Network Working Group
Request for Comments: 1142

David Oran, Editor
Digital Equipment Corp.

February 1990

OSI IS-IS Intra-domain Routing Protocol

Status of this Memo

This RFC is a republication of ISO DP 10589 as a service to the Internet Community. This is not an Internet standard. Distribution of this memo is unlimited.

Contents

Figures

Tables

Introduction

This Protocol is one of a set of International Standards produced to facilitate the interconnection of open systems. The set of standards covers the services and protocols required to achieve such interconnection.

This Protocol is positioned with respect to other related standards by the layers defined in the ISO 7498 and by the structure defined in the ISO 8648. In particular, it is a protocol of the Network Layer. This protocol permits Intermediate Systems within a routeing Domain (as defined in ISO/TR 9575) to exchange configuration and routeing information to facilitate the operation of the routeing and relaying functions of the Network Layer.

The protocol is designed to operate in close conjunction with ISO 9542 and ISO 8473. ISO 9542 is used to establish connectivity and reachability between End Systems and Intermediate Systems on individual Subnetworks. Data is carried by ISO 8473. The related algorithms for route calculation and maintenance are also described.

The intra-domain IS–IS routeing protocol is intended to support large routeing domains consisting of combinations of many types of subnetworks. This includes point-to-point links, multipoint links, X.25 subnetworks, and broadcast subnetworks such as ISO 8802 LANs.

In order to support large routeing domains, provision is made for Intra-domain routeing to be organised hierarchically. A large domain may be administratively divided into *areas*. Each system resides in exactly one area. Routeing within an area is referred to as *Level 1 routeing*. Routeing between areas is referred to as *Level 2 routeing*. Level 2 Intermediate systems keep track of the paths to destination areas. Level 1 Intermediate systems keep track of the routeing within their own area. For an NPDU destined to another area, a Level 1 Intermediate system sends the NPDU to the nearest level 2 IS in its own area, regardless of what the destination area is. Then the NPDU travels via level 2 routeing to the destination area, where it again travels via level 1 routeing to the destination End System.

Information technology — Telecommunications and information exchange between systems — Intermediate system to Intermediate system Intra-Domain routeing exchange protocol for use in Conjunction with the Protocol for providing the Connectionless-mode Network Service (ISO 8473)

1 Scope and Field of Application

This International Standard specifies a protocol which is used by Network Layer entities operating ISO 8473 in Intermediate Systems to maintain routeing information for the purpose of routeing within a single routeing domain. The protocol herein described relies upon the provision of a connectionless-mode underlying service.[1]

This Standard specifies:

a) procedures for the transmission of configuration and routeing information between network entities residing in Intermediate Systems within a single routeing domain;

b) the encoding of the protocol data units used for the transmission of the configuration and routeing information;

c) procedures for the correct interpretation of protocol control information; and

d) the functional requirements for implementations claiming conformance to this Standard.

The procedures are defined in terms of:

a) the interactions between Intermediate system Network entities through the exchange of protocol data units; and

b) the interactions between a Network entity and an underlying service provider through the exchange of subnetwork service primitives.

c) the constraints on route determination which must be observed by each Intermediate system when each has a routeing information base which is consistent with the others.

2 References

2.1 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of apply-

---

[1] See ISO 8473 and its Addenda 1 and 3 for the mechanisms necessary to realise this service on subnetworks based on ISO 8208, ISO 8802, and the OSI Data Link Service.

ing the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 7498:1984, *Information processing systems — Open Systems Interconnection — Basic Reference Model.*

ISO 7498/Add.1:1984, *Information processing systems — Open Systems Interconnection — Basic Reference Model — Addendum 1: Connectionless-mode Transmission.*

ISO 7498/Add.3:1989, *Information processing systems — Open Systems Interconnection — Basic Reference Model — Addendum 3: Naming, including Addressing.*

ISO 7498/Add.4:[2], *Information processing systems — Open Systems Interconnection — Basic Reference Model — Addendum 4: Management Framework.*

ISO 8348:1987, *Information processing systems — Data communications — Network Service Definition.*

ISO 8348/Add.1:1987, *Information processing systems — Data communications — Network Service Definition — Addendum 1: Connectionless-mode transmission.*

ISO 8348/Add.2:1988, *Information processing systems — Data communications — Network Service Definition — Addendum 2: Network layer addressing.*

ISO 8473:1988, *Information processing systems — Data communications — Protocol for providing the connectionless-mode network service.*

ISO 8473/Add.3:1989, *Information processing systems — Telecommunications and information exchange between systems — Protocol for providing the connectionless-mode network service — Addendum 3: Provision of the underlying service assumed by ISO 8473 over subnetworks which provide the OSI data link service.*

ISO 8648:1988, *Information processing systems — Open Systems Interconnection — Internal organisation of the Network Layer.*

ISO 9542:1988, *Information processing systems – Telecommunications and information exchange between systems — End system to Intermediate system Routeing exchange protocol for use in conjunction with the protocol for providing the connectionless -mode network service (ISO 8473).*

ISO 8208:1984, *Information processing systems — Data communications — X.25 packet level protocol for Data terminal equipment*

ISO 8802:1988, *Information processing systems — Telecommunications and information exchange between systems — Local area networks.*

ISO/TR 9575:1989, *Information technology — Telecommunications and information exchange between systems — OSI Routeing Framework.*

ISO/TR 9577:[2], *Information technology — Telecommunications and information exchange between systems — Protocol Identification in the Network Layer.*

2.2 Other References

The following references are helpful in describing some of the routeing algorithms:

McQuillan, J. et. al., *The New Routeing Algorithm for the ARPANET*, IEEE Transactions on Communications, May 1980.

Perlman, Radia, *Fault-Tolerant Broadcast of Routeing Information*, Computer Networks, Dec. 1983. Also in IEEE INFOCOM 83, April 1983.

---

[2] to be published

Aho, Hopcroft, and Ullman, *Data Structures and Algorithms*, P204–208 — Dijkstra algorithm.

3 Definitions

3.1 Reference Model definitions

This International Standard makes use of the following terms defined in ISO 7498:

a) Network Layer
b) Network Service access point
c) Network Service access point address
d) Network entity
e) Routeing
f) Network protocol
g) Network relay
h) Network protocol data unit

3.2 Network Layer architecture definitions

This International Standard makes use of the following terms defined in ISO 8648:

a) Subnetwork
b) End system
c) Intermediate system
d) Subnetwork service
e) Subnetwork Access Protocol
f) Subnetwork Dependent Convergence Protocol
g) Subnetwork Independent Convergence Protocol

3.3 Network Layer addressing definitions

This International Standard makes use of the following terms defined in ISO 8348/Add.2:

a) Subnetwork address
b) Subnetwork point of attachment
c) Network Entity Title

3.4 Local Area Network Definitions

This International Standard makes use of the following terms defined in ISO 8802:

a) Multi-destination address
b) Broadcast medium

3.5 Routeing Framework Definitions

This document makes use of the following terms defined in ISO/TR 9575:

a) Administrative Domain
b) Routeing Domain
c) Hop
d) Black hole

3.6 Additional Definitions

For the purposes of this International Standard, the following definitions apply:

3.6.1 **Area:** A routeing subdomain which maintains detailed routeing information about its own internal composition, and also maintains routeing information which allows it to reach other routeing subdomains. It corresponds to the Level 1 subdomain.

3.6.2 **Neighbour:** An adjacent system reachable by traversal of a single subnetwork by a PDU.

3.6.3 **Adjacency:** The subset of the local routeing information base pertinent to a single neighbour.

3.6.4 **Circuit:**The subset of the local routeing information base pertinent to a single local SNPA.

3.6.5 **Link:** The communication path between two neighbours.

A Link is "up" when communication is possible between the two SNPAs.

3.6.6 **Designated IS:** The Intermediate system on a LAN which is designated to perform additional duties. In particular it generates Link State PDUs on behalf of the LAN, treating the LAN as a pseudonode.

3.6.7 **Pseudonode:** Where a broadcast subnetwork has *n* connected Intermediate systems, the broadcast subnetwork itself is considered to be a *pseudonode*.

The pseudonode has links to each of the *n* Intermediate systems and each of the ISs has a single link to the pseudonode (rather than *n-1* links to each of the other Intermediate systems). Link State PDUs are generated on behalf of the pseudonode by the Designated IS. This is depicted below in figure 1.

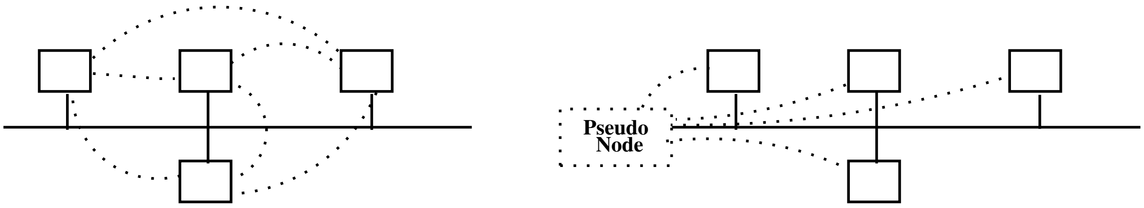


Figure 1 - Use of a Pseudonode to collapse a LAN Topology

3.6.8 **Broadcast subnetwork**: A subnetwork which supports an arbitrary number of End systems and Intermediate systems and additionally is capable of transmitting a single SNPDU to a subset of these systems in response to a single SN_UNITDATA request.

3.6.9 **General topology subnetwork:** A subnetwork which supports an arbitrary number of End systems and Intermediate systems, but does not support a convenient multi-destination connectionless transmission facility, as does a broadcast subnetwork.

3.6.10 **Routeing Subdomain:** a set of Intermediate systems and End systems located within the same Routeing domain.

3.6.11 **Level 2 Subdomain:** the set of all Level 2 Intermediate systems in a Routeing domain.

3.6.12 **Hippity Cost:** the cost, in terms of a routeing metric value, of traversing an Intermediate System. An Intermediate System having an **infinite hippity cost** indicates that no routes are to be calculated passing through that Intermediate System.

4 Symbols and Abbreviations

4.1 Data Units

PDU	Protocol Data Unit
SNSDU	Subnetwork Service Data Unit
NSDU	Network Service Data Unit
NPDU	Network Protocol Data Unit
SNPDU	Subnetwork Protocol Data Unit

4.2 Protocol Data Units

ESH PDU	ISO 9542 End System Hello Protocol Data Unit
ISH PDU	ISO 9542 Intermediate System Hello Protocol Data Unit
RD PDU	ISO 9542 Redirect Protocol Data Unit
IIH	Intermediate system to Intermediate system Hello Protocol Data Unit
LSP	Link State Protocol Data Unit
SNP	Sequence Numbers Protocol Data Unit
CSNP	Complete Sequence Numbers Protocol Data Unit
PSNP	Partial Sequence Numbers Protocol Data Unit

4.3 Addresses

AFI	Authority and Format Indicator
DSP	Domain Specific Part
IDI	Initial Domain Identifier
IDP	Initial Domain Part
NET	Network Entity Title
NSAP	Network Service Access Point
SNPA	Subnetwork Point of Attachment

4.4 Miscellaneous

DA	Dynamically Assigned
DCM	Dynamic Connection Management
DED	Dynamically Established Data link
DTE	Data Terminal Equipment
ES	End System
IS	Intermediate System
L1	Level 1
L2	Level 2
LAN	Local Area Network
MAC	Media Access Control
QoS	Quality of Service
SN	Subnetwork

SNAcP	Subnetwork Access Protocol
SNDCP	Subnetwork Dependent Convergence Protocol
SNICP	Subnetwork Independent Convergence Protocol
SRM	Send Routeing Message
SSN	Send Sequence Numbers Message
SVC	Switched Virtual Circuit

5 Typographical Conventions

This International Standard makes use of the following typographical conventions:

a) Important terms and concepts appear in *italic* type when introduced for the first time;

b) Protocol constants and management parameters appear in sansSerif type with multiple words run together. The first word is lower case, with the first character of subsequent words capitalised;

c) Values of constants, parameters, and protocol fields appear enclosed in “double quotes”.

6 Overview of the Protocol

6.1 System Types

There are the following types of system:

End Systems: These systems deliver NPDUs to other systems and receive NPDUs from other systems, but do not relay NPDUs. This International Standard does not specify any additional End system functions beyond those supplied by ISO 8473 and ISO 9542.

Level 1 Intermediate Systems: These systems deliver and receive NPDUs from other systems, and relay NPDUs from other source systems to other destination systems. They route directly to systems within their own area, and route towards a level 2 Intermediate system when the destination system is in a different area.

Level 2 Intermediate Systems: These systems act as Level 1 Intermediate systems in addition to acting as a system in the subdomain consisting of level 2 ISs. Systems in the level 2 subdomain route towards a destination area, or another routeing domain.

6.2 Subnetwork Types

There are two generic types of subnetworks supported.

a) *broadcast subnetworks*: These are multi-access subnetworks that support the capability of addressing a group of attached systems with a single NPDU, for instance ISO 8802.3 LANs.

b) *general topology subnetworks*: These are modelled as a set of point-to-point links each of which connects exactly two systems.

   There are several generic types of general topology subnetworks:

   1) *multipoint links*: These are links between more than two systems, where one system is a primary (or master) system, and the remaining systems are secondary (or slave) systems. The primary is capable of direct

communication with any of the secondaries, but the secondaries cannot communicate directly among themselves.

2) *permanent point-to-point links*: These are links that stay connected at all times (unless broken, or turned off by system management), for instance leased lines or private links.

3) *dynamically established data links (DEDs):* these are links over connection oriented facilities, for instance X.25, X.21, ISDN, or PSTN networks.

Dynamically established data links can be used in one of three ways:

(i) *static point-to-point (Static):* The call is established upon system management action and cleared only on system management action (or failure).

(ii) *dynamic connection management (DCM):* The call is established upon receipt of traffic, and brought down on timer expiration when idle.

(iii) *dynamically assigned (DA):* Uses dynamic connection management, but in addition, the address to which the call is to be established is determined dynamically.

All subnetwork types are treated by the Subnetwork Independent functions as though they were connectionless subnetworks, using the Subnetwork Dependent Convergence functions of ISO 8473 where necessary to provide a connectionless subnetwork service. The Subnetwork Dependent functions do, however, operate differently on connectionless and connection-oriented subnetworks.

6.3 Topologies

A single organisation may wish to divide its *Administrative Domain* into a number of separate *Routeing Domains.* This has certain advantages, as described in ISO/TR 9575. Furthermore, it is desirable for an intra-domain routeing protocol to aid in the operation of an inter-domain routeing protocol, where such a protocol exists for interconnecting multiple administrative domains.

In order to facilitate the construction of such multi-domain topologies, provision is made for the entering of *static* inter-domain routeing information. This information is provided by a set of *Reachable Address Prefixes* entered by System Management at the ISs which have links which cross routeing domain boundaries. The prefix indicates that any NSAPs whose NSAP address matches the prefix may be reachable via the SNPA with which the prefix is associated. Where the subnetwork to which this SNPA is connected is a general topology subnetwork supporting dynamically established data links, the prefix also has associated with it the required subnetwork addressing information, or an indication that it may be derived from the destination NSAP address (for example, an X.121 DTE address may sometimes be obtained from the IDI of the NSAP address).

The Address Prefixes are handled by the level 2 routeing algorithm in the same way as information about a level 1 area within the domain. NPDUs with a destination address matching any of the prefixes present on any Level 2 Intermediate System within the domain can therefore be relayed (using level 2 routeing) by that IS and delivered out of the domain. (It is assumed that the routeing functions of the other domain will then be able to deliver the NPDU to its destination.)

6.4 Addresses

Within a routeing domain that conforms to this standard, the Network entity titles of Intermediate systems shall be structured as described in 7.1.1.

All systems shall be able to generate and forward data PDUs containing NSAP addresses in any of the formats specified by ISO 8348/Add.2. However, NSAP addresses of End systems should be structured as described in 7.1.1 in order to take full advantage of IS–IS routeing. Within such a domain it is still possible for some End Systems to have addresses assigned which do not conform to 7.1.1, provided they meet the more general requirements of ISO 8348/Add.2, but they may require additional configuration and be subject to inferior routeing performance.

6.5 Functional Organisation

The intra-domain IS–IS routeing functions are divided into two groups

- *Subnetwork Independent Functions*
- *Subnetwork Dependent Functions*

6.5.1 Subnetwork Independent Functions

The Subnetwork Independent Functions supply full-duplex NPDU transmission between any pair of neighbour systems. They are independent of the specific subnetwork or data link service operating below them, except for recognising two generic types of subnetworks:

- **General Topology Subnetworks,** which include HDLC point-to-point, HDLC multipoint, and dynamically established data links (such as X.25, X.21, and PSTN links), and
- **Broadcast Subnetworks**, which include ISO 8802 LANs.

The following Subnetwork Independent Functions are identified

- **Routeing.** The routeing function determines NPDU paths. A path is the sequence of connected systems and links between a source ES and a destination ES.

  The combined knowledge of all the Network Layer entities of all the Intermediate systems within a routeing domain is used to ascertain the existence of a path, and route the NPDU to its destination. The routeing component at an Intermediate system has the following specific functions:

  - It extracts and interprets the routeing PCI in an NPDU.
  - It performs NPDU forwarding based on the destination address.
  - It manages the characteristics of the path. If a system or link fails on a path, it finds an alternate route.
  - It interfaces with the subnetwork dependent functions to receive reports concerning an SNPA which has become unavailable, a system that has failed, or the subsequent recovery of an SNPA or system.
  - It informs the ISO 8473 error reporting function when the forwarding function cannot relay an NPDU, for instance when the destination is unreachable or when the NPDU would have needed to be segmented and the NPDU requested “no segmentation”.
- **Congestion control**. Congestion control manages the resources used at each Intermediate system.

6.5.2 Subnetwork Dependent Functions

The subnetwork dependent functions mask the characteristics of the subnetwork or data link service from the subnetwork independent functions. These include:

- Operation of the Intermediate system functions of ISO 9542 on the particular subnetwork, in order to
  - Determine neighbour Network entity title(s) and SNPA address(es)
  - Determine the SNPA address(s) of operational Intermediate systems
- Operation of the requisite Subnetwork Dependent Convergence Function as defined in ISO 8473 and its Addenda 1 and 3, in order to perform
  - Data link initialisation
  - Hop by hop fragmentation over subnetworks with small maximum SNSDU sizes

- Call establishment and clearing on dynamically established data links

6.6 Design Goals

This International Standard supports the following design requirements. The correspondence with the goals for OSI routeing stated in ISO/TR 9575 are noted.

- **Network Layer Protocol Compatibility.** It is compatible with ISO 8473 and ISO 9542. *(See clause 7.5 of ISO/TR 9575),*
- **Simple End systems:** It requires no changes to end systems, nor any functions beyond those supplied by ISO 8473 and ISO 9542. *(See clause 7.2.1 of ISO/TR 9575),*
- **Multiple Organisations:** It allows for multiple routeing and administrative domains through the provision of static routeing information at domain boundaries. *(See clause 7.3 of ISO/TR 9575),*
- **Deliverability** It accepts and delivers NPDUs addressed to reachable destinations and rejects NPDUs addressed to destinations known to be unreachable.
- **Adaptability.** It adapts to topological changes within the routeing domain, but not to traffic changes, except potentially as indicated by local queue lengths. It splits traffic load on multiple equivalent paths. *(See clause 7.7 of ISO/TR 9575),*
- **Promptness.** The period of adaptation to topological changes in the domain is a reasonable function of the domain diameter (that is, the maximum logical distance between End Systems within the domain) and Data link speeds. *(See clause 7.4 of ISO/TR 9575),*
- **Efficiency.** It is both processing and memory efficient. It does not create excessive routeing traffic overhead. *(See clause 7.4 of ISO/TR 9575),*
- **Robustness.** It recovers from transient errors such as lost or temporarily incorrect routeing PDUs. It tolerates imprecise parameter settings. *(See clause 7.7 of ISO/TR 9575),*
- **Stability.** It stabilises in finite time to “good routes”, provided no continuous topological changes or continuous data base corruptions occur.
- **System Management control.** System Management can control many routeing functions via parameter changes, and inspect parameters, counters, and routes. It will not, however, depend on system management action for correct behaviour.
- **Simplicity.** It is sufficiently simple to permit performance tuning and failure isolation.
- **Maintainability.** It provides mechanisms to detect, isolate, and repair most common errors that may affect the routeing computation and data bases. *(See clause 7.8 of ISO/TR 9575),*
- **Heterogeneity.** It operates over a mixture of network and system types, communication technologies, and topologies. It is capable of running over a wide variety of subnetworks, including, but not limited to: ISO 8802 LANs, ISO 8208 and X.25 subnetworks, PSTN networks, and the OSI Data Link Service. *(See clause 7.1 of ISO/TR 9575),*
- **Extensibility.** It accommodates increased routeing functions, leaving earlier functions as a subset.
- **Evolution.** It allows orderly transition from algorithm to algorithm without shutting down an entire domain.
- **Deadlock Prevention.** The congestion control component prevents buffer deadlock.

- **Very Large Domains.** With hierarchical routeing, and a very large address space, domains of essentially unlimited size can be supported. *(See clause 7.2 of ISO/TR 9575),*
- **Area Partition Repair.** It permits the utilisation of level 2 paths to repair areas which become partitioned due to failing level 1 links or ISs. *(See clause 7.7 of ISO/TR 9575),*
- **Determinism.** Routes are a function only of the physical topology, and not of history. In other words, the same topology will always converge to the same set of routes.
- **Protection from Mis-delivery.** The probability of mis-delivering a NPDU, i.e. delivering it to a Transport entity in the wrong End System, is extremely low.
- **Availability.** For domain topologies with *cut set* greater than one, no single point of failure will partition the domain. *(See clause 7.7 of ISO/TR 9575),*
- **Service Classes.** The service classes of *transit delay, expense*[3]*,* and *residual error probability* of ISO 8473 are supported through the optional inclusion of multiple routeing metrics.

6.6.1 Non-Goals

The following are not within the design scope of the intra-domain IS–IS routeing protocol described in this International Standard:

- **Traffic adaptation.** It does not automatically modify routes based on global traffic load.
- **Source-destination routeing.** It does not determine routes by source as well as destination.
- **Guaranteed delivery.** It does not guarantee delivery of all offered NPDUs.
- **Level 2 Subdomain Partition Repair.** It will not utilise Level 1 paths to repair a level 2 subdomain partition. For correct operation, a connected level 2 subdomain is required.

  EDITOR’S NOTE: The above statement about “correct operation” is not strictly true, and leaves the impression that the protocol will break if the L2 subdomain becomes partitioned. This is not true — the two pieces each work correctly in isolation. The editor suggests that the second sentence above be changed to read: “For full logical connectivity to be available, a connected...”.

- **Equal treatment for all ES Implementations.** The End system poll function defined in 8.4.5 presumes that End systems have implemented the Suggested ES Configuration Timer option of ISO 9542. An End system which does not implement this option may experience a temporary loss of connectivity following certain types of topology changes on its local subnetwork.

6.7 Environmental Requirements

For correct operation of the protocol, certain guarantees are required from the local environment and the Data Link Layer.

The required local environment guarantees are:

a) Resource allocation such that the certain minimum resource guarantees can be met, including

   1) memory (for code, data, and buffers)
   2) processing;

   See 12.2.5 for specific performance levels required for conformance

---

[3] “Expense” is referred to as “cost” in ISO 8473. The latter term is not used here because of possible confusion with the more general usage of the term to indicate path cost according to any routeing metric.

b) A quota of buffers sufficient to perform routeing functions;

c) Access to a timer or notification of specific timer expiration; and

d) A very low probability of corrupting data.

The required subnetwork guarantees for point-to-point links are:

a) Provision that both source and destination systems complete start-up before PDU exchange can occur;

b) Detection of remote start-up;

c) Provision that no old PDUs be received after start-up is complete;

d) The following events are “low probability”, which means that they occur sufficiently rarely so as not to impact performance, on the order of once per thousand NPDUs

    1) Loss, duplication or corruption of NPDUs;

    2) Routeing PDU non-sequentiality; and

e) Reporting of failures and degraded subnetwork conditions.

The required subnetwork guarantees for broadcast links are:

a) Multicast capability, i.e., the ability to address a subset of all connected systems with a single PDU;

b) The following events are “low probability”, which means that they occur sufficiently rarely so as not to impact performance, on the order of once per thousand PDUs

    1) Routeing PDU non-sequentiality,

    2) Routeing PDU loss due to detected corruption; and

    3) Receiver overrun;

c) The following events are “very low probability”, which means performance will be impacted unless they are extremely rare, on the order of less than one event per four years

    1) Delivery of NPDUs with undetected data corruption; and

    2) Non-transitive connectivity, i.e. where system *A* can receive transmissions from systems *B* and *C*, but system *B* cannot receive transmissions from system *C*.

The following services are assumed to be not available from broadcast links:

a) Reporting of failures and degraded subnetwork conditions that result in NPDU loss, for instance receiver failure. The routeing functions are designed to account for these failures.

6.8 Functional Organisation of Subnetwork Independent Components

The Subnetwork Independent Functions are broken down into more specific functional components. These are described briefly in this sub-clause and in detail in clause 7. This International Standard uses a functional decomposition adapted from the model of routeing presented in clause 5.1 of ISO/TR 9575. The decomposition is not identical to that in ISO/TR 9575, since that model is more general and not specifically oriented toward a detailed description of intra-domain routeing functions such as supplied by this protocol.

The functional decomposition is shown below in figure 2.

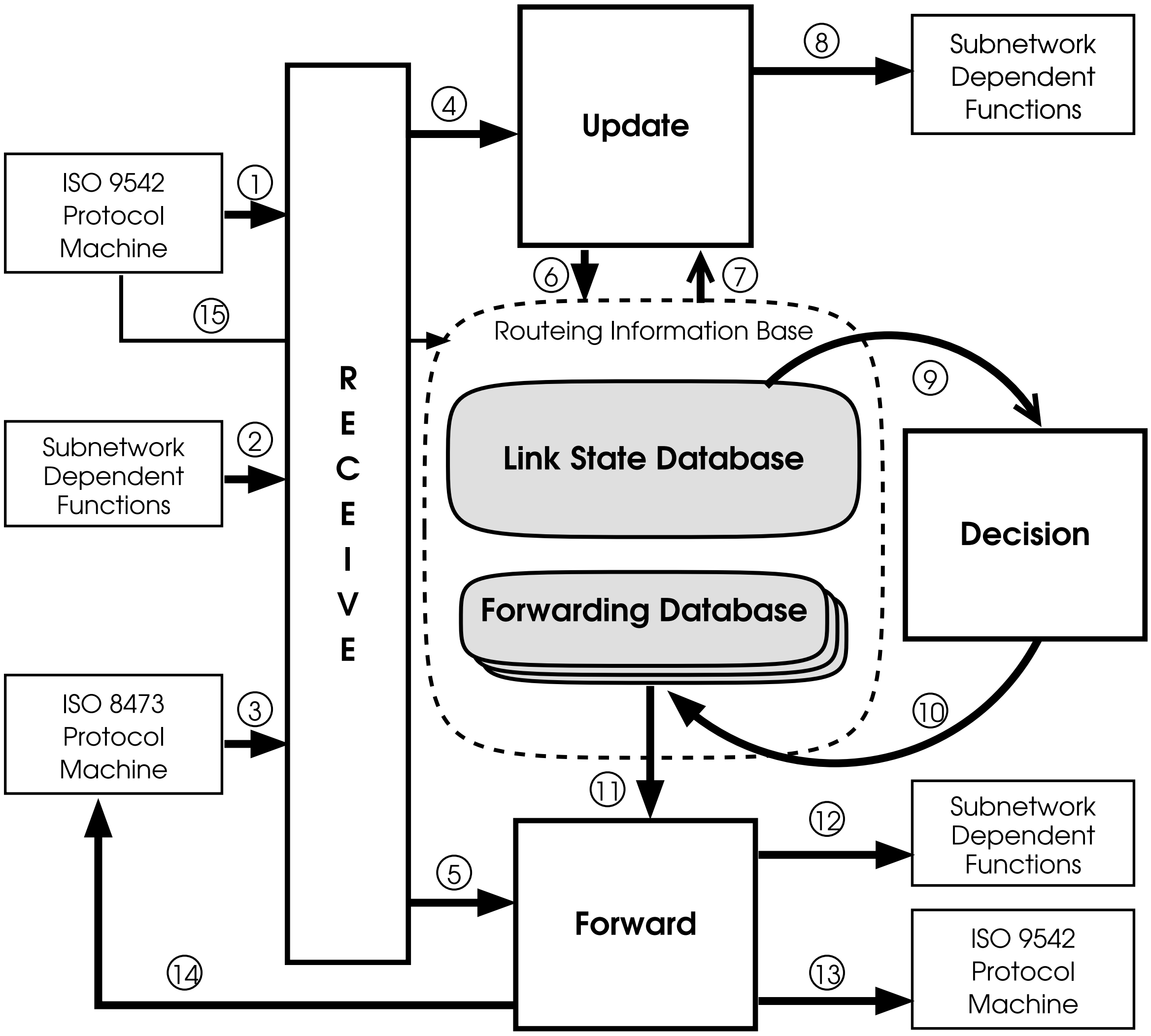


Figure 2 - Decomposition of Subnetwork Independent Functions

6.8.1 Routeing

The routeing processes are:

- *Decision Process*
- *Update Process*

  NOTE — this comprises both the *Information Collection* and *Information Distribution* components identified in ISO/TR 9575.
- *Forwarding Process*
- *Receive Process*

6.8.1.1 Decision Process

This process calculates routes to each destination in the domain. It is executed separately for level 1 and level 2 routeing, and separately within each level for each of the routeing metrics supported by the Intermediate system. It uses the *Link State Database*, which consists of information from the latest Link State PDUs from every other Intermediate system in the area, to compute shortest paths from this IS to all other systems in the area – ⑨ in figure 2. The Link State Data Base is maintained by the Update Process.

Execution of the Decision Process results in the determination of [circuit, neighbour] pairs (known as *adjacencies*), which are stored in the appropriate Forwarding Information base – ⑩ – and used by the Forwarding process as paths along which to forward NPDUs.

Several of the parameters in the routeing data base that the Decision Process uses are set by system management or are determined by the implementation. These include:

- maximum number of Intermediate and End systems within the IS's area;
- maximum number of Intermediate and End system neighbours of the IS, etc.,

so that databases can be sized appropriately. Also parameters such as

- routeing metrics for each circuit; and
- timers

can be adjusted for enhanced performance. The complete list of System Management set-able parameters is listed in clause 11.

6.8.1.2 Update Process

This process constructs, receives and propagates Link State PDUs. Each Link State PDU contains information about the identity and routeing metric values of the adjacencies of the IS that originated the Link State PDU.

The Update Process receives Link State and Sequence Numbers PDUs from the Receive Process — ④ in figure 2. It places new routeing information in the routeing information base — ⑥ and propagates routeing information to other Intermediate systems – ⑦ and ⑧ .

General characteristics of the Update Process are:

- Link State PDUs are generated as a result of topological changes, and also periodically. They may also be generated indirectly as a result of System Management actions (such as changing one of the routeing metrics for a circuit).
- Level 1 Link State PDUs are propagated to all Intermediate systems within an area, but are not propagated out of an area.
- Level 2 Link State PDUs are propagated to all Level 2 Intermediate systems in the domain.
- Link State PDUs are not propagated outside of a domain.
- The update process, through a set of System Management parameters, enforces an upper bound on the amount of routeing traffic overhead it generates.

6.8.1.3 Forwarding Process

This process supplies and manages the buffers necessary to support NPDU relaying to all destinations.

It receives, via the Receive Process, ISO 8473 PDUs to be forwarded – ⑤ in figure 2.

It performs a lookup in the appropriate[4] Forwarding Database – ⑪ – to determine the possible output adjacencies to use for forwarding to a given destination , chooses one adjacency – ⑫ —, generates error indications to ISO 8473 – ⑭ , and signals ISO 9542 to issue Redirect PDUs – ⑬ .

6.8.1.4 Receive Process

The Receive Process obtains its inputs from the following sources

- received PDUs with the NPID of Intra-Domain routeing – ② in figure 2,
- routeing information derived by the ES–IS protocol from the receipt of ISO 9542 PDUs – ① ; (ISO 9542 places its routeing information directly into the routeing information base — ⑮) and
- ISO 8473 data PDUs handed to the routeing function by the ISO 8473 protocol machine – ③ .

It then performs the appropriate actions, which may involve passing the PDU to some other function (e.g. to the Forwarding Process for forwarding – ⑤).

---

[4] The appropriate Forwarding Database is selected by choosing a routeing metric based on fields in the QoS Maintenance option of ISO 8473.

7 Subnetwork Independent Functions

This clause describes the algorithms and associated databases used by the routeing functions. The managed objects and attributes defined for System Management purposes are described in Clause 11.

The following processes and data bases are used internally by the subnetwork independent functions. Following each process or data base title, in parentheses, is the type of systems which must keep the database. The system types are “L2” (level 2 Intermediate system), and “L1” (level 1 Intermediate system). Note that a level 2 Intermediate system is also a level 1 Intermediate system in its home area, so it must keep level 1 databases as well as level 2 databases.

Processes:

- Decision Process (L2, L1)
- Update Process (L2, L1)
- Forwarding Process (L2, L1)
- Receive Process (L2, L1)

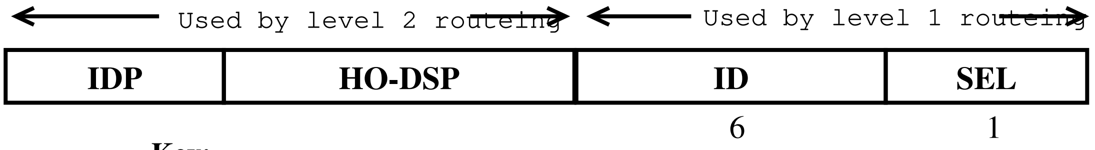
Databases:

- Level 1 Link State data base (L2, L1)
- Level 2 Link State data base (L2)
- Adjacency Database (L2, L1)
- Circuit Database (L2, L1)
- Level 1 Shortest Paths Database (L2, L1)
- Level 2 Shortest Paths Database (L2)
- Level 1 Forwarding Databases — one per routeing metric (L2, L1)
- Level 2 Forwarding Database — one per routeing metric (L2)

7.1 Addresses

7.1.1 Addresses within a routeing domain

Within a routeing domain an address is a variable length quantity conforming to the ISO NSAP address structure described in ISO 8348/Add.2. Addresses used in conjunction with this protocol are further structured as shown below in figure 4. Each field’s definition and use is defined below.



Key:

IDP	Initial Domain Part
HO-DSP	High Order Domain Specific Part
ID	System Identifier
SEL	NSAP Selector

Figure 3 - Address Structure for Intra-domain IS–IS Routeing

IDP, the Initial Domain Part of the address as defined by ISO 8348/Add.2. It consists of subfields AFI and IDI. Any of the AFI values defined by ISO 8348/Add.2 are legal NSAP addresses or Network entity titles when used in conjunction with this protocol.

HO-DSP, the high order part of the DSP as defined by ISO 8348/Add.2. The Intra-domain IS-IS protocol places no restrictions on the usage of this field, however a number of technical considerations apply to the assignment of DSP values in routeing domains conforming to ths protocol. See B.3 for more information.

The concatenation of IDP and high-order DSP is known as the *Area Address*.[5]

ID, a 6 octet *system identification*. Correct operation of the routeing protocol requires that this field be unique within an *area* for End Systems and Level 1 Intermediate systems, and unique within the routeing domain for Level 2 Intermediate systems.

NOTE — The use of an IEEE 802 MAC address, which is guaranteed to be globally unique, as the value of the ID field would ensure that the necessary area-wide uniqueness property is satisfied. If an 802 address is used it may correspond to the actual MAC address of the system on an ISO 8802 LAN, but this correspondence is neither assumed nor required by the routeing algorithms.

EDITOR\S NOTE — The current six octet length of the ID field has yet to be confirmed through member body ballot and comment. The current feeling is that this field should definitely remain of fixed size, in order to ensure that the ability of an Is to do fast relaying is not compromised. Also, the current opinion is that if the field size is changed, it may get larger (perhaps 7 or 8 octets) but not smaller. It is also possible that the length may be fixed on a per routeing domain basis instead of globally by the standard.

SELECTOR *NSAP Selector*, a 1 octet field which acts as a selector for the entity which is to receive the PDU (may be a Transport entity or the Intermediate system Network entity itself.) It is always the last octet of the address. The entire address minus the SELECTOR field is known as the *Network Entity Title* (NET) of the corresponding system.

The Area Address is the entire address minus the last 7 octets. An area address shall be unique within the entire global network addressing domain as defined in ISO 8348/Add.2.

If an address exactly matches one of the area addresses of a system, it is in the system's area, and is routed by level 1 routeing. Otherwise it is routed via level 2 routeing.

Level 2 routeing acts on address prefixes, routeing towards the longest reachable address prefix that matches the destination address.

7.1.2 Address encoding

All full NSAP addresses and Network entity titles are encoded according to the ISO *preferred binary encoding*. This applies not only to addresses contained in ISO 8473 and ISO 9542 PDUs, but also to addresses carried by this protocol's PDUs (such as LSPs and IIH PDUs). For details of the encoding of address prefixes see 7.1.4.

---

[5] An area may have more than one area address. See clause 7.2.11 for more details.

7.1.3 Permissible End System Address Formats

End systems are not required to follow the addressing convention defined above in 7.1.1. However, the following restrictions apply in order for Intermediate systems to route correctly in the presence of such End systems in a routeing domain:

a) Two End systems shall not have addresses that match all but the SELECTOR field. (Intermediate systems will not distinguish between such addresses, since the SELECTOR field is ignored for the purposes of identifying the system to which a PDU is to be delivered. It is used only within the destination system, to identify the entity to which the PDU should be delivered.)

b) An End system whose address does not match the Area Address of a neighbour Level 1 Intermediate system will not be able to use that Level 1 IS as a relay. (Level 1 Intermediate systems route solely based on the ID portion of the address, and only for addresses that match their Area Address.)

c) An End system whose address matches the Area Address of an area in a domain, shall reside in that area of that domain. (Level 1 Intermediate systems in the matching area will assume the location of the system is in that area.)

d) Two End systems residing in the same Level 1 area shall not have addresses with the same ID field.

7.1.4 Encoding of Level 2 Addresses

When a full NSAP address is encoded according to the preferred binary encoding specified in ISO 8348/Add.2, the IDI is padded with leading digits (if necessary) to obtain the maximum IDP length specified for that AFI.

A Level 2 address prefix consists of a leading sub-string of a full NSAP address, such that it matches a set of full NSAP addresses that have the same leading sub-string. However this truncation and matching is performed on the NSAP represented by the abstract syntax of the NSAP address, not on the encoded (and hence padded) form.[6]

Level 2 address prefixes are encoded in LSPs in the same way as full NSAP addresses, except when the end of the prefix falls within the IDP. In this case the prefix is directly encoded as the string of semi-octets with no padding.

7.2 The Decision Process

This process uses the database of Link State information to calculate the forwarding database(s), from which the forwarding process can know the proper next hop for each NPDU. The Level 1 Link State Database is used for calculating the Level 1 Forwarding Database(s), and the Level 2 Link State Database is used for calculating the Level 2 Forwarding Database(s).

