ISO/IEC JTC1/SC6 N 6348

Project:	JTC1.06.41.04
Date:	1990-10-22

ISO/IEC JTC1/SC6
TELECOMMUNICATIONS AND INFORMATION EXCHANGE BETWEEN SYSTEMS

Secretariat: U.S.A. (ANSI)

Title: Text of DIS 10589

Source: Project Editor (D. Oran)

Address reply to:
Secretariat ISO/IEC JTC 1/SC 6 – American National Standards Institute, 1430 Broadway, New York, NY 10018

Tel: 212 642-4931; TX 42 42 96 ANSI UI; FAX 212 302-1286

ISO/IEC JTC 1 Secretariat: USA

Voting begins on Voting Terminates on

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION· ·ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION · ·COMMISSION ELECTROTECHNIQUE INTERNATIONALE

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)

Technologies de l’information — Communication de données et échange d’information entre systèmes — Protocole intra-domain de routage d’un système intermediare à un système intermediare à utiliser conjointement avec le protocole fournissant le service de réseau en mode sans connexion (ISO 8473)

UDC 00000.000 : 000.0000000000

Descriptors:

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENT AND APPROVAL. IT IS THEREFORE SUBJECT TO CHANGE AND MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL ACCEPTED BY ISO AND IEC.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL AS DOCUMENTS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

Contents

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 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 applying 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 7498-3:1989, *Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 3: Naming and Addressing.*

ISO 7498-4:1989, *Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 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*

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

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 9577:1990, *Information technology — Telecommunications and information exchange between systems — Protocol Identification in the Network Layer.*

ISO/IEC DIS 10165-4:, *Information technology — Open systems interconnection — Management Information Services — Structure of Management Information Part 4: Guidelines for the Definition of Managed Objects.*

ISO/IEC 10039:1990, *IPS-T&IEBS — MAC Service Definition.*

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 ARPA*NET, 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.

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) Media access control
c) 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:** A portion of the local routeing information which pertains to the reachability of a single neighbour ES or IS over a single circuit.

Adjacencies are used as input to the Decision Process for forming paths through the routeing domain.

A separate adjacency is created for each neighbour on a circuit, and for each level of routeing (i.e. level 1 and level 2) on a broadcast circuit.

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.

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.

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

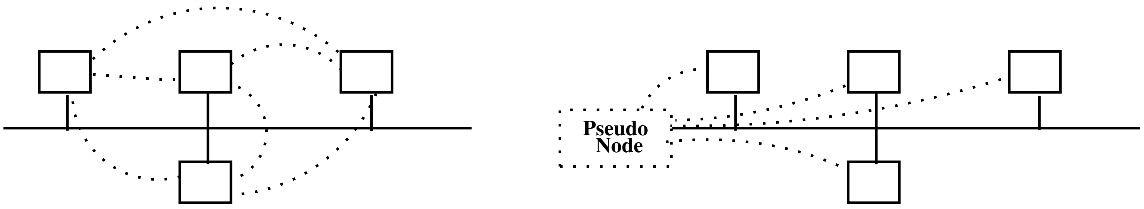


Figure 1 - Use of a pseudonode to collapse a LAN Topology

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
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
NLPID	Network Layer Protocol Identifier
PCI	Protocol Control Information
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) Protocol field names appear in San Serif type with each word capitalised.

d) 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 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 two 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) *dynamically assigned (DA):* The call is established upon receipt of traffic, and brought down on timer expiration when idle. The address to which the call is to be established is determined dynamically from information in the arriving NPDU(s). No IS–IS routeing PDUs are exchanged between ISs on a DA circuit.

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 Addendum 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*[2], and *residual error probability* of ISO 8473 are supported through the optional inclusion of multiple routeing metrics.

- **Authentication.** The protocol is capable of carrying information to be used for the authentication of Intermediate systems in order to increase the security and robustness of a routeing domain. The specific mechanism supported in this International Standard however, only supports a weak form of authentication using passwords, and thus is useful only for protection against accidental misconfiguration errors and does not protect against any serious security threat. In the future, the algorithms may be enhanced to provide stronger forms of authentication than can be provided with passwords without needing to change the PDU encoding or the protocol exchange machinery.

6.6.1 Non-Goals

The following are not within the design scope of the intradomain 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 full logical connectivity to be available, a connected level 2 subdomain is required.

