ISPPP must support a multiplicity of physical and data link layer protocols. Many types of point-to-point links exist. Links can be serial or parallel, synchronous or asynchronous, low speed or high speed, electrical or optical. Standards are required for the transmission of IP datagrams over each type of commonly used link.

The ISPPP must not inhibit the use of any type of link. This includes, but is not limited to, asynchronous, bit-oriented synchronous (HDLC [10] and X.25 LAPB [11]), and byte-oriented synchronous (BISYNC and DDCMP [15]) links.

The ISPPP must initially provide support for at least the following types of links:

Full duplex asynchronous RS-232 [3] links with 8 bits of data and no parity, ranging in speeds from 300 to 19.2k bps or more.

Full duplex bit-oriented synchronous links including RS-422, RS-

423, V.35 and T1.

Other links should be standardized as the need arises.

2.8 Error Detection

The ISPPP must provide some form of basic error detection. Most network and transport layer protocols provide mechanisms to detect corrupted packets. However, some network protocols expect error free transmission and either provide error detection only on a conditional basis or do not provide it at all. It is the consensus of the Internet community that error correction should always be implemented in the end-to-end transport, but that link error detection in the form of a checksum, Cyclic Redundancy Check (CRC) or other frame check mechanism is useful to prevent wasted bandwidth from propagation of corrupted packets. Link level error correction is not required.

2.9 Standardized Maximum Packet Length (MTU)

The ISPPP must have a standardized default maximum packet length for each type of point-to-point link. This standardization helps to promote interoperable implementations. Higher layer protocols must not attempt to transmit packets longer than the MTU. If a higher layer protocol does try to transmit a packet which is too long, the ISPPP must drop the packet and return an error. The MTU may potentially be changed from the default via some sort of explicit negotiation or private agreement, but the default must be enforced in all other cases. The default should be at least 1500 bytes, to efficiently carry common LAN traffic.

2.10 Switched and Non-Switched Media

The ISPPP must be able to support both switched (dynamic) and non- switched (static) point-to-point links. A common example of a non- switched link is a 3-wire asynchronous RS-232 cable which might connect a host to a particular gateway. Switched media may be exemplified by connections over a standard voice network or an Integrated Services Digital Network (ISDN). Links over ISDN are currently rare, but are expected to become increasingly commonplace. To be a viable standard, the ISPPP must be able to effectively support both types of links. Procedures for establishing switched connections are beyond the scope of this document.

2.11 Symmetry

The ISPPP should operate symmetrically to maximize flexibility.

The ISPPP must allow communications among any combination of gateways and hosts. One host may need to communicate directly with another host, or it may be connected to a gateway to gain access to a whole network. A gateway may establish a connection to a single host in order to deliver a packet, or it may connect to another gateway on a permanent or transient basis. Symmetry is destroyed by pre-assigned static roles, such as master and slave or gateway and host. If necessary, roles may be dynamically determined on a per connection basis.

2.12 Connection Liveness

The ISPPP must include a mechanism to automatically determine when a link is functioning properly and when it is defunct. This mechanism should be enabled by default, but the protocol and all implementations must allow this mechanism to be disabled.

When enabled, this mechanism should discover changes in a link's status in a timely fashion -- no more than a few minutes. Continuing to utilize a link which is down often causes routing problems commonly referred to as "black holes". These problems can be hard to find and diagnose. By automatically detecting a failing link, a point-to-point protocol can avoid such problems, and also provide a powerful tool for a network manager trying to locate and remedy the fault.

When a point-to-point connection is not functioning properly, it must be declared "down" for the purposes of routing packets for higher level protocols. In order to certify a link "up", the systems on either end of the link must be able to successfully exchange packets. In other words, the systems at both ends must be able both to transmit and to receive packets, and the link must be able to transport packets in both directions. Links are defined to be "down" at initialization, their liveness must be verified before they may be declared "up".