7.2.1 Input and output

INPUT

- Link State Database – This database is a set of information from the latest Link State PDUs from all known Intermediate systems (within this area, for Level 1, or within the level 2 subdomain, for Level 2). This database is received from the Update Process.
- Notification of an Event – This is a signal from the Update Process that a change to a link has occurred somewhere in the domain.

OUTPUT

- Level 1 Forwarding Databases — one per routeing metric
- (Level 2 Intermediate systems only) Level 2 Forwarding Databases — one per routeing metric

---

[6] An example of prefix matching may be found in annex B, clause B.1.

- (Level 2 Intermediate systems only) The Level 1 Decision Process informs the Level 2 Update Process of the ID of the Level 2 Intermediate system within the area with lowest ID reachable with real level 1 links (as opposed to a virtual link consisting of a path through the level 2 subdomain)

- (Level 2 Intermediate systems only) If this Intermediate system is the “partition designated level 2 Intermediate system” in this partition, the Level 2 Decision Process informs the Level 1 Update Process of the values of the default routeing metric to and ID of the “partition designated level 2 Intermediate system” in each other partition of this area.

7.2.2 Routeing metrics

There are four routeing metrics defined, corresponding to the four possible orthogonal qualities of service defined by the QoS Maintenance field of ISO 8473. Each circuit emanating from an Intermediate system shall be assigned a value for one or more of these metrics by System management. The four metrics are as follows:

a) *Default metric:* This is a metric understood by every Intermediate system in the domain. Each circuit shall have a positive integral value assigned for this metric. The value may be associated with any objective function of the circuit, but by convention is intended to measure the *capacity* of the circuit for handling traffic, for example, its throughput in bits-per-second. Higher values indicate a lower capacity.

b) *Delay metric:* This metric measures the *transit delay* of the associated circuit. It is an optional metric, which if assigned to a circuit shall have a positive integral value. Higher values indicate a longer transit delay.

c) *Expense metric:* This metric measures the *monetary cost* of utilising the associated circuit. It is an optional metric, which if assigned to a circuit shall have a positive integral value[7]. Higher values indicate a larger monetary expense.

d) *Error metric:* This metric measures the *residual error probability* of the associated circuit. It is an optional metric, which if assigned to a circuit shall have a non-zero value. Higher values indicate a larger probability of undetected errors on the circuit.

NOTE - The decision process combines metric values by simple addition. It is important, therefore, that the values of the metrics be chosen accordingly.

Every Intermediate system shall be capable of calculating routes based on the default metric. Support of any or all of the other metrics is optional. If an Intermediate system supports the calculation of routes based on a metric, its update process may report the metric value in the LSPs for the associated circuit; otherwise, the IS shall not report the metric.

When calculating paths for one of the optional routeing metrics, the decision process only utilises LSPs with a value reported for the corresponding metric. If no value is associated with a metric for any of the IS’s circuits the system shall not calculate routes based on that metric.

NOTE - A consequence of the above is that a system reachable via the default metric may not be reachable by another metric.

See 7.4.2 for a description of how the forwarding process selects one of these metrics based on the contents of the ISO 8473 QoS Maintenance option.

7.2.3 Broadcast Subnetworks

Instead of treating a broadcast subnetwork as a fully connected topology, the broadcast subnetwork is treated as a pseudonode, with links to each attached system. Attached systems shall only report their link to the pseudonode. The

[7] The path computation algorithm utilised in this International Standard requires that all circuits be assigned a positive value for a metric. Therefore, it is not possible to represent a “free” circuit by a zero value of the expense metric. By convention, the value 1 is used to indicate a “free” circuit.

designated Intermediate system, on behalf of the pseudonode, shall construct Link State PDUs reporting the links to all the systems on the broadcast subnetwork with a zero value for each supported routeing metric[8].

The pseudonode shall be identified by the sourceID of the Designated Intermediate system, followed by a non-zero pseudonodeID assigned by the Designated Intermediate system. The pseudonodeID is locally unique to the Designated Intermediate system.

Designated Intermediate systems are determined separately for level 1 and level 2. They are known as the *LAN Level 1 Designated IS* and the *LAN Level 2 Designated IS* respectively. See Clause 8.4.4.

An Intermediate system may resign as Designated Intermediate System on a broadcast circuit either because it (or it's SNPA on the broadcast subnetwork) is being shut down or because some other Intermediate system of higher priority has taken over that function. When an Intermediate system resigns as Designated Intermediate System, it shall initiate a network wide purge of its pseudonode Link State PDU(s) by setting their Remaining Lifetime to zero and performing the actions described in 7.3.16.4. A LAN Level 1 Designated Intermediate System purges Level 1 Link State PDUs and a LAN Level 2 Designated Intermediate System purges Level 2 Link State PDUs. An Intermediate system which has resigned as both Level 1 and Level 2 Designated Intermediate System shall purge both sets of LSPs.

When an Intermediate system declares itself as designated Intermediate system and it is in possession of a Link State PDU of the same level issued by the previous Designated Intermediate System for that circuit (if any), it shall initiate a network wide purge of that (or those) Link State PDU(s) as above.

7.2.4 Links

Two Intermediate systems are not considered neighbours unless each reports the other as directly reachable over one of their SNPAs. On a Connection-oriented subnetwork (either point-to-point or general topology), the two Intermediate systems in question shall ascertain their neighbour relationship when a connection is established and hello PDUs exchanged. A malfunctioning IS might, however, report another IS to be a neighbour when in fact it is not. To detect this class of failure the decision process checks that each link reported as “up” in a LSP is so reported by both Intermediate systems. If an Intermediate system considers a link “down” it shall **not** mention the link in its Link State PDUs.

On broadcast subnetworks, this class of failure shall be detected by the designated IS, which has the responsibility to ascertain the set of Intermediate systems that can all communicate on the subnetwork. The designated IS shall include these Intermediate systems (and no others) in the Link State PDU it generates for the pseudonode representing the broadcast subnetwork.

7.2.5 Multiple LSPs for the same system

The Update process is capable of dividing a single logical LSP into a number of separate PDUs for the purpose of conserving link bandwidth and processing (see 7.3.4). The Decision Process, on the other hand, shall regard the LSP with LSP Number zero in a special way. If the LSP with LSP Number zero and remaining lifetime > 0, is not present for a particular system then the Decision Process shall not process any LSPs with non-zero LSP Number which may be stored for that system.

The following information shall be taken only from the LSP with LSP Number zero. Any values which may be present in other LSPs for that system shall be disregarded by the Decision Process.

a) The setting of the “Infinite Hippity Cost” bit.

b) The value of the Intermediate system Type field.

c) The Area Addresses option.

---

[8] They are set to zero metric values since they have already been assigned metrics by the link to the pseudonode. Assigning a non-zero value in the pseudonode LSP would have the effect of doubling the actual value.

7.2.6 Routeing Algorithm Overview

The routeing algorithm used by the Decision Process is a *shortest path first (SPF)* algorithm. Instances of the algorithm are run independently and concurrently by all Intermediate systems in a routeing domain. Intra-Domain routeing of a PDU occurs on a hop-by-hop basis: that is, the algorithm determines only the next hop, not the complete path, that a data PDU will take to reach its destination. To guarantee correct and consistent route computation by every Intermediate system in a routeing domain, this International Standard depends on the following properties:

a) All Intermediate systems in the routeing domain converge to using identical topology information; and

b) Each Intermediate system in the routeing domain generates the same set of routes from the same input topology and set of metrics.

The first property is necessary in order to prevent inconsistent, potentially looping paths. The second property is necessary to meet the goal of determinism stated in 6.6.

A system executes the SPF algorithm to find a set of paths to a destination system in the routeing domain. The set may consist of:

a) a single path of minimum cost: these are termed *minimum cost paths*;

b) a set of paths of equal minimum cost: these are termed *equal minimum cost paths*; or

c) a set of paths which will get a PDU closer to its destination than the local system: these are called *downstream paths.*

Paths which do not meet the above conditions are illegal and shall not be used.

The Decision Process, in determining its paths, also ascertains the identity of the adjacency which lies on the first hop to the destination on each path. These adjacencies are used to form the Forwarding Database, which the forwarding process uses for relaying PDUs.

Separate route calculations are made for each pairing of a level in the routeing hierarchy (i.e. L1 and L2) with a supported routeing metric. Since there are four routeing metrics and two levels some systems may execute multiple instances of the SPF algorithm. For example,

- if an IS is a L2 Intermediate system which supports all four metrics and computes minimum cost routes for all metrics, it would execute the SPF calculation eight times.

- if an IS is a L1 Intermediate system which supports all four metrics, and additionally computes downstream routes, it would execute the algorithm 4×(number of neighbours + 1) times.

Any implementation of an SPF algorithm meeting both the static and dynamic conformance requirements of clause 12 of this International Standard may be used. Recommended implementations are described in detail in Annex C.

7.2.7 Removal of Excess Adjacencies

When there are more than maximumPathSplits minimal cost paths to a destination, this set shall be pruned until only maximumPathSplits remain. The Intermediate system shall discriminate based upon:

NOTE - The precise precedence among the paths is specified in order to meet the goal of determinism define in 6.6.

- **adjacency type:** End system or level 2 reachable address prefix adjacencies are retained in preference to other adjacencies

- **neighbour ID:** where two or more adjacencies are of the same type, an adjacency with a lower neighbour ID is retained in preference to an adjacency with a higher neighbour id.

- **circuit ID:** where two or more adjacencies are of the same type, and same neighbour ID, an adjacency with a lower *circuit ID* is retained in preference to an adjacency with a higher *circuit ID*, where *circuit ID* is the value of:
  - ptPtCircuitID for non-broadcast circuits,
  - L1CircuitID for broadcast circuits when running the Level 1 Decision Process, and
  - L2CircuitID for broadcast circuits when running the Level 2 Decision Process.
- **LANAddress:** where two or more adjacencies are of the same type, same neighbour ID, and same circuit ID (e.g. a system with multiple LAN adapters on the same circuit) an adjacency with a lower LANAddress is retained in preference to an adjacency with a higher LANAddress.

7.2.8 Robustness Checks

The Decision Process shall not utilise a link between two Intermediate Systems unless both ISs report the link.

NOTE - the check is not applicable to links to an End System.

Reporting the link indicates that it has a defined value for at least the default routeing metric. It is permissible for two endpoints to report different defined values of the same metric for the same link. In this case, routes may be asymmetric.

7.2.9 Construction of a Forwarding Database

The information that is needed in the forwarding database for routeing metric *k* is the set of adjacencies for each system *N*.

7.2.9.1 Identification of Nearest Level 2 IS by a Level 1 IS

Level 1 Intermediate systems need one additional piece of information per routeing metric: the next hop to the nearest level 2 Intermediate system according to that routeing metric. A level 1 IS shall ascertain the set, *R*, of "attached" level 2 Intermediate system(s) for metric *k* such that the total cost to *R* for metric *k* is minimal.

If there are more adjacencies in this set than maximumPathSplits, then the IS shall remove excess adjacencies as described in 7.2.7.

7.2.9.2 Setting the Attached Flag in Level 2 Intermediate Systems

If a level 2 Intermediate system discovers, after computing the level 2 routes for metric *k*, that it cannot reach any other areas using that metric, it shall:

- set AttachedFlag for metric *k* to False;
- regenerate its Level 1 LSP with LSP number zero; and
- compute the "nearest level 2 Intermediate system" for metric *k* for insertion in the appropriate forwarding database, according to the algorithm described in 7.2.9.1 for level 1 Intermediate systems.

NOTE - AttachedFlag for each metric *k* is examined by the Update Process, so that it will report the value in the ATT field of its Link State PDUs.

If a level 2 Intermediate system discovers, after computing the level 2 routes for metric *k*, that it can reach at least one other area using that metric, it shall

- set AttachedFlag for metric *k* to True;
- regenerate its Level 1 LSP with LSP number zero; and

- set the level 1 forwarding database entry for metric *k* which corresponds to “nearest level 2 Intermediate system” to SELF.

7.2.10 Information for Repairing Partitioned Areas

An area may become partitioned as a result of failure of one or more links in the area. However, if each of the partitions has a connection to the level 2 subdomain, it is possible to repair the partition via the level 2 subdomain, provided that the level 2 subdomain itself is not partitioned. This is illustrated in Figure 4.

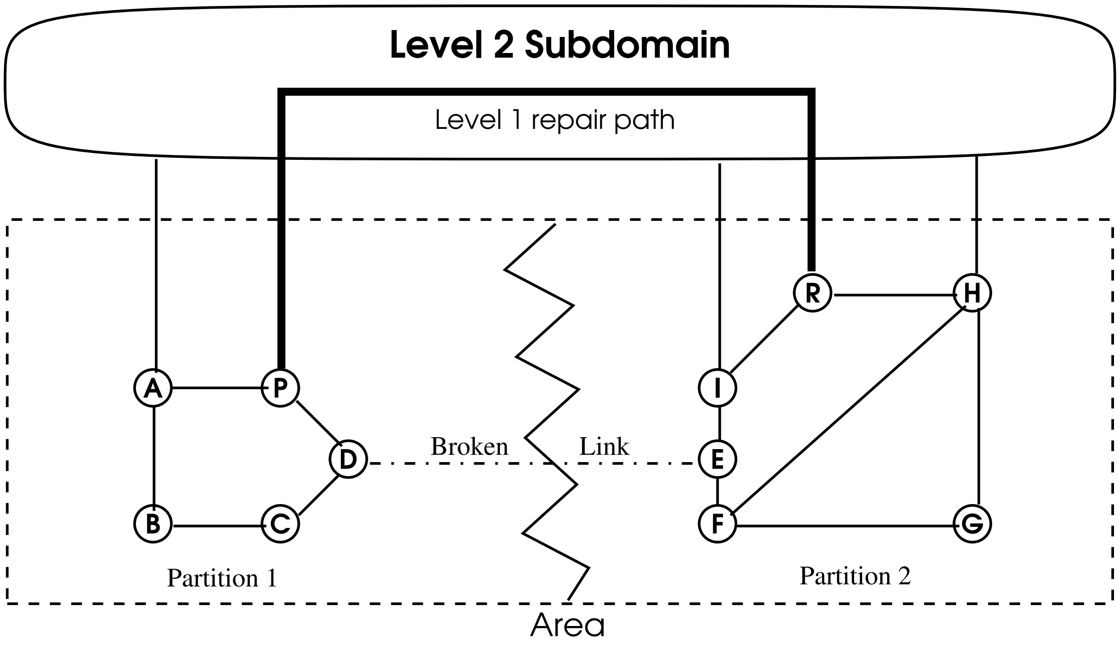


Figure 4 - Repair of partitioned level 1 area

All the systems A – I, R and P are in the same area *n*. When the link between D and E is broken, the area becomes partitioned. Within each of the partitions the *Partition Designated Level 2 Intermediate system* is selected from among the level 2 Intermediate systems in that partition. In the case of partition 1 this is P, and in the case of partition 2 this is R. The level 1 repair path is then established between between these two level 2 Intermediate systems. Note that the repaired link is now between P and R, not between D and E.

The Partition Designated Level 2 Intermediate Systems repair the partition by forwarding NPDUs destined for other partitions of the area through the level 2 subdomain. They do this by acting in their capacity as Level 1 Intermediate Systems and advertising in their Level 1 LSPs adjacencies to each Partition Designated Level 2 Intermediate System in the area. This adjacency is known as a “Virtual Adjacency” or “Virtual Link”. Thus other Level 1 Intermediate Systems in a partition calculate paths to the other partitions through the Partition Designated Level 2 Intermediate System. A Partition Designated Level 2 Intermediate System forwards the Level 1 NPDUs through the level 2 subdomain by encapsulating them in 8473 Data NPDUs with its Virtual Network Entity Title as the source NSAP and the “adjacent” Partition Designated Level 2 Intermediate System’s Virtual Network Entity Title as the destination NSAP. The following subclauses describe this in more detail.

7.2.10.1 Partition Detection and Virtual Level 1 Link Creation

Partitions of a Level 1 area are detected by the Level 2 Intermediate System(s) operating within the area. In order to participate in the partition repair process, these Level 2 Intermediate systems must also act as Level 1 Intermediate systems in the area. A partition of a given area exists whenever two or more Level 2 ISs located in that area are reported in the L2 LSPs as being a ’Partition Designated Level 2 IS’. Conversely, when only one Level 2 IS in an area is reported as being the ’Partition Designated Level 2 IS’, then that area is not partitioned. Partition repair is accomplished by the Partition Designated Level 2 IS. The election of the Partition Designated Level 2 IS as described in the next subsection must be done before the detection and repair process can begin.

In order to repair a partition of a Level 1 area, the Partition designated Level 2 IS creates a "Virtual Network Entity" to represent the partition. The Network Entity Title for this virtual network entity shall be constructed from the first listed Area Address from its Level 2 Link State PDU, and the ID of the Partition Designated Level 2 IS. The IS shall also construct a virtual link (represented by a new Virtual Adjacency managed object) to each Partition Designated Level 2 IS in the area, with the NET of the partition recorded in the Identifier attribute. The virtual links are the repair paths for the partition. They are reported by the Partition Designated Level 2 IS into the entire Level 1 area by adding the ID of each "adjacent" Partition Designated Level 2 IS to the "Intermediate System Neighbors" part of its Level 1 Link State PDU. The "Virtual Flag" shall be set TRUE for these Intermediate System neighbors. The metric value for this virtual link shall be the default metric value d(N) obtained from this system's Level 2 PATHS database, where N is the "adjacent" Partition Designated Level 2 IS via the Level 2 subdomain.

An Intermediate System which operates as the Partition Designated Level 2 Intermediate System shall perform the following steps after completing the Level 2 shortest path computation in order to detect partitions in the Level 1 area and create repair paths:

a) Examine Level 2 Link State PDUs of all Level 2 Intermediate systems. Search Area Addresses for any address that matches any of the addresses in partitionAreaAddresses. If a match is found, and the Partition Designated Level 2 Intermediate system's ID does not equal this system's ID, then inform the level 1 update process at this system of the identity of the Partition Designated Level 2 Intermediate system, together with the path cost for the default routeing metric to that Intermediate system.

b) Continue examining Level 2 LSPs until all Partition Designated Level 2 Intermediate systems in other partitions of this area are found, and inform the Level 1 Update Process of all of the other Partition Designated Level 2 Intermediate systems in other partitions of this area, so that

    1) Level 1 Link State PDUs can be propagated to all other Partition designated level 2 Intermediate systems for this area (via the level 2 subdomain).

    2) All the Partition Designated Level 2 Intermediate systems for other partitions of this area can be reported as adjacencies in this system's Level 1 Link State PDUs.

If a partition has healed, the IS shall destroy the associated virtual network entity and virtual link by deleting the Virtual Adjacency. The Partition Designated Level 2 IS detects a healed partition when another Partition Designated Level 2 IS listed as a virtual link in its Level 1 Link State PDU was not found after running the partition detection and virtual link creation algorithm described above.

If such a Virtual Adjacency is created or destroyed, the IS shall generate a "PartitionVirtualLinkChange" event.

7.2.10.2 Election of Partition Designated Level 2 Intermediate System

The Partition Designated Level 2 IS is a Level 2 IS which:

- reports itself as "attached" by the default metric in its LSPs;
- reports itself as implementing the partiton repair option;
- operates as a Level 1 IS in the area;
- is reachable via Level 1 routeing without traversing any virtual links; and
- has the lowest ID

The election of the Partition Designated Level 2 IS is performed by running the decision process algorithm after the Level 1 decision process has finished, and before the Level 2 decision process to determine Level 2 paths is executed.

In order to guarantee that the correct Partition Designated Level 2 IS is elected, the decision process is run using only the Level 1 LSPs for the area, and by examining only the Intermediate System Neighbors whose Virtual Flag is FALSE. The results of this decision process is a set of all the Level 1 Intermediate Systems in the area that can be reached via Level 1, non-virtual link routeing. From this set, the Partition Designated Level 2 IS is selected by choosing the IS for which

- IS TYPE (as reported in the Level 1 LSP) is “Level 2 Intermediate System”;
- ATT indicates “attatched” by the default metric;
- P indicates support for the partiton repair option; and
- ID is the lowest among the subset of attached Level 2 Intermediate Systems.

7.2.10.3 Computation of Partition Area Addresses

A Level 2 Intermediate System shall compute the set of partitionAreaAddresses, which is the union of all manualAreaAddresses as reported in the Level 1 Link State PDUs of all Level 2 Intermediate systems reachable in the partition by the traversal of non-virtual links. If more than MaximumAreaAddresses are present, the Intermediate system shall retain only those areas with numerically lowest Area Address (as described in 7.2.11.1). If one of the local system’s ManualAreaAddresses is so rejected the event “Manual Address Dropped From Area” shall be generated.

7.2.10.4 Encapsulation of NPDUs Across the Virtual Link

All NPDUs sent over virtual links shall be encapsulated as ISO 8473 Data NPDUs. The encapsulating Data NPDU shall contain the Virtual Network Entity Title of the Partition Designated Level 2 IS that is forwarding the NPDU over the virtual link in the Source Address field, and the Virtual NET of the “adjacent” Partition Designated Level 2 IS in the Destination Address field. The SELECTOR field in both NSAPs shall contain the “IS-IS routeing” selector value. The QoS Maintenence field of the outer PDU shall be set to indicate forwarding via the default routeing metric (see table 1 on page 44).

For Data and Error Report NPDUs the Segmentation Permitted and Error Report flags and the Lifetime parameter of the outer NPDU shall be copied from the inner NPDU. When the inner NPDU is decapsulated, its lifetime parameter shall be decremented by the difference between the existing value of the parameter and the value of the Lifetime parameter in the outer NPDU.

For LSPs and SNPs the Segmentation Permitted flag shall be set to “True” and the Error Report flag shall be set to “False”. The Lifetime Parameter shall be set to 255. When an inner LSP is decapsulated, its remaining lifetime shall be decremented by half the difference between 255 and the value of the Lifetime parameter in the outer NPDU.

Data NPDUs shall not be fragmented before encapsulation, unless the total length of the Data NPDU (including header) exceeds 65535 octets. In that case, the original Data NPDU shall first be fragmented, then encapsulated. In all cases, the encapsulated Data NPDU may need to be fragmented by ISO 8473 before transmission in which case it must be reassembled and decapsulated by the destination Partition Designated Level 2 IS. The encapsulation is further described as part of the forwarding process in clause 7.4.3.2. The decapsulation is described as part of the Receive process in clause 7.4.4.

7.2.11 Computation of Area Addresses

A Level 1 or Level 2 Intermediate System shall compute the values of areaAddresses (the set of Area Addresses for this Level 1 area), by forming the union of the sets of manualAreaAddresses reported in the Area Addresses field of all Level 1 LSPs with LSP number zero in the local Intermediate system’s link state database.

NOTE - This includes all source systems, whether currently reachable or not. It also includes the local Intermediate system’s own Level 1 LSP with LSP number zero.

NOTE - There is no requirement for this set to be updated immediately on each change to the database contents. It is permitted to defer the computation until the next running of the Decision Process.

If more than MaximumAreaAddresses are present, the Intermediate system shall retain only those areas with numerically lowest Area Address (as described in 7.2.11.1). If one of the local system's ManualAreaAddresses is rejected the event "Manual Address Dropped From Area" shall be generated.

7.2.11.1 Comparison of Area Addresses

Numerical comparison of Area Addresses shall be performed on the encoded form of the address (i.e. with any padding of the IDI having been performed) by padding the shorter Area Address with trailing zeros to the length of the longer Area Address and then performing a numerical comparison in which the AFI is most significant.

7.3 The Update Process

The Update Process is responsible for generating and propagating Link State information reliably throughout the routeing domain.

The Link State information is used by the Decision Process to calculate routes.

7.3.1 Input and Output

INPUT

- Adjacency Database – maintained by the Subnetwork Dependent Functions
- Reachable Address managed objects - maintained by Network Management
- Notification of Adjacency Database Change – notification by the Subnetwork Dependent Functions that an adjacency has come up, gone down, or changed cost. (Circuit up, Circuit down, Adjacency Up, Adjacency Down, and Cost change events)
- AttachedFlag – (level 2 Intermediate systems only), a flag computed by the Level 2 Decision Process indicating whether this system can reach (via level 2 routeing) other areas
- Link State PDUs – The Receive Process passes Link State PDUs to the Update Process, along with an indication of which adjacency it was received on.
- Sequence Numbers PDUs – The Receive Process passes Sequence Numbers PDUs to the Update Process, along with an indication of which adjacency it was received on.
- Other Partitions – The Level 2 Decision Process makes available (to the Level 1 Update Process on a Level 2 Intermediate system) a list of ⟨Partition Designated Level 2 Intermediate system, Level 2 default metric value⟩ pairs, for other partitions of this area.

OUTPUT

- Link State Database
- Signal to the Decision Process of an event, which is either the receipt of a Link State PDU with different information from the stored one, or the purging of a Link State PDU from the database. The reception of a Link State PDU which has a different sequence number or Remaining Lifetime from one already stored in the database, but has an identical variable length portion, shall not cause such an event.

NOTE - An implementation may compare the checksum of the stored Link State PDU, modified according to the change in sequence number, with the checksum of the received Link State PDU. If they differ, it may assume that the variable length portions are different and an event signalled to the Decision Process. However, if the checksums are the same, an octet for octet comparison must be made in order to determine whether or not to signal the event.

7.3.2 Generation of Local Link State Information

The Update Process is responsible for constructing a set of Link State PDUs. The purpose of these Link State PDUs is to inform all the other Intermediate systems (in the area, in the case of Level 1, or in the Level 2 subdomain, in the case of Level 2), of the state of the links between the Intermediate system that generated the PDUs and its neighbours.

The Update Process in an Intermediate system shall generate one or more new Link State PDUs under the following circumstances:

a) upon timer expiration;

b) when notified by the Subnetwork Dependent Functions of an Adjacency Database Change;

c) when a change to some Network Management characteristic would cause the information in the LSP to change (for example, a change in manualAreaAddresses).

7.3.3 Use of Manual Routeing Information

Manual routeing information is routeing information entered by system management. It may be specified in two forms.

a) *Manual Adjacencies*

b) *Reachable Addresses*

These are described in the following sub-clauses.

7.3.3.1 Manual Adjacencies

An End system adjacency may be created by System Management. Such an adjacency is termed a *manual End system adjacency.* In order to create a manual End system adjacency, system managements shall specify:

a) the (set of) system IDs reachable over that adjacency; and

b) the corresponding SNPA Address.

These adjacencies shall appear as adjacencies with type “Manual”, adjacencyType “End system” and state “Up”. Such adjacencies provide input to the Update Process in a similar way to adjacencies created through the operation of ISO 9542. When the state changes to “Up” the adjacency information is included in the Intermediate System’s own Level 1 LSPs.

NOTE - Manual End system adjacencies shall not be included in a Level 1 LSPs issued on behalf of a pseudonode, since that would presuppose that all Intermediate systems on a broadcast subnetwork had the same set of manual adjacencies as defined for this circuit.

7.3.3.2 Reachable Addresses

A Level 2 Intermediate system may have a number of Reachable Address managed objects created by System management. When a Reachable Address is in state “On” and its parent Circuit is also in state “On”, the name and each of its defined routeing metrics shall be included in Level 2 LSPs generated by this system.

7.3.4 Multiple LSPs

Because a Link State PDU is limited in size to ReceiveLSPBufferSize, it may not be possible to include information about all of a system’s neighbours in a single LSP. In such cases, a system may use multiple LSPs to convey this information. Each LSP in the set carries the same sourceID field (see clause 9), but sets its own LSP Number field individually. Each of the several LSPs is handled independently by the Update Process, thus allowing distribution of topology updates to be pipelined. However, the Decision Process recognises that they all pertain to a common originating system because they all use the same sourceID.

NOTE - Even if the amount of information is small enough to fit in a single LSP, a system may optionally choose to use several LSPs to convey it; use of a single LSP in this situation is not mandatory.

The maximum sized Level 1 or Level 2 LSP which may be generated by a system is controlled by the values of the management parameters originatingL1LSPBufferSize or originatingL2LSPBufferSize respectively.

NOTE - These parameters should be set consistently by system management. If this is not done, some adjacencies will fail to initialise.

The IS shall treat the LSP with LSP Number zero in a special way, as follows:

a) The following fields are meaningful to the decision process only when they are present in the LSP with LSP Number zero:

   1) The setting of the “Infinite Hippity Cost” bit.
   2) The value of the Intermediate System Type field.
   3) The Area Addresses option. (This is only present in the LSP with LSP Number zero, see below).

b) When the values of any of the above items are changed, an Intermediate System shall re-issue the LSP with LSP Number zero, to inform other Intermediate Systems of the change. Other LSPs need not be reissued.

Once a particular adjacency has been assigned to a particular LSP Number, it is desirable that it not be moved to another LSP Number. This is because moving an adjacency from one LSP to another can cause temporary loss of connectivity to that system. This can occur if the new version of the LSP which originally contained information about the adjacency (which now does not contain that information) is propagated before the new version of the other LSP (which now contains the information about the adjacency). In order to minimise the impact of this, the following restrictions are placed on the assignment of information to LSPs.

a) The Area Addresses option field shall occur only in the LSP with LSP Number zero.

b) Intermediate System Neighbours options shall occur after the Area Addresses option and before any End system (or in the case of Level 2, Prefix) Neighbours options.

c) End system (or Prefix) Neighbour options (if any) shall occur after any Area Address or Intermediate System Neighbour options.

   NOTE — In this context, “after” means at a higher octet number from the start of the same LSP or in an LSP with a higher LSP Number.

   NOTE — An implementation is recommended to ensure that the number of LSPs generated for a particular system is within approximately 10% of the optimal number which would be required if all LSPs were densely packed with neighbour options. Where possible this should be accomplished by re-using space in LSPs with a lower LSP Number for new adjacencies. If it is necessary to move an adjacency from one LSP to another, the SRMflags (see clause 7.3.15) for the two new LSPs shall be set as an atomic action.[9]

When some event requires changing the LSP information for a system, the system shall reissue that (or those) LSPs which would have different contents. It is not required to reissue the unchanged LSPs. Thus a single End system adjacency change only requires the reissuing of the LSP containing the End system Neighbours option referring to that adjacency. The parameters maximumLSPGenerationInterval and minimumLSPGenerationInterval shall apply to each LSP individually.

---

[9] If the two SRMflags are not set atomically, a race condition will exist in which one of the two LSPs may be propagated quickly, while the other waits for an entire propagation cycle. If this occurs, adjacencies will be falsely eliminated from the topology and routes may become unstable for period of time potentially as large as maximumLSPGeneratonInterval.

7.3.5 Periodic LSP Generation

The Update Process shall periodically re-generate and propagate on every circuit with an IS adjacency of the appropriate level (by setting SRMflag on each circuit), all the LSPs (Level 1 and/or Level 2) for the local system and any pseudonodes for which it is responsible. The Intermediate system shall re-generate each LSP at intervals of at most MaximumLSPGenerationInterval seconds, with jitter applied as described in 10.1.

These LSPs may all be generated on expiration of a single timer or alternatively separate timers may be kept for each LSP Number and the individual LSP generated on expiration of this timer.

7.3.6 Event Driven LSP Generation

In addition to the periodic generation of LSPs, an Intermediate system shall generate an LSP when an event occurs which would cause the information content to change. The following events may cause such a change.

- an Adjacency or Circuit Up/Down event

- a change in Circuit metric

- a change in Reachable Address metric

- a change in ManualAreaAddresses

- a change in SystemID

- a change in Designated Intermediate System status

- a change in the waiting status

When such an event occurs the IS shall re-generate changed LSP(s) with a new sequence number. If the event necessitated the generation of an LSP which had not previously been generated (for example, an adjacency “Up” event for an adjacency which could not be accommodated in an existing LSP), the sequence number shall be set to one. The IS shall then propagate the LSP(s) on every circuit by setting SRMflag for each circuit. The timer maximumLSPGenerationInterval shall **not** be reset.

There is a hold-down timer (minimumLSPGenerationInterval) on the generation of each individual LSP.

7.3.7 Generation of Level 1 LSPs (non-pseudonode)

The Level 1 Link State PDU not generated on behalf of a pseudonode contains the following information in its variable length fields.

- In the *Area Addresses* option— the set of manualAreaAddresses for this Intermediate System.

- In the *Intermediate System Neighbours* option — the set of 7-octet Intermediate system IDs of neighbouring Intermediate systems formed from:

  - The set of neighbourSystemIDs with an appended zero octet (indicating non-pseudonode) from adjacencies in the state “Up”, on circuits of type “Point-Point”, “In” or “Out”, with

    x adjacencyType “L1 Intermediate System”

    x adjacencyType “L2 Intermediate System” and L2OnlyMode “False”.

    The metrics shall be set to the values of Level 1 metric_k of the circuit for each supported routeing metric.

  - The set of L1CircuitIDs for all circuits of type “Broadcast”. (i.e. the neighbouring pseudonode IDs) .

    The metrics shall be set to the values of Level 1 metric_k of the circuit for each supported routeing metric.

- The set of IDs with an appended zero octet derived from the Network Entity Titles of all Virtual Adjacencies of this IS. (Note that the Virtual Flag is set when encoding these entries in the LSP — see 7.2.10.)

  The default metric shall be set to the total cost to the virtual NET for the default routeing metric. The remaining metircs shall be set to the value indicating *unsupported*.

- In the *End System Neighbours* option — the set of 6-octet IDs of neighbouring End systems formed from:
  - The SystemID of the Intermediate System itself, with a value of zero for all supported metrics.
  - The set of endSystemIDs from all adjacencies with type "Auto-configured", in state "Up", on circuits of type "Point-to-Point", "In" or "Out", with adjacencyType "End system".

    The metrics shall be set to the values of Level 1 metric$_k$ of the circuit for each supported routeing metric.
  - The set of endSystemIDs from all adjacencies with type "Manual" in state "Up", on all circuits.

    The metrics shall be set to the values of Level 1 metric$_k$ of the circuit for each supported routeing metric.

7.3.8 Generation of Level 1 pseudonode LSPs

An IS shall generate a Level 1 pseudonode Link State PDU for each circuit for which this Intermediate System is the Level 1 LAN Designated Intermediate System. The LSP shall specify the following information in its variable length fields. In all cases a value of zero shall be used for all supported routeing metrics

- The Area Addresses option is not present.

  Note - This information is not required since the set of Area Addresses for the node issueing the pseudonode LSP will already have been made available via its own non-pseudonode LSP.

- In the Intermediate System Neighbours option — the set of 7-octet Intermediate System IDs of neighbouring Intermediate Systems on the circuit for which this pseudonode LSP is being generated formed from:
  - The Designated Intermediate System's own SystemID with an appended zero octet (indicating non-pseudonode).
  - The set of neighbourSystemIDs with an appended zero octet (indicating non-pseudonode) from adjacencies on this circuit in the state "Up", with adjacencyType "L1 Intermediate System".
- In the End system Neighbours option — the set of 6 octet IDs of neighbouring End systems formed from:
  - The set of endsystemIDs from all adjacencies with type "Auto-configured", in state "Up", on the circuit for which this pseudonode is being generated, with adjacencyType "End system".

7.3.9 Generation of Level 2 LSPs (non-pseudonode)

The Level 2 Link State PDU not generated on behalf of a pseudonode contains the following information in its variable length fields:

- In the Area Addresses option — the set of areaAddresses for this Intermediate system computed as described in 7.2.11.
- In the Partition Designated Level 2 IS option — the ID of the Partition Designated Level 2 Intermediate System for the partition.
- In the Intermediate System Neighbours option — the set of 7-octet Intermediate system IDs of neighbouring Intermediate systems formed from:
  - The set of neighbourSystemIDs with an appended zero octet (indicating non-pseudonode) from adjacencies in the state "Up", on circuits of type "Point-to-Point", "In" or "Out", with adjacencyType "L2 Intermediate System".

  - The set of L2CircuitIDs for all circuits of type “Broadcast”. (i.e. the neighbouring pseudonode IDs)

    The metrics shall be set to the values of Level 2 metric_k of the circuit for each supported routeing metric.

- In the Prefix Neighbours option — the set of variable length prefixes formed from—
  - The set of names of all Reachable Address managed objects in state “On”, on all circuits in state “On”.

    The metrics shall be set to the values of Level 2 metric_k for the reachable address.

7.3.10 Generation of Level 2 pseudonode LSPs

A Level 2 pseudonode Link State PDU is generated for each circuit for which this Intermediate System is the Level 2 LAN Designated Intermediate System and contains the following information in its variable length fields. In all cases a value of zero shall be used for all supported routeing metrics.

- The Area Addresses option is not present.

  Note - This information is not required since the set of Area Addresses for the node issueing the pseudonode LSP will already have been made available via its own non-pseudonode LSP.

- In the Intermediate System Neighbours option — the set of 7 octet Intermediate System IDs of neighbouring Intermediate Systems on the circuit for which this pseudonode LSP is being generated formed from:
  - The Designated Intermediate System’s own SystemID with an appended zero octet (indicating non-pseudonode).
  - The set of neighbourSystemIDs with an appended zero octet (indicating non-pseudonode) from adjacencies on this circuit in the state “Up” with adjacencyType “L2 Intermediate System”.
- The prefix neighbours option is not present.

7.3.11 Generation of the Checksum

This International Standard makes use of the checksum function defined in ISO 8473.

The source IS shall compute the LSP Checksum when the LSP is generated. The checksum shall never be modified by any other system. The checksum allows the detection of memory corruptions and thus prevents both the use of incorrect routeing information and its further propagation by the Update Process.

The checksum shall be computed over all fields in the LSP which appear after the Remaining Lifetime field. This field (and those appearing before it) are excluded so that the LSP may be aged by systems without requiring re-computation.

As an additional precaution against hardware failure, when the source computes the Checksum, it shall start with the two checksum variables (C0 and C1) initialised to what they would be after computing for the SystemID portion (i.e. the first 6 octets) of its SOURCE ID. (This value is computed and stored when the Network entity is enabled and whenever SystemID changes.) The IS shall then resume Checksum computation on the contents of the PDU after the first 6 octets of the SOURCE ID field.

NOTE - All Checksum calculations on the LSP are performed treating the SOURCE ID field as the first octet. This procedure prevents the source from accidentally sending out Link State PDUs with some other system’s ID as source.

7.3.12 Initiating Transmission

The IS shall store the generated Link State PDU in the Link State Database, overwriting any previous Link State PDU with the same LSP Number generated by this system. The IS shall then set all SRMFlags for that Link State PDU, indicating it is to be propagated on all circuits with Intermediate System adjacencies.

An Intermediate System shall ensure (by reserving resources or otherwise) that it will always be able to store and internalise its own (and its own pseudonode) LSPs.

7.3.13 Preservation of order

When an existing Link State PDU is re-transmitted (with the same or a different sequence number), but with the same information content (i.e. the variable length part) as a result of there having been no changes in the local topology databases, the order of the information in the variable length part shall be the same as that in the previously transmitted LSP.

> NOTE - If a sequence of changes result in the state of the database returning to some previous value, there is no requirement to preserve the ordering. It is only required when there have been no changes whatever. This allows the receiver to detect that there has been no change in the information content by performing an octet for octet comparison of the variable length part, and hence not re-run the decision process.

7.3.14 Propagation of LSPs

The update process is responsible for propagating Link State PDUs throughout the domain (or in the case of Level 1, throughout the area).

The basic mechanism is flooding, in which each Intermediate system propagates to all its neighbour Intermediate systems except that neighbour from which it received the PDU. Duplicates are detected and dropped.