- **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

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) Provision that no PDUs transmitted after a particular startup is complete are delivered out of sequence;

e) Provision that failure to deliver a specific subnetwork SDU will result in the timely disconnection of the subnetwork connection in both directions and that this failure will be reported to both systems; and

f) Reporting of other subnetwork 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*.

[2] “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.

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 intradomain routeing functions such as supplied by this protocol.

The functional decomposition is shown below in figure 2.

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

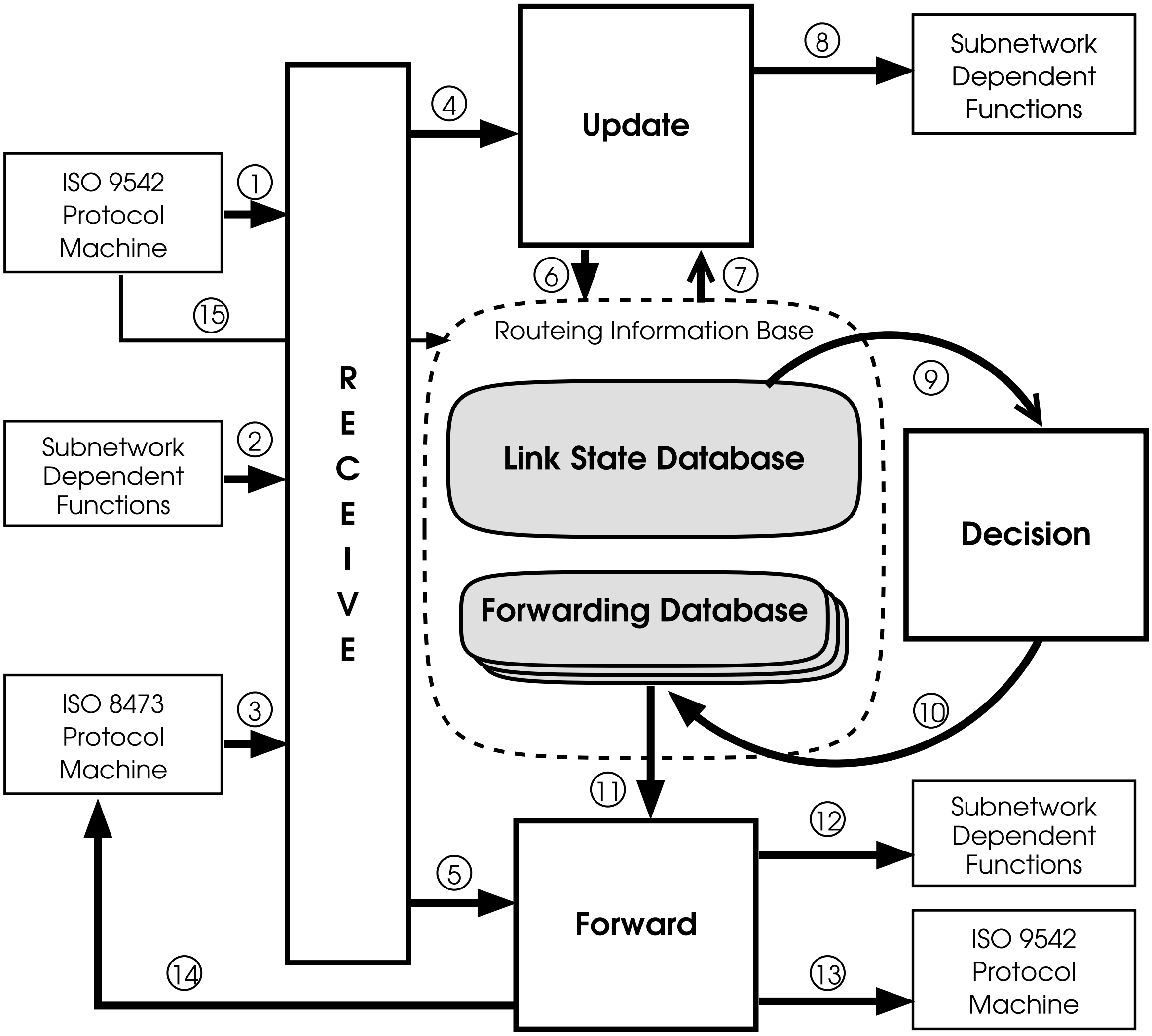

Update
8
Subnetwork Dependent Functions
4
ISO 9542 Protocol Machine
1
15
6
7
Routeing Information Base
9
RECEIVE
Subnetwork Dependent Functions
2
Link State Database
Decision
Forwarding Database
ISO 8473 Protocol Machine
3
10
11
12
Subnetwork Dependent Functions
5
Forward
ISO 9542 Protocol Machine
14
13

Figure 2 - Decomposition of Subnetwork Independent Functions

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 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[3] 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 ; 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).