This feature may be disabled in situations where connection status determination is "expensive". For example, a link may traverse a Public Data Network (such as TELENET or TYMNET) which accounts for bandwidth utilization. Constant pinging would result in charges being accrued even in the absence of useful communications.

2.13 Loopback Detection

The ISPPP must be capable of automatically detecting a looped-back link without operator assistance. Modems and other communications gear are often placed in a loopback mode to aid in diagnosis of circuit failures. Detection of this condition must take no longer

than one period of the liveness protocol. While the link is in loopback mode, each end of the link must declare the other end to be unreachable. However, to aid in diagnosis, each end of the link may declare itself reachable for any higher-level protocol which distinguishes between the two ends of the link.

2.14 Misconfiguration Detection

The ISPPP must be able to quickly detect misconfigured point-to-point connections. A connection which is misconfigured must never be declared to be up. Many systems, gateways in particular, have more than one point-to-point connection. When many cables terminate within a small area, the possibility for confusion abounds. It becomes very easy to mistakenly plug a cable into the wrong connector, or even to swap cables. The protocol should do its best to provide protection against these errors by verifying the remote end's identity whenever possible before marking an interface as operational. The purpose of this verification is not rigorous authentication but the detection of simple errors.

2.15 Network Layer Address Negotiation

The ISPPP must allow network layer (such as IP) addresses to be negotiated. The negotiation algorithm should be as simple as possible and must be guaranteed to terminate in all cases. Many network layer protocols and implementations are required to know the addresses at both ends of a point-to-point link before packets may be routed. These addresses may be statically configured, but it may sometimes be necessary or convenient for these addresses be dynamically ascertained at connection establishment. This is especially important when switched media are used. For example, a dial-up IP gateway must know the IP address of its peer before packets can be successfully routed. This address can be either statically or dynamically configured. In the former case, the gateway's peer must therefore learn the static address (static with respect to the gateway). In the latter situation, the gateway must dynamically learn the address used by its peer.

2.16 Data Compression Negotiation

The ISPPP must provide a way to negotiate the use of data compression algorithms. This mechanism should be as simple as possible and must be guaranteed to terminate in all cases. The protocol is not required to standardize any data compression algorithms; conforming implementations of the protocol therefore may refuse to do data compression when negotiating (refusal to do data compression always takes precedence over an offer to do it). However, to allow the use of data compression between consenting systems, the point-to-point

protocol must not impede the use of data compression. In fact, it should be possible to use multiple, independent data compression schemes simultaneously. Because data compression algorithms are still very experimental in the Internet environment, it is likely that many different algorithms will be tried. The negotiation protocol must distinguish between these different algorithms to ensure that data compression is not enabled unless the same algorithm or algorithms are used at both ends of the connection. The number of such supported algorithms must be easily extensible.

2.17 Extensibility and Option Negotiation

The ISPPP must allow for future extensions in a flexible way. The Internet will never cease to evolve. Changes in technology and user demands create new requirements. To function effectively as a standard, the protocol must have the ability to evolve along with its environment.

To accomplish this, the ISPPP should be designed to be as extensible as possible and to allow for experimentation within the guidelines of the other requirements presented in this document. A proposed solution is to specify an option negotiation protocol. The option negotiation protocol could be used for the negotiation of network layer addresses, data compression schemes, MTU, encryption, etc. The option negotiation protocol must itself be extensible; it should allow the negotiation of a large number of future options and it should allow the use of other types of point-to-point links and encapsulation schemes.

3. Features Not Required

This section discusses functionality which is explicitly not required. These functions may potentially be included in implementations as long as the inclusion does not violate any of the requirements itemized in the previous section.

3.1 Error Correction

As discussed above in the sections on Simplicity and Error Detection, error correction is the responsibility of the transport layer and is not required in a point-to-point protocol. However, on links with high error rates, performance may be increased by adding error correction at the data link level. Therefore, the ISPPP must not prevent the addition of error correction by private agreement, even though such mechanisms are not required in the basic implementation.