Link state PDUs are received from the Receive Process. The maximum size control PDU (Link State PDU or Sequence Numbers PDU) which a system expects to receive shall be ReceiveLSPBufferSize octets. (i.e. the Update process must provide buffers of at least this size for the reception, storage and forwarding of received Link State PDUs and Sequence Numbers PDUs.) If a control PDU larger than this size is received, it shall be treated as if it had an invalid checksum (i.e. ignored by the Update Process and a “Corrupted LSP Received” event generated).

Upon receipt of a Link State PDU the Update Process shall perform the following functions:

a) Level 2 Link State PDUs shall be propagated on circuits which have at least one Level 2 adjacency.

b) Level 1 Link State PDUs shall be propagated on circuits which have at least one Level 1 adjacency or at least one Level 2 adjacency not marked “Level 2 only”.

c) When propagating a Level 1 Link State PDU on a broadcast subnetwork, the IS shall transmit to the multi-destination subnetwork address “All-L1-Intermediate systems”.

d) When propagating a Level 2 Link State PDU on a broadcast subnetwork, the IS shall transmit to the multi-destination subnetwork address “All-L2-Intermediate systems”.

> NOTE — When propagating a Link State PDU on a general topology subnetwork the Data Link Address is unambiguous (because Link State PDUs are not propagated across Dynamically Assigned circuits).

e) An Intermediate system receiving a Link State PDU with an incorrect LSP Checksum or with an invalid PDU syntax shall

   1) log a circuit event, “Corrupted Link State PDU Received”,

   2) overwrite the Checksum and Remaining Lifetime with 0, and

   3) treat the Link State PDU as though its Remaining Lifetime had expired (see 7.3.16.4.)

f) A Intermediate system receiving a Link State PDU which is new (as identified in 7.3.16) shall

   1) store the Link State PDU into Link State database, and

   2) mark it as needing to be propagated upon all circuits except that upon which it was received.

g) When a Intermediate system receives a Link State PDU from source *S*, which it considers older than the one stored in the database for *S*, it shall set the SRM flag for *S*'s Link State PDU associated with the circuit from which the older Link State PDU was received. This indicates that the stored Link State PDU needs to be sent on the link from which the older one was received.

h) When a system receives a Link State PDU which is the same (not newer or older) as the one stored, the Intermediate system shall

   1) acknowledge it if necessary, as described in 7.3.17, "Making the Update Reliable", and

   2) clear the SRM flag for that circuit for that Link State PDU.

i) A Link State PDU received with a zero checksum shall be treated as if the Remaining Lifetime were 0. The age, if not 0, shall be overwritten with 0.

The Update Process scans the Link State Database for Link State PDUs with SRM Flags set. When one is found, provided the timestamp lastSent indicates that it was propagated no more recently than minimumLSPTransmissionInterval, the IS shall

a) transmit it on all circuits with SRM Flags set, and

b) update lastSent.

7.3.15 Manipulation of SRM and SSN Flags

For each Link State PDU, and for each circuit over which routeing messages are to be exchanged (i.e. not on DA circuits), there are two flags:

> *Send Routeing Message (SRM*) – if set, indicates that Link State PDU should be transmitted on that circuit. On broadcast circuits SRM is cleared as soon as the LSP has been transmitted, but on non-broadcast circuits SRM is only cleared on reception of a Link State PDU or Sequence Numbers PDU as described below.
>
> SRM shall never be set for an LSP with sequence number zero. (See 7.3.15.2)
>
> *Send Sequence Numbers (SSN*) – if set, indicates that information about that Link State PDU should be included in a Partial Sequence Numbers PDU transmitted on that circuit. When the Sequence Numbers PDU has been transmitted SSN is cleared. Note that the Partial Sequence Numbers PDU serves as an acknowledgement that a Link State PDU was received.

7.3.15.1 Action on Receipt of a Link State PDU

When a Link State PDU is received on a circuit *C*, the IS shall perform the following functions:

a) If the LSP has zero Remaining Lifetime, perform the actions described in 7.3.16.4.

b) If the source *S* of the LSP is an IS or pseudonode for which the first 6 octets are equal to the SystemID of the receiving Intermediate System, and the receiving Intermediate System does not have that LSP in its database, or has that LSP, but no longer considers it to be in the set of LSPs generated by this system (e.g. it was generated by a previous incarnation of the system), then initiate a network wide purge of that LSP as described in 7.3.16.4.

c) If the source *S* of the LSP is a system (pseudonode or otherwise) for which the first 6 octets are equal to the SystemID of the receiving Intermediate system, and the receiving Intermediate system has an LSP in the set of currently generated LSPs from that source in its database (i.e. it is an LSP generated by this Intermediate system), perform the actions described in 7.3.16.1.

d) Otherwise, (the source *S* is some other system),

1) If the LSP is newer than the one in the database, or if an LSP from that source does not yet exist in the database:
   (i) Store the new LSP in the database, overwriting the existing database LSP for that source (if any) with the received LSP.
   (ii) Set SRMflag for that LSP for all circuits other than *C*.
   (iii) Clear SRMflag for *C*.
   (iv) If *C* is a non-broadcast circuit, set SSNflag for that LSP for *C*.
   (v) Clear SSNflag for that LSP for the circuits other than *C*.
2) If the LSP is equal to the one in the database (same Sequence Number, Remaining Lifetimes both zero or both non-zero, same checksums):
   (i) Clear SRMflag for *C*.
   (ii) If *C* is a non-broadcast circuit, set SSNflag for that LSP for *C*.
3) If the LSP is older than the one in the database:
   (i) Set SRMflag for *C*.
   (ii) Clear SSNflag for *C*.

When storing a new LSP, the Intermediate system shall first ensure that it has sufficient memory resources to both store the LSP and generate whatever internal data structures will be required to process the LSP by the Update Process. If these resources are not available the LSP shall be ignored. It shall neither be stored nor acknowledged. When an LSP is ignored for this reason the IS shall enter the “Waiting State”. (See 7.3.19).

7.3.15.2 Action on Receipt of a Sequence Numbers PDU

When a Sequence Numbers PDU (Complete or Partial, see clause 7.3.17) is received on circuit *C* the IS shall perform the following functions:

a) For each LSP reported in the Sequence Numbers PDU:
   1) If the reported value equals the database value and *C* is a non-broadcast circuit, Clear SRMflag for *C* for that LSP.
   2) If the reported value is older than the database value, Clear SSNflag, and Set SRMflag.
   3) If the reported value is newer than the database value, Set SSNflag, and if *C* is a non-broadcast circuit Clear SRMflag.
   4) If no database entry exists for the LSP, and the reported Remaining Lifetime, checksum and sequence number of the LSP are all non-zero, create an entry with sequence number *0* (see 7.3.16.1.), and set SSNflag for that entry and circuit *C*. Under no circumstances shall SRMflag be set for such an LSP with zero sequence number.

      NOTE - This is because possessing a zero sequence number LSP is semantically equivalent to having no information about that LSP. If such LSPs were propagated by setting SRMflag it would result in an unnecessary consumption of both bandwidth and memory resources.

b) If the Sequence Numbers PDU is a Complete Sequence Numbers PDU, Set SRMflags for *C* for all LSPs in the database (except those with zero sequence number or zero remaining lifetime) with LSPIDs within the range specified for the CSNP by the START LSP ID and END LSP ID fields, which were not mentioned in the Complete Sequence Numbers PDU (i.e. LSPs this system has, which the neighbour does not claim to have).

7.3.15.3 Action on expiration of Complete SNP Interval

The IS shall perform the following actions every completeSNPInterval seconds for circuit *C*:

a) If *C* is a broadcast circuit, then

   1) If this Intermediate system is a Level 1 Designated Intermediate System on circuit *C*, transmit a complete set of Level 1 Complete Sequence Numbers PDUs on circuit *C*. Ignore the setting of SSNflag on Level 1 Link State PDUs.

   2) If this Intermediate system is a Level 2 Designated Intermediate System on circuit *C*, transmit a complete set of Level 2 Complete Sequence Numbers PDUs on circuit *C*. Ignore the setting of SSNflag on Level 2 Link State PDUs.

   A complete set of CSNPs is a set whose startLSPID and endLSPID ranges cover the complete possible range of LSPIDs. (i.e. there is no possible LSPID value which does not appear within the range of one of the CSNPs in the set). Where more than one CSNP is transmitted on a broadcast circuit, they shall be separated by an interval of at least minimumBroadcastLSPTransmissionInterval seconds.

   NOTE — An IS is permitted to transmit a small number of CSNPs (no more than 10) with a shorter separation interval, (or even “back to back”), provided that no more than *1000*/minimumBroadcastLSPTransmissionInterval CSNPs are transmitted in any one second period.

b) Otherwise (*C* is a point to point circuit, including non-DA DED circuits and virtual links), do nothing. CSNPs are only transmitted on point to point circuits at initialisation.

7.3.15.4 Action on expiration of Partial SNP Interval

The maximum sized Level 1 or Level 2 PSNP which may be generated by a system is controlled by the values of originatingL1LSPBufferSize or originatingL2LSPBufferSize respectively. An Intermediate system shall perform the following actions every PartialSNPInterval seconds for circuit *C* with jitter applied as described in 10.1:

a) If *C* is a broadcast circuit, then

   1) If this Intermediate system is a Level 1 Intermediate System or a Level 2 Intermediate System with ManualL2OnlyMode “False”, but is **not** a Level 1 Designated Intermediate System on circuit *C*, transmit a Level 1 Partial Sequence Numbers PDU on circuit *C*, containing entries for as many Level 1 Link State PDUs with SSNflag set as will fit in the PDU, and then clear SSNflag for these entries. To avoid the possibility of starvation, the scan of the LSP database for those with SSNflag set shall commence with the next LSP which was not included in the previous scan. If there were no Level 1 Link State PDUs with SSNflag set, do not transmit a Level 1 Partial Sequence Numbers PDU.

   2) If this Intermediate system is a Level 2 Intermediate System, but is **not** a Level 2 Designated Intermediate System on circuit *C*, transmit a Level 2 Partial Sequence Numbers PDU on circuit *C*, containing entries for all as many Level 2 Link State PDUs with SSNflag set as will fit in the PDU, and then clear SSNflag for these entries. To avoid the possibility of starvation, the scan of the LSP database for those with SSNflag set shall commence with the next LSP which was not included in the previous scan. If there were no Level 2 Link State PDUs with SSNflag set, do not transmit a Level 2 Partial Sequence Numbers PDU.

b) Otherwise (*C* is a point to point circuit, including non-DA DED circuits and virtual links), for Level 1 and Level 2 (according to Intermediate system type) transmit a Partial Sequence Numbers PDU on circuit *C*, containing entries for as many Link State PDUs with SSNflag set as will fit in the PDU, and then clear SSNflag for these entries. To avoid the possibility of starvation, the scan of the LSP database for those with SSNflag set shall commence with the next LSP which was not included in the previous scan. If there were no Link State PDUs with SSNflag set, do not transmit a Partial Sequence Numbers PDU.

7.3.15.5 Action on expiration of Minimum LSP Transmission Interval

An IS shall perform the following actions every minimumLSPTransmissionInterval seconds with jitter applied as described in 10.1.

a) For all Point to Point circuits *C* transmit all LSPs that have SRMflag set on circuit *C*, but **do not** clear the SRMflag. The SRMflag will subsequently be cleared by receipt of a Complete or Partial Sequence Numbers PDU.

The interval between two consecutive transmissions of the same LSP shall be at least MinimumLSPTransmissionInterval. Clearly, this can only be achieved precisely by keeping a separate timer for each LSP. This would be an unwarranted overhead. Any technique which ensures the interval will be between MinimumLSPTransmissionInterval and *2* * MinimumLSPTransmissionInterval is acceptable.

7.3.15.6 Controlling the Rate of Transmission on Broadcast Circuits

The Routeing characteristic minimumBroadcastLSPTransmissionInterval indicates the minimum interval between PDU arrivals which can be processed by the slowest Intermediate System on the LAN.

Setting SRMFlags on an LSP for a broadcast circuit does not cause the LSP to be transmitted immediately. Instead the Intermediate system shall scan the LSP database every minimumBroadcastLSPTransmissionInterval (with jitter applied as described in 10.1), and from the set of LSPs which have SRMFlags set for this circuit, one LSP shall be chosen at random. This LSP shall be multicast on the circuit, and SRMFlags cleared.

NOTE - In practice it would be very inefficient to scan the whole database at this rate, particularly when only a few LSPs had SRMFlags set. Implementations may require additional data structures in order to reduce this overhead.

NOTE - An IS is permitted to transmit a small number of LSPs (no more than 10) with a shorter separation interval, (or even “back to back”), provided that no more than *1000/*minimumBroadcastLSPTransmissionInterval LSPs are transmitted in any one second period.

In addition, the presence of any LSPs which have been received on a particular circuit and are queued awaiting processing shall inhibit transmission of LSPs on that circuit. However, LSPs may be transmitted at a minimum rate of one per second even in the presence of such a queue.

7.3.16 Determining the Latest Information

The Update Process is responsible for determining, given a received link state PDU, whether that received PDU represents new, old, or duplicate information with respect to what is stored in the database.

It is also responsible for generating the information upon which this determination is based, for assigning a sequence number to its own Link State PDUs upon generation, and for correctly adjusting the Remaining Lifetime field upon broadcast of a link state PDU generated originally by any system in the domain.

7.3.16.1 Sequence Numbers

The sequence number is a 4 octet unsigned value. Sequence numbers shall increase from zero to SequenceModulus. When a system initialises, it shall start with sequence number 1 for its own Link State PDUs.[10]

The sequence numbers the Intermediate system generates for its Link State PDUs with different values for LSP number are independent. The algorithm for choosing the numbers is the same, but operationally the numbers will not be synchronised.

If an Intermediate system *R* somewhere in the domain has information that the current sequence number for source *S* is greater than that held by *S*, *R* will return to *S* a Link State PDU for *S* with *R*’s value for the sequence number. When *S* receives this LSP it shall change its sequence number to be the next number greater than the new one received, and shall generate a link state PDU.

If an Intermediate system needs to increment its sequence number, but the sequence number is already equal to SequenceModulus, the event “Attempt to Exceed Maximum Sequence Number” shall be generated and the Routeing Module shall be disabled for a period of at least MaxAge + ZeroAgeLifetime, in order to be sure that any versions of this LSP with the high sequence number have expired. When it is re-enabled the IS shall start again with sequence number 1.

---

[10] It starts with 1 rather than 0 so that the value 0 can be reserved to be guaranteed to be less than the sequence number of any actually generated Link State PDU. This is a useful property for Sequence Numbers PDUs.

7.3.16.2 LSP Confusion

It is possible for an LSP generated by a system in a previous incarnation to be alive in the domain and have the same sequence number as the current LSP.

To ensure database consistency among the Intermediate Systems, it is essential to distinguish two such PDUs. This is done efficiently by comparing the checksum on a received LSP with the one stored in memory.

If the sequence numbers match, but the checksums do not and the LSP is not in the current set of LSPs generated by the local system, then the system that notices the mismatch shall treat the LSP as if its Remaining Lifetime had expired. It shall store one of the copies of the LSP, with zero written as the Remaining Lifetime, and flood the LSP.

If the LSP is in the current set of LSPs generated by the local system then the IS shall change the LSP's sequence number to be the next number greater than that of the received LSP and regenerate the LSP.

7.3.16.3 Remaining Lifetime Field

When the source generates a link state PDU, it shall set the Remaining Lifetime to MaxAge.

When a system holds the information for some time before successfully transmitting it to a neighbour, that system shall decrement the Remaining Lifetime field according to the holding time. Before transmitting a link state PDU to a neighbour, a system shall decrement the Remaining Lifetime in the PDU being transmitted by at least 1, or more than 1 if the transit time to that neighbour is estimated to be greater than one second. When the Remaining Lifetime field reaches 0, the system shall purge that Link State PDU from its database. In order to keep the Intermediate Systems' databases synchronised, the purging of an LSP due to Remaining Lifetime expiration is synchronised by flooding an expired LSP. See 7.3.16.4.

If the RemainingLifetime of the received LSP is zero it shall be processed as described in 7.3.16.4. If the Remaining Lifetime of the received LSP is non-zero, but there is an LSP in the database with the same sequence number and zero Remaining Lifetime, the LSP in the database shall be considered most recent. Otherwise, the PDU with the larger sequence number shall be considered the most recent.

If the value of Remaining Lifetime is greater than MaxAge, the LSP shall be processed as if there were a checksum error.

7.3.16.4 LSP Expiration Synchronisation

When the Remaining Lifetime on an LSP in memory becomes zero, the IS shall

a) set all SRMFlags for that LSP, and

b) retain only the LSP header.

c) record the time at which the Remaining Lifetime for this LSP became zero. When ZeroAgeLifetime has elapsed since the LSP Remaining Lifetime became zero, the LSP header shall be purged from the database.

NOTE - A check of the checksum of a zero Remaining Lifetime LSP succeeds even though the data portion is not present

When a purge of an LSP with non-zero Remaining Lifetime is initiated, the header shall be retained for MaxAge.

If an LSP from source *S* with zero Remaining Lifetime is received on circuit *C* :

a) If no LSP from *S* is in memory, then the IS shall

   1) send an acknowledgement of the LSP on circuit *C*, but

   2) shall not retain the LSP after the acknowledgement has been sent.

b) If an LSP from *S* is in the database, then

1) If the received LSP is newer than the one in the database (i.e. received LSP has higher sequence number, or same sequence number and database LSP has non-zero Remaining Lifetime) the IS shall:
    (i) overwrite the database LSP with the received LSP, and note the time at which the zero Remaining Lifetime LSP was received, so that after ZeroAgeLifetime has elapsed, that LSP can be purged from the database,
    (ii) set SRMflag for that LSP for all circuits other than *C*,
    (iii) clear SRMflag for *C*,
    (iv) if *C* is a non-broadcast circuit, set SSNflag for that LSP for *C*, and
    (v) clear SSNflag for that LSP for the circuits other than *C*.
2) If the received LSP is equal to the one in the database (i.e. same Sequence Number, Remaining Lifetimes both zero) the IS shall:
    (i) clear SRMflag for *C*, and
    (ii) if *C* is a non-broadcast circuit, set SSNflag for that LSP for *C*.
3) If the received LSP is older than the one in the database (i.e. received LSP has lower sequence number) the IS shall:
    (i) set SRMflag for *C*, and
    (ii) clear SSNflag for *C*.

c) If this system (or pseudonode) is *S* and there is an un-expired LSP from *S* (i.e. its own LSP) in memory, then the IS:
    1) shall not overwrite with the received LSP, but
    2) shall change the sequence number of the unexpired LSP from *S* as described in Clause 7.3.16.1,
    3) generate a new LSP; and
    4) set SRMflag on all circuits.

7.3.17 Making the Update Reliable

The update process is responsible for making sure the latest link state PDUs reach every reachable Intermediate System in the domain.

a) On point-to-point links the Intermediate system shall send an explicit acknowledgement encoded as a Partial Sequence Numbers PDU (PSNP)containing the following information:
    1) source's ID
    2) PDU type (Level 1 or 2)
    3) sequence number
    4) Remaining Lifetime
    5) checksum

    This shall be done for all received link state PDUs which are newer than the one in the database, or duplicates of the one in the database. Link state PDUs which are older than that stored in the database are answered instead by a newer link state PDU, as specified in 7.3.14 above.

b) On broadcast links, instead of explicit acknowledgements for each link state PDU by each Intermediate system, a special PDU known as a Complete Sequence Numbers PDU (CSNP), shall be multicast periodically by the Designated Intermediate System. The PDU shall contain a list of all LSPs in the database, together with enough information so that Intermediate systems receiving the CSNP can compare with their LSP database to determine whether they and the CSNP transmitter have synchronised LSP databases. The maximum sized Level 1 or Level 2 Sequence Numbers PDU which may be generated by a system is controlled by the values of originatingL1LSPBufferSize or originatingL2LSPBufferSize respectively. In practice, the information required to be transmitted in a single CSNP may be greater than will fit in a single PDU. Therefore each CSNP carries an inclusive range of LSPIDs to which it refers. The complete set of information shall be conveyed by transmitting a series of individual CSNPs, each referring to a subset of the complete range. The ranges of the complete set of CSNPs shall be contiguous (though not necessarily transmitted in order) and shall cover the entire range of possible LSPIDs.

The LAN Level 1 Designated Intermediate System shall periodically multicast complete sets of Level 1 CSNPs to the multi-destination address AllL1ISs. The LAN Level 2 Designated Intermediate System shall periodically multicast complete sets of Level 2 CSNPs to the multi-destination address AllL2ISs.

Absence of an LSPID from a Complete Sequence Numbers PDU whose range includes that LSPID indicates total lack of information about that LSPID.

If an Intermediate system, upon receipt of a Complete Sequence Numbers PDU, detects that the transmitter was out of date, the receiver shall multicast the missing information.

NOTE — Receipt of a link state PDU on a link is the same as successfully transmitting the Link State PDU on that link, so once the first Intermediate system responds, no others will, unless they have already transmitted replies.

If an Intermediate system detects that the transmitter had more up to date information, the receiving Intermediate system shall multicast a Partial Sequence Numbers PDU (PSNP), containing information about LSPs for which it has older information. This serves as an implicit request for the missing information. Although the PSNP is multicast, only the Designated Intermediate System of the appropriate level shall respond to the PSNP.

NOTE — This is equivalent to the PSNP being transmitted directly to the Designated Intermediate System, in that it avoids each Intermediate System unnecessarily sending the same LSP(s) in response. However, it has the advantage of preserving the property that all routeing messages can be received on the multi-destination addresses, and hence by a LAN adapter dedicated to the multi-destination address.

c) When a non-broadcast circuit (re)starts, the IS shall:

   1) set SRMflag for that circuit on all LSPs, and

   2) send a Complete set of Complete Sequence Numbers PDUs on that circuit.

7.3.18 Validation of Databases

An Intermediate System shall not continue to operate for an extended period with corrupted routeing information. The IS shall therefore operate in a *fail-stop* manner. If a failure is detected, the Intermediate system Network entity shall be disabled until the failure is corrected. In the absence of an implementation-specific method for ensuring this, the IS shall perform the following checks at least every maximumLSPGenerationInterval seconds.

a) On expiration of this timer the IS shall re-check the checksum of every LSP in the LSP database (except those with a Remaining Lifetime of zero) in order to detect corruption of the LSP while in memory. If the checksum of any LSP is incorrect, the event “Corrupted LSP Detected” shall be logged, and as a minimum the entire Link State Database shall be deleted and action taken to cause it to be re-acquired. One way to achieve this is to disable and re-enable the IS Network entity.

   NOTE — On point to point links, this requires at least that a CSNP be transmitted.

b) On completion of these checks the decision process shall be notified of an event (even if any newly generated LSPs have identical contents to the previous ones). This causes the decision process to be run and the forwarding databases re-computed, thus protecting against possible corruption of the forwarding databases in memory, which would not otherwise be detected in a stable topology.

c) The IS shall reset the timer for a period of MaximumLSPGenerationInterval with jitter applied as described in 10.1.

7.3.19 LSP Database Overload

As a result of network mis-configuration, or certain transitory conditions, it is possible that there may be insufficient memory resources available to store a received Link State PDU. When this occurs, an IS needs to take certain steps to ensure that if its LSP database becomes inconsistent with the other ISs', that these ISs do not rely on forwarding paths through the overloaded IS.

7.3.19.1 Entering the Waiting State

When an LSP cannot be stored, the LSP shall be ignored and Waiting State shall be entered. A timer shall be started for WaitingTime seconds, and the Intermediate System shall generate and flood its own LSP with zero LSP number with the "Infinite Hippity Cost" Bit set. This prevents this Intermediate system from being considered as a forwarding path by other Intermediate Systems.

It is possible that although there are sufficient resources to store an LSP and permit the operation of the Update Process on that LSP, the Decision Process may subsequently require further resources in order to complete. If these resources are not available, the Intermediate system shall then (i.e. during the attempt to run the Decision Process) enter Waiting State until such time as they are available and WaitingTime seconds have elapsed since the last LSP was ignored by the Update Process.

An implementation shall partition the available memory resources between the Level 1 and Level 2 databases. An overload condition can therefore exist independently for Level 1 or Level 2 (or both). The status attributes L1State and L2State indicate the condition for the Level 1 and Level 2 databases respectively. On entering Level 1 "Waiting State" the IS shall generate the "LSP L1 Database Overload" event, and on entering Level 2 "Waiting State" the IS shall generate the "LSP L2 Database Overload" event.

7.3.19.2 Actions in Level 1 Waiting State

While in Level 1 "waiting" state

a) If a Link State PDU cannot be stored, the IS shall ignore it and restart the timer for WaitingTime seconds.

b) The IS shall continue to run the Decision and Forwarding processes as normal.

c) When the WaitingTime timer expires, the IS shall:

   1) Generate an "LSP L1 Database Overload (recovered)" event.

   2) Clear the "Infinite Hippity Cost" bit in its own Level 1 LSP with zero LSP number and re-issue it.

   3) Set the L1State to "On".

   4) Resume normal operation.

7.3.19.3 Actions in Level 2 Waiting State

While in Level 2 "waiting" state

a) If a Link State PDU cannot be stored, the IS shall ignore it and restart the timer for WaitingTime seconds.

b) The IS shall continue to run the Decision and Forwarding processes as normal.

c) When the WaitingTime timer expires, the IS shall:

   1) Generate an "LSP L2 Database Overload (recovered)" event.
   2) Clear the "Infinite Hippity Cost" bit in its own Level 2 LSP with zero LSP number and re-issue it.
   3) Set the L2State to "On".
   4) Resume normal operation.

7.3.20 Use of the Link State Database

The only portion of the database relevant to the Decision Process is the data portion of the Link State PDUs.

The Update Process additionally uses the fields sequence number, Remaining Lifetime, and SRMFlag. The Remaining Lifetimes in the stored link state PDUs can either be periodically decremented, or converted upon receipt into an internal timestamp, and converted back into a Remaining Lifetime upon transmission.

7.3.20.1 Synchronisation with the Decision Process

Since the Update Process and the Decision Process share the Link State Database, care must be taken that the Update Process does not modify the Link State Database while the Decision Process is running.

There are two approaches to this. In one approach, the Decision Process signals when it is running. During this time, the Update Process queues incoming Link State PDUs, and does not write them into the Link State Database. If more Link State PDUs arrive than can fit into the queue allotted while the Decision Process is running, the Update Process drops them and does not acknowledge them.

Another approach is to have two copies of the Link State Database — one in which the Decision Process is computing, and the other in which the Update Process initially copies over the first database, and in which all new Link State PDUs are written. Additionally, depending on the hashing scheme, it is likely that a second copy of the address hash table will be required, so that the Update Process can do a rehash occasionally for efficiency.

When the Decision Process is ready to run again, it locks the new copy of the Link State Database, leaving the Update Process to copy over the information into the first area, and write new updates while the Decision Process runs again.

The advantage of the first approach is that it takes less memory. The advantage of the second approach is that Link State PDUs will never need to be dropped.

NOTE - If the decision process is implemented according to the specification in C.2, a finer level of parallelism is possible, as described below.

Arrival of a Link State PDU for a system before that system has been put into TENT is permitted. The new Link State PDU is used when that system is eventually put into TENT. Similarly, arrival of a new Link State PDU for a system after that system has been put into PATHS is permitted. That system has already been completely processed. The arrival of the new Link State PDU is noted and the decision process re-executed when the current execution has completed. An in-progress execution of the decision process shall not be abandoned, since this could prevent the decision process from ever completing.

Arrival of a Link State PDU for a system between that system being put on TENT and being transferred to PATHS shall be treated as equivalent to one of the previous two cases (for example, by buffering, or taking some corrective action).

7.3.20.2 Use of Buffers and Link Bandwidth

Implementations shall have a buffer management strategy that does not prevent other clients of the buffering service from acquiring buffers due to excessive use by the Update Process. They shall also ensure that the Update Process does not consume all the available bandwidth of links. In particular no type of traffic should experience starvation for longer than its acceptable latency. Acceptable latencies are approximately as follows:

- Hello traffic – Hello timer × 0.5
- Data Traffic – 10 seconds.

NOTE - The first of these requirements can be met by restricting the Update process to the use of a single buffer on each circuit for transmission. This may also cause the second requirement to be met, depending on the processor speed.

7.3.21 Parameters

MaxAge – This is the amount of time that may elapse since the estimated origination of the stored Link State PDU by the source before the LSP is considered expired. The expired LSP can be deleted from the database after a further ZeroAgeLifetime has expired. MaxAge shall be larger than maximumLSPGenerationInterval, so that a system is not purged merely because of lack of events for reporting Link State PDUs.

MaxAge is an architectural constant equal to 20 minutes.

ZeroAgeLifetime - This is the minimum amount of time for which the header of an expired LSP shall be retained after it has been flooded with zero Remaining Lifetime. A very safe value for this would be 2 × MaxAge. However all that is required is that the header be retained until the zero Remaining Lifetime LSP has been safely propagated to all the neighbours.

ZeroAgeLifetime is an architectural constant with a value of 1 minute.

maximumLSPGenerationInterval – This is the maximum amount of time allowed to elapse between generation of Link State PDUs by a source. It shall be less than MaxAge.

Setting this parameter too fast adds overhead to the algorithms (a lot of Link State PDUs). Setting this parameter too slow (and not violating constraints) causes the algorithm to wait a long time to recover in the unlikely event that incorrect Link State information exists somewhere in the domain about the system.

A reasonable setting is 15 minutes.

minimumLSPGenerationInterval – This is the minimum time interval between generation of Link State PDUs. A source Intermediate system shall wait at least this long before re-generating one of its own Link State PDUs.

Setting this too large causes a delay in reporting new information. Setting this too small allows too much overhead.

A reasonable setting is 30 seconds.

minimumLSPTransmissionInterval – This is the amount of time an Intermediate system shall wait before further propagating another Link State PDU from the same source system.

Setting this too large causes a delay in propagation of routeing information and stabilisation of the routeing algorithm. Setting this too small allows the possibility that the routeing algorithm, under low probability circumstances, will use too many resources (CPU and bandwidth).

Setting minimumLSPTransmissionInterval greater than minimumLSPGenerationInterval makes no sense, because the source would be allowed to generate LSPs more quickly than they'd be allowed to be broadcast. Setting minimumLSPTransmissionInterval smaller than minimumLSPGenerationInterval is desirable to recover from lost LSPs.

A reasonable value is 5 seconds.

CompleteSNPInterval – This is the amount of time between periodic transmissions of a complete set of Sequence Number PDUs by the Designated Intermediate system on a broadcast link. Setting this too low slows down the convergence of the routeing algorithm when Link State PDUs are lost due to the datagram environment of the Data Link layer on the broadcast link.

Setting this too high results in extra control traffic overhead.

A reasonable value is 10 seconds.

7.4 The Forwarding Process

The forwarding process is responsible for transmitting NPDUs originated by this system, and performing NPDU forwarding.

7.4.1 Input and Output

INPUT

- NPDUs from the ISO 8473 protocol machine
- PDUs from Update Process
- PDUs from Receive Process
- Forwarding Databases (Level 1 and 2) — one for each routeing metric

OUTPUT

- PDUs to Data Link Layer

7.4.2 Routeing Metric Selection

The Forwarding process selects a forwarding database for each NPDU to be relayed based on:

- the level at which the forwarding is to occur: level 1 or level 2; and
- a mapping of the ISO 8473 QoS Maintenance field onto one of the Intermediate system's supported routeing metrics.

The former selection is made by examining the Destination Address field of the NPDU.

The latter selection is made as follows:

a) If the QoS Maintenance field is not present in the NPDU, then the IS shall select the forwarding database calculated for the *default metric*.

b) If the QoS Maintenance field is present, the IS shall examine bits 7 and 8 of the parameter value octet. If these two bits specify any combination other than "1 1" (meaning globally unique QoS), then the IS shall select the forwarding database calculated for the *default metric*, otherwise

c) The IS shall select a forwarding database by mapping the values of bits 3, 2 and 1 of the parameter value as shown below in table 1 and shall proceed as follows:

   1) If the IS does not support the selected routeing metric, the IS shall forward based upon the *default metric;*

   2) If the forwarding database for one of the optional routeing metrics is selected and the database either does not contain an entry for the Destination Address in the NPDU being relayed, or contains an entry indicating that the destination is unreachable using that metric, then the IS shall attempt to forward based upon the *default metric*;

   3) Otherwise, forward based on the selected optional metric.

Table 1 - QoS Maintenance bits to routeing metric mappings

bit 1	*bit 2*	*bit 3*	*Selected routeing metric*
0	0	0	expense metric
1	0	0	default metric
0	1	1	default metric
0	1	0	expense metric
0	0	1	delay metric
1	1	0	error metric
1	0	1	delay metric
1	1	1	error metric

7.4.3 Forwarding Decision

7.4.3.1 Basic Operation

Let DEST = the Network Layer destination address of the PDU to be forwarded, or the next entry in the source routeing field, if present. It consists of sub-fields AREA ADDRESS, ID, and SELECTOR.

NOTE - The SELECTOR field in the destination address is not examined by Intermediate Systems. It is used by End Systems to select the proper Transport entity to which to deliver NSDUs.

This system's (the one examining this PDU for proper forwarding decision) address consists of sub-fields AREA ADDRESS and ID.

a) If the local system type is a level 1 Intermediate system, or the local system type is a level 2 Intermediate system and AttachedFlag_k = False, then:

  1) If the area address matches then

    CALL Forward(Level 1,$\text{metric}_{k,}$ DEST; ADJ),

    and forward on adjacency ADJ.

  2) ELSE

    CALL Forward(Level 1,metric_k, 0; ADJ)

    and forward on adjacency ADJ. (i.e. pass to nearest Level 2 Intermediate system.)

b) If the local system type is Level 2, and AttachedFlag_k = True then:

  1) If the area address matches then

    CALL Forward(Level 1,metric_k, DEST; ADJ), and forward on adjacency ADJ.

  2) ELSE

    CALL Forward(Level 2, metric_k,DEST; ADJ)

    and forward on adjacency ADJ. i.e. Forward according to normal Level 2 routeing

7.4.3.2 Encapsulation for Partition Repair

If this Intermediate system is the Partition Designated Level 2 IS for this partition, and the PDU is being forwarded onto the special adjacency to a Partition Designated Level 2 Intermediate system in a different partition of this area, encapsulate the complete PDU as the data field of a data NPDU (i.e., with an additional layer of header), making this system the Source address and the other Partition Designated Level 2 Intermediate system (obtained from the identifier attribute of the virtual Adjacency managed object) the Destination Address in the outer PDU header. Set the QoS Maintenance field of the outer PDU to indicate forwarding via the default routeing metric (see table 1). Then forward the encapsulated PDU onto an adjacency ADJ, obtained by calling Forward(Level 2, *default metric,* outer PDU DEST; ADJ).

7.4.3.3 The Procedure Forward

```
PROCEDURE Forward(
        level: (level1, level2),
        metric: (default,delay,expense,error);
        dest: NetworkLayerAddress,
        VAR adj: POINTER TO adjacency
        ) : BOOLEAN
```

This procedure chooses, from a Level 1 forwarding database – if level is level1, or from a Level 2 forwarding database – if level is level2, an adjacency on which to forward NPDUs for destination dest. A pointer to the adjacency is returned in adj, and the procedure returns the value “True”. A destination of “0” at level 1 selects the adjacency for the nearest level 2 IS computed as described in 7.2.9.1.

If there are multiple possible adjacencies, as a result of multiple minimum cost paths, then one of those adjacencies shall be chosen. An implementation may chose the adjacency at random, or may use the possible adjacencies in “round robin” fashion.

If there is no entry in the selected forwarding database for the address dest, and the NPDU originated from the a local Transport entity and the system has one or more Intermediate System adjacencies, then one of those is chosen at random (or in “round robin” fashion) and the procedure returns the value “True”. Otherwise the procedure returns the value “False”.[11]

NOTE - Since the local adjacency database is pre-loaded into the decision process, there will always be an entry in the forwarding database for destinations to which an adjacency exists.

NOTE - The PDU to be forwarded may require fragmentation, depending on which circuit it is to be forwarded over. First the circuit is chosen for the PDU then, if necessary, the PDU is fragmented into multiple NPDUs which are all transmitted over that circuit.

7.4.3.4 Generating Redirect PDUs

In addition to forwarding an NPDU, the IS shall inform the local ISO 9542 protocol machine to generate a *Redirect PDU* if the PDU is being forwarded onto the same circuit from which it came, and if the source SNPA address of the NPDU indicates that the NPDU was received from an End System.

7.4.4 The Receive Process

The Receive Process is passed information from any of the following sources.

- received PDUs with the NPID of Intra-Domain routeing,

---

[11] This is done so that a system in the overloaded state will still be able to originate or forward NPDUs. If a system with a partial routeing information base were prohibited from attempting to forward to an unknown destination, system management would be unable to either communicate with this system, or route through it, for the purpose of diagnosing and/or correcting the underlying fault.

- configuration information from the ISO 9542 protocol machine,
- ISO 8473 data PDUs handed to the routeing function by the ISO 8473 protocol machine.

When an area is partitioned, a level 2 path is used as a level 1 link to repair the partitioned area. When this occurs, all PDUs (between the neighbours which must utilise a multi-hop path for communication) shall be encapsulated in a data NPDU, addressed to the “Intra-Domain routeing” SELECTOR. Control traffic (LSPs, Sequence Numbers PDUs) shall also be encapsulated, as well as data NPDUs that are to be passed between the “neighbours”.

NOTE - It is not necessary to transmit encapsulated IIH PDUs over a virtual link, since virtual adjacencies are established and monitored by the operation of the Decision Process and not the Subnetwork Dependent functions

The Receive Process shall perform the following functions:

- If it is a data NPDU, addressed to *this system* with SELECTOR = “Intra-Domain routeing”, then
  - decapsulate the NPDU (remove the outer NPDU header).
  - If the decapsulated PDU is a data NPDU, move the “congestion” indications to the decapsulated NPDU, and pass it to the ISO 8473 protocol machine.
  - Otherwise, if the decapsulated PDU is not an ISO 8473 PDU, perform the following steps on the decapsulated PDU:
- If it is a Link State PDU, pass it to the Update Process
- If it is a Sequence Numbers PDU, pass it to the Update Process
- If it is a IIH PDU, pass it to the appropriate Subnetwork Dependent Function
- If it is a data NPDU or Error Report for another destination, pass it to the Forwarding Process
- Otherwise, ignore the PDU

7.5 Routeing Parameters

The routeing Parameters set-able by System Management are listed for each managed object in clause 11.

7.5.1 Architectural Constants

The architectural constants are described in Table 2.