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.

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

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

The NSAP addresses and NETs of systems are variable length quantities that conform to the requirements of ISO 8348/Add.2. The corresponding NPAI contained in ISO 8473 PDUs and in this protocol's PDUs (such as LSPs and IIHs) must use the preferred binary encoding; the underlying syntax for this information may be either abstract binary syntax or abstract decimal syntax. Any of the AFIs and their corresponding DSP syntax may be used with this protocol.

7.1.1 NPAI Of Systems Within A Routeing Domain

Figure 3 illustrates the structure of an encoded NSAP address or NET.

The structure of the NPAI will be interpreted in the following way by the protocol described in this international standard:

Area Address
address of one area within a routeing domain — a variable length quantity consisting of the entire high-order part of the NPAI, excluding the ID and SEL fields, defined below.

ID System identifier — a variable length field from 1 to 8 octets (inclusive). Each routeing domain employing this protocol shall select a single size for the ID field and all Intermediate systems in the routeing domain shall use this length for the system IDs of all systems in the routeing domain.

The set of ID lengths supported by an implementation is an implementation choice, provided that at least one value in the permitted range can be accepted. The routeing domain administrator must ensure that all ISs included in a routeing domain are able to use the ID length chosen for that domain.

SEL NSAP Selector — a 1-octet field which acts as a selector for the entity which is to receive the PDU(this may be a Transport entity or the Intermediate system Network entity itself). It is the least significant (last) octet of the NPAI.

7.1.2 Deployment of Systems

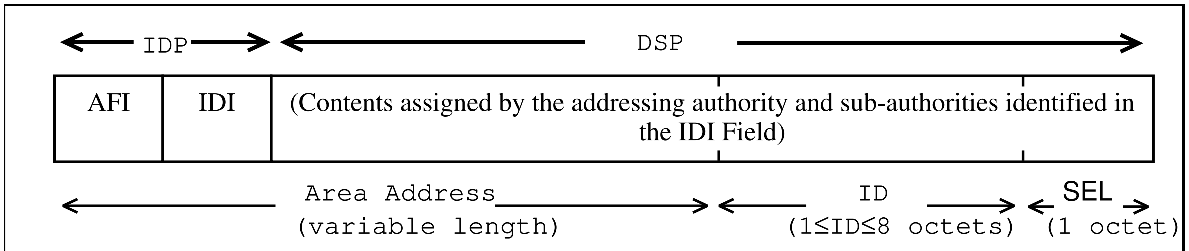
For correct operation of the routeing protocol defined in this international standard, systems deployed in a routeing domain must meet the following requirements:

a) For all systems:

1) Each system in an area must have a unique systemID: that is, no two systems (IS or ES) in an area can use the same ID value.

2) Each area address must be unique within the global OSIE: that is, a given area address can be associated with only one area.

3) All systems having a given value of area address must be located in the same area.


IDP
DSP
AFI
IDI
(Contents assigned by the addressing authority and sub-authorities identified in the IDI Field)
Area Address
(variable length)
ID
(1≤ID≤8 octets)
SEL
(1 octet)

NOTE - The diagram shows how this international standard will parse the NPAI (that is, an address or NET encoded by the preferred binary encoding of ISO 8348/Add.2). This protocol uses the fields "Area Address", "ID", and "SEL".

To emphasize that address information is constructed according to ISO 8348/Add.2, the quantities "IDP", "DSP", "AFI", and "IDI" are also shown; however, they are not explicitly used by this protocol.

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

b) Additional Requirements for Intermediate systems:

   1) Each Level 2 Intermediate system within a routeing domain must have a unique value for its ID field: that is, no two level 2 ISs in a routeing domain can have the same value in their ID fields.

c) Additional Requirements for End systems:

   1) No two End systems in an area may have addresses that match in all but the SEL fields.

d) An End system can be attached to a level 1 IS only if its area address matches one of the entries in the adjacent IS's manualAreaAddresses parameter.