3.2 Flow Control

Flow control (such as XON/XOFF) is not required. Any implementation of the ISPPP is expected to be capable of receiving packets at the full rate possible for the particular data link and physical layers used in the implementation. If higher layers cannot receive packets at the full rate possible, it is up to those layers to discard packets or invoke flow control procedures. As discussed above, end- to-end flow control is the responsibility of the transport layer. Including flow control within a point-to-point protocol often causes violation of the simplicity requirement.

3.3 Sequencing

Sequencing of packets is not required. The ISPPP need provide no more service than the IP protocol, an unreliable datagram service which is free to reorder packets. In fact, it is specifically allowed to reorder packets based upon some type-of-service criteria implemented in higher-level protocols.

3.4 Backward Compatibility

There is no requirement for the ISPPP to provide backward compatibility with any other point-to-point protocol. First, there are no official Internet Standards with which backward compatibility must be maintained. Second, attempting to maintain backward compatibility may lead to needless restrictions on the new protocol. However, there is no need for the designers of the ISPPP to go out of their way to inhibit backward compatibility.

3.5 Multi-Point Links

There is no requirement for supporting multi-point links. Many features which are required are only valid between two peers. These links are sufficiently rare that the benefits of supporting them are outweighed by the added complexity their support would introduce into the ISPPP.

Historical Note: The original rationale also stated: "Furthermore, it is unlikely that many new types of multi-point links will be introduced in the foreseeable future." Since this was written, considerable effort has been expended in new multi-point links, including Switched Multimegabit Data Service, Frame Relay, and Asynchronous Transfer Mode. However, it is clear that these are considerably more complex than ISPPP.

3.6 Half-Duplex or Simplex Links

Support for half-duplex or simplex links is not required. These types of links are not in common use in the current Internet. Half- duplex links require some method of turning the line around. The ISPPP need not have an explicit mechanism for handling line turn- around. Such support might possibly be added in the future via the required extension mechanism.

3.7 7-bit Asynchronous RS-232 Links

The use of asynchronous RS-232 need not support 7-bit links. 8-bit links are predominant in the Internet environment and supporting 7- bit links introduces unnecessary complexity.

4. Prior Work On PPP Protocols

This section reviews a number of existing point-to-point and data link layer protocols and points out which of our requirements are not satisfied.

4.1 Internet Protocols

4.1.1 RFC 891 - DCN Local-Network Protocols, Appendix A

In Appendix A of RFC 891, "DCN Local-Network Protocols" [4], D.L. Mills describes the data link layer packet formats used by the Fuzzball system for asynchronous, character-oriented synchronous, DDCMP, HDLC, ARPANET 1822, X.25 LAPB and ethernet links. These protocols meet the stated requirements for simplicity, transparency, packet framing and efficiency, but fall short of many of the others. Most of these protocols assume the use of the IP protocol, and do not include any type of protocol demultiplexing field. No error detection mechanism is provided except when necessary to comply with another standard such as ethernet. RFC 891 does not mention the MTU used for any of these links. Other requirements such as loopback detection and misconfiguration detection are not discussed. Finally, no option negotiation scheme is defined; without a protocol demultiplexing field it would be difficult or impossible to include one.

4.1.2 RFC 914 - Thinwire Protocols

RFC 914, "Thinwire Protocols" [5], discusses the use of low speed links in the Internet. This document places its main emphasis on decreasing round-trip delay and increasing link efficiency with the help of header compression (vs. data compression) techniques. Three "Thinwire" protocols are discussed, Thinwire I, Thinwire II and

Thinwire III. The latter two protocols require the use of a reliable data link layer protocol; one such protocol, "SLIP" (not to be confused with Rick Adams' SLIP), is proposed in Appendix D of the RFC. As proposed, "SLIP" does not meet many of the stated requirements. Although not terribly complex, as a reliable, error detecting and correcting protocol, it is not "simple". The 32 octet packet size makes it inefficient for large or uncompressed packets, requiring complex fragmentation and reassembly. The use of other than asynchronous links is not mentioned. The entire reliable link layer would be redundant over LAPB links. There is no mechanism for option negotiation or future extensibility.