Table 2 - Routeing architectural constants

Name	*Value*	*Description*
MaxLinkMetric	63.	Maximum value of a routeing metric assignable to a circuit
MaxPathMetric	1023.	Maximum total metric value for a complete path
AllL1ISs	TBS by Digital	The multi-destination address “All Level 1 Intermediate Systems”
AllL2ISs	09-00-2B-02-00-00	The multi-destination address “All Level 2 Intermediate Systems”
AllIntermediateSystems	09-00-2B-00-00-05	The multi-destination address “All Intermediate Systems” used by ISO 9542
AllEndSystems	09-00-2B-00-00-04	The multi-destination address “All End Systems” used by ISO 9542
ISO-SAP	FE	The SAP for ISO Network Layer on ISO 8802-3 LANs
IntradomainRouteing-PD	TBS by TR9577	The Network Layer Protocol Discriminator assigned by ISO/TR 9577 for this Protocol
IntradomainRouteing-Selector	0.	The NSAP selector for the Intermediate System Network entity
SequenceModulus	2^{32}	Size of the sequence number space used by the Update Process
ReceiveLSPBufferSize	1492.	The size of LSP which all Intermediate systems must be capable of receiving.
MaxAge	1200.	Number of seconds before LSP considered expired.
ZeroAgeLifetime	60.	Number of seconds that an LSP with zero Remaining Lifetime shall be retained after propagating a purge.
MaximumAreaAddresses	3.	The maximum number of Area Addresses which may exist for a single area.
HoldingMultiplier	3.	The number by which to multiply helloTimer to obtain Holding Timer for ISH PDUs and for Point to Point IIH PDUs.
ISISHoldingMultiplier	10.	The number by which to multiply ISIShelloTimer to obtain Holding Timer for Level 1 and Level 2 LAN IIH PDUs.
Jitter	25.	The percentage of jitter which is applied to the generation of periodic PDUs.

8 Subnetwork Dependent Functions

The *Subnetwork Dependent Functions* mask the characteristics of the different kinds of Subnetworks from the *Subnetwork Independent Routeing Functions*. The only two types of circuits the Subnetwork Independent Functions recognise are *broadcast* and *general topology*.

The Subnetwork Dependent Functions include:

- The use of the ISO 8473 *Subnetwork Dependent Convergence Functions (SNDCF)* so that this protocol may transmit and receive PDUs over the same subnetwork types, using the same techniques, as does ISO 8473.
- Co-ordination with the operation of the ES–IS protocol (ISO 9542) in order to determine the Network layer addresses (and on Broadcast subnetworks, the subnetwork points of attachment) and identities (End System or Intermediate System) of all adjacent neighbours. This information is held in the *Adjacency* data base. It is used to construct Link State PDUs.
- The exchange of IIH PDUs. While it is possible for an Intermediate System to identify that it has an Intermediate System neighbour by the receipt of an ISO 9542 ISH PDU, there is no provision within ISO 9542 to indicate whether the neighbour is a Level 1 or a Level 2 Intermediate System. Specific PDUs (*LAN Level 1, LAN Level 2* and *Point to point IIH PDUs*) are defined to convey this information.

8.1 Multi-destination Circuits on ISs at a Domain Boundary

Routeing information (Link State PDUs) is not exchanged across a routeing domain boundary. All routeing information relating to a circuit connected to another routeing domain is therefore entered via the Reachable Address managed objects. This information is disseminated to the rest of the routeing domain via Link State PDUs as described in 7.3.3.2. This has the effect of causing NPDUs destined for NSAPs which are included in the addressPrefixes of the Reachable Addresses to be relayed to that Intermediate System at the domain boundary. On receipt of such an NPDU the Intermediate system shall forward it onto the appropriate circuit, based on its own Link State information. However in the case of multi-destination subnetworks (such as an ISO 8208 subnetwork using Dynamic Assignment, a broadcast subnetwork, or a connectionless subnetwork) it is necessary to ascertain additional subnetwork dependent addressing information in order to forward the NPDU to a suitable SNPA. (This may be the target End system or an Intermediate system within the other domain.)

In general the SNPA address to which an NPDU is to be forwarded can be derived from the destination NSAP of the NPDU. It may be possible to perform some algorithmic manipulation of the NSAP address in order to derive the SNPA address. However there may be some NSAPs where this is not possible. In these cases it is necessary to have pre-configured information relating an address prefix to a particular SNPA address.

This is achieved by additional information contained in the Reachable Address subordinate managed object. The mappingType characteristic may be specified as “Manual”, in which case a particular SNPA address or set of SNPA addresses is specified in the SNPA Address characteristic. Alternatively the name of an SNPA address extraction algorithm may be specified.

8.2 Point to Point Subnetworks

This clause describes the identification of neighbours on point to point links and on both Static and DCM DED circuits.

The IS shall operate the ISO 9542 protocol, shall be able to receive ISO 9542 ISH PDUs from other ISs, and shall store the information so obtained in the adjacency database.

8.2.1 Receipt of ESH PDUs – Database of End Systems

An IS shall enter an End system into the adjacency database when an ESH PDU is received on a circuit. If an ESH PDU is received on the same circuit, but with a different NSAP address, the new address shall be added to the adjacency, with a separate timer. A single ESH PDU may contain more than one NSAP address. When a new data link address or NSAP address is added to the adjacency database, the IS shall generate an “Adjacency State Change (Up)” event on that adjacency.

The IS shall set a timer for the value of “Holding Time” in the received ESH PDU. If another ESH PDU is not received from the ES before that timer expires, the ES shall be purged from the database, provided that the Subnetwork Independent Functions associated with initialising the adjacency have been completed. Otherwise the IS shall clear the adjacency as soon as those functions are completed.

When the adjacency is cleared, the Subnetwork Independent Functions shall be informed of an “Adjacency Change (Down)” event, and the adjacency can be re-used after the Subnetwork Independent Functions associated with bringing down the adjacency have been completed.

8.2.2 Receiving ISH PDUs by an Intermediate System

On receipt of an ISH PDU by an Intermediate System, the IS shall create an adjacency (with state “Initialising” and adjacencyType “Unknown”), if one does not already exist, and then perform the following actions:.

a) If the Adjacency state is “Up” and the ID portion of the NET field in the ISH PDU does not match the neighbourID of the adjacency then the IS shall:

    1) generate an “Adjacency State Change (Down)” event;

    2) delete the adjacency; and

    3) create a new adjacency with:

        (i) state set to “Initialising”, and

        (ii) adjacencyType set to “Unknown”.

    4) perform the following actions..

b) If the Adjacency state is “Initialising”, and the adjacencyType status is “Intermediate System”, the ISH PDU shall be ignored.

c) If the Adjacency state is “Initialising” and the adjacencyType status is not “Intermediate System”, a point to point IIH PDU shall be transmitted as described in Clause 8.2.3.

d) The adjacencyType status shall be set to “Intermediate System” indicating that the neighbour is an Intermediate system, but the type (L1 or L2) is, as yet, unknown.

8.2.3 Sending Point to Point IIH PDUs

When required, an IS shall construct and transmit a Point to Point IIH PDU as follows.

a) The Circuit Type field shall be set according to Table 3.

b) The Local Circuit ID field shall be set to a value assigned by this Intermediate system when the circuit is created. This value shall be unique among all the circuits of this Intermediate system.

c) The first Point to Point IIH PDU (i.e. that transmitted as a result of receiving an ISH PDU, rather than as a result of timer expiration) shall be padded (with trailing PAD options containing arbitrary valued octets) so that the SNSDU containing the IIH PDU has a length of at least *maxsize - 1* octets[12] where *maxsize* is the maximum of

    - dataLinkBlocksize

---

[12] The minimum length of PAD which may be added is 2 octets, since that is the size of the option header. Where possible the PDU should be padded to maxsize, but if the PDU length is *maxsize- 1* octets no padding is possible (or required).

- originatingL1LSPBufferSize
- originatingL2LSPBufferSize

This is done to ensure that an adjacency will only be formed between systems which are capable of exchanging PDUs of length up to *maxsize* octets. In the absence of this check, it would be possible for an adjacency to exist with a lower maximum block size, with the result that some LSPs and SNPs (i.e. those longer than this maximum, but less than *maxsize*) would not be exchanged.

NOTE - It is necessary for the manager to ensure that the value of dataLinkBlocksize on a circuit which will be used to form an Intermediate system to Intermediate system adjacency is set to a value greater than or equal to the maximum of the LSPBufferSize characteristics listed above. If this is not done, the adjacency will fail to initialise. It is not possible to enforce this requirement, since it is not known until initialisation time whether or not the neighbour on the circuit will be an End system or an Intermediate system. An End system adjacency may operate with a lower value for dataLinkBlocksize.

Table 3 - Setting the value of the Circuit Type field

Routeing Type	Circuit manualL2OnlyMode	Circuit Type Field
Level 1	—	Level 1 only (1)
Level 2	“True”	Level 2 only (2)
Level 2	“False”	Level 1 and 2 (3)

8.2.4 Receiving Point to Point IIH PDUs

When a Point to Point IIH PDU is received by an Intermediate system, the area addresses of the two Intermediate Systems shall be compared to ascertain the validity of the adjacency. If the two Intermediate systems have an area address in common, the adjacency is valid for all combinations of Intermediate system types (except where a Level 1 Intermediate system is connected to a Level 2 Intermediate system with manualL2OnlyMode set “True”). However, if they have no area address in common, the adjacency is only valid if both Intermediate systems are Level 2, and the IS shall mark the adjacency as Level 2 Only. This is described in more detail below.

On receipt of a Point to Point IIH PDU, each of the Area Addresses from the PDU shall be compared with the set of Area Addresses in the manualAreaAddresses characteristic.

a) If a match is detected between any pair the following actions are taken.

   1) If the local system is of Routeing Type “L1IntermediateSystem” the IS shall perform the action indicated by Table 4.

   2) If the local system is of Routeing Type “L2IntermediateSystem” and the Circuit manualL2OnlyMode has the value “False”, the IS shall perform the action indicated by Table 5.

   3) If the local system is of Routeing Type “L2IntermediateSystem” and the Circuit manualL2OnlyMode has the value “True”, the IS shall perform the action indicated by Table 6.

b) If a no match is detected between any pair, the following actions shall be performed.

   1) If the local system is of Routeing Type “L1IntermediateSystem” and the adjacency is not in state “Up”, the IS shall delete the adjacency (if any) and generate an “Initialisation Failure (Area Mismatch)” Event.

   2) If the local system is of Routeing Type “L1IntermediateSystem” and the adjacency is in state “Up”, the IS shall delete the adjacency and generate an “Adjacency State Change (Down – Area Mismatch)” Event .

   3) If the local system is of Routeing Type “L2IntermediateSystem” the IS shall perform the action indicated by Table 7 (irrespective of the value of manualL2OnlyMode for this circuit).

c) If the action taken is “Up”, as detailed in the tables referenced above, the IS shall compare the Source ID field of the PDU with the local SystemID.

1) If the local Intermediate system has the higher Source ID, the IS shall set the Circuit CircuitID status to the concatenation of the local SystemID and the Local Circuit ID (as sent in the Local Circuit ID field of point to point IIH PDUs from this Intermediate System) of this circuit.

2) If the remote Intermediate system has the higher Source ID, the IS shall set the Circuit CircuitID status to the concatenation of the remote system's Source ID (from the Source ID field of the PDU), and the remote system's Local Circuit ID (from the Local Circuit ID field of the PDU).

3) If the two source IDs are the same (i.e. the system is initialising to itself), the local SystemID is used.

NOTE — The circuitID status is not used to generate the Local Circuit ID to be sent in the Local Circuit ID field of IIH PDUs transmitted by this Intermediate system. The Local Circuit ID value is assigned once, when the circuit is created and is not subsequently changed.

d) If the action taken is "Accept" and the new value computed for the circuitID is different from that in the existing adjacency, the IS shall

1) generate an "Adjacency State Change(Down)" event, and

2) delete the adjacency.

e) If the action taken is "Up" or "Accept" the IS shall

1) copy the Adjacency neighbourAreas entries from the PDU,

2) set the holdingTimer to the value of the "Holding Time" from the PDU, and

3) set the neighbourSystemID to the value of the "Source ID" from the PDU.

Table 4 - Level 1 State table for matching areas

Circuit Type[1]	Adjacency Type		
	none[2]	Level 1[3]	Level 2[4]
Level 1 only	Up[5] L1[6]	Accept	Down[7] (Wrong system)
Level 2 only	Reject[8] (Wrong system)	Down[7] (Wrong system)	Down[7] (Wrong system)
Level 1 & 2	Up[5] L2[9]	Down[7] (Wrong system)	Accept

[1] The value of the "Circuit Type" field in the received PDU.

[2] The adjacency is not in state "Up"

[3] The adjacency is in state "Up" and the Adjacency adjacencyType is "L1 Intermediate System".

[4] The adjacency is in state "Up" and the Adjacency adjacencyType is "L2 Intermediate System".

[5] The adjacency is accepted and an "Adjacency State Change (Up)" event is generated. If the Adjacency adjacencyType was "Unknown" (i.e. no ISH PDU has yet been received), a point to point IIH PDU is also transmitted.

[6] The adjacencyType status is set to "L1 Intermediate System".

[7] An "Adjacency State Change (Down)" event is generated, with the specified reason, and the adjacency deleted.

[8] An "Initialization Failure" event is generated with the specified reason.

[9] The adjacencyType status is set to "L2 Intermediate System".

Table 5 - Level 2 State table for matching areas

Circuit Type[1]	Adjacency Type			
	none[2]	Level 1[3]	Level 2[4]	Level 2 Only[5]
Level 1 only	Up[6] L1[7]	Accept	Down[8] (Wrong system)	Down[8] (Wrong system)
Level 2 only	Up[6] L2O[9]	Down[8] (Wrong system)	Down[8] (Wrong system)	Accept
Level 1 & 2	Up[6] L2[10]	Down[8] (Wrong system)	Accept	Down[8] (Wrong system)

[1] The value of the “Circuit Type” field in the received PDU.

[2] The adjacency is not in state “Up”

[3] The adjacency is in state “Up” and the Adjacency adjacencyType is “L1 Intermediate System”.

[4] The adjacency is in state “Up” and the Adjacency adjacencyType is “L2 Intermediate System, with L2OnlyMode “False”.

[5] The adjacency is in state “Up” and the Adjacency adjacencyType is “L2 Intermediate System, with L2OnlyMode “True”.

[6] The adjacency is accepted and an “Adjacency State Change (Up)” event is generated. If the Adjacency adjacencyType was “Unknown” (i.e. no ISH PDU has yet been received), a point to point IIH PDU is also transmitted.

[7] The adjacencyType status is set to “L1 Intermediate System”.

[8] An “Adjacency State Change (Down)” event is generated, with the specified reason, and the adjacency deleted.

[9] .The adjacencyType status is set to “L2 Intermediate System”, and the L2OnlyMode is set to “True”.

[10] The adjacencyType status is set to “L2 Intermediate System”, and the L2OnlyMode is set to “False”.

Table 6 - Level 2 Only State table for matching areas

Circuit Type[1]	Adjacency Type		
	none[2]	Level 2[3]	Level 2 Only[4]
Level 1 only	Reject[5] (Wrong system)	Down[6] (Wrong system)	Down[6] (Wrong system)
Level 2 only	Up[7] L2O[8]	Down[6] (Wrong system)	Accept
Level 1 & 2	Up[7] L2O[8]	Down[6] (Wrong system)	Accept

[1] The value of the “Circuit Type” field in the received PDU.

[2] The adjacency is not in state “Up”

[3] The adjacency is in state “Up” and the Adjacency adjacencyType is “L2 Intermediate System”, with L2OnlyMode “False”.

[4] The adjacency is in state “Up” or and the Adjacency adjacencyType is “L2 Intermediate System” with L2OnlyMode “True”.

[5] An “Initialization Failure” event is generated with the specified reason.

[6] An “Adjacency State Change (Down)” event is generated, with the specified reason, and the adjacency deleted.

[7] The adjacency is accepted and an “Adjacency State Change (Up)” event is generated. If the Adjacency adjacencyType was “Unknown” (i.e. no ISH PDU has yet been received), a point to point IIH PDU is also transmitted.

[8] The adjacencyType status is set to “L2 Intermediate System” with L2OnlyMode “True”.

Table 7 - Level 2 State table for non-matching areas

8.2.5 Monitoring Point-to-point Adjacencies

The IS shall keep a holding time (adjacency holdingTimer) for the point-to-point adjacency. The value of the hold is 15 minutes bour is not heard from in that time, the IS shall

a) purge it from the database; and

b) generate an "Adjacency State Change (Down)" event.

8.3.2 SVC Establishment

8.3.2.1 Use of ISO 8473 Subnetwork Dependent Convergence Functions

SVCs shall be established according to the procedures defined in the ISO 8208 Subnetwork Dependent Convergence Functions of ISO 8473 (this may be on system management action or on arrival of data depending on the type of circuit). The Call Request shall contain a Protocol Discriminator specifying ISO 8473 in the first octet of Call User-data.

In the case of a STATIC circuit, an SVC shall be established only upon system management action. The IS shall use neighbourSNPAAddress as the called SNPA address.

In the case of a DCM circuit, the call establishment procedures are initiated by the arrival of traffic for the circuit.

8.3.2.2 Dynamically Assigned Circuits

A dynamically assigned circuit has multiple adjacencies, and can therefore establish SVCs to multiple SNPAs. In general the SNPA address to which a call is to be established can be derived from the NSAP to which an NPDU is to be forwarded. In the case where all the NSAPs accessible over the ISO 8208 subnetwork have IDIs which are their SNPA addresses, the correct SNPA can be ascertained by extracting the IDI. However there may be some NSAPs, which it is required to reach over the ISO 8208 subnetwork, whose IDI does not correspond to the SNPA address of their point of attachment to the ISO 8208 subnetwork. The IDI may refer to some other SNPA address which is sub-optimally connected to the target NSAP (or not even connected at all), or the IDP may not contain an X.121 address at all (e.g. ISO DCC scheme). In these cases the IS shall have pre-configured information relating an IDP (or address prefix) to a particular SNPA address to call.

This is achieved, as described in 8.1, by additional information contained in the Reachable Address managed object. The address extraction algorithm may be specified to extract the IDI portion where the IDI is the required X.121 address. An example of a set of Reachable Addresses is shown in Table 8.

Table 8 - Example of address prefixes

Address Prefix	SNPA Address
`39 123`	*X*
`37 aaaaa`	*B*
`37 D`	*Y*
`37`	Extract X.121 SNPA address
`*`	*R, S, T*

This is interpreted as follows:

a) For the ISO DCC prefix `39 123`, call the SNPA address *X*.

b) For the X.121 IDI address prefix `37 aaaaa`, don't call *aaaaa*, but call *B* instead.

c) For all IDPs based on SNPAs with DNIC *D* (i.e. with address prefix `37 D`), call the address *Y* (which would probably be a gateway to a subnetwork with DNIC *D*).

d) For any other X.121 IDI (i.e. address prefix `37`) – call the SNPA whose address is used as the IDI.

e) Anything else ("*" in table 8) – call one of the SNPA addresses *R*, *S* or *T*. These would typically be the SNPA addresses of Level 2 Intermediate Systems through which any other addresses could potentially be reached.

NOTE - If a DA circuit is defined with a reachable address prefix which includes the addresses reachable over a DCM or STATIC circuit, the cost(s) for the DA circuit must be greater than those of the DCM or STATIC circuit. If this is not the case, the DA circuit may be used to establish a call to the remote SNPA supporting the DCM or STATIC circuit, which would then (wrongly) assume it was the DCM/STATIC circuit.

8.3.2.3 Initiating Calls (Level 2 Intermediate Systems)

When an NPDU is to be forwarded on a dynamically assigned circuit, for destination NSAP address *D*, the IS shall:

a) Calculate *D*'s subnetwork address, either as explicitly stated in the circuit database, or as extracted from the IDP.

  1) If this system is an ES and there is an entry in the RedirectCache or ReversePathCache for *D*, use the subnetwork address in the cache entry.

  2) If this system is an ES or Level 2 Intermediate system, and the address matches one of the listed reachable address prefixes (including "*", if present), the subnetwork address is that specified according to the mappingType characteristic (either Manual, indicating that the set of addresses in the SNPAAddresses attribute of that Reachable Address are to be used, or Algorithm, indicating that it is to be extracted from the IDP using the specified algorithm). If multiple SNPA addresses are specified, and there is already an adjacency up to one of those SNPA addresses, then choose that subnetwork address, otherwise choose the subnetwork address with the oldest timestamp as described in 8.3.2.4.

  3) If the address does not match one of the listed reachable address prefixes (and there is no "*" entry), invoke the ISO 8473 Discard PDU function.

b) Scan the adjacencies for one already open to *D*'s subnetwork address (i.e. reserveTimer has not yet expired). If one is found, transmit the NPDU on that adjacency.

c) If no adjacency has a call established to the required subnetwork address, but there is a free adjacency, attempt to establish the call using that subnetwork address.

d) If there is no free adjacency invoke the ISO 8473 Discard PDU function.

NOTE — Where possible, when an adjacency is reserved (when an SVC has been cleared as a result of the idleTimer expiring, but the reserveTimer has not yet expired), resources within the subnetwork service provider should be reserved, in order to minimise the probability that the adjacency will not be able to initiate a call when required.

8.3.2.4 Call Attempt Failures

The Reachable Address managed objects may contain a set of SNPA addresses, each of which has an associated time-stamp. The time-stamps shall be initialised to "infinitely old".

Some of the SNPAs in this set may be unreachable. If a call attempt fails to one of the SNPA addresses listed, the IS shall mark that entry in the list with the time of the latest failed attempt. When an SNPA address is to be chosen from the list, the IS shall choose the one with the oldest time-stamp , unless the oldest time-stamp is more recent than recallTimer. If the oldest time-stamp is more recent than recallTimer, all SNPAs in the set shall be assumed temporarily unreachable and no call attempt is made. The IS shall instead invoke the ISO 8473 Discard PDU function.

When attempting to establish a connection to a single specific subnetwork address (not through one of a set of SNPA addresses), if a call attempt to a particular SNPA address, *A*, fails for any reason, the IS shall invoke the ISO 8473 Discard PDU function. Additionally the adjacency on which the call attempt was placed shall be placed in "Failed" state, and the recall timer set. Until it expires, the IS shall not attempt call establishment for future NPDUs to be forwarded over subnetwork address *A*, but instead the IS shall invoke the ISO 8473 Discard PDU function.

When the recall timer expires, the IS shall free the adjacency for calls to a different destination or retry attempts to subnetwork address *A*.

NOTE - If an implementation can store the knowledge of SNPA addresses that have failed along with the time since the attempt was made in a location other than the adjacency on which the call was attempted, then that adjacency can be used for other calls.

8.3.3 Reverse Path Forwarding on DA Circuits

Where a subdomain is attached to a Connection-oriented subnetwork by two or more SNPAs, the IDP for the addresses within the subdomain may be chosen to be constructed from the address of one of the points of attachment. (It need not be. The whole subdomain could be multi-homed by using both SNPA addresses, or some other IDP could be chosen; e.g. ISO DCC.) Traffic to the subdomain from some other SNPA will cause a call to be established to the SNPA corresponding to the IDP of the addresses in the subdomain. Traffic *from* the subdomain may use either of the SNPAs depending on the routeing decisions made by the subdomain. This is illustrated in the diagram below (figure 5).

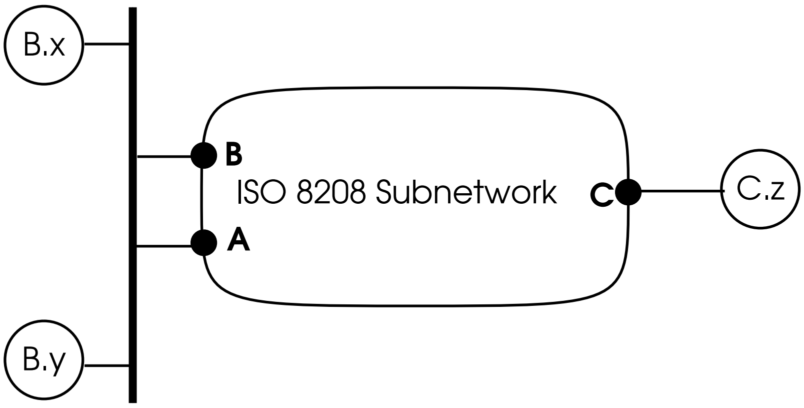


Figure 5 - Example for reverse path forwarding

The subdomain is attached to the connection-oriented subnetwork via SNPAs *A* and *B*. The addresses on the subdomain are constructed using the SNPA address of *B* as the IDI. If traffic for **C.z** is sent from **B.x**, a call will be established from *A* to *C*. The reverse traffic from **C.z** to **B.x** will cause another call to be established from *C* to *B*. Thus two SVCs have been established where only one is required.

This problem is prevented by the local system retaining a cache (known as the ReversePathCache) of NSAP addresses from which traffic has been received over each adjacency. When it has traffic to forward over the connection-oriented subnetwork, the IS shall it first check to see if the destination NSAP is in the cache of any of its adjacencies, and if so forwards the traffic over that adjacency. An NSAP shall only be added to the cache when the remote SNPA address of the adjacency over which it is received differs from the SNPA address to be called which would be generated by checking against the Circuit Reachable Addresses managed objects. If the cache is full, the IS shall overwrite the least recently used entry. The ReversePathCache shall have a size of at least one entry. The IS shall purge the cache when the adjacency is taken down (i.e. when the reserve timer expires).

8.3.4 Use of ISO 9542 on ISO 8208 subnetworks

STATIC and DCM circuits are equivalent to point to point links, and as such permit the operation of ISO 9542 as described for point to point links in 8.2.

For DA circuits, it is impractical to use ISO 9542 to obtain configuration information, such as the location of Intermediate systems, since this would require calls to be established to all possible SNPA addresses.

The IS shall not send ISO 9542 ISH PDUs on a DA circuit. The IS shall take no action on receipt of an ESH PDU or ISH PDU, and the circuit shall complete initialisation without waiting for their arrival.

The IS shall not send Point to point IIH PDU on DA circuits. The IS shall ignore receipt of a point-point IIH PDU. (This would only occur if a STATIC or DCM circuit became erroneously connected to an SVC being used for a DA circuit.)

8.3.5 Interactions with the Update Process

A dynamically assigned circuit contains a list of ⟨reachable address prefix, cost, SNPA address⟩ tuples. When the circuit is Enabled, the Subnetwork Dependent Functions in an Intermediate system shall report (to the Update Process) “adjacency cost change” events for all address prefixes in the circuit Reachable Address managed objects, together with the Reachable Address metric$_k$+ callEstablishmentMetric$_k$Increment. This causes this information to be included in subsequently generated LSPs as described in Clause 7.3.3.2.

Routeing PDUs (LSPs and Sequence number PDUs) shall not be sent on dynamically assigned circuits.

NOTE - In the following sub-clauses, it is assumed that the Reachable Addresses referenced are only those which have been enabled (i.e. that have state “On”), and whose parent circuit is also in state “On”.

8.3.5.1 Adjacency Creation

After an SVC to SNPA address *D* is successfully established and a new adjacency created for it (whether it was initiated by the local or the remote system), if callEstablishmentMetric$_k$Increment is greater than 0, the IS shall scan the circuit Reachable Address managed objects for all addressPrefixes listed with *D* as (one of) the SNPAAddress(es).

For Reachable Addresses with mappingType “Algorithm”, the IS shall construct an *implied address prefix*[13] from the actual remote SNPA address *D* and the address extraction algorithm.

The IS shall generate an Adjacency cost change event for each such address prefix (both actual and implied) with the Reachable Address metric$_k$ (without the added callEstablishmentMetric$_k$Increment). This causes information that those address prefixes are reachable with the lower cost to be included in subsequently generated LSPs. The effect of this is to encourage the use of already established SVCs where possible.

8.3.5.2 Adjacency Deletion

When the adjacency with SNPAAddress *D* is freed (Reserve Timer has expired, or the adjacency is deleted by System Management action) then if callEstablishmentMetric$_k$Increment is greater than 0, the IS shall scan the Circuit Reachable Address managed objects for all those with mappingType “Manual" and (one of) their SNPAAddresses equal to *D*. The IS shall generate “Adjacency cost change" events to the Update Process for all such address prefixes with the Reachable Address + callEstablishmentMetric$_k$Increment. For Reachable Addresses with mappingType “X.121" for which it is possible to construct an *implied address prefix* as above, the IS shall generate an “Adjacency State Change" event for that implied prefix.

A cost change event shall only be generated when the count of the number of subnetwork addresses which have an established SVC changes between 1 and 0.

8.3.5.3 Circuit Call Establishment Increment Change

On a dynamically assigned circuit, when system management changes the Circuit callEstablishmentMetric$_k$Increment for that circuit, the IS shall generate “adjacency cost change events" for all address prefixes affected by the change (i.e. those for which calls are not currently “established").

The IS shall scan all the Reachable Address managed objects of that Circuit. If the Reachable Address has mappingType “X.121", the IS shall generate an “adjacency cost change" event for that name with the Reachable Address metric$_k$+ the new value of callEstablishmentMetric$_k$Increment. If the Reachable Address has mappingType “Manual", the IS shall scan all the Adjacencies of the Circuit for an Adjacency with SNPAAddress equal to (one of) the SNPAAddresses of that Reachable Address. If no such adjacency is found the IS shall generate an “adjacency cost change" event for that name with the Reachable Address metric$_k$ + the new value of callEstablishmentMetric$_k$Increment.

---

[13] i.e. some address prefix which matches the addressPrefix of the Reachable Address, and which would generate the SNPA Address D when the extraction algorithm is applied

8.3.5.4 Reachable Address Cost Change

When the metric$_k$ characteristic of a Reachable Address in state “On" is changed by system management, the IS shall generate cost change events to the Update Process to reflect this change.

If the Reachable Address has mappingType “Manual", the IS shall scan all the Adjacencies of the Circuit for an Adjacency with SNPAAddress equal to (one of) the SNPAAddresses of that Reachable Address. If one or more such adjacencies are found, the IS shall generate an “adjacency cost change" event for that name with the new Reachable Address metric$_k$. If no such adjacency is found the IS shall generate an “adjacency cost change" event or that name with the new Reachable Address metric$_k$.

If the Reachable Address has mappingType “X.121", the IS shall generate an “adjacency cost change" event for that name with the new Reachable Address metric$_k$+ callEstablishmentMetric$_k$Increment. In addition, for all Adjacencies of the Circuit with an SNPAAddress for which an implied address prefix can be generated for this Reachable Address, the IS shall generate an “adjacency cost change" event for that implied address prefix and the new Reachable Address metric$_k$.

8.3.5.5 Disabling a Reachable Address

When a Reachable Address managed object is disabled via management action, the IS shall generate an “Adjacency Down" event to the Update Process for the name of that Reachable Address and also for any implied prefixes associated with that Reachable Address. In addition it shall generate an “Adjacency State Change (Down)" system management event. Any associated adjacencies shall be deleted as follows.

The IS shall scan the Adjacencies on this circuit for any whose SNPAAddress neither matches (one of) the SNPAAddresses of any remaining Reachable Addresses with mappingType “Manual", nor could have an implied address prefix generated from a Reachable Address with mappingType “X.121". For any such adjacencies, the IS shall clear any existing call and delete the adjacency.

8.3.5.6 Enabling a Reachable Address

When a Reachable Address is enabled via system management action, the IS shall generate “Adjacency cost change" events as described for *Reachable Address cost change* in Clause 8.3.5.4 above. In addition it shall generate an “Adjacency State Change(Up)" system management event.

8.4 Broadcast Subnetworks

8.4.1 Broadcast Subnetwork IIH PDUs

All Intermediate systems on broadcast circuits (both Level 1 and Level 2) shall transmit *LAN IIH PDUs* as described in 8.4.3. Level 1 Intermediate systems shall transmit only Level 1 LAN IIH PDUs. Level 2 Intermediate Systems on circuits with manualL2OnlyMode set to the value “True”, shall transmit only Level 2 LAN IIH PDUs. Level 2 Intermediate systems on circuits with manualL2OnlyMode set to the value “False”, shall transmit both.

Level *n* LAN IIH PDUs contain the transmitting Intermediate system’s ID, holding timer, Level *n* Priority and manualAreaAddresses, plus a list containing the LANAddresses of all the adjacencies of adjacencyType “L*n* Intermediate System” (in state “Initialising” or “Up”) on this circuit.

LAN IIH PDUs shall be padded (with trailing PAD options containing arbitrary valued octets) so that the SNSDU containing the IIH PDU has a length of at least *maxsize- 1* octets[14] where *maxsize* for Level 1 IIH PDUs is the maximum of

- dataLinkBlocksize
- originatingL1LSPBufferSize

---

[14] The minimum length of PAD which may be added is 2 octets, since that is the size of the option header. Where possible the PDU should be padded to maxsize, but if the PDU length is *maxsize- 1* octets no padding is possible (or required).

and for Level 2 IIH PDUs is the maximum of

- dataLinkBlocksize
- originatingL2LSPBufferSize

This is done to ensure that an adjacency will only be formed between systems which are capable of exchanging PDUs of length up to *maxsize* octets. In the absence of this check, it would be possible for an adjacency to exist with a lower maximum block size, with the result that some LSPs and SNPs (i.e. those longer than this maximum, but less than *maxsize*) would not be exchanged.

> NOTE - An example of a topology where this could occur is one where an extended LAN is constructed from LAN segments with different maximum block sizes. If, as a result of misconfiguration or some dynamic reconfiguration, a path exists between two Intermediate systems on separate LAN segments having a large maximum block size, which involves transit of a LAN segment with a smaller maximum block size, loss of larger PDUs will occur if the Intermediate systems continue to use the larger maximum block size. It is better to refuse to bring up the adjacency in these circumstances.

Level 1 Intermediate systems shall transmit Level 1 LAN IIH PDUs to the multi-destination address AllL1ISs, and also listen on that address. They shall also listen for ESH PDUs on the multi-destination address AllIntermediateSystems. The list of neighbour Intermediate systems shall contain only Level 1 Intermediate Systems within the same area. (i.e. Adjacencies of adjacencyType “L1 Intermediate System”.)

Level 2 Only Intermediate systems (i.e. Level 2 Intermediate systems which have the Circuit manualL2OnlyMode characteristic set to the value “True”) shall transmit Level 2 LAN IIH PDUs to the multi-destination address AllL2ISs, and also listen on that address. The list of neighbour Intermediate systems shall contain only Level 2 Intermediate systems. (i.e. Adjacencies of adjacencyType “L2 Intermediate System”.)

Level 2 Intermediate systems (with manualL2OnlyMode “False”) shall perform both of the above actions. Separate Level 1 and Level 2 LAN IIH PDUs shall be sent to the multi-destination addresses AllL1ISs and AllL2ISs describing the neighbour Intermediate systems for Level 1 and Level 2 respectively. Separate adjacencies shall be created by the receipt of Level 1 and Level 2 LAN IIH PDUs.

8.4.1.1 Receipt of Level 1 IIH PDUs

On receipt of a Level 1 LAN IIH PDU on the multi-destination address AllL1ISs, the IS shall compare each of the Area Addresses, from the received IIH PDU with the set of area addresses in the manualAreaAddresses characteristic. If a match is not found between any pair (i.e. the local and remote system have no area address in common), the IS shall reject the adjacency and generate an “Initialisation failure (area mismatch)” event. Otherwise (a match is found) the IS shall accept the adjacency and set the Adjacency adjacencyType to “L1 Intermediate System”.

8.4.1.2 Receipt of Level 2 IIH PDUs

On receipt of a Level 2 LAN IIH PDU on the multi-destination address AllL2ISs, the IS shall accept the adjacency, and set the Adjacency adjacencyType to “L2 Intermediate System”.

8.4.1.3 Existing Adjacencies

When a Level *n* LAN IIH PDU (Level 1 or Level 2) is received from an Intermediate system for which there is already an adjacency with

a) the Adjacency LANAddress equal to the MAC Source address of the PDU, **and**

b) the Adjacency neighbourSystemID equal to the Source ID field from the PDU **and**

c) the adjacencyType equal to “L*n* Intermediate System”,

the IS shall update the holding timer, LAN Priority and neighbourAreas according to the values in the PDU.

8.4.1.4 New Adjacencies

When

a) a Level *n* LAN IIH PDU (Level 1 or Level 2) is received (from Intermediate system *R*), and

b) there is no adjacency for which the Adjacency LANAddress is equal to the MAC Source address of the PDU; and

c) the Adjacency neighbourSystemID is equal to the Source ID field from the PDU, and

d) adjacencyType is "L*n* Intermediate System",

the IS shall create a new adjacency. However, if there is insufficient space in the adjacency database, to permit the creation of a new adjacency the IS shall instead perform the actions described in 8.4.2.

The IS shall

a) set adjacencyType status to "L*n* Intermediate System" (where *n* is the level of the IIH PDU),

b) set the holding timer, LAN Priority, neighbour ID and neighbourAreas according to the values in the PDU., and

c) set the LANAddress according to the MAC source address of the PDU.

The IS shall set the state of the adjacency to "initialising", until it is known that the communication between this system and the source of the PDU (*R*) is two-way. However *R* shall be included in future Level *n* LAN IIH PDUs transmitted by this system.

When *R* reports this circuit's LANAddress in its Level *n* LAN IIH PDUs, the IS shall

a) set the adjacency's state to "Up", and

b) generate an "Adjacency State Change (Up)" event.

The IS shall keep a separate Holding Time (Adjacency holdingTimer) for each "L*n* Intermediate System" adjacency. The value of holdingTimer shall be set to the Holding Time as reported in the Holding Timer field of the Level *n* LAN IIH PDUs. If a neighbour is not heard from in that time, the IS shall

a) purge it from the database; and

b) generate an "Adjacency State Change (Down)" event.

If a Level *n* LAN IIH PDU is received from neighbour *N*, and this system's LANAddress is no longer in *N*'s IIH PDU, the IS shall

a) set the adjacency's state to "initialising", and

b) generate an "Adjacency State Change (Down)" event.

8.4.2 Insufficient Space in Adjacency Database

If an IS needs to create a new Intermediate system adjacency, but there is insufficient space in the adjacency database, the adjacency of adjacencyType "L*n* Intermediate System" with lowest LANPriority (or if more than one adjacency has the lowest priority, the adjacency with the lowest LANAddress, from among those with the lowest priority) shall be purged from the database. If the new adjacency would have the lowest priority, it shall be ignored, and a "Rejected Adjacency" event generated.

NOTE - A 48 bit LAN address is treated in numerical comparisons as an unsigned 48 bit integer, with the first byte transmitted, and the leftmost byte as written, treated as the least significant byte.

If an old adjacency must be purged, the IS shall generate an “Adjacency State Change (Down)” event for that adjacency. After the Subnetwork Independent Functions issue an “adjacency down complete”, the IS may create a new adjacency.

8.4.3 Transmission of LAN IIH PDUs

An IS shall transmit a LAN IIH PDU immediately when the circuit has been enabled. A Level 2 Intermediate System (not marked Level2Only) shall transmit both a Level 1 and a Level 2 LAN IIH PDU.

The IS shall also transmit a LAN IIH PDU when at least 1 second has transpired since the last transmission of a LAN IIH PDU of the same type on this circuit by this system and:

a) ISISHelloTimer seconds have elapsed[16] since the last periodic LAN IIH PDU transmission

   The Holding Time is set to ISISHoldingMultiplier × ISISHelloTimer. For a Designated Intermediate System the value of DRISISHelloTimer[17] is used instead of ISIShelloTimer. The Holding Time for this PDU shall therefore be set to ISISHoldingMultiplier × DRISISHelloTimer seconds. This permits failing Designated Intermediate Systems to be detected more rapidly,

   or

b) the contents of the next IIH PDU to be transmitted would differ from the contents of the previous IIH PDU transmitted by this system, or

c) this system has determined that it is to become or resign as LAN Designated Intermediate System for that level.

To minimise the possibility of the IIH PDU transmissions of all Intermediate systems on the LAN becoming synchronised, the Hello Time timer shall only be reset when a IIH PDU is transmitted as a result of timer expiration, or on becoming or resigning as Designated Intermediate System.