It is the responsibility of the routeing domain's administrative authority to enforce the requirements of 7.1.2. The protocol defined in this international standard assumes that these requirements are met, but has no means to verify compliance with them.

7.1.3 Manual area addresses

The use of several synonymous area addresses by an IS is accommodated through the use of the management parameter manualAreaAddresses. This parameter is set locally for each level 1 IS by system management; it contains a list of all synonymous area addresses associated with the IS, including the IS's area address as contained in its own NET.

Each level 1 IS distributes its manualAreaAddresses in its Level 1 LSP's Area Addresses field, thus allowing level 2 ISs to create a composite list of all area addresses supported within a given area. Level 2 ISs in turn advertise the composite list throughout the level 2 subdomain by including it in their Level 2 LSP's Area Addresses field, thus distributing information on all the area addresses associated with the entire routeing domain. The procedures for establishing an adjacency between two level 1 ISs require that there be at least one area address in common between their two manualAreaAddresses lists, and the procedures for establishing an adjacency between a level 1 Is and an End system require that the End system's area address must match an entry in the IS's manualAreaAddresses list. Therefore, it is the responsibility of System Management to ensure that each area address associated with an IS is included: in particular, system management must ensure that the area addresses of all ESs and Level 1 ISs adjacent to a given level 1 IS are included in that IS's manualAreaAddresses list.

If the area address field for the destination address of an 8473 PDU — or for the next entry in its source routeing field, when present — is not listed in the parameter areaAddresses of a level 1 IS receiving the PDU, then the destination system does not reside in the IS's area. Such PDUs will be routed by level-2 routeing.

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.[1]

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.

7.1.5 Comparison of Addresses

Unless otherwise stated, numerical comparison of addresses shall be performed on the encoded form of the address, by padding the shorter address with trailing zeros to the length of the longer address, and then performing a numerical comparison.

The addresses to which this precedure applies include NSAP addresses, Network Entity Titles, and SNPA addresses.

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

- (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

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

(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[2]. 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.

Each of the four metrics described above may be of two types: an *Internal metric* or an *External metric*. Internal metrics are used to describe links/routes to destinations internal to the routeing domain. External metrics are used to describe links/routes to destinations outside of the routeing domain. These two types of metrics are not directly comparable, except the internal routes are always preferred over external routes. In other words an internal route will always be selected even if an external route with lower total cost exists.

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 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[3].

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 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.

[2] 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.

[3] 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.

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 LSP Database Overload bit.

b) The value of the IS Type field.

c) The Area Addresses option.

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 legal paths to a destination system in the routeing domain. The set may consist of:

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

b) a set of paths of equal minimum metric sum: 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 paths 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 paths, 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 Paths

When there are more than maximumPathSplits legal 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 defined in 6.6.

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

- **metric sum:** Paths having a lesser metric sum are retained in preference to paths having a greater metric sum. By metric sum is understood the sum of the metrics along the path to the destination.

- **neighbour ID:** where two or more paths are associated with adjacencies 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 paths are associated with adjacencies 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

7.2.8.1 Computing Routes through Overloaded Intermediate systems

The Decision Process shall not utilise a link to an Intermediate system neighbour from an IS whose LSPs have the LSP Database Overload indication set. Such paths may introduce loops since the overloaded IS does not have a complete routeing information base. The Decision Process shall, however utilise the link to reach End system neighbours since these paths are guaranteed to be non-looping.

7.2.8.2 Two-way connectivity check

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.