4.1.3 RFC 916 - Reliable Asynchronous Transfer Protocol

RFC 916 [6] presents RATP, the Reliable Asynchronous Transfer Protocol. RATP provides error detection and correction, sequencing and flow control across a point-to-point connection. It is directed towards full duplex RS-232 links although it is useful for other point-to-point links. Although the author claims that RATP is not as complex as some other protocols, it is far from simple. RATP solves many of the problems which we have labeled non-requirements and fails to solve many of our stated requirements. Specifically, RATP does not support option negotiation and has no mechanism for future extensibility. Since RFC 916 was published, no consensus has emerged advocating RATP. For these reasons RATP is not recommended as the ISPPP.

4.1.4 RFC 935 - Reliable Link Layer Protocols

RFC 935 [7] is a rebuttal to the protocols proposed in RFCs 914 and 916. J. Robinson discusses existing and widely-used national and international standards which meet the needs addressed by the two prior RFCs. The standards reviewed include character-oriented asynchronous and synchronous (bisynch) protocols and bit-oriented synchronous protocols. RFC 935 does not present any higher level issues such as option negotiation or extensibility.

4.1.5 RFC 1009 - Requirements for Internet Gateways

Section 3 of RFC 1009, "Constituent Network Interfaces" [8], briefly discusses requirements for transmission of IP datagrams over a number of types of point-to-point links including X.25 LAPB, HDLC framed synchronous links, Xerox Synchronous Point-to-Point synchronous lines and the MIT Serial Line Framing Protocol for asynchronous lines. RFC 1009 merely mentions these as reasonable candidates and does not go into depth on any of them. All are discussed further in this document.

4.1.6 RFC 1055 - Serial Line IP

Rick Adams' Serial Line IP (SLIP) protocol [9] has become something of a de facto standard due to the popularity of the 4.2 and 4.3BSD UNIX operating systems. SLIP is easily added to 4.2 systems and is included with 4.3. Many other TCP/IP implementation have added SLIP implementations in order to be compatible. Yet SLIP is not a real standard; the protocol was only recently published in RFC form. Before RFC 1055 it was specified in the SLIP source code. SLIP does not meet most of the requirements set forth above. SLIP certainly meets the requirement for simplicity, and also meets the requirements for transparency and bandwidth efficiency. But SLIP only provides for sending IP packets over asynchronous serial lines. Since it provides no higher level protocol field for demultiplexing, SLIP cannot support multiple concurrent higher level protocols. Providing only a framing protocol, SLIP would be entirely redundant when used with a LAPB synchronous link. SLIP includes absolutely no mechanism for error detection, not even parity. Again due to its lack of a protocol type field, SLIP does not support any type of option negotiation or extensibility.

4.2 International Protocols

4.2.1 ISO 3309 - HDLC Frame Structure

ISO 3309 [10], the HDLC frame structure, is a simple data link layer protocol which provides framing of packets transmitted over bit- oriented synchronous links. Special flag sequences mark the beginning and end of frames and bit stuffing allows data containing flag characters to be transmitted. A 16-bit Frame Check Sequence provides error detection.

By itself, the HDLC frame structure does not meet most of the requirements. HDLC does not provide protocol multiplexing, standard MTUs, fault detection or option negotiation. There is no mechanism for future extensibility.

Given the HDLC frame structure's wide acceptance and simplicity, it may be an ideal building block for the ISPPP.

4.2.2 ISO 6256 - HDLC Balanced Class of Procedures

ISO 6256, the HDLC Balanced Class of Procedures, specifies a data link layer protocol which provides error correction, sequencing and flow control. ISO 6256 builds on ISO 3309 and ISO 4335, HDLC Elements of Procedures.