Where an Intermediate system is transmitting both Level 1 and Level 2 LAN IIH PDUs, it shall maintain a separate timer (separately jittered) for the transmission of the Level 1 and Level 2 IIH PDUs. This avoids correlation between the Level 1 and Level 2 IIH PDUs and allows the reception buffer requirements to be minimised.

8.4.4 LAN Designated Intermediate Systems

A LAN Designated Intermediate System is the highest priority Intermediate system in a particular set on the LAN, with numerically highest MAC source LANAddress breaking ties. (See above for definition of numerical comparison of LAN addresses.)

There are, in general, two LAN Designated Intermediate Systems on each LAN, namely the LAN Level 1 Designated Intermediate System and the LAN Level 2 Designated Intermediate System. They are elected as follows.

a) Level 1 Intermediate systems elect the LAN Level 1 Designated Intermediate System.

b) Level 2 Only Intermediate systems (i.e. Level 2 Intermediate Systems which have the Circuit manualL2OnlyMode characteristic set to the value “True”) elect the LAN Level 2 Designated Intermediate System.

c) Level 2 Intermediate systems (with manualL2OnlyMode “False”) elect both the LAN Level 1 and LAN Level 2 Designated Intermediate Systems.

---

[16] Jitter is applied as described in Clause 10.1.

[17] In this case jitter is not applied, since it would result in intervals of less than one second.

The set of Intermediate systems to be considered includes the local Intermediate system, together with all Intermediate systems of the appropriate type as follows.

a) For the LAN Level 1 Designated Intermediate System, it is the set of Intermediate systems from which LAN Level 1 IIH PDUs are received and to which Level 1 adjacencies exist in state "Up". When the local system either becomes or resigns as LAN Level 1 Designated Intermediate System, the IS shall generate a "LAN Level 1 Designated Intermediate System Change" event. In addition, when it becomes LAN Level 1 Designated Intermediate System, it shall perform the following actions.

    1) Generate and transmit Level 1 Pseudonode LSPs using the existing End system configuration.

    2) Purge the Level 1 Pseudonode LSPs issued by the previous LAN Level 1 Designated Intermediate System (if any) as described in 7.2.3.

    3) Solicit the new End system configuration as described in Clause 8.4.5.

b) For the LAN Level 2 Designated Intermediate System, it is the set of Intermediate systems from which LAN Level 2 IIH PDUs are received and to which Level 2 adjacencies exist in state "Up". When the local system either becomes or resigns as LAN Level 2 Designated Intermediate System, the IS shall generate a "LAN Level 2 Designated Intermediate System Change" event. In addition, when it becomes LAN Level 2 Designated Intermediate System, it shall perform the following actions.

    1) Generate and transmit a Level 2 Pseudonode LSP.

    2) Purge the Level 2 Pseudonode LSPs issued by the previous LAN Level 2 Designated Intermediate System (if any) as described in 7.2.3.

When an Intermediate system resigns as LAN Level 1 or Level 2 Designated Intermediate System it shall perform the actions on Link State PDUs described in 7.2.3.

When the broadcast circuit is enabled on an Intermediate system the IS shall perform the following actions.

a) Commence sending IIH PDUs with the LAN ID field set to the concatenation of its own SystemID and its locally assigned one octet Local Circuit ID.

b) Solicit the End system configuration as described in Clause 8.4.5.

c) Start listening for ISO 9542 ISH PDUs and ESH PDUs and acquire adjacencies as appropriate. Do not run the Designated Intermediate System election process.

d) After waiting ISISHelloTimer * 2 seconds, run the Level 1 and or the Level 2 Designated Intermediate System election process depending on the Intermediate system type. This shall be run subsequently whenever an IIH PDU is received or transmitted as described in 8.4.3. (For these purposes, the transmission of the system's own IIH PDU is equivalent to receiving it). If there has been no change to the information on which the election is performed since the last time it was run, the previous result can be assumed. The relevant information is

    1) the set of Intermediate system adjacency states,

    2) the set of Intermediate System priorities (including this system's) and

    3) the existence (or otherwise) of at least one "Up" End system (not including Manual Adjacencies) or Intermediate system adjacency on the circuit.

An Intermediate system shall not declare itself to be a LAN Designated Intermediate system of any type until it has at least one "Up" End system (not including Manual Adjacencies) or Intermediate system adjacency on the circuit. (This prevents an Intermediate System which has a defective receiver and is incapable of receiving any PDUs from erroneously electing itself LAN Designated Intermediate System.)

The LAN ID field in the LAN IIH PDUs transmitted by this system shall be set to the value of the LAN ID reported in the LAN IIH PDU (for the appropriate level) received from the system which this system considers to be the Designated Intermediate System. This value shall also be passed to the Update Process, as the pseudonode ID, to enable Link State PDUs to be issued for this system claiming connectivity to the pseudonode.

If this system, as a result of the Designated Intermediate System election process, considers itself to be designated Intermediate System, the LAN ID field shall be set to the concatenation of this system's own ID and the locally assigned one octet Local Circuit ID.

8.4.5 Soliciting the ES configuration

When there is a change in the topology or configuration of the LAN (for example the partitioning of a LAN into two segments by the failure of a repeater or bridge), it is desirable for the (new) Designated Intermediate System to acquire the new End system configuration of the LAN as quickly as possible in order that it may generate Link State PDUs which accurately reflect the actual configuration. This is achieved as follows.

When the circuit is enabled, or the Intermediate system detects a change in the set of Intermediate systems on the LAN, or a change in the Designated Intermediate System ID, the IS shall initiate a poll of the ES configuration by performing the following actions.

a) Delay a random interval between 0 and PollESHelloRate seconds. (This is to avoid synchronisation with other Intermediate systems which have detected the change.)

b) If (and only if) an Intermediate System had been removed from the set of Intermediate systems on the LAN, reset the entryRemainingTime field in the endsystemIDs adjacency database record of all adjacencies on this circuit to the value

   (ISISHelloTimer + PollESHelloRate) × HoldingMultiplier

   or the existing value whichever is the lower. (This causes any End systems which are no longer present on the LAN to be rapidly timed out, but not before they have a chance to respond to the poll.)

c) Transmit HoldingMultiplier ISH PDUs for each NET possessed by the Intermediate system with a Suggested ES Configuration Timer value of PollESHelloRate at an interval between them of ISISHelloTimer seconds and a holding time of helloTimer * HoldingMultiplier.

d) Resume sending ISH PDUs at intervals of helloTimer seconds with a Suggested ES Configuration Timer value of DefaultESHelloTimer.

8.4.6 Receipt of ESH PDUs – Database of End Systems

An IS shall enter an End system into the adjacency database when an ESH PDU is received from a new data link address. If an ESH PDU is received with the same data link address as a current adjacency, but with a different NSAP address, the new address shall be added to the adjacency, with a separate timer. A single ESH PDU may contain more than one NSAP address. When a new data link address or NSAP address is added to the adjacency database, the IS shall generate an "Adjacency State Change (Up)" event on that adjacency.

The IS shall set a timer for the value of "Holding Time" in the received ESH PDU. If another ESH PDU is not received from the ES before that timer expires, the ES shall be purged from the database, provided that the Subnetwork Independent Functions associated with initialising the adjacency have been completed. Otherwise the IS shall clear the adjacency as soon as those functions are completed.

When the adjacency is cleared, the Subnetwork Independent Functions shall be informed of an "Adjacency Change (Down)" event, and the adjacency can be re-used after the Subnetwork Independent Functions associated with bringing down the adjacency have been completed.

9 Structure and Encoding of PDUs

This clause describes the PDU formats of the Intra-Domain Routeing protocol.

Octets in a PDU are numbered starting from 1, in increasing order. Bits in a octet are numbered from 1 to 8, where bit 1 is the least significant bit and is pictured on the right. When consecutive octets are used to represent a number, the lower octet number has the most significant value.

Fields marked “Reserved” are transmitted as zero, and ignored on receipt, unless otherwise noted.

Values are given in decimal. All numeric fields are unsigned integers, unless otherwise noted.

Network Layer addresses (NSAP addresses, NETs, Area Addresses and Address Prefixes) are encoded in PDUs according to the preferred binary encoding specified in ISO 8348/Add.2; the entire address, taken as a whole is represented explicitly as a string of binary octets. This string is conveyed in its entirety in the address fields of the PDUs. The rules governing the generation of the preferred binary encoding are described in ISO 8348/Add.2. The address so generated is encoded with the most significant octet (i.e. the AFI) of the address being the first octet transmitted, and the more significant semi-octet of each pair of semi-octets in the address is encoded in the more significant semi-octet of each octet (i.e. in the high order 4 bits). Thus the address `/371234` is encoded as

		No. of Octets
3	7	1
1	2	1
3	4	1

Figure 6 - Address encoding example

The types of PDUs are:

- Level 1 LAN IS to IS Hello PDU
- Level 2 LAN IS to IS Hello PDU
- Point-to-Point IS to IS Hello PDU
- Level 1 Link State PDU
- Level 2 Link State PDU
- Level 1 Complete Sequence Numbers PDU
- Level 2 Complete Sequence Numbers PDU
- Level 1 Partial Sequence Numbers PDU
- Level 2 Partial Sequence Numbers PDU

9.1 Level 1 LAN IS to IS Hello PDU

This PDU is multicast by Intermediate systems on broadcast circuits to the multi-destination address AllL1ISs. The purpose of this PDU is for Intermediate systems on broadcast circuits to discover the identity of other Level 1 Intermediate systems on that circuit. Trailing pad options are inserted to make PDU LENGTH equal to at least *maxsize - 1* where *maxsize* is the maximum of

- dataLinkBlocksize
- originatingL1LSPBufferSize

(see 8.4.1).

	No. of Octets
INTRADOMAIN ROUTEING PROTOCOL DISCRIMINATOR	1
LENGTH INDICATOR	1
VERSION/PROTOCOL ID EXT	1
RESERVED	1
R R R TYPE	1
VERSION	1
ECO	1
USER ECO	1
RESERVED/CIRCUIT TYPE	1
SOURCE ID	6
HOLDING TIMER	2
PDU LENGTH	2
RES PRIORITY	1
LAN ID	7
VARIABLE LENGTH FIELDS	VARIABLE

- INTRADOMAIN ROUTEING PROTOCOL DISCRIMINATOR – architectural constant
- LENGTH INDICATOR – Header Length in octets (27.)
- VERSION/PROTOCOL ID EXTENSION – 1
- RESERVED – transmitted as 0, ignored on receipt
- TYPE (bits 1 through 5) – 15. Note bits 6, 7 and 8 are Reserved, which means they are transmitted as 0 and ignored on receipt.
- VERSION – 1

- ECO — transmitted as zero, ignored on receipt
- USER ECO — transmitted as zero, ignored on receipt
- RESERVED/CIRCUIT TYPE – Most significant 6 bits reserved (Transmitted as zero, ignored on receipt). Low order bits (bits 1 and 2) indicate:
    - 0 – reserved value (if specified the entire PDU shall be ignored)
    - 1 – Level 1 only
    - 2 – Level 2 only (sender is Level 2 Intermediate system with manualL2OnlyMode set “True” for this circuit, and will use this link only for Level 2 traffic)
    - 3 – both Level 1 and Level 2 (sender is Level 2 Intermediate system, and will use this link both for Level 1 and Level 2 traffic)

  NOTE — In a LAN Level 1 IIH PDU the CIRCUIT TYPE shall be either 1 or 3.

- SOURCE ID – 6 octet unique ID of transmitting Intermediate system
- HOLDING TIMER – Holding Timer to be used for this Intermediate system
- PDU LENGTH – Entire length of this PDU, in octets, including header
- RES/PRIORITY – Bit 8 reserved (Transmitted as zero, ignored on receipt). Bits 1 through 7 – priority for being LAN Level 1 Designated Intermediate System. Higher number has higher priority for being LAN Level 1 Designated Intermediate System. Unsigned integer.
- LAN ID – 7 octet field composed from the 6 octet unique ID of the LAN Level 1 Designated Intermediate System, plus low order octet assigned by LAN Level 1 Designated Intermediate System. Copied from LAN Level 1 Designated Intermediate System’s IIH PDU.
- VARIABLE LENGTH FIELDS – fields of the form:

	No. of Octets
CODE	1
LENGTH	1
VALUE	LENGTH

Any codes in a received PDU that are not recognised shall be ignored.

Currently defined codes are:

- Area addresses – the set of manualAreaAddresses of this Intermediate System.
    - * CODE – 1
    - * LENGTH – total length of the value field.

  - VALUE –

	No. of Octets
ADDR LENGTH	1
AREA ADDRESS	ADDR LENGTH
ADDR LENGTH	1
AREA ADDRESS	ADDR LENGTH

    - x ADDR LENGTH – Length of Area Address in octets.
    - x AREA ADDRESS – Area address.

- Intermediate System Neighbours – The set of Intermediate systems on this LAN to which adjacencies of adjacencyType “L1 Intermediate System” exist in state “Up” or “Initialising” (i.e. those from which Level 1 IIH PDUs have been heard).
  - CODE – 6
  - LENGTH – total length of the value field.
  - VALUE –

	No. of Octets
LANADDRESS	6
LAN ADDRESS	6

    - x LAN ADDRESS – 6 octet MAC Address of Intermediate System neighbour.

- Padding – This option may occur multiple times. It is used to pad the PDU to at least *maxsize* - 1.
  - CODE – 8.
  - LENGTH – total length of the value field (may be zero).
  - VALUE – LENGTH octets of arbitrary value.

9.2 Level 2 LAN IS to IS Hello PDU

This PDU is multicast by Intermediate systems on broadcast circuits to the multi-destination address AllL2ISs. The purpose of this PDU is for Intermediate systems on broadcast circuits to discover the identity of other Level 2 Intermediate systems on that circuit. Trailing pad options are inserted to make PDU LENGTH equal to at least *maxsize - 1* where

- dataLinkBlocksize
- originatingL2LSPBufferSize

(see Clause 8.4.1).

	No. of Octets
INTRADOMAIN ROUTEING PROTOCOL DISCRIMINATOR	1
LENGTH INDICATOR	1
VERSION/PROTOCOL ID EXT	1
RESERVED	1
R R R TYPE	1
VERSION	1
ECO	1
USER ECO	1
RESERVED/CIRCUIT TYPE	1
SOURCE ID	6
HOLDING TIMER	2
PDU LENGTH	2
RES PRIORITY	1
LAN ID	7
VARIABLE LENGTH FIELDS	VARIABLE

- INTRADOMAIN ROUTEING PROTOCOL DISCRIMINATOR – architectural constant
- LENGTH INDICATOR – Header Length in octets (27.)
- VERSION/PROTOCOL ID EXTENSION – 1
- RESERVED – transmitted as 0, ignored on receipt
- TYPE (bits 1 through 5) – 16. Note bits 6, 7 and 8 are Reserved, which means they are transmitted as 0 and ignored on receipt.
- VERSION – 1
- ECO — transmitted as zero, ignored on receipt

- USER ECO — transmitted as zero, ignored on receipt
- RESERVED/CIRCUIT TYPE – Most significant 6 bits reserved (Transmitted as zero, ignored on receipt). Low order bits (bits 1 and 2) indicate:
    - 0 – reserved value (if specified the entire PDU shall be ignored)
    - 1 – Level 1 only
    - 2 – Level 2 only (sender is Level 2 Intermediate System with manualL2OnlyMode set “True” for this circuit, and will use this link only for Level 2 traffic)
    - 3 – both Level 1 and Level 2 (sender is Level 2 Intermediate System, and will use this link both for Level 1 and Level 2 traffic)

NOTE — In a LAN Level 2 IIH PDU the CIRCUIT TYPE shall be either 2 or 3.

- SOURCE ID – 6 octet unique ID of transmitting Intermediate System
- HOLDING TIMER – Holding Timer to be used for this Intermediate System
- PDU LENGTH – Entire length of this PDU, in octets, including header
- RES/PRIORITY – Bit 8 reserved (Transmitted as zero, ignored on receipt). Bits 1 through 7 – priority for being LAN Level 2 Designated Intermediate System. Higher number has higher priority for being LAN Level 2 Designated Intermediate System. Unsigned integer.
- LAN ID – 7 octet field composed from the 6 octet unique ID of the LAN Level 2 Designated Intermediate System, plus low order octet assigned by LAN Level 2 Designated Intermediate System. Copied from LAN Level 2 Designated Intermediate System’s IIH PDU.
- VARIABLE LENGTH FIELDS – fields of the form:

	No. of Octets
CODE	1
LENGTH	1
VALUE	LENGTH

Any codes in a received PDU that are not recognised shall be ignored.

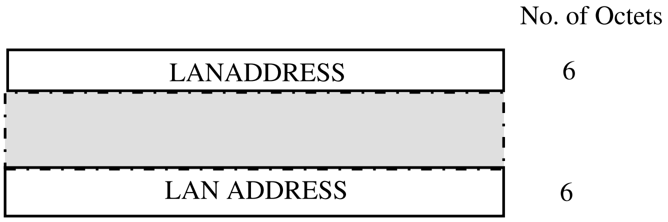
Currently defined codes are:

- Area addresses – the set of manualAreaAddresses of this Intermediate system.
    - * CODE – 1
    - * LENGTH – total length of the value field.
    - * VALUE –

	No. of Octets
ADDR LENGTH	1
AREA ADDRESS	ADDR LENGTH
ADDR LENGTH	1
AREA ADDRESS	ADDR LENGTH

    - x ADDR LENGTH – Length of Area Address in octets.
    - x AREA ADDRESS – Area address.

- **Intermediate System Neighbours** – The set of Intermediate systems on this LAN to which adjacencies of **adjacencyType** “L2 Intermediate System” exist in **state** “Up” or “Initialising” (i.e. those from which Level 2 IIH PDUs have been heard).
  - * CODE – 6
  - * LENGTH – total length of the value field.
  - * VALUE –



  - * LAN ADDRESS – 6 octet MAC Address of Intermediate System neighbour

- **Padding** – This option may occur multiple times. It is used to pad the PDU to at least *maxsize* 1.
  - * CODE – 8.
  - * LENGTH – total length of the value field (may be zero).
  - * VALUE – LENGTH octets of arbitrary value.

9.3 Point-to-Point IS to IS Hello PDU

This PDU is transmitted by Intermediate systems on non-broadcast circuits, after receiving an ISH PDU from the neighbour system. Its purpose is to determine whether the neighbour is a Level 1 or a Level 2 Intermediate System. Trailing pad options are inserted to make PDU LENGTH equal to at least *maxsize 1* where *maxsize* is the maximum of

- dataLinkBlocksize
- originatingL1LSPBufferSize
- originatingL2LSPBufferSize

(see Clause 8.2.3).

	No. of Octets
INTRADOMAIN ROUTEING PROTOCOL DISCRIMINATOR	1
LENGTH INDICATOR	1
VERSION/PROTOCOL ID EXT	1
RESERVED	1
R | R | R | TYPE	1
VERSION	1
ECO	1
USER ECO	1
RESERVED/CIRCUIT TYPE	1
SOURCE ID	6
HOLDING TIMER	2
PDU LENGTH	2
LOCAL CIRCUIT ID	1
VARIABLE LENGTH FIELDS	VARIABLE

- INTRADOMAIN ROUTEING PROTOCOL DISCRIMINATOR – architectural constant
- LENGTH INDICATOR – Header Length in octets (20.)
- VERSION/PROTOCOL ID EXTENSION – 1
- RESERVED – transmitted as 0, ignored on receipt
- TYPE — (bits 1 through 5) – 17. Note bits 6, 7 and 8 are Reserved, which means they are transmitted as 0 and ignored on receipt.
- VERSION – 1

- ECO — transmitted as zero, ignored on receipt
- USER ECO — transmitted as zero, ignored on receipt
- RESERVED/CIRCUIT TYPE – Most significant 6 bits reserved (Transmitted as zero, ignored on receipt). Low order bits (bits 1 and 2) indicate:
  - 0 – reserved value (if specified the entire PDU shall be ignored)
  - 1 – Level 1 only
  - 2 – Level 2 only (sender is Level 2 Intermediate system with manualL2OnlyMode set “True” for this circuit, and will use this link only for Level 2 traffic)
  - 3 – both Level 1 and Level 2 (sender is Level 2 Intermediate system and will use this link both for Level 1 and Level 2 traffic)
- SOURCE ID – 6 octet unique ID of transmitting Intermediate system
- HOLDING TIMER – Holding Timer to be used for this Intermediate system
- PDU LENGTH – Entire length of this PDU, in octets, including header
- LOCAL CIRCUIT ID – 1 octet unique ID assigned to this circuit when it is created by this Intermediate system. The actual ID by which the circuit is known to both ends of the link is determined by the Intermediate system with the lower SOURCE ID.
- VARIABLE LENGTH FIELDS – fields of the form:

	No. of Octets
CODE	1
LENGTH	1
VALUE	LENGTH

Any codes in a received PDU that are not recognised shall be ignored.

Currently defined codes are:

- Area addresses – the set of manualAreaAddresses of this Intermediate system
  - * CODE – 1
  - * LENGTH – total length of the value field.
  - * VALUE –

	No. of Octets
ADDR LENGTH	1
ARE A ADDRESS	ADDR LENGTH
ADDR LENGTH	1
AREA ADDRESS	ADDR LENGTH

x ADDR LENGTH – Length of Area Address in octets.
x AREA ADDRESS – Area address.

- Padding – This option may occur multiple times. It is used to pad the PDU to at least *maxsize* 1.

- CODE – 8.
- LENGTH – total length of the value field (may be zero).
- VALUE – LENGTH octets of arbitrary value.

9.4 Level 1 Link State PDU

Level 1 Link State PDUs are generated by Level 1 and Level 2 Intermediate systems, and propagated throughout an area. The contents of the Level 1 Link State PDU indicates the state of the adjacencies to neighbour Intermediate Systems, or pseudonodes, and End systems of the Intermediate system that originally generated the PDU.

	No. of Octets
INTRADOMAIN ROUTEING PROTOCOL DISCRIMINATOR	1
LENGTH INDICATOR	1
VERSION/PROTOCOL ID EXT	1
RESERVED	1
R R R TYPE	1
VERSION	1
ECO	1
USER ECO	1
PDU LENGTH	2
REMAINING LIFETIME	2
LSP ID	8
SEQUENCE NUMBER	4
CHECKSUM	2
P ATT HIPPITY IS TYPE	1
VARIABLE LENGTH FIELDS	VARIABLE

- INTRADOMAIN ROUTEING PROTOCOL DISCRIMINATOR – architectural constant
- LENGTH INDICATOR – Header Length in octets (27.)
- VERSION/PROTOCOL ID EXTENSION – 1
- RESERVED – Transmitted as 0, ignored on receipt
- TYPE (bits 1 through 5) – 18. Note bits 6, 7 and 8 are Reserved, which means they are transmitted as 0 and ignored on receipt.
- VERSION – 1
- ECO — transmitted as zero, ignored on receipt
- USER ECO — transmitted as zero, ignored on receipt
- PDU LENGTH – Entire Length of this PDU, in octets, including header
- REMAINING LIFETIME – Number of seconds before LSP considered expired

- LSP ID – 8 octet ID of the source of the Link State PDU. It is structured as follows:

	No. of Octets
SOURCE ID	6
PSEUDONODE ID	1
LSP NUMBER	1

- SEQUENCE NUMBER – sequence number of LSP
- CHECKSUM – Checksum of contents of LSP from SOURCE ID to end. Checksum is computed as described in 7.3.11.
- P/ATT/HIPPITY/IS TYPE
  - P – Bit 8, indicates when set that the issuing Intermediate System supports the Partition Repair optional function.
  - ATT - Bits 7-4 indicate, when set, that the issuing Intermediate System is ‘attached’ to other areas using:
    - * Bit 4 - the Default Metric
    - * Bit 5 - the Delay Metric
    - * Bit 6 - the Expense Metric
    - * Bit 7 - the Error Metric.
  - HIPPITY – Bit 3 – A value of 0 indicates zero hippity cost, and a value of 1 indicates infinite hippity cost. An LSP with this bit set will not be used by any decision process to calculate routes through the originating system.
  - Intermediate System TYPE – Bits 1 and 2 indicate the type of Intermediate System – One of the following values:
    - * 0 – Unused value
    - * 1 – (i.e. bit 1 set) Level 1 Intermediate system
    - * 2 – Unused value
    - * 3 – (i.e. bits 1 and 2 set) Level 2 Intermediate system.
- VARIABLE LENGTH FIELDS – fields of the form:

	No. of Octets
CODE	1
LENGTH	1
VALUE	LENGTH

Any codes in a received LSP that are not recognised are ignored and passed through unchanged.

Currently defined codes are:

- Area addresses – the set of manualAreaAddresses of this Intermediate system. For LSPs not generated on behalf of the pseudonode this option shall always be present in the LSP with LSP number zero, and shall never be present in an LSP with non-zero LSP number. It shall appear before any Intermediate System Neighbours or End system Neighbours options. This option shall never be present in pseudonode LSPs.
  - * CODE – 1
  - * LENGTH – total length of the value field.

* VALUE –

	No. of Octets
ADDR LENGTH	1
AREA ADDRESS	ADDR LENGTH
ADDR LENGTH	1
AREA ADDRESS	ADDR LENGTH

x ADDR LENGTH – Length of Area Address in octets.
x AREA ADDRESS – Area address.

- Intermediate System Neighbours – Intermediate system and pseudonode neighbours.

This is permitted to appear multiple times, and in an LSP with any LSP number. However, all the Intermediate system Neighbours options shall precede the End system Neighbours options. i.e. they shall appear before any End system Neighbour options in the same LSP and no End system Neighbour options shall appear in an LSP with lower LSP number.

* CODE – 2.

* LENGTH – 1. plus a multiple of 11.

* VALUE –

			No. of Octets
VIRTUAL FLAG			1
0	R	DEFAULT METRIC	1
S	R	DELAY METRIC	1
S	R	EXPENSE METRIC	1
S	R	ERROR METRIC	1
Neighbour ID			7
0	R	DEFAULT METRIC	1
S	R	DELAY METRIC	1
S	R	EXPENSE METRIC	1
S	R	ERROR METRIC	1
Neighbour ID			7

x VIRTUAL FLAG is a Boolean. If equal to 1, this indicates the link is really a Level 2 path to repair an area partition. (Level 1 Intermediate Systems would always report this octet as 0 to all neighbours).

x DEFAULT METRIC is the value of the default metric for the link to the listed neighbour. Bits 8 and 7 of this field are reserved.

x DELAY METRIC is the value of the delay metric for the link to the listed neighbour. If this IS does not support this metric it shall set the bit “S” to 1 to indicate that the metric is unsupported. Bit 7 of this field is reserved.

- x EXPENSE METRIC is the value of the expense metric for the link to the listed neighbour. If this IS does not support this metric it shall set the bit “S” to 1 to indicate that the metric is unsupported. Bit 7 of this field is reserved.
- x ERROR METRIC is the value of the error metric for the link to the listed neighbour. If this IS does not support this metric it shall set the bit “S” to 1 to indicate that the metric is unsupported. Bit 7 of this field is reserved.
- x Neighbour ID is 7 octets. For Intermediate System neighbours, the first 6 octets are the neighbour’s ID, and the 7th octet is 0. For pseudonode neighbours, the first 6 octets is the LAN Level 1 Designated Intermediate System’s ID, and the 7th octet is a non-zero quantity defined by the LAN Level 1 Designated Intermediate System.

\- End System Neighbours – End system neighbours

This may appear multiple times, and in an LSP with any LSP number. See the description of the Intermediate System Neighbours option above for the relative ordering constraints. Only adjacencies with identical costs can appear in the same list.

- * CODE – 3.
- * LENGTH – 4. plus a multiple of 6.
- * VALUE –

			No. of Octets
0	R	DEFAULT METRIC	1
S	R	DELAY METRIC	1
S	R	EXPENSE METRIC	1
S	R	ERROR METRIC	1
Neighbour ID			6
Neighbour ID			6

- x DEFAULT METRIC is the value of the default metric for the link to each of the listed neighbours. Bits 8 and 7 of this field are reserved.
- x DELAY METRIC is the value of the delay metric for the link to each of the listed neighbours. If this IS does not support this metric it shall set the bit “S” to 1 to indicate that the metric is unsupported. Bit 7 of this field is reserved.
- x EXPENSE METRIC is the value of the expense metric for the link to each of the listed neighbours. If this IS does not support this metric it shall set the bit “S” to 1 to indicate that the metric is unsupported. Bit 7 of this field is reserved
- x ERROR METRIC is the value of the error metric for the link to each of the listed neighbour. If this IS does not support this metric it shall set the bit “S” to 1 to indicate that the metric is unsupported. Bit 7 of this field is reserved
- x Neighbour ID – 6 octet ID of End system neighbour.

9.5 Level 2 Link State PDU

Level 2 Link State PDUs are generated by Level 2 Intermediate systems, and propagated throughout the level 2 domain. The contents of the Level 2 Link State PDU indicates the state of the adjacencies to neighbour Level 2 Intermediate Systems, or pseudonodes, and to reachable address prefixes of the Intermediate system that originally generated the PDU.

<table>
<tr><th colspan="4"></th><th>No. of Octets</th></tr>
<tr><td colspan="4">INTRADOMAIN ROUTEING PROTOCOL DISCRIMINATOR</td><td>1</td></tr>
<tr><td colspan="4">LENGTH INDICATOR</td><td>1</td></tr>
<tr><td colspan="4">VERSION/PROTOCOL ID EXT</td><td>1</td></tr>
<tr><td colspan="4">RESERVED</td><td>1</td></tr>
<tr><td>R</td><td>R</td><td>R</td><td>TYPE</td><td>1</td></tr>
<tr><td colspan="4">VERSION</td><td>1</td></tr>
<tr><td colspan="4">ECO</td><td>1</td></tr>
<tr><td colspan="4">USER ECO</td><td>1</td></tr>
<tr><td colspan="4">PDU LENGTH</td><td>2</td></tr>
<tr><td colspan="4">REMAINING LIFETIME</td><td>2</td></tr>
<tr><td colspan="4">LSP ID</td><td>8</td></tr>
<tr><td colspan="4">SEQUENCE NUMBER</td><td>4</td></tr>
<tr><td colspan="4">CHECKSUM</td><td>2</td></tr>
<tr><td>P</td><td>ATT</td><td>HIPPITY</td><td>IS TYPE</td><td>1</td></tr>
<tr><td colspan="4">VARIABLE LENGTH FIELDS</td><td>VARIABLE</td></tr>
</table>

- INTRADOMAIN ROUTEING PROTOCOL DISCRIMINATOR – architectural constant

- LENGTH INDICATOR – Header Length in octets (27.)

- VERSION/PROTOCOL ID EXTENSION – 1

- RESERVED – Transmitted as 0, ignored on receipt

- TYPE (bits 1 through 5) – 20. Note bits 6, 7 and 8 are Reserved, which means they are transmitted as 0 and ignored on receipt.

- VERSION – 1

- ECO — transmitted as zero, ignored on receipt

- USER ECO — transmitted as zero, ignored on receipt

- PDU LENGTH – Entire Length of this PDU, in octets, including header.

- REMAINING LIFETIME – Number of seconds before LSP considered expired

- LSP ID – 8 octet ID of the source of the Link State PDU. It is structured as follows:

	No. of Octets
SOURCE ID	6
PSEUDONODE ID	1
LSP NUMBER	1

- SEQUENCE NUMBER – sequence number of LSP
- CHECKSUM – Checksum of contents of LSP from SOURCE ID to end. Checksum is computed as described in Clause 7.3.11.
- P/ATT/HIPPITY/IS TYPE
  - P – Bit 8, indicates when set that the issuing Intermediate System supports the Partition Repair optional function.
  - ATT - Bits 7-4 indicate, when set, that the issuing Intermediate System is ‘attached’ to other areas using:
    - Bit 4 - the Default Metric
    - Bit 5 - the Delay Metric
    - Bit 6 - the Expense Metric
    - Bit 7 - the Error Metric.
  - HIPPITY – Bit 3 – A value of 0 indicates zero hippity cost, and a value of 1 indicates infinite hippity cost. An LSP with this bit set will not be used by any decision process to calculate routes through the originating system.
  - Intermediate System TYPE – Bits 1 and 2 indicate the type of Intermediate System – One of the following values:
    - 0 – Unused value
    - 1 – (i.e. bit 1 set) Level 1 Intermediate system
    - 2 – Unused value
    - 3 – (i.e. bits 1 and 2 set) Level 2 Intermediate system.

  Note — In a Level 2 Link State PDU, IS TYPE shall be 3.

- VARIABLE LENGTH FIELDS – fields of the form:

	No. of Octets
CODE	1
LENGTH	1
VALUE	LENGTH

Any codes in a received LSP that are not recognised are ignored and passed through unchanged.

Currently defined codes are:

- Area addresses – the set of partitionAreaAddresses of this Intermediate system. For non-pseudonode LSPs this option shall always be present in the LSP with LSP number zero, and shall never be present in an LSP with non-zero LSP number. It shall appear before any Intermediate System Neighbours or Prefix Neighbours options. This option shall never be present in pseudonode LSPs.
  - CODE – 1

- * LENGTH – total length of the value field.
- * VALUE –

	No. of Octets
ADDR LENGTH	1
AREA ADDRESS	ADDR LENGTH
ADDR LENGTH	1
AREA ADDRESS	ADDR LENGTH

- x ADDR LENGTH – Length of Area Address in octets.
- x AREA ADDRESS – Area address.

- **Partition Designated Level 2 Intermediate System** – ID of Designated Level 2 Intermediate System for the partition. For non-pseudonode LSPs issued by Intermediate Systems which support the partition repair optional function this option shall always be present in the LSP with LSP number zero, and shall never be present in an LSP with non-zero LSP number. It shall appear before any Intermediate System Neighbours or Prefix Neighbours options. This option shall never be present in pseudonode LSPs.
  - * CODE – 4.
  - * LENGTH – 6
  - * VALUE – ID of Partition Designated Level 2 Intermediate System for the partition.
- **Intermediate System Neighbours** – Intermediate system and pseudonode neighbours.

  This is permitted to appear multiple times, and in an LSP with any LSP number. However, all the Intermediate System Neighbours options shall precede the Prefix Neighbours options. i.e. they shall appear before any Prefix Neighbour options in the same LSP and no Prefix Neighbour options shall appear in an LSP with lower LSP number.
  - * CODE – 2.
  - * LENGTH – 1. plus a multiple of 11.
  - * VALUE –

			No. of Octets
RESERVED			1
0	R	DEFAULT METRIC	1
S	R	DELAY METRIC	1
S	R	EXPENSE METRIC	1
S	R	ERROR METRIC	1
Neighbour ID			7
0	R	DEFAULT METRIC	1
S	R	DELAY METRIC	1
S	R	EXPENSE METRIC	1
S	R	ERROR METRIC	1
Neighbour ID			7

- x RESERVED — transmitted as zero ignored on receipt.
- x DEFAULT METRIC is the value of the default metric for the link to the listed neighbour. Bits 8 and 7 of this field are reserved.
- x DELAY METRIC is the value of the delay metric for the link to the listed neighbour. If this IS does not support this metric it shall set bit “S” to 1 to indicate that the metric is unsupported. Bit 7 of this field is reserved.
- x EXPENSE METRIC is the value of the expense metric for the link to the listed neighbour. If this IS does not support this metric it shall set bit “S” to 1 to indicate that the metric is unsupported. Bit 7 of this field is reserved.
- x ERROR METRIC is the value of the error metric for the link to the listed neighbour. If this IS does not support this metric it shall set bit “S” to 1 to indicate that the metric is unsupported. Bit 7 of this field is reserved.
- x Neighbour ID is 7 octets. For Intermediate System neighbours, the first 6 octets are the neighbour’s ID, and the 7th octet is 0. For pseudonode neighbours, the first 6 octets is the LAN Level 2 Designated Intermediate System’s ID, and the 7th octet is a non-zero quantity defined by the LAN Level 2 Designated Intermediate System.

- Prefix Neighbours – reachable address prefix neighbours

This may appear multiple times, and in an LSP with any LSP number. See the description of the Intermediate System Neighbours option above for the relative ordering constraints. Only adjacencies with identical costs can appear in the same list.

- * CODE – 5.
- * LENGTH – Total length of the VALUE field.
- * VALUE –

			No. of Octets
0	R	DEFAULT METRIC	1
S	R	DELAY METRIC	1
S	R	EXPENSE METRIC	1
S	R	ERROR METRIC	1
ADDRESS PREFIX LENGTH			1
ADDRESS PREFIX			⌈ADDRESS PREFIX LENGTH /2⌉
ADDRESS PREFIX LENGTH			1
ADDRESS PREFIX			⌈ADDRESS PREFIX LENGTH /2⌉

- x DEFAULT METRIC is the value of the default metric for the link to each of the listed neighbours. Bits 8 and 7 of this field are reserved.
- x DELAY METRIC is the value of the delay metric for the link to each of the listed neighbours. If this IS does not support this metric it shall set the bit “S” to 1 to indicate that the metric is unsupported.
- x EXPENSE METRIC is the value of the expense metric for the link to each of the listed neighbours. If this IS does not support this metric it shall set the bit “S” to 1 to indicate that the metric is unsupported.
- x ERROR METRIC is the value of the error metric for the link to each of the listed neighbour. If this IS does not support this metric it shall set the bit “S” to 1 to indicate that the metric is unsupported.
- x ADDRESS PREFIX LENGTH is the length in semi-octets of the following prefix. A length of zero indicates a prefix that matches all NSAPs.

x ADDRESS PREFIX is a reachable address prefix encoded as described in Clause 7.1.4. If the length in semi-octets is odd, the prefix is padded out to an integral number of octets with a trailing zero semi-octet.

Note that the area addresses listed in the Area Addresses option of Level 2 Link State PDU with LSP number zero, are understood to be reachable address neighbours with cost 0. They are not listed separately in the Prefix Neighbours options.

9.6 Level 1 Complete Sequence Numbers PDU

	No. of Octets
INTRADOMAIN ROUTEING PROTOCOL DISCRIMINATOR	1
LENGTH INDICATOR	1
VERSION/PROTOCOL ID EXT	1
RESERVED	1
R | R | R | TYPE	1
VERSION	1
ECO	1
USER ECO	1
PDU LENGTH	2
SOURCE ID	7
START LSP ID	8
END LSP ID	8
VARIABLE LENGTH FIELDS	VARIABLE

- INTRADOMAIN ROUTEING PROTOCOL DISCRIMINATOR – architectural constant
- LENGTH INDICATOR – Header Length in octets (33.)
- VERSION/PROTOCOL ID EXTENSION – 1
- RESERVED – transmitted as 0, ignored on receipt
- TYPE (bits 1 through 5) – 24. Note bits 6, 7 and 8 are Reserved, which means they are transmitted as 0 and ignored on receipt.
- VERSION – 1
- ECO — transmitted as zero, ignored on receipt
- USER ECO — transmitted as zero, ignored on receipt
- PDU LENGTH – Entire Length of this PDU, in octets, including header
- SOURCE ID – 7 octet ID of Intermediate System (with zero Circuit ID) generating this Sequence Numbers PDU.
- START LSP ID – 8 octet ID of first LSP in the range covered by this Complete Sequence Numbers PDU.