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 be-

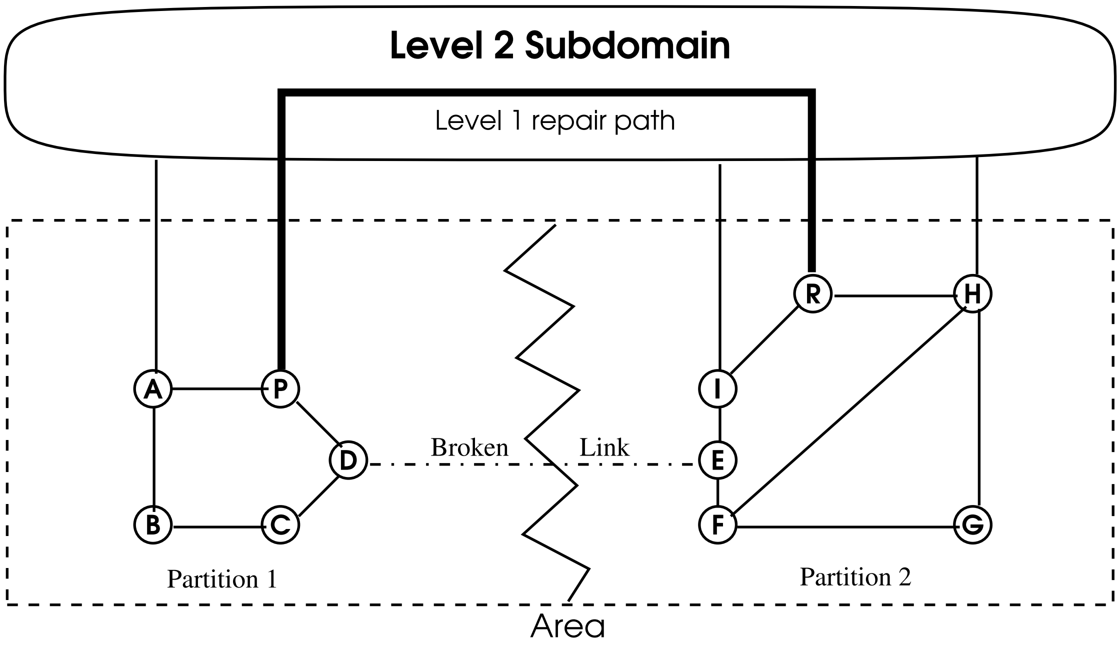


Figure 4 - Repair of partitioned level 1 area

comes 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 sub-clauses 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 Neighbours field of its Level 1 Link State PDU. The Virtual Flag shall be set “True” for these Intermediate System neighbours. 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 areaAddresses 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 notification.

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 partition 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 Neighbours 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 "attached" by the default metric;

- P indicates support for the partition 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 7.1.5). If one of the local system's manualAreaAddresses is so rejected the notification manualAddressDroppedFromArea 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 SEL field in both NSAPs shall contain the "IS-IS routeing" selector value. The QoS Maintenance field of the outer PDU shall be set to indicate forwarding via the default routeing metric (see table 1 on page 32).

For Data and Error Report NPDUs the Segmentation Permitted and Error Report flags and the Lifetime field of the outer NPDU shall be copied from the inner NPDU. When the inner NPDU is decapsulated, its Lifetime field shall be set to the value of the Lifetime field 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 field 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 field 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 7.4.3.2. The decapsulation is described as part of the Receive process in 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 7.1.5). If one of the local system's manualareaAddresses is rejected the notification manualAddressDroppedFromArea shall be generated.

7.2.12 Order of Preference of Routes

If an Intermediate system takes part in level 1 routeing, and determines (by looking at the area address) that a given destination is reachable within its area, then that destination will be reached exclusively by use of level 1 routeing. In particular:

a) Level 1 routeing is always based on internal metrics.

b) Amongst routes in the area, routes on which the requested QoS (if any) is supported are always preferred to routes on which the requested QoS is not supported.

c) Amongst routes in the area of the same QoS, the shortest routes are preferred. For determination of the shortest path, if a route with specific QoS support is available, then the specified QoS metric is used, otherwise the default metric is used.

d) Amongst routes of equal cost, load splitting may be performed.

If an Intermediate system takes part in level 1 routeing, does not take part in level 2 routeing, and determines (by looking at the area address) that a given destination is *not* reachable within its area, and at least one attached level 2 IS is reachable in the area, then that destination will be reached by routeing to a level 2 Intermediate system as follows:

a) Level 1 routeing is always based on internal metrics.

b) Amongst routes in the area to attached level 2 ISs, routes on which the requested QoS (if any) is supported are always preferred to routes on which the requested QoS is not supported.

c) Amongst routes in the area of the same QoS to attached level 2 ISs, the shortest route is preferred. For determination of the shortest path, if a route on which the specified QoS is available, then the specified QoS metric is used, otherwise the default metric is used.

d) Amongst routes of equal cost, load splitting may be performed.

If an Intermediate system takes part in level 2 routeing and is attached, and the IS determines (by looking at the area address) that a given destination is not reachable within its area, then that destination will be reached as follows:

a) Routes on which the requested QoS (if any) is supported are always preferred to routes on which the requested QoS is not supported.

b) Amongst routes of the same QoS, routes are prioritised as follows:

   1) Highest precedence: routes matching the area address of any area in the routeing domain

   2) Medium precedence: Routes matching a reachable address prefix with an internal metric. For destinations matching multiple reachable address prefix entries all with internal metrics, the longest prefix shall be preferred.