As far as meeting our requirements is concerned, ISO 6256 does not

provide any more utility than does ISO 3309. The capabilities that are provided are all considered unnecessary and overly complex.

4.2.3 CCITT X.25 and X.25 LAPB

CCITT recommendation X.25 [11] describes a network layer protocol providing error-free, sequenced, flow controlled virtual circuits. X.25 includes a data link layer, X.25 LAPB, which uses ISO 3309, 4335 and 6256. Neither X.25 LAPB or full LAPB meet any more of our requirements than the ISO protocols.

4.2.4 CCITT I.441 LAPD

CCITT I.441 LAPD [12] defines the Link Access Procedure on the ISDN D-Channel. The data link layer of LAPD is very similar to that of LAPB and fails to meet the same requirements.

4.3 Other Protocols

4.3.1 Cisco Systems point-to-point protocols

The Cisco Systems gateway supports both asynchronous links using SLIP and synchronous links using either simple HDLC framing, X.25 LAPB or full X.25. The HDLC framing procedure includes a four byte header. The first octet (address) is either 0x0F (unicast intent) or 0x8F (multicast intent). The second octet (control byte) is left zero and is not checked on reception. The third and fourth octets contain a standard 16 bit Ethernet protocol type code.

A "keepalive" or "beaconing" protocol is used to ensure the two-way connectivity of the serial line. Each end of the link periodically sends two 32 bit sequence numbers to the other side. One sequence number is the local side's sequence number, the other is the sequence number received from the other side. Hearing the local sequence number from the other side indicates that the link is working in both directions.

The keepalive protocol is extensible. One extension is used to default IP addresses on serial lines of systems without non-volatile memory. A request for address is sent to the remote side. The remote side responds with its own IP address and a subnet mask. When the querying side receives the reply, it checks if the host portion of the remote address is either 1 or 2. If so, the opposite address is chosen for the local address. If not, the protocol cannot be used and we must rely on other address resolution means. This protocol assumes that each serial link uses one subnet or network number.

LAPB assuming IP is another possible encapsulation. A multi-protocol

extension of LAPB (multi-LAPB) includes a 16 bit Ethernet type code after the address and control bytes and in front of the actual protocol data. DDN X.25 and Commercial X.25 encapsulations are also supported. Multiple protocols are supported by making protocol dependent CALL REQUEST's.

4.3.2 MIT PC/IP framing protocol

The MIT PC/IP framing protocol [13] provides a mechanism for the transmission of IP datagrams over asynchronous links. The low-level protocol (LLP) sublayer provides encapsulation while the local net protocol provides multiplexing of IP datagrams and IP address request packets. The protocol only allows host-to-gateway connections. Host-to-gateway flow control is provided by requiring the host to transmit request packets to the gateway until an acknowledgment is received. Rudimentary IP address negotiation requires the host to ascertain its IP address from the gateway.

The protocol does not implement error detection, connection status determination, fault detection or option negotiation. Only asynchronous links are supported.

4.3.3 Proteon p4200 point-to-point protocol

The Proteon p4200 multi-protocol router supports transmission of packets over bit-oriented synchronous links with a wide range of speeds (zero to 2 Mb/sec). The p4200 point-to-point protocol encapsulates packets inside HDLC frames but does not use the HDLC address or control fields; these two octets are instead used for a 16-bit type field. The p4200 does use the HDLC frame check sequence trailer. Protocol type numbers are ad hoc and do not correspond to any existing standard. A simple liveness protocol detects dead connections.

Although the Proteon protocol does meet many of our requirements, it does not meet our requirements for option negotiation.

4.3.4 Ungermann Bass point-to-point protocol

The Ungermann Bass router supports synchronous links using simple HDLC framing. Neither the HDLC address or control field are used, IP datagrams are placed immediately after the HDLC flag.