- END LSP ID – 8 octet ID of last LSP in the range covered by this Complete Sequence Numbers PDU.
- VARIABLE LENGTH FIELD – a list of LSP entries of the form:

	No. of Octets
REMAINING LIFETIME	2
LSP ID	8
LSP SEQ NUMBER	4
CHECKSUM	2
REMAINING LIFETIME	2
LSP ID	8
LSP SEQ NUMBER	4
CHECKSUM	2

- REMAINING LIFETIME – Remaining Lifetime of LSP.
- LSP ID – 8 octet ID of the LSP to which this entry refers.
- LSP SEQ NUMBER – Sequence number of LSP.
- CHECKSUM – Checksum reported in LSP.

The entries shall be sorted into ascending LSPID order (the LSP number octet of the LSPID is the least significant octet).

9.7 Level 2 Complete Sequence Numbers PDU

	No. of Octets
INTRADOMAIN ROUTEING PROTOCOL DISCRIMINATOR	1
LENGTH INDICATOR	1
VERSION/PROTOCOL ID EXT	1
RESERVED	1
R R R TYPE	1
VERSION	1
ECO	1
USER ECO	1
PDU LENGTH	2
SOURCE ID	7
START LSP ID	8
END LSP ID	8
VARIABLE LENGTH FIELDS	VARIABLE

- INTRADOMAIN ROUTEING PROTOCOL DISCRIMINATOR – architectural constant
- LENGTH INDICATOR – Header Length in octets (33.)
- VERSION/PROTOCOL ID EXTENSION – 1
- RESERVED – transmitted as 0, ignored on receipt
- TYPE (bits 1 through 5) – 25. Note bits 6, 7 and 8 are Reserved, which means they are transmitted as 0 and ignored on receipt.
- VERSION – 1
- ECO — transmitted as zero, ignored on receipt
- USER ECO — transmitted as zero, ignored on receipt
- PDU LENGTH – Entire Length of this PDU, in octets, including header
- SOURCE ID – 7 octet ID of Intermediate System (with zero Circuit ID) generating this Sequence Numbers PDU.
- START LSP ID – 8 octet ID of first LSP in the range covered by this Complete Sequence Numbers PDU.
- END LSP ID – 8 octet ID of last LSP in the range covered by this Complete Sequence Numbers PDU.

- VARIABLE LENGTH FIELD – a list of LSP entries of the form:

	No. of Octets
REMAINING LIFETIME	2
LSP ID	8
LSP SEQ NUMBER	4
CHECKSUM	2
REMAINING LIFETIME	2
LSP ID	8
LSP SEQ NUMBER	4
CHECKSUM	2

  - REMAINING LIFETIME – Remaining Lifetime of LSP.
  - LSP ID – 8 octet ID of the LSP to which this entry refers.
  - LSP SEQ NUMBER – Sequence number of LSP.
  - CHECKSUM – Checksum reported in LSP.

  The entries shall be sorted into ascending LSPID order (the LSP number octet of the LSPID is the least significant octet).

9.8 Level 1 Partial Sequence Numbers PDU

	No. of Octets
INTRADOMAIN ROUTEING PROTOCOL DISCRIMINATOR	1
LENGTH INDICATOR	1
VERSION/PROTOCOL ID EXT	1
RESERVED	1
R | R | R | TYPE	1
VERSION	1
ECO	1
USER ECO	1
PDU LENGTH	2
SOURCE ID	7
VARIABLE LENGTH FIELDS	VARIABLE

- INTRADOMAIN ROUTEING PROTOCOL DISCRIMINATOR – architectural constant
- LENGTH INDICATOR – Header Length in octets (17.)
- VERSION/PROTOCOL ID EXTENSION – 1
- RESERVED – transmitted as 0, ignored on receipt
- TYPE (bits 1 through 5) – 26. Note bits 6, 7 and 8 are Reserved, which means they are transmitted as 0 and ignored on receipt.
- VERSION – 1
- ECO — transmitted as zero, ignored on receipt
- USER ECO — transmitted as zero, ignored on receipt
- PDU LENGTH – Entire Length of this PDU, in octets, including header
- SOURCE ID – 7 octet ID of Intermediate system (with zero Circuit ID) generating this Sequence Numbers PDU.

- VARIABLE LENGTH FIELD – a list of LSP entries of the form:

	No. of Octets
REMAINING LIFETIME	2
LSP ID	8
LSP SEQ NUMBER	4
CHECKSUM	2
REMAINING LIFETIME	2
LSP ID	8
LSP SEQ NUMBER	4
CHECKSUM	2

  - REMAINING LIFETIME – Remaining Lifetime of LSP.
  - LSP ID – 8 octet ID of the LSP to which this entry refers.
  - LSP SEQ NUMBER – Sequence number of LSP.
  - CHECKSUM – Checksum reported in LSP.

The entries shall be sorted into ascending LSPID order (the LSP number octet of the LSPID is the least significant octet).

9.9 Level 2 Partial Sequence Numbers PDU

	No. of Octets
INTRADOMAIN ROUTEING PROTOCOL DISCRIMINATOR	1
LENGTH INDICATOR	1
VERSION/PROTOCOL ID EXT	1
RESERVED	1
R | R | R | TYPE	1
VERSION	1
ECO	1
USER ECO	1
PDU LENGTH	2
SOURCE ID	7
VARIABLE LENGTH FIELDS	VARIABLE

- INTRADOMAIN ROUTEING PROTOCOL DISCRIMINATOR – architectural constant
- LENGTH INDICATOR – Header Length in octets (17.)
- VERSION/PROTOCOL ID EXTENSION – 1
- RESERVED – transmitted as 0, ignored on receipt
- TYPE (bits 1 through 5) – 27. Note bits 6, 7 and 8 are Reserved, which means they are transmitted as 0 and ignored on receipt.
- VERSION – 1
- ECO — transmitted as zero, ignored on receipt
- USER ECO — transmitted as zero, ignored on receipt
- PDU LENGTH – Entire Length of this PDU, in octets, including header
- SOURCE ID – 7 octet ID of Intermediate system (with zero Circuit ID) generating this Sequence Numbers PDU.

- VARIABLE LENGTH FIELD – a list of LSP entries of the form:

	No. of Octets
REMAINING LIFETIME	2
LSP ID	8
LSP SEQ NUMBER	4
CHECKSUM	2
REMAINING LIFETIME	2
LSP ID	8
LSP SEQ NUMBER	4
CHECKSUM	2

  - REMAINING LIFETIME – Remaining Lifetime of LSP.
  - LSP ID – 8 octet ID of the LSP to which this entry refers.
  - LSP SEQ NUMBER – Sequence number of LSP.
  - CHECKSUM – Checksum reported in LSP.

  The entries shall be sorted into ascending LSPID order (the LSP number octet of the LSPID is the least significant octet).

10 System Environment

10.1 Generating Jitter on Timers

When PDUs are transmitted as a result of timer expiration, there is a danger that the timers of individual systems may become synchronised. The result of this is that the traffic distribution will contain peaks. Where there are a large number of synchronised systems, this can cause overloading of both the transmission medium and the systems receiving the PDUs. In order to prevent this from occurring, all periodic timers, the expiration of which can cause the transmission of PDUs, shall have "jitter" introduced as defined in the following algorithm.

```
CONSTANT
    Jitter = 25;            (* The percentage jitter as defined in the architectural constant Jitter *)
    Resolution = 100;       (* The timer resolution in milliseconds *)

PROCEDURE Random(max : Integer): Integer;
     (* This procedure delivers a Uniformly distributed random integer R such that 0 < R < max  *)

PROCEDURE DefineJitteredTimer(baseTimeValueInSeconds : Integer; expirationAction : Procedure);

VAR
     baseTimeValue, maximumTimeModifier, waitTime : Integer;
     nextexpiration : Time;

BEGIN
    baseTimeValue := baseTimeValueInSeconds * 1000 / Resolution;
    maximumTimeModifier := baseTimeValue * Jitter / 100; (* Compute maximum possible jitter *)

    WHILE running DO
        BEGIN
            (* First compute next expiration time *)
        randomTimeModifier := Random(maximumTimeModifier);
        waitTime := baseTimeValue - randomTimeModifier;
        nextexpiration := CurrentTime + waitTime;
            (* Then perform expiration Action *)
        expirationAction;
        WaitUntil(nextexpiration);
        END (* of Loop *)
END (* of DefineJitteredTimer *)
```

Thus the call "DefineJitteredTimer(HelloTime, SendHelloPDU);" where "HelloTime" is 10 seconds, will cause the action "SendHelloPDU" to be performed at random intervals of between 7.5 and 10 seconds. The essential point of this algorithm is that the value of "randomTimeModifier" is randomised within the inner loop. Note that the new expiration time is set immediately on expiration of the last interval, rather than when the expiration action has been completed.

The time resolution shall be less than or equal to 100 milliseconds. It is recommended to be less than or equal to 10 milliseconds. The time resolution is the maximum interval that can elapse without there being any change in the value of the timer. The periodic transmission period shall be random or pseudo-random in the specified range, with uniform distribution across similar implementations.

10.2 Resolution of Timers

All timers specified in units of seconds shall have a resolution of no less than ±1 second.

All timers specified in units of milliseconds shall have a resolution of no less than ±10 milliseconds

10.3 Requirements on the Operation of ISO 9542

This International Standard places certain requirements on the use of ISO 9542 by Intermediate systems which go beyond those mandatory requirements stated in the conformance clause of ISO 9542. These requirements are:

a) The IS shall operate the Configuration Information functions on all types of subnetworks supported by the IS. This includes the reception of ESH PDUs, and the reception and transmission of ISH PDUs.

b) The IS shall enable the “All Intermediate Systems” multi-destination subnetwork address.

11 System Management

11.1 General

The operation of the Intra-domain IS–IS routeing functions may be monitored and controlled using System Management. This clause is the management specification for the *Routeing Managed Object* and presents the management facilities provided by routeing.

11.1.1 Data Types

```
FROM CommonMgmt IMPORT
    NSAPAddress, NetworkEntityTitle, AreaAddress, AddressPrefix, SystemID, Unsigned, Counter,
    VersionNumber, ManagedObjectName, Latin1String, OctetString, SimpleName, LocalMOName,
    LANAddress, SNPAAddress, BinaryAbsoluteTime;

TYPE
    adjacencyType =
        (Unknown(0), End system(1), Intermediate System(2), L1 Intermediate System(3), L2 Intermediate
        System(4);

     IDEntry = RECORD
        ID : SystemID;                              (* The System ID of a neighbour End system as reported
                              in its ESH PDUs. *)
        entryHoldingTime: Unsigned [1..65535]        (* The holding time in seconds as reported in the
                              neighbours hellos. *)
        END;

    NETEntry = RECORD
        NET: NetworkEntityTitle;                    (* The NET of a neighbour Intermediate system as
                              reported in its IIH PDUs. *)
        entryHoldingTime: Unsigned [1..65535]        (* The holding time in seconds as reported in the
                              neighbours hellos. *)
        END;

    CircuitType =
        (802.3(1), Pt-Pt(2), X.25 Static Incoming(3), X.25 Static Outgoing(4), X.25 DA(5), X.25 DCM(6));
    Intermediate SystemType = (L1Intermediate System(1), L2Intermediate System(2));
```

11.1.2 Support

The attributes of the Routeing managed object and its subordinate managed objects are defined in the following subclauses. However, not all the attributes that are defined apply to all instances of the particular managed object class. For example, the routeing characteristic attribute maximumVirtualAdjacencies is only defined if the type of the Routeing managed object is L2Intermediate System. The existence of these variants is indicated by the *Support* specified for the attribute using the abbreviations defined in Table 9.

Table 9 - Abbreviations used for Support

Abbreviation	*Selecting Attribute*		*Values*
	managed object	*Attribute*	
L1	Routeing	Type	L1Intermediate System
L2	Routeing	Type	L2Intermediate System
Broadcast	Circuit	Type	802.3
non-Broadcast	Circuit	Type	**not** 802.3
Pt-Pt	Circuit	Type	Pt-Pt **or** In **or** Out
In	Circuit	Type	X.25 Static Incoming
Out	Circuit	Type	X.25 Static Outgoing
DA	Circuit	Type	X.25 DA
X.25	Circuit	Type	X.25 Static Incoming **or** X.25 Static Outgoing **or** X.25 DA
Intermediate System adj	Adjacency	AdjacencyType	Any Intermediate System adjacency
End system adj	Adjacency	AdjacencyType	Any End system adjacency
Manual	Adjacency	Type	Manual

Where multiple values for the *same* selecting attribute are specified, they are considered to be **or**ed together. Where values for *different* selecting attributes are specified, the resultant values for each selecting attribute are considered to be **and** edtogether. Thus the expression indicates that the attribute is defined if and only if

L1, L2
In, Out

((Routeing Type = L1Intermediate System) **or** (Routeing Type = L2Intermediate System))
and
((Circuit Type = X.25 Static Incoming) **or** (Circuit Type = X.25 Static Outgoing))

Where an expression has no value specified for a particular selecting attribute, the attribute is defined for *all* values of that selecting attribute. Thus the above example, if applied to an attribute of the Adjacency managed object, would indicate that the attribute was defined for all values of the adjacency type attribute.

11.1.3 Resetting of Timers

Many of the characteristic attributes of the Routeing managed object and its subordinate managed objects represent the values of timers. They specify the interval between certain events in the operation of the Routeing state machines. If the value of one of these characteristics is changed to a new value *t* while the Routeing state machine is in operation the implementation shall take the necessary actions to ensure that for any time interval which was in progress when the corresponding attribute was changed, the next expiration of that interval takes place *t* seconds from the original start of that interval, or immediately, whichever is the later.

Where this action is necessary it is indicated in the following tables by a footnote.

11.1.4 Resource Limiting Characteristics

Certain attributes place limits on some “resource”, such as maximumSVCAdjacencies. In general implementations may allocate memory resources up to this limit when the routeing managed object is enabled and it may be impossible to change the allocation without first disabling and re-enabling the Network entity. Therefore this International Standard only requires that system management shall be able to change these characteristics when the managed object is disabled (i.e. in the state off).

However some implementations may be able to change the allocation of resources without first disabling the Network entity. In this case it is permitted to *increase* the value of the characteristic at any time, but it shall not be *decreased* below the currently “used” value of the resource. For example, maximumSVCAdjacencies shall not be decreased below the current number of SVCs which have been created.

Characteristics of this type are indicated in the following tables by a footnote.

11.2 The Routeing managed object

Each system includes one Routeing managed object. The Routeing managed object is subordinate to the *Connectionless Network Service* managed object.

The routeing managed object has the classes of subordinate subordinate managed object shown in Figure 7. These are described in subsequent clauses.

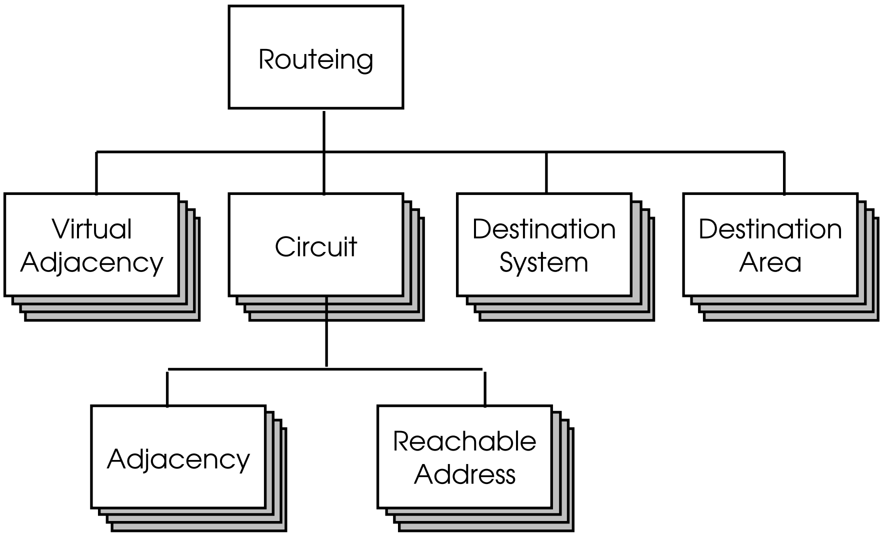


Figure 7 - Classes of Managed Object Subordinate to Routeing

In addition to subordinate managed objects, the Routeing managed object has Characteristic, Status and Counter Attributes, and Events.

11.2.1 Identifier Attribute

The Routeing managed object has no Identifier Attribute.

11.2.2 Characteristic Attributes

The Routeing managed object has the Characteristic Attributes shown in Tables 10–11.

Table 10 - Routeing Characteristic Attributes (part 1)

Keyword	*Code*	*Support*	*Syntax*	*Default Value*	*Description*
Version[1]	0	L1, L2	VersionNumber	1	The version number of this International Standard to which the implementation conforms.
Type[1]	1	L1, L2	Intermediate SystemType	*none*	The Routeing type of this system. This may only be set when the Routeing managed object is created.
Maximum PathSplits[5]	3	L1, L2	Unsigned [1..32]	2	Maximum number of paths with equal routeing metric value which it is permitted to split between.
Maximum Buffers[2]	4	L1, L2	Unsigned [1..∞]	Imp. Specific	Maximum guaranteed number of buffers for forwarding. This is the number of forwarding buffers that is to be reserved, more may be used if they are available. (See D.1.1).
Minimum LSP Transmission Interval[3]	5	L1, L2	Unsigned [5..30]	5	Minimum interval, in seconds, between re-transmissions of an LSP.
Maximum LSP Generation Interval[3]	6	L1, L2	Unsigned [60..900]	900	Maximum interval, in seconds, between generated LSPs by this system.
Minimum Broadcast LSP Transmission Interval	7	L1, L2	Unsigned [1..65535]	33	Minimum interval, in *milliseconds*, between transmission of LSPs on a broadcast circuit (See 7.3.15.6).
Complete SNP Interval[3]	64	L1, L2	Unsigned [1..600]	10	Interval, in seconds, between generation of Complete Sequence Numbers PDUs by a Designated Intermediate System on a broadcast circuit.
Originating L1 LSP Buffer Size[5]	45	L1, L2	Unsigned [512..receiveLSPBufferSize]	receiveLSPBufferSize	The maximum size of Level 1 LSPs and SNPs originated by this system.
Originating L2 LSP Buffer Size[5]	46	L2	Unsigned [512..receiveLSPBufferSize]	receiveLSPBufferSize	The maximum size of Level 2 LSPs and SNPs originated by this system.

[1] This Characteristic is read-only.

[2] This Characteristic may only be changed as described in Clause 11.1.4.

[3] When this Characteristic is changed, the timer must be reset as described in Clause 11.1.3.

[5] This Characteristic may only be changed when the managed object is in the state **off**.

Table 11 - Routeing Characteristic Attributes (part 2)

Keyword	*Code*	*Support*	*Syntax*	*Default Value*	*Description*
Redirect Holding Time	48	L1, L2	Unsigned [1..65535]	600	The holding time (in seconds) to be specified in Redirect PDUs generated by this system.
Manual Area Addresses	15	L1, L2	Set of Area-Address	{}	Area Addresses to be used for this Intermediate System. At least one value must be supplied. The maximum number of Area Addresses which may exist in the set is MaximumAreaAddresses.
Maximum Virtual Adjacencies	8	L2	Unsigned [0..32]	2	Maximum number of Virtual Adjacencies which may be created to repair partitioned Level 1 domains.
Minimum LSP Generation Interval	9	L1,L2	Unsigned [5..300]	30	Maximum interval in seconds between successive generation of LSPs with the same LSPID by this IS
Default ESHello Timer	76	L1,L2	Unsigned [1..65535]	600	The value to be used for the suggested ES configuration timer in ISH PDUs when not soliciting the ES configuration.
Poll ESHello Rate	78	L1,L2	Unsigned [1..65535]	50	The value to be used for the suggested ES configuration timer in ISH PDUs when soliciting the ES configuration.
Partial SNP Interval	72	L1,L2	Unsigned [1..65535]	2	Minimum interval between sending Partial Sequence Number PDUs.
Waiting Time	73	L1,L2	Unsigned [1..65535]	60	Number of seconds to delay in “waiting” state before entering “On” state.
DR ISISHello Timer	79	L1,L2	Unsigned [1..65535]	1	The interval in seconds between the generation of IIH PDUs by the designated router on a LAN.

11.2.3 Status Attributes

The Routeing managed object has the Status Attributes shown in Table 12.

Table 12 - Routeing Status Attributes

Keyword	*Code*	*Support*	*Syntax*	*Description*
State	24	L1, L2	{Off(0), On(1)}	The state of the Routeing managed object.
L1 State	25	L1, L2	{Off(0), On(1), Waiting(2)}	The state of the Level 1 database.
L2 State	26	L2	{Off(0), On(1), Waiting(2)}	The state of the Level 2 database.
Partition Area Addresses	27	L2	Set of AreaAddress	The set union of all manualAreaAddresses of all Intermediate systems in the partition reachable by non-virtual links (calculated from their Level 1 LSPs).
Partition Designated L2 Intermediate System	28	L2	SystemID	The ID of the Partition Designated Level 2 Intermediate System for this system.
Area Addresses	31	L1, L2	Set of AreaAddress	The union of the sets of manualAreaAddresses reported in all Level 1 Link State PDUs received by this Intermediate System.

11.2.4 Counter Attributes

The Routeing managed object has the Counter Attributes shown in Table 13.

Table 13 - Routeing Counter Attributes

Keyword	*Code*	*Support*	*Syntax*	*Description*
PDU Format Errors	35	L1, L2	Counter	Number of times the PDU Format Error Event has been generated.
Partition Virtual Link Changes	36	L1, L2	Counter	Number of times the Partition Virtual Link Change Event has been generated.
Corrupted LSPs Detected	40	L1, L2	Counter	Number of Corrupted LSP Detected events generated.
LSP L1 Database Overloads	41	L1, L2	Counter	Number of times the LSP L1 Database Overload event has been generated.
LSP L2 Database Overloads	42	L2	Counter	Number of times the LSP L2 Database Overload event has been generated.
Manual Addresses Dropped From Area	43	L2	Counter	Number of times the Manual Addresses Dropped From Area event has been generated.
Attempts to Exceed Maximum Sequence Number	63	L1, L2	Counter	Number of times the Attempt To Exceed Maximum Sequence Number event has been generated.
Sequence Number Skips	64	L1, L2	Counter	Number of times the Sequence Number Skipped event has been generated.
Own LSP Purges	65	L1, L2	Counter	Number of times the Own LSP Purged event has been generated.

11.2.5 Events

The Routeing managed object has the events shown below. The arguments are further described in Table 14. Arguments identifying the originating managed object and the time at which the Event was generated are always automatically added by the Agent.

EVENT PDU Format Error = 2
  SUPPORTED BY L1, L2
  COUNTED AS PDUFormatErrors
  ARGUMENTS Receiving Adjacency, PDU Header, Discard reason
END PDU Format Error;

The PDU Format Error Event is generated when a data NPDU is discarded due to a format error.

EVENT Partition Virtual Link Change = 4
  SUPPORTED BY L2
  COUNTED AS partitionVirtualLinkChanges
  ARGUMENTS Virtual Link Change, Virtual Link Address
END Partition Virtual Link Change;

The Partition Virtual Link Change Event is generated when a virtual link (for the purposes of Level 1 partition repair) is either created or deleted. The relative order of events relating to the same Virtual Link must be preserved.

EVENT Corrupted LSP Detected = 6
    SUPPORTED BY L1, L2
    COUNTED AS corruptedLSPsDetected
    ARGUMENTS NONE
END Corrupted LSP Detected;

The Corrupted LSP Detected Event is generated when a corrupted Link State PDU is detected in memory

EVENT LSP L1 Database Overload = 7
    SUPPORTED BY L1, L2
    COUNTED AS LSPL1DatabaseOverloads
    ARGUMENTS stateChange, sourceID
END LSP L1 Database Overload;

The LSP L1 Database Overload Event is generated when the L1State of the system changes between “On” and “Waiting” or “Waiting” and “On”. The stateChange argument is set to indicate the resulting state, and in the case of “Waiting” the sourceID is set to indicate the source of the LSP which precipitated the overload.

EVENT LSP L2 Database Overload = 8
    SUPPORTED BY L2
    COUNTED AS LSPL2DatabaseOverloads
    ARGUMENTS stateChange, sourceID
END LSP L2 Database Overload;

The LSP L2 Database Overload Event is generated when the L2State of the system changes between “On” and “Waiting” or “Waiting” and “On”. The stateChange argument is set to indicate the resulting state, and in the case of “Waiting” the sourceID is set to indicate the source of the LSP which precipitated the overload.

EVENT Manual Address Dropped From Area = 9
    SUPPORTED BY L1,L2
    COUNTED AS manualAddressesDroppedFromArea
    ARGUMENTS areaAddress
END Manual Address Dropped From Area;

The Manual Address Dropped From Area Event is generated when one of the manualAreaAddresses (specified on this system) is ignored when computing partitionAreaAddresses or areaAddresses because there are more than MaximumAreaAddresses distinct Area Addresses. The areaAddress argument is set to the ignored Area Address. It is generated once for each Area Address in manualAreaAddresses which is dropped. It is not logged again for that Area Address until after it has been reinstated into areaAddresses (i.e. it is only the action of dropping the Area Address and not the state of being dropped, which causes the event to be generated).

EVENT Attempt To Exceed Maximum Sequence Number = 16
    SUPPORTED BY L1, L2
    COUNTED AS attemptsToExceedMaximumSequenceNumber
    ARGUMENTS NONE
END Attempt To Exceed Maximum Sequence Number;

The Attempt To Exceed Maximum Sequence Number Event is generated when an attempt is made to increment the sequence number of an LSP beyond the maximum sequence number. Following the generation of this event Routeing is disabled for at least (MaxAge + ZeroAgeLifetime) seconds.

EVENT Sequence Number Skipped = 17
SUPPORTED BY L1, L2
COUNTED AS SequenceNumberSkips
ARGUMENTS NONE
END Sequence Number Skipped;

The Sequence Number Skipped Event is generated when the sequence number of an LSP is incremented by more than one.

EVENT Own LSP Purged = 18
SUPPORTED BY L1, L2
COUNTED AS ownLSPPurges
ARGUMENTS NONE
END Own LSP Purged;

The Own LSP Purged Event is generated when a zero aged copy of a system's own LSP is received from some other system. This represents an erroneous attempt to purge the local system's LSP.

Table 14 - Routeing managed object Event Arguments

Keyword	*Code*	*Support*	*Syntax*	*Description*
Receiving Adjacency	0	L1, L2	SimpleName	This is the Local managed object Name of the Adjacency upon which the NPDU was received.
PDU Header	2	L1, L2	OctetString	This is the Data NPDU header causing this event.
Discard Reason	3	L1, L2	Unsigned [0..255]	This is the reason the event was generated, and is the value of the Reason for Discard parameter which would be placed in an Error Report NPDU resulting from the discard of this NPDU. Note that the event is generated and the Discard Reason ascertained whether or not an Error Report NPDU was sent or requested.
Area Address	4	L2	AreaAddress	This is the Area Address which caused MaximumAreaAddresses to be exceeded.
Source ID	6	L1, L2	OctetString [8]	The source ID of the LSP.
Source Address	7	L1, L2	NSAPAddress	The source NSAP address of the data NPDU.
Virtual Link Change	9	L2	{Created(1), Deleted(0)}	This indicates whether the event was generated as a result of the creation or deletion of Virtual Level 1 link between two Level 2 Intermediate Systems. This is caused by the operation of the Level 1 domain partition repair algorithm.
Virtual Link Address	10	L2	NetworkEntityTitle	The NET of the Level 2 Intermediate System at the remote end of the virtual link.

11.3 Circuit Managed Objects

The Routeing managed object has a class of subordinate managed object called *Circuit.* A Circuit contains information about a subnetwork or a data link, such as a point-to-point connection to another system, a dynamically established data link, or a broadcast link.

Each circuit has Identifier, Characteristic, Counter and Status Attributes, and Events. It also has Adjacency and Reachable Address subordinate managed objects. These are described in subsequent clauses.

11.3.1 Identifier Attribute

The Identifier Attribute for each Circuit is shown in Table 15.

Table 15 - Circuit Identifier Attribute

Keyword	*Code*	*Syntax*	*Description*
Name	0	SimpleName	A string which is the Identifier for the Circuit and which is unique amongst the set of Circuits maintained by Routeing. It must be specified by the System Manager when the Circuit is created.

11.3.2 Characteristic Attributes

The Circuit managed object has the characteristic attributes shown in Tables 16–19.

Table 16 - Circuit Characteristic Attributes (part 1)

Keyword	*Code*	*Support*	*Syntax*	*Default Value*	*Description*
Type[1]	1	L1, L2	Circuit-Type	*none*	The type of the circuit. This attribute may only be set when the Circuit is created. Subsequently it is read-only.
Hello Timer[4]	3	L1, L2	Unsigned [1..21845]	10	The period, in seconds, between ISH PDUs.
ISIS Hello Timer[4]	4	L1, L2	Unsigned [1..21845]	3	The period, in seconds, between LAN Level 1 and Level 2 IIH PDUs. It is also used as the period between ISH PDUs when polling the ES configuration.

[1] This Characteristic is read-only.

[4] When this Characteristic is changed, the timer must be reset as described in Clause 11.1.3.

Table 17 - Circuit Characteristic Attributes (part 2)

Keyword	*Code*	*Support*	*Syntax*	*Default Value*	*Description*
L1 Default Metric	6	L1, L2	Unsigned [1..MaxLinkMetric]	20	The *default metric* value of this circuit for Level 1 traffic.
L1 Delay Metric	60	L1, L2	Unsigned [1..MaxLinkMetric]	*none*	The *delay metric* value of this circuit for Level 1 traffic.
L1 Expense Metric	61	L1, L2	Unsigned [1..MaxLinkMetric]	*none*	The *expense metric* value of this circuit for Level 1 traffic.
L1 Error Metric	62	L1, L2	Unsigned [1..MaxLinkMetric]	*none*	The *error metric* value of this circuit for Level 1 traffic.
L2 Default Metric	7	L2	Unsigned [1..MaxLinkMetric]	20	The *default metric* value of this circuit for Level 2 traffic.
L2 Delay Metric	63	L2	Unsigned [1..MaxLinkMetric]	*none*	The *delay metric* value of this circuit for Level 2 traffic.
L2 Expense Metric	64	L2	Unsigned [1..MaxLinkMetric]	*none*	The *expense metric* value of this circuit for Level 2 traffic.
L2 Error Metric	65	L2	Unsigned [1..MaxLinkMetric]	*none*	The *error metric* value of this circuit for Level 2 traffic.
Call Establishmend Default Metric Increment	8	L2 DA	Unsigned [0..MaxLinkMetric]	0	Additional value to be reported for the default metric value of unestablished DA adjacencies.
Call Establishmend Delay Metric Increment	66	L2 DA	Unsigned [0..MaxLinkMetric]	0	Additional value to be reported for the default metric value of unestablished DA adjacencies.
Call Establishmend Expense Metric Increment	67	L2 DA	Unsigned [0..MaxLinkMetric]	0	Additional value to be reported for the default metric value of unestablished DA adjacencies.
Call Establishmend Error Metric Increment	68	L2 DA	Unsigned [0..MaxLinkMetric]	0	Additional value to be reported for the default metric value of unestablished DA adjacencies.

Table 18 - Circuit Characteristic Attributes (part 3)

Keyword	*Code*	*Support*	*Syntax*	*Default Value*	*Description*
L1 Intermediate System Priority	10	L1, L2 Broadcast	Unsigned [1..127]	64	Priority for becoming LAN Level 1 Designated Intermediate System.
L2 Intermediate System Priority	11	L2 Broadcast	Unsigned [1..127]	64	Priority for becoming LAN Level 2 Designated Intermediate System.
Neighbour SNPA Address[2]	18	L1, L2 Out, In, DCM	SNPA-Address	0	SNPA Address to call, or SNPA Address from which to accept call.
Recall Timer[4]	19	L1, L2 Out, In, DCM	Unsigned [0..65535]	60	Number of seconds that must elapse between a call failure on a DED circuit and a recall.
Maximum Call Attempts	20	L1, L2 Out, In, DCM	Unsigned [0..255]	10	Maximum number of successive call failures before halting. (A value of zero means infinite retries.)
Idle Timer[4]	21	L1, L2 Out, In, DCM	Unsigned [0..65535]	30	Number of seconds of idle time before call is cleared.

[4] When this Characteristic is changed, the timer must be reset as described in Clause 11.1.3.

Table 19 - Circuit Characteristic Attributes (part 4)

Keyword	*Code*	*Support*	*Syntax*	*Default Value*	*Description*
Initial Minimum Timer[4]	22	L1, L2 DCM, DA	Unsigned [1..65535]	55	Number of seconds that a call remains connected after being established, irrespective of traffic. (Note. This should be set small enough so that the call is cleared before the start of the next charging interval.)
Reserve Timer[4]	23	L1, L2 DCM, DA	Unsigned [1..65535]	600	Number of seconds, after call is cleared due to lack of traffic, during which the SVC remains reserved for the previous SNPA address.
Maximum SVC Adjacencies[6]	24	L1, L2 DA	Unsigned [1..65535]	1	Number of Adjacencies to reserve for SVCs for this circuit. This is the maximum number of simultaneous calls which are possible on this circuit.
Reserved Adjacency[2]	25	DA	Boolean	False	When “True”, indicates that one SVC must be reserved for a connection to a Intermediate System.
Manual L2 Only Mode[2]	26	L2	Boolean	False	When “True”, indicates that this Circuit is to be used only for Level 2.

[2] This Characteristic may only be changed when the managed object is in the state **off**.

[4] When this Characteristic is changed, the timer must be reset as described in Clause 11.1.3.

[6] This Characteristic may only be changed as described in Clause 11.1.4.

11.3.3 Status Attributes

Each Circuit managed object has the Status Attributes shown in Table 20.

Table 20 - Circuit Status Attributes

Keyword	*Code*	*Support*	*Syntax*	*Description*
State	29	L1, L2	{On(1), Off(0)}	The current state of the circuit.
PtPt Circuit ID	31	L1, L2 Non-broadcast	Octet String [0..6]	The 7 octet ID of the circuit allocated during initialization. If no value has been negotiated (either because the adjacency is to an End system, or because initialization has not yet successfully completed), this attribute has the value which would be proposed for this circuit. (i.e. the concatenation of the local system ID and the one octet local Circuit ID for this circuit.)
L1Circuit ID	62	L1, L2 Broadcast	Octet String [0..6]	The 7 octet LAN ID allocated by the LAN Level 1 Designated Intermediate System. Where this system is not aware of the value (because it is not participating in the Level 1 Designated Intermediate System election), this attribute has the value which would be proposed for this circuit. (i.e. the concatenation of the local system ID and the one octet local Circuit ID for this circuit.)
L2 Circuit ID	63	L2 Broadcast	Octet String [0..6]	The 7 octet LAN ID allocated by the LAN Level 2 Designated Intermediate System. Where this system is not aware of the value (because it is not participating in the Level 2 Designated Intermediate System election), this attribute has the value which would be proposed for this circuit. (i.e. the concatenation of the local system ID and the one octet local Circuit ID for this circuit.)
L1 Designated Intermediate System	33	L1, L2 Broadcast	SystemID	The ID[2] of the LAN Level 1 Designated Intermediate System on this circuit.
L2 Designated Intermediate System	34	L2 Broadcast	SystemID	The ID[2] of the LAN Level 2 Designated Intermediate System on this circuit.

[2] If, for any reason this system is not partaking in the relevant Designated Intermediate System election process, then the value returned is “00-00-00-00-00-00”

11.3.4 Counter Attributes

The Circuit managed object has the Counter attributes shown in Tables 21 – 22.

Table 21 - Circuit Counter Attributes (part 1)

Keyword	*Code*	*Support*	*Syntax*	*Description*
Calls Placed	41	L1, L2 Out, DCM, DA	Counter	Number of Call attempts (successful or unsuccessful).
Calls Failed	42	L1, L2 Out, DCM, DA	Counter	Number of Unsuccessful Call attempts.
Circuit Changes	43	L1, L2	Counter	Number of times this Circuit state changed between On and Off and vice versa.
Changes in Adjacency State	46	L1, L2	Counter	Number of Adjacency State Change events generated.

Table 22 - Circuit Counter Attributes (part 2)

Keyword	*Code*	*Support*	*Syntax*	*Description*
Initialization Failures	47	L1, L2	Counter	Number of Initialization Failure events generated.
Rejected Adjacencies	50	L1, L2	Counter	Number of Rejected Adjacency events generated.
Corrupted LSPs Received	51	L1, L2	Counter	Number of Corrupted LSP Received events generated.
LAN L2 Designated Intermediate System Changes	53	L2	Counter	Number of LAN L2 Designated Intermediate System Change events generated.
LAN L1 Designated Intermediate System Changes	52	L1, L2	Counter	Number of LAN L1 Designated Intermediate System Change events generated.
Exceeded Maximum Call Attempts	55	L1, L2 Out, DCM	Counter	Number of Exceeded Maximum Call Attempts events generated.
Exceeded Maximum SVC Adjacencies	56	L1, L2 DA	Counter	Number of Exceeded Maximum SVC Adjacencies events generated.
Control Message PDUs Sent	57	L1, L2	Counter	Number of control message PDUs sent on this circuit.
Control Message PDUs Received	58	L1, L2	Counter	Number of control message PDUs received on this circuit.

11.3.5 Events

The Circuit managed object has the events shown below. The arguments are further described in Table 23. When the argument is Reason, the ReasonType is coded as shown in Table 24. Where *Additional Arguments* are specified in Table 24 for a particular ReasonType, those arguments (as defined in Table 23) must be included in the argument list for the event generating the Reason argument.

Arguments identifying the originating managed object and the time at which the Event was generated are always automatically added by the Agent.

EVENT Circuit Change = 0
  SUPPORTED BY L1, L2
  COUNTED AS circuitChanges
  ARGUMENTS New Circuit State
END Circuit Change;

The Circuit Change Event is generated when the state of the Circuit changes from On to Off or from Off to On. The relative order of events relating to the same Circuit must be preserved.

EVENT Adjacency State Change = 1
  SUPPORTED BY L1, L2
  COUNTED AS changesInAdjacencyState
  ARGUMENTS New Adjacency State, Adjacent system, Reason
END Adjacency State Change;

The Adjacency State Change Event is generated when the state of an Adjacency on the Circuit changes from Up to Down or Down to Up (in the latter case the Reason argument is omitted). For these purposes the states Up and Up/dormant are considered to be Up, and any other state is considered to be Down. The relative order of events relating to the same Adjacency must be preserved.

EVENT Initialisation Failure = 2
  SUPPORTED BY L1, L2
  COUNTED AS initialisationFailures
  ARGUMENTS Adjacent system, Reason
END Initialisation Failure;

The Initialisation Failure Event is generated when an attempt to initialise with an adjacent system fails as a result of either Version Skew or Area Mismatch. In the case of Version Skew, the Adjacent system argument is not present.

EVENT Rejected Adjacency = 5
  SUPPORTED BY L1, L2
  COUNTED AS rejectedAdjacencies
  ARGUMENTS Adjacent system, Reason
END Rejected Adjacency;

The Rejected Adjacency Event is generated when an attempt to create a new adjacency is rejected, because of a lack of resources.

EVENT Corrupted LSP Received = 8
  SUPPORTED BY L1, L2
  COUNTED AS corruptedLSPsReceived
  ARGUMENTS Adjacent system
END Corrupted LSP Received;

The Corrupted LSP Received Event is generated when a corrupted Link State PDU is received.