   3) Lowest precedence: Routes matching a reachable address prefix with an external metric. For destinations matching multiple reachable address prefix entries all with external metrics, the longest prefix shall be preferred.

c) For routes with equal precedence as specified above, the shortest path shall be preferred. For determination of the shortest path, a route supporting the specified QoS is used if available; otherwise a route using the default metric shall be used. Amongst routes of equal cost, load splitting may be performed.

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 System 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", neighbourSystemType "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.

Metrics assigned to Manual adjacencies must be Internal metrics.

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.

Metrics assigned to Reachable Address managed objects may be either Internal or External.

A reachable address is considered to be "active" when all the following conditions are true:

a) The parent circuit is in state "On";

b) the Reachable Address is in state "On"; and

c) the parent circuit is of type broadcast or is in data link state "Running".

Whenever a reachable address changes from being "inactive" to "active" a signal shall be generated to the Update process to cause it to include the Address Prefix of the reachable address in the Level 2 LSPs generated by that system as described in 7.3.9.

Whenever a reachable address changes from being "active" to "inactive", a signal shall be generated to the Update process to cause it to cease including the Address Prefix of the reachable address in the Level 2 LSPs.

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.

NOTE - In order to minimise the transmission of redundant information, it is advisable for an IS to group Reachable Address Prefix information by the circuit with which it is associated. Doing so will ensure that the minimum number of LSP fragments need be transmitted if a circuit to another routeing domain changes state.

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 LSP Database Overload bit.

   2) The value of the IS 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 7.3.15) for the two new LSPs shall be set as an atomic action.[4]

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.

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.

[4] 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.

7.3.6 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 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 neighbourSystemType "L1 Intermediate System"

    x neighbourSystemType "L2 Intermediate System" and adjacencyUsage "Level 2" or "Level1 and 2".

    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 metrics shall be set to the value indicating *unsupported*.
- In the End System Neighbours option — the set of 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 neighbourSystemType "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.
- In the Authentication Information field — if the system's areaTransmitPassword is non-null, include the Authentication Information field containing an Authentication Type of "Password", and the value of the areaTransmitPassword.

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 issuing the pseudonode LSP will already have been made available via its own non-pseudonode LSP.
- In the Intermediate System Neighbours option — the set of 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
    - x neighbourSystemType "L1 Intermediate System"
    - x "L2 Intermediate System" and adjacencyUsage "Level 1".
- In the End System Neighbours option — the set of 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 neighbourSystemType "End system".
- In the Authentication Information field — if the system's areaTransmitPassword is non-null, include the Authentication Information field containing an Authentication Type of "Password", and the value of the areaTransmitPassword.

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 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 neighbourSystemType "L2 Intermediate System".
  - The set of l2CircuitIDs for all circuits of type "Broadcast". (i.e. the neighbouring pseudonode IDs)

    The metric and metric type 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.
- In the Authentication Information field — if the system's domainTransmitPassword is non-null, include the Authentication Information field containing an Authentication Type of "Password", and the value of the domainTransmitPassword.

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 issuing the pseudonode LSP will already have been made available via its own non-pseudonode LSP.
- In the Intermediate System Neighbours option — the set of 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 neighbourSystemType "L2 Intermediate System".
- The Prefix Neighbours option is not present.
- In the Authentication Information field — if the system's domainTransmitPassword is non-null, include the Authentication Information field containing an Authentication Type of "Password", and the value of the domainTransmitPassword.

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 recomputation.

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 ID Length 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 non-pseudonode zeroth LSP. In the event that it is not capable of storing and internalising one of its own LSPs it shall enter the overloaded state as described in 7.3.19.1.

NOTE - It is recommended that an Intermediate system ensure (by reserving resources, or otherwise) that it will always be able to store and internalise all its own (zero and non-zero, pseudonode and non-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 corruptedLSPReceived notification 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 AllL1IS.

d) When propagating a Level 2 Link State PDU on a broadcast subnetwork, the IS shall transmit to the multi-destination subnetwork address AllL2IS.