The U-B protocol does not meet any of our requirements for fault detection or option negotiation. No mechanism for future extensibility is currently defined.

4.3.5 Wellfleet point-to-point protocol

The Wellfleet router supports synchronous links using simple HDLC framing. The HDLC framing procedure uses the HDLC address and places the Unnumbered Information (UI) command in all frames. A simple header following the UI command provides a two octet type field using the same values as Ethernet.

The Wellfleet protocol does not meet any of our requirements for fault detection or option negotiation. No mechanism for future extensibility is currently defined, although one could be added.

4.3.6 XNS Synchronous Point-to-Point Protocol

The Xerox Network Systems Synchronous Point-to-Point protocol (XNS PPP) [14] was designed to address most of the same issues that an ISPPP must address. In particular, it addresses the issues of simplicity, transparency, efficiency, packet framing, protocol multiplexing, error detection, standard MTUs, symmetry, switched and non-switched media, connection status, network address negotiation and future extensibility. However, the XNS SPPP does not meet our requirements for multiple data link layer protocols, fault detection and data compression negotiation. Although protocol multiplexing is provided, the packet type field has only 8 bits which is too few.

References

[1] Postel, J., "Internet Protocol", STD 5, RFC 791, USC/Information Sciences Institute, September 1981.

[2] Postel, J., "User Datagram Protocol", STD 6, RFC768, USC/Information Sciences Institute, August 1980.

[3] Electronic Industries Association, EIA Standard RS-232-C, "Interface Between Data Terminal Equipment and Data Communications Equipment Employing Serial Binary Data Interchange", August 1969.

[4] Mills, D. L., "DCN Local-Network Protocols", STD 44, RFC 891, University of Delaware, December 1983.

[5] Farber, David J., Delp, Gary S., and Conte, Thomas M., "A Thinwire Protocol for Connecting Personal Computers to the Internet", RFC 914, University of Delaware, September 1984.

[6] Finn, G., "Reliable Asynchronous Transfer Protocol (RATP)", RFC 916, USC/Information Sciences Institute, October 1984.

[7] Robinson, J., "Reliable Link Layer Protocols", RFC 935, BBN, January 1985.

[8] Braden, R., and J. Postel, "Requirements for Internet Gateways", STD 4, RFC1009, USC/Information Sciences Institute, June 1987.

[9] Romkey, J., "A Nonstandard for the Transmission of IP Datagrams Over Serial Lines: SLIP", STD 47, RFC 1055, June 1988. STD 4, RFC 1009, June 1987.

[10] ISO International Standard (IS) 3309, "Data Communications - High-level Data Link Control Procedures - Frame Structure", 1979.

[11] CCITT Recommendation X.25, "Interface Between Data Terminal Equipment (DTE) and Data Circuit Terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks", Vol. VIII, Fascicle VIII.2, Rec. X.25.

[12] CCITT Recommendation Q.921 "ISDN User-Network Interface Data Link Layer Specification".

[13] Romkey, J.L., "PC/IP Programmer's Manual", Massachussetts Institute of Technology Laboratory for Computer Science, January 1986.

[14] Xerox Corporation, "Synchronous Point-to-Point Protocol", Xerox System Integration Standard, Stamford, Connecticut, XSIS 158412, December 1984.

[15] "Digital Data Communications Message Protocol", Digital Equipment Corporation.

Security Consideration

Security issues are not discussed in this memo.

Chair's Address

The working group can be contacted via the current chair:

Fred Baker Advanced Computer Communications 315 Bollay Drive Santa Barbara, California 93117

EMail: fbaker@acc.com

Author's Address

Questions about this memo can also be directed to:

Drew Perkins 4015 Holiday Park Drive Murrysville, PA 15668

EMail: perkins+@cmu.edu

Editor's Address

Typographic revision and historical notes by:

William Allen Simpson 1384 Fontaine Madison Heights, Michigan 48071

EMail: Bill.Simpson@um.cc.umich.edu