EVENT LAN L1 Designated Intermediate System Change = 10
  SUPPORTED BY L1, L2
  COUNTED AS LANL1DesignatedIntermediate SystemChanges

ARGUMENTS Designated Intermediate System Change
END LAN L1 Designated Intermediate System Change;

The LAN L1 Designated Intermediate System Change Event is generated when the local system either elects itself or resigns as being the LAN L1 Designated Intermediate System on this circuit. The relative order of these events must be preserved.

EVENT LAN L2 Designated Intermediate System Change = 11
SUPPORTED BY L2
COUNTED AS LANL2DesignatedIntermediate SystemChanges
ARGUMENTS Designated Intermediate System Change
END LAN L2 Designated Intermediate System Change;

The LAN L2 Designated Intermediate System Change Event is generated when the local system either elects itself or resigns as being the LAN L2 Designated Intermediate System on this circuit. The relative order of these events must be preserved.

EVENT Exceeded Maximum Call Attempts = 13
SUPPORTED BY L1, L2 Out, DCM
COUNTED AS exceededMaximumCallAttempts
ARGUMENTS NONE
END Exceeded Maximum Call Attempts;

The Exceeded Maximum Call Attempts Event is generated when recallCount becomes equal to maximumCallAttempts.

EVENT Exceeded Maximum SVC Adjacencies = 14
SUPPORTED BY L1, L2 DA
COUNTED AS exceededMaximumSVCAdjacencies
ARGUMENTS NONE
END Exceeded Maximum SVC Adjacencies;

The Exceeded Maximum SVC Adjacencies Event is generated when there is no free adjacency on which to establish an SVC for a new destination.(see 8.3.2.3)

Table 23 - Circuit Event Arguments

Keyword	*Code*	*Support*	*Syntax*	*Description*
New Circuit State	0	L1, L2	{On(1), Off(0)}	This is the direction of the Circuit state change specified as the resulting state. i.e. a change from On to Off is specified as Off.
New Adjacency State	1	L1, L2	{Up(1), Down(0)}	This is the direction of the Adjacency state change specified as the resulting state. i.e. a change from Up to Down is specified as Down. For these purposes the states Up and Up/dormant are considered to be Up, and any other state is considered to be Down.
Adjacent System	2	L1, L2	Set of SystemID	The system ID of the adjacent system. In the case of End systems, there may be more than one ID present.
Reason	3	L1, L2	ReasonType	The associated reason. This is an enumerated type as specified in Table 24.
Designated Intermediate System Change	4	L1, L2	{Elected(1), Resigned-(0)}	The direction of the change in designated Intermediate System status of this system.
PDU Header	6	L1, L2	OctetString	The header of the PDU which caused the event to be generated.
Called Address	7	L1, L2 X.25	SNPA-Address	The SNPA Address which was being called when the Adjacency was taken down as a result of a call reject.
Version	8	L1, L2	Version-Number	The version number reported by the other system

Table 24 - Reason Type Definition

Code	*Reason*	*Support*	*Additional Arguments*
0	Holding Timer Expired	L1, L2	*none*
1	Checksum Error (Routeing message received with checksum error)	L1, L2	PDU Header
2	One Way Connectivity (Hello received indicating connectivity became one-way)	L1, L2 (Broadcast)	*none*
3	Call Rejected	L1, L2 (X.25)	Called Address
4	Reserve Timer Expired	L2 (DA)	*none*
5	Circuit Disabled (Adjacency taken down because circuit disabled)	L1, L2	*none*
6	Version Skew	L1, L2	Version
7	Area mismatch	L1, L2	PDU Header
8	Maximum Broadcast Intermediate Systems Exceeded	L1, L2 (Broadcast)	*none*
9	Maximum Broadcast End Systems Exceeded	L1, L2 (Broadcast)	*none*
10	Wrong System Type	L2 (Non-Broadcast)	*none*

11.4 Adjacency managed objects

The Circuit managed object has a class of subordinate managed object called Adjacency. An Adjacency contains information about a particular neighbour system accessible over that Circuit. Adjacencies are normally created as result of auto-configuration with a neighbour, but they may be created manually where such auto-configuration is not possible.

Each Adjacency has Identifier, Characteristic, and Status Attributes. These are described in subsequent clauses.

11.4.1 Identifier Attribute

The Identifier Attribute for each Adjacency is shown in Table 25.

Table 25 - Adjacency Identifier Attribute

Keyword	*Code*	*Syntax*	*Description*
Name	0	SimpleName	A string which is the Identifier for the Adjacency and which is unique amongst the set of Adjacencies maintained by Routeing for this Circuit. If this is a manually created adjacency (i.e. the type is Manual) it is set by the System Manager when the Adjacency is created, otherwise it is generated by the implementation such that it is unique. The set of identifier containing the leading string RTG$ are reserved for Automatic Adjacencies. An attempt to create a Manual Adjacency with such an identifier will cause an exception to be raised.

11.4.2 Characteristic Attributes

The Adjacency managed object has the characteristic attributes shown in Table 26.

Table 26 - Adjacency Characteristic Attributes

Keyword	*Code*	*Support*	*Syntax*	*Default Value*	*Description*
Type[1]	1	L1, L2	{Manual(0), Autoconfigured(1)}	*none*	Indicates whether the adjacency has been manually created and will not autoconfigure, or has been created as a result of the receipt of hello PDU's and will autoconfigure the data link address (SNPAAddress or LANAddress) and Intermediate SystemNETs (and hence areaAddresses) status attributes. This characteristic is not directly set-able. It will always have the value Manual when the adjacency has been created manually, otherwise it will have the value Autoconfigured.

[1] This Characteristic is read-only.

11.4.3 Status Attributes

Each Adjacency managed object has the Status Attributes shown in Tables 27–28.

Table 27 - Adjacency Status Attributes (part 1)

Keyword	*Code*	*Support*	*Syntax*	*Description*
State	2	L1, L2	{Initialising(1), Up(2), Failed(4)}	The state of the adjacency.
LAN Address	3	L1, L2 Broadcast	LANAddress	The MAC address of the neighbour system on a broadcast circuit.
SNPA Address	4	L1, L2 DA	SNPA-Address	The SNPA Address of the neighbour system on an X.25 circuit.
Adjacency Type	5	L1, L2 not DA	Adjacency-Type	The type of the adjacency – one of:- - Unknown - End system - Intermediate system - L1 Intermediate system - L2 Intermediate system
LAN Priority	6	L1, L2 Broadcast Intermediate System Adj	Unsigned [0..127]	Priority of neighbour on this adjacency for becoming LAN Level 1 Designated Intermediate System if adjacencyType is “L1 Intermediate System” or LAN Level 2 Designated Intermediate System if adjacencyType is “L2 Intermediate System”.

Table 28 - Adjacency Status Attributes (part 2)

Keyword	*Code*	*Support*	*Syntax*	*Description*
Holding Timer	8	L1, L2 Intermediate System Adj	Unsigned [0..65535]	Holding time for this adjacency updated from the IIH PDUs.
Neighbour SystemID	9	L1, L2 Intermediate System Adj	SystemID	The SystemID of the neighbouring Intermediate system from the “Source ID” field of the neighbour’s IIH PDU. The Intermediate System ID for this neighbour is derived by appending zero to this value.
L2 Only Mode	10	L2 Pt–Pt Intermediate System adj	Boolean	When “True” indicates that this adjacency is to be used only for Level 2.
End System IDs	11	L1, L2 End system Adj	Set of IDEntry	This contains the system ID(s) of a neighbour End system. Where (in a Intermediate System) an adjacency has been created manually, these will be the set of IDs given in the manualIDs parameter of the create directive.
Neighbour Areas	13	L1, L2 Intermediate System adj not DA	Set of Area-Address	This contains the Area Addresses of a neighbour Intermediate System from the IIH PDU..

11.4.4 Counter

The Adjacency managed object has no Counter Attributes.

11.4.5 Events

The Adjacency managed object has no Events.

11.5 Reachable Address managed objects

The Circuit managed object has a class of subordinate managed object called Reachable Address. A Reachable Address contains information about a manually entered address prefix accessible over that Circuit. It exists only on Level 2 Intermediate systems.

Each Reachable Address has Identifier, Characteristic and Status Attributes. These are described in subsequent subclauses.

11.5.1 Identifier Attribute

The Identifier Attribute for each Reachable Address is shown in Table 29.

Table 29 - Reachable Address Identifier Attribute

Keyword	*Code*	*Syntax*	*Description*
Name	0	AddressPrefix	The address prefix to which this Reachable Address refers.

11.5.2 Characteristic Attributes

The Reachable Address managed object has the characteristic attributes shown in Table 30.

Table 30 - Reachable Address Characteristic Attributes (part 1)

Keyword	*Code*	*Support*	*Syntax*	*Default Value*	*Description*
Default Metric	2	L1, L2	Unsigned [1..MaxLinkMetric]	20	The *default metric value for* reaching the specified prefix over this circuit. If this attribute is changed while both the Reachable Address and the Circuit are Enabled (i.e. state On), the actions described in 8.3.5.4 must be taken.
Delay Metric	7	L1, L2	Unsigned [1..MaxLinkMetric]	*none*	The *delay metric value for* reaching the specified prefix over this circuit. If this attribute is changed while both the Reachable Address and the Circuit are Enabled (i.e. state On), the actions described in 8.3.5.4 must be taken.
Expense Metric	8	L1, L2	Unsigned [1..MaxLinkMetric]	*none*	The *expense metric value for* reaching the specified prefix over this circuit. If this attribute is changed while both the Reachable Address and the Circuit are Enabled (i.e. state On), the actions described in 8.3.5.4 must be taken.
Error Metric	9	L1, L2	Unsigned [1..MaxLinkMetric]	*none*	The *error metric value for* reaching the specified prefix over this circuit. If this attribute is changed while both the Reachable Address and the Circuit are Enabled (i.e. state On), the actions described in 8.3.5.4 must be taken.

Table 31 - Reachable Address Characteristic Attributes (part 2)

Keyword	*Code*	*Support*	*Syntax*	*Default Value*	*Description*
Mapping Type[2]	3	L2 X.25 or L2 Broadcast	{Manual(0), X.121(1)[3]}	X.121	The type of mapping to be employed to ascertain the SNPA Address to which a call should be placed for this prefix. **X.121** indicates that the X.121 address extraction algorithm is to be employed. This will extract the SNPA address from the IDI of an X.121 format IDP of the NSAP address to which the NPDU is to be forwarded. **Manual** indicates that the set of addresses in the SNPAAddresses or LANAddresses characteristic are to be used. For Broadcast circuits, only the value **Manual** is permitted.
SNPA Addresses[2]	4	L2 X.25	SNPA-Address	Set of SNPA-Address	A set of SNPA addresses to which a call may be directed in order to reach an address which matches the address prefix of the Reachable Address. Associated with each SNPA Address, but not visible to System Management, is a variable lastFailure of Type BinaryAbsoluteTime.
LAN Addresses[2]	5	L2 Broadcast	LAN-Addresses	00-00-00-00-00-00	Asingle LAN addresses to which an NPDU may be directed in order to reach an address which matches the address prefix of the Reachable Address. An exception is raised if an attempt is made to enable the Reachable Address with the default value.

[2] This Characteristic may only be changed when the managed object is in the state **Off**.

[3] **Temporary footnote.** More algorithm types may be added as required.

11.5.3 Status Attributes

The Reachable Address managed object has the Status Attributes shown in Table 32.

Table 32 - Reachable Address Status Attributes

Keyword	*Code*	*Support*	*Syntax*	*Description*
State	6	L2	{On(1), Off(0)}	The state of the Reachable Address managed object.

11.5.4 Counter Attributes

The Reachable Address managed object has no Counter Attributes.

11.5.5 Events

The Reachable Address managed object has no Events.

11.6 Destination System managed objects

The Routeing managed object has a class of subordinate managed object called Destination System. This subordinate managed object contains information about a particular destination system within a Level 1 domain. It provides a means of obtaining information about the contents of the Level 1 Shortest Paths Database (see C.1.5).

A Destination System managed object is created and deleted by the operation of the Network entity. It does not have a separate existence, but provides a means for System Management to inspect the shortest paths database.

Each Destination System managed object has Identifier and Status Attributes only. These are described in subsequent clauses.

11.6.1 Identifier Attribute

The Identifier Attribute for each Destination System is shown in Table 33.

Table 33 - Destination System Identifier Attribute

Keyword	*Code*	*Syntax*	*Description*
Name	0	NetworkEntityTitle	A Network entity Title of a system in the local domain (i.e. the Area Address portion matches one of the areaAddresses of the domain).

11.6.2 Characteristic Attributes

The Destination System managed object has no Characteristic Attributes.

11.6.3 Status Attributes

The Destination System managed object has the status attributes shown in Table 34.

Table 34 - Destination System Status Attributes

Keyword	*Code*	*Support*	*Syntax*	*Description*
Default Metric Path Cost	1	L1, L2	Unsigned [0..MaxPathMetric]	Cost of least cost path(s) using the *default metric* to destination system
Default Metric Output Adjacencies	2	L1, L2	Set of {LocalEntityName}	The set of Circuit and Adjacency managed object identifiers representing the forwarding decisions based upon the *default metric* for the destination system.
Delay Metric Path Cost	3	L1, L2	Unsigned [0..MaxPathMetric]	Cost of least cost path(s) using the *delay metric* to destination system
Delay Metric Output Adjacencies	4	L1, L2	Set of {LocalEntityName}	The set of Circuit and Adjacency managed object identifiers representing the forwarding decisions based upon the *delay metric* for the destination system.
Expense Metric Path Cost	5	L1, L2	Unsigned [0..MaxPathMetric]	Cost of least cost path(s) using the *expense metric* to destination system
Expense Metric Output Adjacencies	6	L1, L2	Set of {LocalEntityName}	The set of Circuit and Adjacency managed object identifiers representing the forwarding decisions based upon the *expense metric* for the destination system.
Error Metric Path Cost	7	L1, L2	Unsigned [0..MaxPathMetric]	Cost of least cost path(s) using the *expense metric* to destination system
Error Metric Output Adjacencies	8	L1, L2	Set of {LocalEntityName}	The set of Circuit and Adjacency managed object identifiers representing the forwarding decisions based upon the *error metric* for the destination system.

11.6.4 Counter Attributes

The Destination system managed object has no Counter Attributes.

11.6.5 Events

The Destination system managed object has no Events.

11.7 Destination Area Managed Objects

The Routeing managed object has a class of subordinate managed object called Destination Area. This managed object contains information about a particular destination area Prefix. It exists only in Level 2 Intermediate systems. It provides a means of obtaining information about the contents of the Level 2 Shortest Paths Database (see C.1.6).

A Destination Area managed object is created and deleted by the operation of the Network entity. It does not have a separate existence, but provides a means for System Management to inspect the shortest paths database.

Each Destination Area managed object has Identifier and Status Attributes only. These are described in subsequent sub-clauses.

11.7.1 Identifier Attribute

The Identifier Attribute for each Destination Area is shown in Table 35.

Table 35 - Destination Area Identifier Attribute

Keyword	*Code*	*Syntax*	*Description*
Name	0	AddressPrefix	An Area Address (or prefix) of a destination area.

11.7.2 Characteristic Attributes

The Destination Area managed object has no Characteristic Attributes.

11.7.3 Status Attributes

The Destination Area managed object has the status attributes shown in Table 36.

Table 36 - Destination Area Status Attribute

Keyword	*Code*	*Support*	*Syntax*	*Description*
Default Metric Path Cost	1	L1, L2	Unsigned [0..MaxPathMetric]	Cost of least cost path(s) using the *default metric* to destination area.
Default Metric Output Adjacencies	2	L1, L2	Set of {LocalEntityName}	The set of Circuit and Adjacency (or Reachable Address) managed object identifiers representing the forwarding decisions based upon the *default metric* for the destination area.
Delay Metric Path Cost	3	L1, L2	Unsigned [0..MaxPathMetric]	Cost of least cost path(s) using the *delay metric* to destination area.
Delay Metric Output Adjacencies	4	L1, L2	Set of {LocalEntityName}	The set of Circuit and Adjacency (or Reachable Address) managed object identifiers representing the forwarding decisions based upon the *delay metric* for the destination area.
Expense Metric Path Cost	5	L1, L2	Unsigned [0..MaxPathMetric]	Cost of least cost path(s) using the *expense metric* to destination area.
Expense Metric Output Adjacencies	6	L1, L2	Set of {LocalEntityName}	The set of Circuit and Adjacency (or Reachable Address) managed object identifiers representing the forwarding decisions based upon the *expense metric* for the destination area.
Error Metric Path Cost	7	L1, L2	Unsigned [0..MaxPathMetric]	Cost of least cost path(s) using the *expense metric* to destination area.
Error Metric Output Adjacencies	8	L1, L2	Set of {LocalEntityName}	The set of Circuit and Adjacency (or Reachable Address) managed object identifiers representing the forwarding decisions based upon the *error metric* for the destination area.

11.7.4 Counter Attributes

The Destination Area managed object has no Counter Attributes.

11.7.5 Events

The Destination Area managed object has no Events.

11.8 Virtual Adjacency Managed Objects

The Routeing managed object has a class of subordinate managed object called Virtual Adjacency. It exists only in Level 2 Intermediate Systems. It contains information about a Virtual Adjacency created by the operation of the Level 2 Routeing Algorithm to repair Level 1 partitions.

A Virtual Adjacency managed object is created and deleted by the operation of the Network entity.

Each Virtual Adjacency managed object has Identifier and Status Attributes only. These are described in subsequent sub-clauses.

11.8.1 Identifier Attribute

The Identifier Attribute for each Virtual Adjacency is shown in Table 37.

Table 37 - Virtual Adjacency Identifier Attribute

Keyword	*Code*	*Syntax*	*Description*
Name	0	NetworkEntityTitle	The Network entity Title which is the destination of a Virtual link being used to repair a partitioned Level 1 area (see 7.2.10).

11.8.2 Characteristic Attributes

The Virtual Adjacency managed object has no Characteristic Attributes.

11.8.3 Status Attributes

The Virtual Adjacency managed object has the status attributes shown in Table 38.

Table 38 - Virtual Adjacency Status Attributes

Keyword	*Code*	*Support*	*Syntax*	*Description*
Metric	1	L2	Unsigned [0..MaxPath Metric]	Cost of least cost L2 path(s) to destination area based on the *default metric.*

11.8.4 Counter Attributes

The Virtual Adjacency managed object has no Counter Attributes.

11.8.5 Events

The Virtual Adjacency managed object has no Events.

12 Conformance

TEMPORARY NOTE - The conformance requirements have been drafted on the assumption that the following protocol functions are optional.

a) L1 Routeing on a L2 IS

TEMPORARY NOTE — Some technical changes and topological restrictions may be necessary in order to accomodate the optioanlity of L1 Routeing on a L2 IS. Member bodies are requested to comment on this issue as part of their response to the DP ballot.

b) L2 Routeing on a L1 IS
c) Equal cost path splitting
d) Area Partition Repair
e) Multiple routeing metrics
f) Downstream Splitting (Danish Proposal)
g) LAN Subnetwork dependent functions
h) 8208 Subnetwork dependent functions
i) Point-to-point Subnetwork dependent functions
j) Reachable Address prefix definition and SNPA extraction/mapping

12.1 Static Conformance Requirements

12.1.1 Protocol Implementation Conformance Statement

A Protocol Implementation Conformance Statement (PICS) shall be completed in respect of any claim for conformance of an implementation to this International Standard: the PICS shall be produced in accordance with the relevant PICS pro-forma in Annex A.

12.1.2 Static Conformance for all ISs

A system claiming conformance to this International Standard shall be capable of:

a) calculating a single minimum cost route to each destination according to 7.2.6 for the default metric specified in 7.2.2;

b) utilising Link State information from a system only when an LSP with LSP number 0 and remaining lifetime>0 is present according to 7.2.5;

c) removing excess adjacencies according to 7.2.7

d) performing the robustness checks according to 7.2.8;

e) constructing a forwarding database according to 7.2.9;

f) if (and only if) Area Partition Repair is supported,

    1) performing the operations according to 7.2.10;

    2) performing the encapsulation operations in the forwarding process according to 7.4.3.2; and

    3) performing the decapsulation operations in the receive process according to 7.4.4;

    TEMPORARY NOTE — may need to reorganise clause 7.4.4 in order to make it crystal clear what is required in the receive process in the presence/absence of partition repair

g) computing Area Addresses according to 7.2.11;

h) generating local Link State information as required by 7.3.2;

i) including information from Manual Adjacencies according to 7.3.3.1;

j) if (and only if) Reachable Addresses are supported, including information from Reachable Addresses according to 7.3.3.2;

k) generating multiple LSPs according to 7.3.4;

l) generating LSPs periodically according to 7.3.5;

m) generating LSPs on the occurrence of events according to 7.3.6;

n) generating an LSP checksum according to 7.3.11;

o) operating the Update Process according to 7.3.12–7.3.17 including controlling the rate of LSP transmission only for each broadcast circuit (if any) according to 7.3.15.6;

p) operating the LSP database overload procedures according to 7.3.19.1;

q) selecting the appropriate forwarding database according to 7.4.2;

r) forwarding ISO 8473 PDUs according to 7.4.3.1 and 7.4.3.3;

s) operating the receive process according to 7.4.4;

> TEMPORARY NOTE item 1 of the second bulleted list is only required if you implement partition repair. We need to reorganise the structure so we can pull this out.

t) forming and maintaining adjacencies on each supported point to point circuit (if any) according to 8.2;

u) performing on each supported ISO 8208 circuit (if any)

   1) SVC establishment according to 8.3.2.1 using the network layer protocols according to 8.3.1;

   2) If (and only if) Reachable Addresses are supported, the operations specified in 8.3.2.2– 8.3.5.6.

v) performing on each supported broadcast circuit (if any)

   1) the pseudonode operations according to 7.2.3;

   2) controlling the rate of LSP transmission according to 7.3.15.6;

   3) the operations specified in 8.4.1–8.4.4 and 8.4.6;

   4) the operations specified in 8.4.5.

w) constructing and correctly parsing all PDUs according to clause 9;

x) providing a system environment in accordance with clause 10;

y) being managed via the system management attributes defined in clause 11. The support of all attributes indicated by the “Support” column in the tables is mandatory.

12.1.3 Static Conformance Requirements for level 1 ISs

A system claiming conformance to this International Standard as a level 1 IS shall conform to the requirements of clause 12.1.2 and in addition shall be capable of

a) identifying the nearest Level 2 IS according to 7.2.9.1;

b) generating Level 1 LSPs according to 7.3.7;

c) generating Level 1 Pseudonode LSPs for each supported broadcast circuit (if any) according to 7.3.8;

d) performing the actions in Level 1 Waiting State according to 7.3.19.2

12.1.4 Static Conformance Requirements for level 2 ISs

A system claiming conformance to this International Standard as a level 2 IS shall conform to the requirements of clause 12.1.2 and in addition shall be capable of

a) setting the attached flag according to 7.2.9.2;

b) generating Level 2 LSPs according to 7.3.9;

c) generating Level 2 Pseudonode LSPs for each supported broadcast circuit (if any) according to 7.3.10;

d) performing the actions in Level 2 Waiting State according to 7.3.19.3.

12.2 Dynamic Conformance

12.2.1 Receive Process Conformance Requirements

Any protocol function supported shall be implemented in accordance with 7.4.4.

12.2.2 Update Process Conformance Requirements

Any protocol function supported shall be implemented in accordance with 7.3 and its subclauses.

Any PDU transmitted shall be constructed in accordance with the appropriate subclauses of 9.

12.2.3 Decision Process Conformance Requirements

Any protocol function supported shall be implemented in accordance with 7.2 and its subclauses.

12.2.4 Forwarding Process Conformance Requirements

Any protocol function supported shall be implemented in accordance with 7.4 and its subclauses.

12.2.5 Performance Requirements

This International Standard requires that the following performance criteria be met. These requirements apply regardless of other demands on the system; if an Intermediate system has other tasks as well, those will only get resources not required to meet these criteria.

12.2.5.1 Performance requirement on the Update process

Each Intermediate system implementation shall specify (in its PICS):

a) the maximum number of other Intermediate systems it can handle. (For L1 Intermediate systems that means Intermediate systems in the area; for L2 Intermediate systems that is the sum of Intermediate systems in the area and Intermediate systems in the L2 subdomain.) Call this limit *N*.

b) the maximum supported forwarding rate in ISO 8473 PDUs per second.

The implementation shall guarantee the update process enough resources to process *N* LSPs per 30 seconds. (Resources = CPU, memory, buffers, etc.)

A maximum propagation time for a single LSP of one second per hop shall be maintained.

12.2.5.2 Performance requirement on the Decision process

The implementation shall guarantee the decision process enough resources to complete (i.e. start to finish) within 5 seconds. In a stable topology while forwarding at the maximum rate. (For L2 Intermediate Systems, this applies to the two levels together, not each level separately.)

12.2.5.3 Reception and Processing of PDUs

An ideal Intermediate system would be able to correctly process all PDUs, both control and data, with which it was presented, while simultaneously running the decision process and responding to management requests. However, in the implementations of real Intermediate systems some compromises must be made. The way in which these compromises are made can dramatically affect the correctness of operation of the Intermediate system. The following general principles apply.

a) A stable topology should result in stable routes when forwarding at the maximum rated forwarding rate.

b) Some forwarding progress should always be made (albeit over incorrect routes) even in the presence of a maximally unstable topology.

In order to further characterise the required behaviour, it is necessary to identify the following types of traffic.

a) IIH traffic. This traffic is important for maintaining Intermediate system adjacencies and hence the Intermediate system topology. In order to prevent gratuitous topology changes it is essential that Intermediate system adjacencies are not caused to go down erroneously. In order to achieve this no more than ISISHoldingMultiplier - 1 IIH PDUs may be dropped between any pair of Intermediate systems. A safer requirement is that *no* IIH PDUs are dropped.

   The rate of arrival of IIH PDUs is approximately constant and is limited on Point–to-Point links to 1/ISISHelloTimer and on LANs to a value of approximately 2(*n*/ISISHelloTimer) + 2, where *n* is the number of Intermediate systems on the LAN (assuming the worst case that they are all Level 2 Intermediate systems).

b) ESH PDU traffic. This traffic is important for maintaining End system adjacencies, and has relatively low processing latency. As with IIH PDUs, loss of End system adjacencies will cause gratuitous topology changes which will result in extra control traffic.

   The rate of arrival of ESH PDUs on Point–to-Point links is limited to approximately 1/DefaultESHelloTimer under all conditions. On LANs the background rate is approximately *n*/DefaultESHelloTimer where *n* is the number of End systems on the LAN. The maximum rate during polling is limited to approximately *n*/PollESHelloRate averaged over a period of about 2 minutes. (Note that the actual peak arrival rate over a small interval may be much higher than this.)

c) LSP (and SNP) traffic. This traffic will be retransmitted indefinitely by the update process if it is dropped, so there is no requirement to be able to process every received PDU. However, if a substantial proportion are lost, the rate of convergence to correct routes will be affected, and bandwidth and processing power will be wasted.

   On Point-to-Point links the peak rate of arrival is limited only by the speed of the data link and the other traffic flowing on that link. The maximum average rate is determined by the topology.

   On LANs the rate is limited at a first approximation to a maximum rate of 1000/minimumBroadcastLSPTransmissionInterval, however it is possible that this may be multiplied by a factor of up to *n*, where *n* is the number of Intermediate systems on the LAN, for short periods. A Intermediate system shall be able to receive and process at least the former rate without loss, even if presented with LSPs at the higher rate. (i.e. it is permitted to drop LSPs, but must process at least 1000/minimumBroadcastLSPTransmissionInterval per second of those presented.)

   The maximum background rate of LSP traffic (for a stable topology) is dependent on the maximum supported configuration size and the settings of maximumLSPGenerationInterval. For these purposes the default value of 900 seconds can be assumed. The number of LSPs per second is then very approximately $(n_1 + n_2$

$+n_e/x)/900$ where n_1 is the number of level 1 Intermediate systems, n_2 the number of level 2 Intermediate systems, n_e the number of End system IDs and x the number of ID which can be fitted into a single LSP.

NOTE — This gives a value around 1 per second for typical maximum configurations of

- x 4000 IDs
- x 100 L1 Intermediate systems per area
- x 400 L2 Intermediate systems

d) Data Traffic. This is theoretically unlimited and can arrive at the maximum data rate of the Point–to-Point link or LAN (for ISO 8802.3 this is 14,000 PDUs per second). In practice it will be limited by the operation of the congestion avoidance and control algorithms, but owing to the relatively slow response time of these algorithms, substantial peaks are likely to occur.

An Intermediate system shall state in its PICS its maximum forwarding rate. This shall be quoted under at least the following conditions.

1) A stable topology of maximum size.

2) A maximally unstable topology. This figure shall be non-zero, but may reasonably be as low as 1 PDU per second.

The following constraints must be met.

a) The implementation shall be capable of receiving the maximum rate of ISH PDUs without loss whenever the following conditions hold

1) The data forwarding traffic rate averaged over any period of one second does not exceed the rate which the implementation claims to support

2) The ESH and LSP rates do not exceed the background (stable topology) rate.

b) If it is unavoidable that PDUs are dropped, it is a goal that the order of retaining PDUs shall be as follows (i.e. It is least desirable for IIH PDUs to be dropped).

1) IIH PDUs

2) ESH PDUs

3) LSPs and SNPs

4) data PDUs.

However, no class of traffic shall be completely starved. One way to achieve this is to allocate a queue of suitable length to each class of traffic and place the PDUs onto the appropriate queue as they arrive. If the queue is full the PDUs are discarded. Processor resources shall be allocated to the queues to ensure that they all make progress with the same priorities as above. This model assumes that an implementation is capable of receiving PDUs and selecting their correct queue at the maximum possible data rate (14,000 PDUs per second for a LAN). If this is not the case, reception of data traffic at a rate greater than some limit (which must be greater than the maximum rated limit) will cause loss of some IIH PDUs even in a stable topology. This limit shall be quoted in the PICS if it exists.

NOTE - Starting from the stable topology condition at maximum data forwarding rate, an increase in the arrival rate of data PDUs will initially only cause some data NPDUs to be lost. As the rate of arrival of data NPDUs is further increased a point may be reached at which random PDUs are dropped. This is the rate which must be quoted in the PICS

12.2.5.4 Transmission

Sufficient processor resources shall be allocated to the transmission process to enable it to keep pace with reception for each PDU type. Where prioritisation is required, the same order as for reception of PDU types applies.

Annex A
PICS Proforma

(This annex is normative)

TEMPORARY NOTE - The editor has included boilerplate for a PICS as part of the DP text, which reflects the current SC6 draft guidelines on PICS proformas. It is included so that member bodies may see the PICS guidelines applied in the context of an evolving standard, and hence comment both on the particular material here as part of the DP ballot, and on the guidelines themselves.

A.1 Introduction

The supplier of a protocol implementation which is claimed to conform to International Standard ISO 10589, whether as a level 1 or level 2 Intermediate system implementation, shall complete the applicable Protocol Implementation Conformance Statement (PICS) proforma.

A completed PICS proforma is the PICS for the implementation in question. The PICS is a statement of which capabilities and options of the protocol have been implemented. The PICS can have a number of uses, including use:

- by the protocol implementor, as a check-list to reduce the risk of failure to conform to the standard through oversight;
- by the supplier and acquirer — or potential acquirer — of the implenetation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;
- by the user — or potential user — of the implementation, as a basis for initially checking the possibility of interworking with another implementation (note that, while interowrking can never be guaranteed, failure to interwork can often be predicted from imcompatible PICS's);
- by a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

A.2 Abbreviations and Special Symbols

A.2.1 Status-related symbols

M	mandatory
O	optional
O.<n>	optional, but support of at least one of the group of options labelled by the same numeral <n> is required.
X	prohibited
–	not applicable

c.<p>	conditional requirement, according to condition <p>

A.3 Instructions for Completing the PICS Proformas

A.3.1 General structure of the PICS proforma

The first part of the PICS proforma — Implementation Identification and Protocol Summary — is to be completed as indicated with the information necessary to identify fully both the supplier and the implementation.

The main part of the PICS proforma is a fixed-format questionaire divided into subclauses each containing a group of individual items. Answers to the questionaire items are to be provided in the rightmost column, either by simply marking an answer to indicate a restricted choice (usually Yes or No), or by entering a value or a set or range of values. (Note that there are some tiems where two or morew choices from a set of possible answers can apply: all relevant choices are to be marked.)

Each item is identified by an item reference in the first column; the second column contains the question to be answered; the third column contains the reference or references to the material that specifies the item in the main body of the standard. the remaining columns record the status of the item — whether support is mandatory, optional or conditional — and procide the space for the answers: see A.3.4 below.

A supplier may also provide — or be required to provide — further information, categorised as either Additional Information or Exception Information. When present, each kind of further information is to be provided in a further subclause of items labelled A<i> or X<i> respectively for cross-referencing purposes, where <i> is any unambiguous identification for the item (e.g. simply a sumber): there are no other restrictions on its format and presentation.

A completed PICS proforma, including any Additonal Information and Exception Information, is the Protocol Implementation Conformance Statement for the implementation in question.

NOTE - Where an implementation is capable of being configured in more than one way, a single PICS may be able to describe all such configurations. However, the supplier has the choice of providing more than one PICS, each covering some subset of the implementation's configuration capabilities, in case this makes for easier and clearer presentation of the information.

A.3.2 Additional Information

Items of Additional Information allow a supplier to provide further information intended to assist the interpretation of the PICS. It is not intended or expected that a lerge quantity will be supplied, and a PICS can be considered complete without any such information. Examples might be an outline of the ways in which a (single) implementation can be set up to operate in a variety of environments and configurations.

References to items of Additional information may be entered next to any answer in the questionaire, and may be included in items of Exception Information.

A.3.3 Exception Information

It may occasionally happen that a supplier will wish to answer an item with mandatory or prohibited status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No pre-printed answer will be found in the Support column for this, but the Supplier may write the desired answer into the Support column. If this is done, the supplier is required to provide an item of Excpetion Information containing the appropriate rationale, and a cross-reference from the inserted answer to the Exception item.

An implementation for which an Exception item is required in this way does not conform to ISO 10589.

NOTE - A possible reason for the situation described above is that a defect report is being progressed, which is expected to change the requirement that is not met by the implementatkn.

A.3.4 Conditional Status

A.3.4.1 Conditional items

The PICS proforma contains a number of conditional items. These are items for which the status — mandatory, optional or prohibited — that applies is dependent upon whether or not certain other items are supported, or upon the values supported for other items. In many cases, whether or not the item applies at all is conditional in this way, as well as the status when the item does apply.

Individual conditional items are indicated by a conditional symbol in the Status column as described in A.3.4.2 below. Where a group of items are subject to the same condition for applicability, a separate preliminary question about the condition appears at the head of the group, with an instruction to skip to a later point in the questionaire if the "Not Applicable" answer is selected.

A.3.4.2 Conditional symbols and conditions

A conditional symbol is of the form c.<n> or c.G<n> where <n> is a numeral. For the first form, the numeral identifies a condition appearing in a list at the end of the subclause containing the item. For the second form, c.G<n>, the numeral identifies a condition appearing in the list of global conditions at the end of the PICS.

A simple condition is of the form:

if <p> then <s1> else <s2>

where <p> is a predicate (see A.3.4.3 below), and <s1> and <s2> are either basic status symbols (M,O,O.<n>, or X) or the symbol "–". An extended condition is of the form

if <p1> then <s1> else <s2>
else if <p2> then <s2>
[else if <p3> ...]
else <s$_n$>

where <p1> etc. are predicates and <s1> etc. are basic status symbols or "–".

The status symbol applicable to an item governed by a simple condition is <s1> if the predicate of the condition is true, and <s2> otherwise; the status symbol applicable to an item governed by an extended condition is <s$_i$> where <p$_i$> is the first true predicate, if any, in the sequence <p1>, <p2>..., and <s$_n$> if no predicate is true.

A.3.4.3 Predicates

A simple predicate in a condition is either

a) a single item reference; or

b) a relation containing a comparison operator (=, <, etc.) with one (or both) of tis operands being an item reference for an item taking numerical values as its answer.

In case (a) the predicate is true if the item referred to is marked as supported, and false otherwise. In case (b), the predicate is true if the relation holds when each item reference is replaced by the value entered in the Support column as answer to the item referred to.

Compound predicates are boolean expressions constructed by combining simple predicates using the boolean operators AND, OR and NOT, and parentheses, in the usual way. A compound predicate is true if and only if the boolean expression evaluates to true when the simple predicates are interpreted as described above.

Items whose references are used in predicates are indicated by an asterisk in the Item column.

A.3.4.4 Answering conditional items

To answer a conditional item, the predicate(s) of the condition is (are) evaluated as described in A.3.4.3 above, and the applicable status symbol is determined as described in A.3.4.2. If the status symbol is "–" this indicates that the item is to be marked in thisa case; otherwise, the Support column is to be completed in the usual way.

When two or more basic status symbols appear in a condition for an item, the Support column for the item contains one line for each such symbol, labelled by the relevant symbol. the answer for the item is to be marked in the line labelled by the symbol selected according to the value of the condition (unselected lines may be crossed out for added clarity).

For example, in the item illustrated below, the N/A column would be marked if neither predicate were true; the answer line lebelled "M:" would be marked if item A4 was marked as supported, and the answer line labelled "O:" would be marked if the condition including items D1 and B52 applied.

Item		References	Status	N/A	Support
H3	Is ... supported?	42.3(d)	c.2	☐	M: Yes ☐ O: Yes ☐ No ☐

c.2: if A4 then M
else if D1 AND (B52 < 3) then O else –

A.4 Identification

A.4.1 Implementation Identification

Supplier	
Contact point for queries about this PICS	
Implementation Name(s) and Version(s)	
Operating system Name(s and Version(s)	
Other Hardware and Operating Systems Claimed	
System Name(s) (if different)	

Notes:

a) Only the first three items are required for all implementations; others may be completed as appropriate in meeting the requirements for full identification.

b) The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (using, e.g., Type, Series, Model)

A.4.2 Protocol Summary: ISO 10589:19xx

Protocol Version	
Addenda Implemented (if applicable)	
Ammendments Implemented	

Have any Exception items been required (see A.3.3)? No ☐ Yes ☐
(The answer Yes means that the implementation does not conform to ISO 10589)

Date of Statement	

PICS Proforma: ISO 10589:19xx

...TBS...

Annex B
Supporting Technical Material

(This annex is informative)

B.1 Matching of Address Prefixes

The following example shows how address prefixes may be matched according to the rules defined in 7.1.4.

The prefix
37-123

matches both the full NSAP addresses
37-1234::AF… and
37-123::AF…

which are encoded as
3700000000001234AF… and
3700000000000123AF…

respectively.

This can be achieved by first converting the address to be compared to an internal *decoded* form (i.e. any padding, as indicated by the particular AFI, is removed), which corresponds to the external representation of the address. The position of the end of the IDP must be marked, since it can no longer be deduced. This is done by inserting the semi-octet "F" after the last semi-octet of the IDP. (There can be no confusion, since the abstract syntax of the IDP is decimal digits).

Thus the examples above become in *decoded* form
371234FAF… and
37123FAF…

and the prefix 37-123 matches as a leading sub-string of both of them.

For comparison purposes the prefix is converted to the internal *decoded* form as above.