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 notification, corruptedLSPReceived,

   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 SRMflag 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, and

   2) clear the SRMflag 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 SRMflags 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 SRMflags 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 (*SRMflag) – if set, indicates that Link State PDU should be transmitted on that circuit. On broadcast circuits SRMflag is cleared as soon as the LSP has been transmitted, but on non-broadcast circuits SRMflag is only cleared on reception of a Link State PDU or Sequence Numbers PDU as described below.

SRMflag shall never be set for an LSP with sequence number zero, nor on a circuit whose externalDomain attribute is "True" (See 7.3.15.2).

Send Sequence Numbers (SSNflag) – 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 SSNflag is cleared. Note that the Partial Sequence Numbers PDU serves as an acknowledgement that a Link State PDU was received.

SSNflag shall never be set on a circuit whose externalDomain attribute is "True".

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) Perform the following PDU acceptance tests:

   1) If the LSP was received over a circuit whose externalDomain attribute is "True", the IS shall discard the PDU.

   2) If the ID Length field of the PDU is not equal to the value of the IS's routeingDomainIDLength, the PDU shall be discarded and an iDFieldLengthMismatch notification generated.

   3) If this is a level 1 LSP, and the set of areaReceivePasswords is non-null, then perform the following tests:

      i) If the PDU does not contain the Authentication Information field then the PDU shall be discarded and an authenticationFailure notification generated.

      ii) If the PDU contains the Authentication Information field, but the Authentication Type is not equal to "Password", then the PDU shall be accepted unless the IS implements the authenticatiion procedure indicated by the Authentication Type. In this case whether the IS accepts or ignores the PDU is outside the scope of this International Standard.

      iii) Otherwise, the IS shall compare the password in the received PDU with the passwords in the set of areaReceivePasswords, augmented by the value of the areaTransmitPassword. If the value in the PDU matches any of these passwords, the IS shall accept the PDU for further processing. If the value in the PDU does not match any of the above values, then the IS shall ignore the PDU and generate an authenticationFailure notification.

   4) If this is a level 2 LSP, and the set of domainReceivePasswords is non-null, then perform the following tests:

      i) If the PDU does not contain the Authentication Information field then the PDU shall be discarded and an authenticationFailure notification generated.

      ii) If the PDU contains the Authentication Information field, but the Authentication Type is not equal to "Password", then the PDU shall be accepted unless the IS implements the authenticatiion procedure indicated by the Authentication Type. In this case whether the IS accepts or ignores the PDU is outside the scope of this International Standard.

      iii) Otherwise, the IS shall compare the password in the received PDU with the passwords in the set of domainReceivePasswords, augmented by the value of the domainTransmitPassword. If the value in the PDU matches any of these passwords, the IS shall accept the PDU for further processing. If the value in the PDU does not match any of the above values, then the IS shall ignore the PDU and generate an authenticationFailure notification.

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

c) If the source *S* of the LSP is an IS or pseudonode for which all but the last octet 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.

d) If the source *S* of the LSP is a system (pseudonode or otherwise) for which the first ID Length 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.

e) 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).

When attempting to store a new version of an existing LSP (with the same LSPID), which has a length less than or equal to that of the existing LSP, the existing LSP shall be removed from the routeing information base and the new LSP stored as a single atomic action. This ensures that such an LSP (which may be carrying the LSP Database Overload indication from an overloaded IS) will never be ignored as a result of a lack of memory resources.

7.3.15.2 Action on Receipt of a Sequence Numbers PDU

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

a) Perform the following PDU acceptance tests:

   1) If the SNP was received over a circuit whose externalDomain attribute is “True”, the IS shall discard the PDU.

   2) If the ID Length field of the PDU is not equal to the value of the IS’s routeingDomainIDLength, the PDU shall be discarded and an iDFieldLengthMismatch notification generated.

   3) If this is a level 1 SNP and the set of areaReceivePasswords is non-null, then perform the following tests:

      i) If the PDU does not contain the Authentication Information field then the PDU shall be discarded and an authenticationFailure notification generated.

      ii) If the PDU contains the Authentication Information field, but the Authentication Type is not equal to “Password”, then the PDU shall be accepted unless the IS implements the authenticatiion procedure indicated by the Authentication Type. In this case whether the IS accepts or ignores the PDU is outside the scope of this International Standard.

      iii) Otherwise, the IS shall compare the password in the received PDU with the passwords in the set of areaReceivePasswords, augmented by the value of the areaTransmitPassword. If the value in the PDU matches any of these passwords, the IS shall accept the PDU for further processing. If the value in the PDU does not match any of the above values, then the IS shall ignore the PDU and generate an authenticationFailure notification.