B.2 Addressing and Routeing

In order to ensure the unambiguous identification of Network and Transport entities across the entire OSIE, some form of address administration is mandatory. ISO 8348/Add.2 specifies a hierarchical structure for network addresses, with a number of top-level domains responsible for administering addresses on a world-wide basis. These address registration authorities in turn delegate to sub-authorities the task of administering portions of the address space. There is a natural tendency to repeat this sub-division to a relatively fine level of granularity in order to ease the task of each sub-authority, and to assign responsibility for addresses to the most “localised” administrative body feasible. This results in (at least in theory) reduced costs of address administration and reduced danger of massive address duplication through administrative error. Furthermore, political factors come into play which require the creation of sub-authorities in order to give competing interests the impression of “hierarchical parity”. For example at the top level of the ISO geographic address space, every country is assigned an equally-sized portion of the address space even though some countries are small and might in practice never want to undertake administration of their own addresses. Other examples abound at lower levels of the hierarchy, where divisions of a corporation each wish to operate as an independent address assignment authority even though this is inefficient operationally and may waste monumental amounts of potential address space.

If network topologies and traffic matrices aligned naturally with the hierarchical organisation of address administration authorities, this profligate use of hierarchy would pose little problem, given the large size (20 octets) of the N-address space. Unfortunately, this is not usually the case, especially at higher levels of the hierarchy. Network topologies may cross address administration boundaries in many cases, for example:

- Multi-national Corporations with a backbone network that spans several countries
- Community-of-interest networks, such as academic or research networks, which span organisations and geographies
- Military networks, which follow treaty alignments rather than geographic or national administrations
- Corporate networks where divisions at times operate as part of a contractor’s network, such as with trade consortia or government procurements.

These kinds of networks also exhibit rich internal topologies and large scale (10^5 systems), which require sophisticated routeing technology such as that provided by the proposed Intra-domain IS–IS routeing protocol. In order to deploy such networks effectively, a considerable amount of address space must be left over for assignment in a way which produces efficient routes without undue consumption of memory and bandwidth for routeing overhead[18].

Similarly important is the inter-connection of these networks via Inter-domain routeing technology. If all of the assignment flexibility of the addressing scheme is exhausted in purely administrative hierarchy (at the high-order end of the address) and in Intra-Domain routeing assignment (at the low end of the address) there may be little or no address space left to customise to the needs of inter-domain routing. Since the work on Inter-Domain routing in ISO is still at a very early stage, it is difficult to predict how seriously the need for routeing efficiency will collide with the need for administrative hierarchy. I suspect that if some amount of address space is not left unassigned the ability to construct reasonably rich inter-domain inter-connections will be seriously constrained.

The considerations for how addresses should be structured for the Intra- and Inter-domain cases are discussed in more detail in the following two clauses.

B.2.1 Address Structure for Intra-domain Routeing

The IS-IS Intra-domain routeing protocol uses a “preferred” addressing scheme. There are a number of reasons the designers of this protocol chose to specify a single address structure, rather than leaving the matter entirely open to the address assignment authorities and the routeing domain administrators:

a) If one address structure is very common and known a priori, the forwarding functions can be made much faster;

b) If part of the address is known to be assigned locally to an end system, then the routeing can be simpler, use less memory, and be potentially faster, by not having to discriminate based on that portion of the address.

c) If part of the address can be designated as globally unique by itself (as opposed to only the entire address having this property) a number of benefits accrue:

    1) Errors in address administration causing duplicate addresses become much less likely

---

[18] This is just a fancy way of saying that hierarchical routing, with its natural effect on address assignment, is a mandatory requirement for such networks.

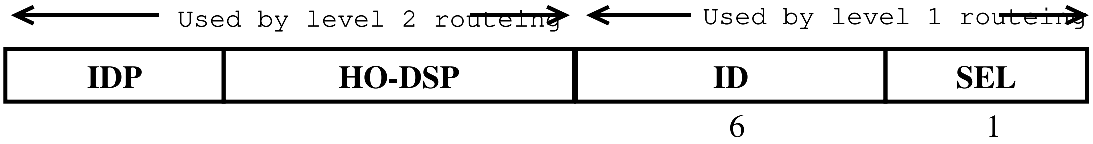
2) Automatic and dynamic NSAP address assignment becomes feasible without global knowledge or synchronisation

3) Routeing on this part of the address can be made simple and fast, since no address collisions will occur in the forwarding database.

d) If a part of the address can be reserved for assignment purely on the basis of topological efficiency (as opposed to political or address administration ease), hierarchical routeing becomes much more memory and bandwidth efficient, since the addresses and the topology are in close correspondence.

e) If an upper bound can be placed on the amount of address space consumed by the Intra-domain routeing scheme, then the use of address space by Inter-domain routeing can be made correspondingly more flexible.

The preferred address format of the Intra-domain IS–IS protocol achieves these goals by being structured into two fixed-sized fields as follows:



Key:

IDP	Initial Domain Part
HO-DSP	High Order Domain Specific Part
ID	System Identifier
SEL	NSAP Selector

Figure 8 - Preferred Address Format

The field marked **IDP** in the figure is precisely the IDP specified in ISO 8348/Add.2. The field marked **HO-DSP** is that portion of the DSP from ISO 8348/Add.2 whose structure, assignment, and meaning are *not* specified or constrained by the Intra-domain IS–IS routeing protocol. However, the design presumes that the routeing domain administrator has at least some flexibility in assigning a protion of te HO-DSP field. The purpose and usage of the fields specified by the Intra-domain IS–IS routeing protocol is explained in the following paragraphs.

B.2.1.1 The IDP + HO-DSP

Since the Intra-domain IS–IS protocol is customised for operation with ISO 8473, all addresses are specified to use the preferred binary encoding of ISO 8348/Add.2. Furthermore, the protocol required that the AFI specify binary DSP encoding in order to unnecessary encoding transformations. This is not considered a serious restriction, since decimal syntax DSPs have not (to this editor’s knowledge) been used with ISO 8473, and a simple transformation can render them valid addresses using an AFI that indicates a binary DSP.

B.2.1.2 The *Selector* (SEL) Field

The **SEL** field is intended for two purposes. Its main use is to allow for multiple higher-layer entities in End systems (such as multiple transport entities) for those systems which need this capability. This allows up to 256 NSAPS in a single End system. The advantage of reserving this field exclusively for local system administration the Intra-domain routing functions need not store routeing information about, nor even look at this field. If each individual NSAP were represented explicitly in routing tables, the size of these tables would grow with the number of NSAPs, rather than with the number of End systems. Since Intra-domain routing routes to *systems*, explicit recording of each NSAP brings no efficiency benefit and potentially consumes large amounts of memory in the Intermediate systems.

A second use for the **SEL** field is in Intermediate systems. Certain IS–IS functions require that PDUs be encapsulated and sent to the Network Entity in an Intermediate system rather than to an NSAP and upward to a Transport entity. An example of this is the Partition Repair function of the Intra-domain IS–IS protocol. In order to use a level 2 path as if it were a single subnetwork in a level 1 area, PDUs are encapsulated and addressed to an IS on the other side of the partition[19]. By reserving certain values of the **SEL** field in Intermediate systems for direct addressing of Intermediate system Network entities, the normal addressing and relaying functions of other Intermediate systems can be transparently used for such purposes.

B.2.1.3 The *Identifier* (ID) Field

The **ID** field is a "flat", large identifier space for identifying OSI systems. The purpose of this field is to allow very fast, simple routeing to a large (but not unconstrained) number of End systems in a routeing domain. The Intra-Domain IS–IS protocol uses this field for routeing within a *area.* While this field is only required to be unambiguous within a single area, if the values are chosen to be globally unambiguous the Intra-domain IS–IS design can exploit this fact in the following ways.

First, a certain amount of parallelism can be obtained during relaying. An IS can be simultaneously processing the **ID** field along with other fields (i.e. **IDP**, **HO-DSP**, **LOC-AREA**). If the **ID** is found in the forwarding table, the IS can initiate forwarding while checking to make sure that the other fields have the expected value. Conversely, if the **ID** is not found the IS can assume that either the addressed NSAP is unreachable or exists only in some other area or routeing domain. In the case where the **ID** is not globally unique, the forwarding table can indicate this fact and relaying delayed until the entire address is analysed and the route looked up.

Second, a considerable savings can be obtained in manual address administration for all systems in the routeing domain. If the **ID** is chosen from the ISO 8802 48-bit address space, the **ID** is known to be globally unique. Furthermore, since LAN systems conforming to ISO 8802 often have their 48-bit MAC address stored in ROM locally, each system can be guaranteed to have a globally unambiguous NET and NSAP(s) without centralised address administration at the area level.[20] This not only eliminates administrative overhead, but also drastically reduces the possibility of duplicate NSAP addresses, which are illegal, difficult to diagnose, and often extremely difficult to isolate.

An alternative to a large, flat space for the lowest level of routeing would be to hierarchically subdivide this field to allow more levels of routeing within a single routeing domain. The designers of the Intra-domain IS–IS protocol considered that this would lead to an inferior routeing architecture, since:

a) The cost of memory in the ISs was sufficiently reasonable that large (e.g. 10^4 system) areas were quite feasible, thus requiring at least 2 octets per level to address

b) Two levels of routeing within a routeing domain were sufficient (allowing domains of 10^6–10^7 systems) because it was unlikely that a single organisation would wish to operate and manage a routeing domain much larger than that.

c) Administrative boundaries often become the dominant concern once routeing domains reach a certain size.

d) The additional burdens and potential for error in manual address assignment were deemed serious enough to warrant the use of a large, flat space.

B.3 Use of the HO-DSP field in Intra-domain routeing

Use of a portion of the **HO-DSP** field provides for hierarchical routeing within a routeing domain. A value is assigned to a set of ISs in order to group the ISs into a single area for the usual benefits of hierarchical routeing:

a) Limiting the size of routeing tables in the ISs;

b) conserving bandwidth by hierarchical summarisation of routeing information;

---

[19] This is a gross oversimplification for the purpose of illustrating the need for the **SEL** field. See the Intra-domain IS–IS specification for a full explanation.

[20] Note, however, that the use of the ISO 8802 addresses does *not* avoid the necessity to run ISO 9542 or to maintain tables mapping NSAP addresses to MAC (i.e. SNPA) addresses on the ISO 8802 subnetwork. This is because there is no guarantee that a particular MAC address is always enabled (the LAN controller may be turned off) or that a system has only a single MAC address.

c) designating portions of the network which are to have optimal routeing within themselves; and

d) moderate firewalling of portions of the routeing domain from failures in other portions.

It is important to note that the assignment of **HO-DSP** values is intended to provide the routeing domain administrator with a mechanism to optimise the routeing within a large routeing domain. The Intra-domain IS–IS designers did *not* intend the **HO-DSP** to be entirely consumed by many levels of address registration authority. Reserving the assignment of a portion of the **HO-DSP** field to the routeing domain administrator also allows the administrator to start with a single assigned **IDP+HO-DSP** and run the routing domain as a single area. As the routeing domain grows, the routeing domain administrator can then add areas without the need to go back to the address administration authority for further assignments. Areas can be added and re-assigned within the routeing domain without involving the external address administration authority.

A useful field to reserve as part of the **HO-DSP** would be 2 octets,permiting up to 65,536 areas in a routeing domain. This is viewed as a reasonable compromise between routeing domain size and address space consumption. The field should be specified as flat for the same reasons that the **ID** field is similarly specified.

B.3.1 Addressing considerations for Inter-domain Routeing

Given that the work on Inter-domain routeing is still in preliminary stages, it is difficult to make concrete statements on how addressing and Inter-domain routeing will interact at the detailed level. However, since the Intra-domain IS–IS protocol supports limited Inter-domain routeing capability (via static routeing), some preliminary observations are possible.

Firstly, it is in the Inter-domain arena where the goals of routeing efficiency and administrative independence collide most strongly. Although the OSI Routeing Framework explicitly gives priority in Inter-domain routeing to considerations of autonomy and firewalls over efficiency, it must be *feasible* to construct an Inter-Domain topology that both produces isolable domains and relays data at acceptable cost. Since no routeing information is exchanged across domain boundaries with static routeing, the practicality of a given Inter-domain topology is essentially determined by the size of the routeing tables that are present at the boundary ISs. If these tables become too large, the memory needed to store them, the processing needed to search them, and the bandwidth needed to transmit them within the routeing domain all combine to disallow certain forms of interconnection.

Inter-domain routeing primarily computes routes to other routeing domains[21]. If there is no correspondence between the address registration hierarchy and the organisation of routeing domains (and their interconnection) then the task of static table maintenance quickly becomes a nightmare, since each and every routeing domain in the OSIE would need a table entry potentially at every boundary IS of every other routeing domain. Luckily, there is some reason to believe that a natural correspondence exists, since at least at the global level the address registration authorities fall within certain topological regions. For example, most of the routeing domains which obtained their **IDP+HO-DSP** from a hierarchy of French authorities are likely to reside in France and be more strongly connected with other routeing domains in France that with routeing domains in other countries.

There are enough exceptions to this rule, however, to be a cause for concern. The scenarios cited in clause <routeingAndAddressing> all exist today and may be expected to remain common for the foreseeable future. Consider as a practical case the High Energy Physics Network (HEPnet), which contains some 17000 End systems, and an unknown number of intermediate systems[22]. This network operates as a single routeing domain in order to provide a known set of services to a known community of users, and is funded and cost-justified on this basis. This network is international in scope (at least 10 countries in North America, Europe, and the far east) and yet its topology does not map well onto existing national boundaries. Connectivity is richer between CERN and FERMIlab, for example than between many points within the U.S.

More importantly, this network has rich connectivity with a number of other networks, including the PDNs of the various countries, the NSFnet in the U.S., the international ESnet (Energy Sciences Network), the general research

---

[21] The Intra-domain IS–IS protocol also uses static Inter-domain tables for routeing to individual End systems across dynamically assigned circuits, and also to End systems whose addresses do not conform to the address construction rules.

[22] The number of ISs is hard to estimate since some ISs and links are in fact shared with other networks, such as the similarly organised NASA Space Physics network, or “SPAN”.

Internet, and military networks in the U.S. and elsewhere. None of these other networks shares a logical part of the NSAP address hierarchy with HEPnet[23] . If the only method of routing from the HEPnet to these other networks was to place each within one and only one of the existing registration authorities, and to build static tables showing these relationships, the tables would clearly grow as $O(n^2)$.

It seems therefore, that some means must be available to assign addresses in a way that captures the Inter-Domain topology, and which co-exists cleanly with both the administrative needs of the registration authorities, and the algorithms employed by both the Intra- and Inter-domain routeing protocols. As alluded to in an earlier clause, it seems prudent to leave some portion of the address space (most likely from the **HO-DSP** part) sufficiently undefined and flexible that various Inter-domain topologies may be efficiently constructed. It would behoove ISO to advise national and international registration authorities on this matter and to encourage them to avoid over-defining the NSAP address structure for the portion of the address space which they administer. This extends also to the GOSIPs of the various countries.

---

[23] It is conceivable that ISO would sanction such networks by assigning a top-level IDI from the ISO non-geographic AFI, but this is unlikely and would only exacerbate the problem if many such networks were assigned top-level registrations.

Annex C
Implementation Guidelines and Examples

(This annex is informative)

C.1 Routeing Databases

Each database contains records as defined in the following sub-clauses. The following datatypes are defined.

```
FROM CommonMgmt IMPORT NSAPAddress, AddressPrefix, BinaryAbsoluteTime;
TYPE
    lspID = ARRAY [0..7] OF Octet;
    systemID = ARRAY [0..5] OF Octet;
    octetTimeStamp = BinaryAbsoluteTime;
```

C.1.1 Level 1 Link State Database

This database is kept by Level 1 and Level 2 Intermediate Systems, and consists of the latest Level 1 Link State PDUs from each Intermediate System (or pseudonode) in the area. The Level 1 Link State PDU lists Level 1 links to the Intermediate System that originally generated the Link State PDU.

```
RECORD
    adr: lspID;                          (* 8 octet ID of LSP originator *)
    type: (Level1IntermediateSystem, AttachedLevel2IntermediateSystem,
    UnattachedLevel2IntermediateSystem);
    seqnum: [0..SequenceModulus];
    LSPage: [0..MaxAge];                 (*Remaining Lifetime *)

    expirationTime: TimeStamp;           (*Time at which LSP age became zero (see 7.3.16.4). *)
    SRMflags: ARRAY[1..(maximumCircuits + maximumVirtualAdjacencies)]OF BOOLEAN;
        (*Indicates this LSP to be sent on this circuit. Note that level 2 Intermediate systems may send level
        1 LSPs to other partitions (if any exist). Only one level 2 Intermediate system per partition does
        this. For level 1 Intermediate Systems the array is just maximumCircuits long. *)
    SSNflags: ARRAY[1..maximumCircuits + maximumVirtualAdjacencies] OF BOOLEAN;
        (*Indicates that information about this LSP shall be included in the next partial sequence number
        PDU transmitted on this circuit. *)
    POINTER TO LSP;                      (*The received LSP *)
    END;
```

C.1.2 Level 2 Link State Database

This database is kept by Level 2 Intermediate Systems, and consists of the latest Level 2 Link State PDUs from each Level 2 Intermediate System (or pseudonode) in the domain. The Level 2 Link State PDU lists Level 2 links to the Intermediate System that originally generated the Link State PDU.

```
RECORD
   adr: lspID;                              (* 8 octet ID of LSP originator *)
   type: (AttachedLevel2IntermediateSystem, UnattachedLevel2IntermediateSystem);
      seqnum: [0..SequenceModulus];
      LSPage: [0..MaxAge];                        (*Remaining Lifetime *)

      expirationTime: TimeStamp;                  (*Time at which LSP age became zero (see
                       7.3.16.4). *)
      SRMflags: ARRAY[1..(maximumCircuits)] OF BOOLEAN;
         (*Indicates this LSP to be sent on this circuit. *)
      SSNflags: ARRAY[1..maximumCircuits] OF BOOLEAN;
         (*Indicates that information about this LSP must be included in the next partial sequence number
         PDU transmitted on this circuit. *)
      POINTER TO LSP;                       (*The received LSP *)
      END;
```

C.1.3 Adjacency Database

This database is kept by all systems. Its purpose is to keep track of neighbours.

For Intermediate systems, the adjacency database comprises a database with an entry for each:

- Adjacency on a Point to Point circuit.
- Broadcast Intermediate System Adjacency. (Note that both a Level 1 and a Level 2 adjacency can exist between the same pair of systems.)
- Broadcast End system Adjacency.
- potential SVC on a DED circuit (maximumSVCAdjacencies for a DA circuit, or 1 for a Static circuit).
- Virtual Link Adjacency.

Each entry contains the parameters in Clause 11 for the Adjacency managed object. It also contains the variable used to store the remaining holding time for each Adjacency IDEntry and NETEntry entry, as defined below.

```
IDEntry = RECORD
   ID: SystemID;
      (* The 6 octet System ID of a neighbour End system extracted from the SOURCE ADDRESS field of its
      ESH PDUs. *)
   entryRemainingTime: Unsigned [1..65535]
      (* The remaining holding time in seconds for this entry. This value is not accessible to system
      management. An implementation may choose to implement the timer rules without an explicit
      remainingTime being maintained. For example by the use of asynchronous timers. It is present here
      in order to permit a consistent description of the timer rules. *)
   END

NETEntry = RECORD
   NET: NetworkEntityTitle;                       (* The NET of a neighbour Intermediate system
      as reported in its IIH PDUs. *)
   entryRemainingTime: Unsigned [1..65535]
       (* The remaining holding time in seconds for this entry. This value is not accessible to system
```

```
    management.  An implementation may choose to implement the timer rules without an explicit
    remainingTime being maintained.  For example by the use of asynchronous timers. It is present here
    in order to permit a consistent description of the timer rules. *)
  END;
```

C.1.4 Circuit Database

This database is kept by all systems. Its purpose is to keep information about a circuit. It comprises an **ARRAY**[1..maximumCircuits].

Each entry contains the parameters in Clause 11 for a Circuit managed object (see 11.3). It also contains the remainingHelloTime (WordUnsigned [1..65535] seconds) variable for the Circuit. This variable **not** accessible to system management. An implementation may choose to implement the timer rules without an explicit remainingHelloTime being maintained. For example by the use of asynchronous timers. It is present here in order to permit a consistent description of the timer rules. Additionally, for Circuits of type "X.25 Static Outgoing" or "X.25 DCM", it contains the recallCount (Unsigned[0..255]) variable for the Circuit. This variable is **not** accessible to system management. It used to keep track of recall attempts.

C.1.5 Level 1 Shortest Paths Database

This database is kept by Level 1 and Level 2 Intermediate Systems (unless each circuit is "Level 2 Only"). It is computed by the Level 1 Decision Process, using the Level 1 Link State Database. The Level 1 Forwarding Database is a subset of this database.

```
RECORD
  adr: systemId;                              (*6 octet ID of destination system *)
  cost: [1..MaxPathMetric];                   (*Cost of best path to destination system *)
  adjacencies: ARRAY[1..maximumPathSplits] OF POINTER TO Adjacency;
    (*Pointer to adjacency for forwarding to system adr *)
  END;
```

C.1.6 Level 2 Shortest Paths Database

This database is kept by Level 2 Intermediate Systems. It is computed by the Level 2 Decision Process, using the Level 2 Link State Database. The Level 2 Forwarding Database is a subset of this database.

```
RECORD
  adr: AddressPrefix;                         (*destination prefix *)
  cost: [1..MaxPathMetric];                   (*Cost of best path to destination prefix *)
  adjacencies: ARRAY[1..maximumPathSplits] OF  POINTER TO Adjacency;
    (*Pointer to adjacency for forwarding to prefix adr *)
  END;
```

C.1.7 Level 1 Forwarding Database

This database is kept by Level 1 and Level 2 Intermediate Systems (unless each circuit is "Level 2 Only"). It is used to determine where to forward a data NPDU with destination within this system's area. It is also used to determine how to reach a Level 2 Intermediate System within the area, for data PDUs with destinations outside this system's area.

```
RECORD
  adr:systemId;                               (*6 octet ID of destination system. Destination "0" is
  special, meaning"nearest level 2 Intermediate system *)
```

```
    splits: [0..maximumPathSplits];              (* Number of valid output adj's for reachingadr (0
    indicates it is unreachable) *)
     nextHop: ARRAY[1..maximumPathSplits] OF POINTER TO adjacency;
        (*Pointer to adjacency for forwarding to destination system *)
    END;
```

C.1.8 Level 2 Forwarding Database

This database is kept by Level 2 Intermediate systems. It is used to determine where to forward a data NPDU with destination outside this system's area.

```
RECORD
    adr: AddressPrefix;                          (*address of destination area. *)
    splits: [0..maximumPathSplits];              (*Number of valid output adj's for reaching adr (0
    indicates it is unreachable) *)
    nextHop: ARRAY[1..maximumPathSplits] OF POINTER TO adjacency;
        (*Pointer to adjacency for forwarding to destination area. *)
    END;
```

C.2 SPF Algorithm for Computing Equal Cost Paths

An algorithm invented by Dijkstra (see references) known as *shortest path first (SPF)*, is used as the basis for the route calculation. It has a computational complexity of the square of the number of nodes, which can be decreased to the number of links in the domain times the log of the number of nodes for sparse networks (networks which are not highly connected).

Two additional optimisations are possible:

a) If the routeing metric is defined over a small finite field (as in this International Standard), the factor of *log n* may be removed by using data structures which maintain a separate list of systems for each value of the metric rather than sorting the systems by logical distance.

b) Updates can be performed incrementally without requiring a complete recalculation. However, a full update must be done periodically to recover from data corruption, and studies suggest that with a very small number of link changes (perhaps 2) the expected computation complexity of the incremental update exceeds the complete recalculation. Thus, this International Standard specifies the algorithm only for the full update.

The original SPF algorithm does not support load splitting over multiple paths. The algorithm in this International Standard does permit load splitting by identifying a set of equal cost paths to each destination rather than a single least cost path.

C.2.1 Databases

PATHS – This represents an acyclic directed graph of shortest paths from the system *S* performing the calculation. It is stored as a set of triples of the form ⟨*N,d(N),{Adj(N)}*⟩, where:

N is a system Identifier. In the level 1 algorithm, *N* is a 7 octet ID. For a non-pseudonode it is the 6 octet system ID, with a 0 appended octet. For a pseudonode it is a true 7 octet quantity, comprised of the 6 octet Designated Intermediate System ID and the extra octet assigned by the Designated Intermediate System. In the level 2 algorithm it is either a 7 octet Intermediate System or pseudonode ID (as in the level 1 algorithm), or it is a variable length address prefix (which will always be a leaf, i.e. "End system", in PATHS).

d(N) is *N*'s distance from *S* (i.e. the total metric value from *N* to *S*).

{Adj(N)} is a set of valid adjacencies that *S* may use for forwarding to *N*.

When a system is placed on PATHS, the path(s) designated by its position in the graph is guaranteed to be a shortest path.

TENT – This is a list of triples of the form ⟨*N,d(N),{Adj(N)}*⟩, where *N, d(N)* and *{Adj(N)}* are as defined above for PATHS.

TENT can intuitively be thought of as a tentative placement of a system in PATHS. In other words, the triple ⟨*N,x,{A}*⟩ in TENT means that if *N* were placed in PATHS, *d(N)* would be *x*, but *N* cannot be placed on PATHS until it is guaranteed that no path shorter than *x* exists.

The triple ⟨*N,x,{A,B}*⟩ in TENT means that if *N* were placed in PATHS, *d(N)* would be *x* via either adjacency *A* or *B*

NOTE - As described above, (see 7.2.6), it is suggested that the implementation keep the database TENT as a set of lists of triples of the form ⟨**,Dist,**⟩, for each possible distance Dist. In addition it is necessary to be able to process those systems which are pseudonodes before any non-pseudonodes at the same distance Dist.

C.2.2 Overview of the Algorithm

The basic algorithm, which builds PATHS from scratch, starts out by putting the system doing the computation on PATHS (no shorter path to SELF can possibly exist). TENT is then pre-loaded from the local adjacency database.

Note that a system is not placed in PATHS unless no shorter path to that system exists. When a system *N* is placed in PATHS, the path to each neighbour *M* of *N*, through *N*, is examined, as the path to *N* plus the link from *N* to *M*. If ⟨*M,*,**⟩ is in PATHS, this new path will be longer, and thus ignored.

If ⟨*M,*,**⟩ is in TENT, and the new path is shorter, the old entry is removed from TENT and the new path is placed in TENT. If the new path is the same length as the one in TENT, *N* is added as a potential parent in the triple in TENT. If *M* is not in TENT, then the path is added to TENT.

Next the algorithm finds the triple ⟨*N,x,{Adj(N)}*⟩ in TENT, with minimal *x*.

NOTE - This is done efficiently because of the optimisation described above. When the list of triples for distance Dist is exhausted, the algorithm then increments Dist until it finds a list with a triple of the form ⟨*,Dist,*⟩.

N is placed in PATHS. We know that no path to *N* can be shorter than *x* at this point because all paths through systems already in PATHS have already been considered, and paths through systems in TENT will have to be greater than *x* because *x* is minimal in TENT.

When TENT is empty, PATHS is complete.

C.2.3 The Algorithm

The Decison Process Algorithm must be run once for each supported routeing metric. A Level 1 Intermediate System runs the algorithm using the Level 1 LSP database to compute Level 1 paths. In addition a Level 2 Intermediate System runs the algorithm using the Level 2 LSP database to compute Level 2 paths.

If this system is a Level 2 Intermediate System which supports the partition repair optional function the Decision Process algorithm for computing Level 1 paths must be run twice for the default metric. The first execution is done to determine which of the area's ManualAreaAddresses are reachable in this partition, and elect a Partition Designated Level 2 Intermediate System for the partition. The Partition Designated Level 2 Intermediate System will determine if the area is partitioned and will create virtual Level 1 links to the other Partition Designated Level 2 Intermediate Systems in the area in order to repair the Level 1 partition. This is further described in clause 7.2.10.

Step 0: Initialise TENT and PATHS to empty. Initialise tentlength to 1.

(tentlength is the pathlength of elements in TENT we are examining.)

a) Add ⟨SELF, 0, W⟩ to PATHS, where W is a special value indicating traffic to SELF is passed up to Transport (rather than forwarded).

b) Now pre-load TENT with the local adjacency database. (Each entry made to TENT must be marked as being either an End system or an Intermediate System to enable the check at the end of Step 2 to be made correctly.) For each adjacency $Adj(N)$, (including Manual Adjacencies, or for Level 2 enabled Reachable Addresses) on enabled circuits, to system N of SELF in state “Up”, compute

   d(N) = cost of the parent circuit of the adjacency (N), obtained from $metric_k$, where k = one of *default metric, delay metric, monetary metric, error metric*.

   Adj(N) = the adjacency number of the adjacency to N

c) If a triple $\langle N,x,\{Adj(M)\}\rangle$ is in TENT, then:

   If $x = d(N)$, then $Adj(M) \leftarrow \{Adj(M)\} \cup Adj(N)$.

d) If there are now more adjacencies in $\{Adj(M)\}$ than maximumPathSplits, then remove excess adjacencies as described in Clause 7.2.7.

e) If $x < d(N)$, do nothing.

f) If $x > d(N)$, remove $\langle N,x,\{Adj(M)\}\rangle$ from TENT and add the triple $\langle N,d(N),Adj(N)\rangle$.

g) If no triple $\langle N, x,\{Adj(M)\}\rangle$ is in TENT, then add $\langle N, d(N),Adj(N)\rangle$ to TENT.

h) Now add any systems to which the local Intermediate system does not have adjacencies, but which are mentioned in neighbouring pseudonode LSPs. The adjacency for such systems is set to that of the Designated Intermediate System.

i) For all broadcast circuits in state “On”, find the LSP with LSP number zero and with the first 7 octets of LSPID equal to the LnCircuitID for that circuit (i.e. pseudonode LSP for that circuit). If it is present, for all the neighbours N reported in all the LSPs of this pseudonode which do not exist in TENT add an entry $\langle N,d(N),Adj(N)\rangle$ to TENT, where

   $d(N) = metric_k$ of the circuit.

   $Adj(N)$ = the adjacency number of the adjacency to the DR.

j) Go to **Step 2**.

Step 1: Examine the zeroth Link State PDU of P, the system just placed on PATHS (i.e. the Link State PDU with the same first 7 octets of LSPID as P, and LSP number zero).

a) If this LSP is present, and the “Infinite Hippity Cost” bit is clear, then for each LSP of P (i.e. all the Link State PDUs with the same first 7 octets of LSPID as P, irrespective of the value of LSP number) compute

$$dist(P,N) = d(P) + metric_k(P,N).$$

   for each neighbour N (both Intermediate System and End system) of the system P. If the “Infinite Hippity Cost” bit is set, only consider the End system neighbours of the system P. $d(P)$ is the second element of the triple

$$\langle P,d(P),\{Adj(P)\rangle$$

   and $metric_k(P,N)$ is the cost of the link from P to N as reported in P’s Link State PDU

b) If $dist(P,N) >$ MaxPathMetric, then do nothing.

c) If $\langle N,d(N),\{Adj(N)\}\rangle$ is in PATHS, then do nothing.

NOTE — $d(N)$ must be less than $dist(P,N)$, or else N would not have been put into PATHS. An additional sanity check may be done here to ensure $d(N)$ is in fact less than $dist(P,N)$.

d) If a triple $\langle N,x,\{Adj(N)\}\rangle$ is in TENT, then:

  1) If $x = dist(P,N)$, then $Adj(\mathrm{N}) \leftarrow \{Adj(\mathrm{N})\} \cup Adj(P)$.

  2) If there are now more adjacencies in $\{Adj(N)\}$ than maximumPathSplits, then remove excess adjacencies, as described in Clause 7.2.7.

  3) If $x < dist(P,N)$, do nothing.

  4) If $x > dist(P,N)$, remove $\langle N,x,\{Adj(N)\}\rangle$ from TENT and add $\langle N,dist(P,N),\{Adj(P)\}\rangle$.

e) If no triple $\langle N, x,\{Adj(N)\}\rangle$ is in TENT, then add $\langle N, dist(P,N),\{P\}\rangle$ to TENT.

Step 2: If TENT is empty, stop, else:

a) Find the element $\langle P,x,\{Adj(P)\}\rangle$, with minimal x as follows:

  1) If an element $\langle *,\mathrm{tentlength},*\rangle$ remains in TENT in the list for tentlength, choose that element. If there are more than one elements in the list for tentlength, choose one of the elements (if any) for a system which is a pseudonode in preference to one for a non-pseudonode. If there are no more elements in the list for tentlength increment tentlength and repeat Step 2.

  2) Remove $\langle P,\mathrm{tentlength},\{Adj(P)\}\rangle$ from TENT.

  3) Add $\langle P,d(P),\{Adj(P)\}\rangle$ to PATHS.

  4) If this is the Level 2 Decision Process running, and the system just added to PATHS listed itself as Partition Designated Level 2 Intermediate system, then additionally add $\langle \mathrm{APEA(P)}, d(P), \{adj(P)\}\rangle$ to PATHS, where AREA.P is the Network Entity Title of the other end of the Virtual Link, obtained by taking the first AREA listed in P's Level 2 LSP and appending P's ID.

  5) If the system just added to PATHS was an End system, go to Step 2, Else go to Step 1.

NOTE - In the Level 2 context, the “End systems” are the set of Reachable Address Prefixes and the set of Area Addresses with zero cost.

C.3 Forwarding Process

C.3.1 Example pseudo-code for the forwarding procedure described in 7.4.3

This procedure chooses, from the Level 1 forwarding database – if level is level1, or from the Level 2 forwarding database – if level is level2, an adjacency on which to forward PDUs for destination dest. A pointer to the adjacency is returned in adj, and the procedure returns the value “True". If no suitable adjacency exists the procedure returns the value “False", in which case a call should be made to Drop(“Destination Address Unreachable", octetNumber).

If queue length values are available to the forwarding process, the minimal queue length of all candidate circuits is chosen, otherwise, they are used in round robin fashion.

```
PROCEDURE Forward(
                    level: (level1, level2),
                    dest: NetworkLayerAddress,
                    VAR adj: POINTER TO adjacency) : BOOLEAN

VAR
    adjArray: ARRAY OF ForwardingDatabaseRecords;
    temp, index, minQueue: CARDINAL;

BEGIN
    (*Set adjArray to appropriate database} *)
    IF level = level1 THEN
        adjArray := level1ForwardingDatabase
    ELSE
        adjArray := level2ForwardingDatabase
    END;
     (*Perform appropriate hashing function to obtain an index into the database *)
     IF Hash(level, dest, index) THEN
        IF adjArray[index].splits > 0 THEN
            (*Find minimum queue size for all equal cost paths *)
            minQueue := MaxUnsigned;
        temp := adjArray[index].lastChosen + 1;        (*start off after last time *)
            FOR i := 1 TO adjArray[index].splits DO
                (*for all equal cost paths to dest *)
                IF temp > adjArray[index].splits THEN
                    (*after end of valid entries, wrap to first *)
                    temp := 1
                ELSE
                    temp := temp + 1
                END;
                IF QueueSize(adjArray[index].nextHop[temp]) < minQueue THEN
                    minQueue := QueueSize(adjArray[index].nextHop[temp]);
                    adj := adjArray[index].nextHop[temp];
                    adjArray[index].lastChosen := temp;
                END;
                Forward := true
            END;
        ELSE
            Forward := false (*There must be at least one valid output adjacency *)
        END
    ELSE
        Forward := false (*Hash returned destination unknown *)
    END
END forward;
```

Annex D
Congestion Control and Avoidance

(This annex is informative)

D.1 Congestion Control

The transmit management subroutine handles congestion control. Transmit management consists of the following components:

> **Square root limiter**. Reduces buffer occupancy time per PDU by using a square root limiter algorithm. The square root limiter also queues PDUs for an output circuit, and prevents buffer deadlock by discarding PDUs when the buffer pool is exhausted. Clause D.1.1 specifies the Square Root Limiter Process.
>
> **Originating PDU limiter.** Limits originating NPDU traffic when necessary to ensure that transit NPDUs are not rejected. An originating NPDU is an NPDU resulting from an NSDU from the Transport at this ES. A transit NPDU is an NPDU from another system to be relayed to another destination ES.
>
> **Flusher.** Flushes PDUs queued for an adjacency that has gone down.

Information for higher layer (Transport) congestion control procedures is provided by the setting of the "congestion experienced" bit in the forwarded data NPDUs as described in <mark:bufferManagement>.

D.1.1 Square Root Limiter

The square root limiter discards a data NPDU by calling the ISO 8473 discard PDU function with the reason "PDU Discarded due to Congestion" when the number of data NPDUs on the circuit output queue exceeds the discard threshold, U_d. U_d is given as follows:

$$U_d = \left\lceil \frac{N_b}{\sqrt{N_c}} \right\rceil$$

where:

N_b = Number of Routeing Layer buffers (maximumBuffers) for all output circuits.

N_c = Number of active output circuits (i.e. Circuits in state "On").

The output queue is a queue of buffers containing data NPDUs which have been output to that circuit by the forwarding process, and which have not yet been transmitted by the circuit. It does **not** include NPDUs which are held by the data link layer for the purpose of retransmission.

Where a data NPDU is to be fragmented by this Intermediate system over this circuit, each fragment shall occupy a separate buffer and shall be counted as such in the queue length. If the addition of all the buffers required for the fragmentation of a single input data NPDU would cause the discard threshold for that queue to be exceeded, it is recommended that all those fragments (including those which could be added without causing the threshold to be exceeded) be discarded.

D.1.2 Originating PDU Limiter

TEMPORARY NOTE - Strictly this function is an End System function. However it is closely coupled to the routeing function, particularly in the case of real systems which are performing the functions of both an Intermediate System and an End System (i.e. systems which can both initiate and terminate data NPDUs and perform relaying functions). Therefore, until a more appropriate location for this information can be determined, this function is described here.

The originating PDU limiter first distinguishes between *originating* NPDUs and *transit* NPDUs. It then imposes a limit on the number of buffers that originating NPDUs can occupy on a per circuit basis. In times of heavy load, originating NPDUs may be rejected while transit NPDUs continue to be routed. This is done because originating NPDUs have a relatively short wait, whereas transit NPDUs, if rejected, have a long wait – a transport retransmission period.

The originating PDU limiter accepts as input:

- An NSDU received from Transport Layer
- A transmit complete signal from the circuit for an ISO 8473 Data PDU.

The originating PDU limiter produces the following as output:

- PDU accepted
- PDU rejected
- Modifications to originating PDU counter

There is a counter, *N*, and an originating PDU limit, originatingQueueLimit, for each active output circuit. Each *N* is initialised to 0. The originatingQueueLimit is set by management to the number of buffers necessary to prevent the circuit from idling.

D.1.3 Flusher

The flusher ensures that no NPDU is queued on a circuit whose state is not ON, or on a non-existent adjacency, or one whose state is not Up.

D.2 Congestion Avoidance

D.2.1 Buffer Management

The Forwarding Process supplies and manages the buffers necessary for relaying. PDUs shall be discarded if buffer thresholds are exceeded. If the average queue length on the input circuit **or** the forwarding processor **or** the output circuit exceeds QueueThreshold, the “congestion experienced” bit shall be set in the QoS maintenance option of the forwarded data PDU (provided the QoS maintenance option is present).

13 Security Considerations

Security issues are not discussed in this memo.

14 Author's Address

David R. Oran
Digital Equipment Corporation
LKG 1-2/A19
550 King Street
Littleton, MA 01460

Email: Oran@Oran.enet.dec.com

Phone: (508) 486-7377