   4) If this is a level 2 SNP, and the set of domainReceivePasswords is non-null, then perform the following tests:

      i) If the PDU does not contain the Authentication Information field then the PDU shall be discarded and an authenticationFailure notification generated.

      ii) If the PDU contains the Authentication Information field, but the Authentication Type is not equal to “Password”, then the PDU shall be accepted unless the IS implements the authenticatiion procedure indicated by the Authentication Type. In this case whether the IS accepts or ignores the PDU is outside the scope of this International Standard.

iii) Otherwise, the IS shall compare the password in the received PDU with the passwords in the set of domainReceivePasswords, augmented by the value of the domainTransmitPassword. If the value in the PDU matches any of these passwords, the IS shall accept the PDU for further processing. If the value in the PDU does not match any of the above values, then the IS shall ignore the PDU and generate an authenticationFailure notification.

b) 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 fields 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.

c) 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 LSPID and End LSPID 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.

If the value of the IS's areaTransmitPassword is non-null, then the IS shall include the Authentication Information field in the transmitted CSNP, indicating an Authentication Type of "Password" and containing the areaTransmitPassword as the authentication value.

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.

If the value of the IS's domainTransmitPassword is non-null, then the IS shall include the Authentication Information field in the transmitted CSNP, indicating an Authentication Type of "Password" and containing the domainTransmitPassword as the authentication value.

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.

If the value of the IS's areaTransmitPassword is non-null, then the IS shall include the Authentication Information field in the transmitted PSNP, indicating an Authentication Type of "Password" and containing the areaTransmitPassword as the authentication value.

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 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.

   If the value of the IS's domainTransmitPassword is non-null, then the IS shall include the Authentication Information field in the transmitted PSNP, indicating an Authentication Type of "Password" and containing the domainTransmitPassword as the authentication value.

b) Otherwise (*C* is a point to point circuit, including non-DA DED circuits and virtual links)

   1) If this system is a Level 1 Intermediate system, 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 Partial Sequence Numbers PDU.

      If the value of the IS's areaTransmitPassword is non-null, then the IS shall include the Authentication Information field in the transmitted PSNP, indicating an Authentication Type of "Password" and containing the areaTransmitPassword as the authentication value.

   2) If this system is a Level 2 Intermediate system, transmit a Level 2 Partial Sequence Numbers PDU on circuit *C*, containing entries for 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 Partial Sequence Numbers PDU.

      If the value of the IS's domainTransmitPassword is non-null, then the IS shall include the Authentication Information field in the transmitted PSNP, indicating an Authentication Type of "Password" and containing the domainTransmitPassword as the authentication value.

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 attribute 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 - 1). When a system initialises, it shall start with sequence number 1 for its own Link State PDUs.[5]

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 – 1, the notification attemptToExceedMaximumSequenceNumber 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.

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.

[5] 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.

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 unexpired 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 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.

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:

a) source's ID

b) PDU type (Level 1 or 2)

c) sequence number

d) Remaining Lifetime

e) 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.

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 re-

ceived on the multi-destination addresses, and hence by a LAN adapter dedicated to the multi-destination address.

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

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

b) 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 notification corruptedLSPDetected 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 LSP Database Overload 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 lSPL1DatabaseOverload notification, and on entering Level 2 “Waiting State” the IS shall generate the lSPL2DatabaseOverload notification.

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 lSPL1DatabaseOverload (recovered)” notification.

   2) Clear the LSP Database Overload 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 lSPL2DatabaseOverload (recovered) notification.

   2) Clear the LSP Database Overload 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 variable 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 both for transmitting NPDUs originated by this system, and for forwarding NPDUs originated by other systems

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 SEL.

NOTE - The SEL 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 in the PDU to be forwarded matches any one of the area addresses of this IS, then consult the level 1 forwarding database to determine the adjacency which is the next hop on the path to the NPDU's destination. Forward the NPDU on this adjacency.

2) Otherwise, consult the level 1 forwarding database to determine the adjacency which is the next hop on the path to the nearest level 2 is in the area, and forward the NPDU on this adjacency.

b) If the local system type is Level 2, and AttachedFlag$_k$ = "True" then:

1) If the Area Address in the PDU to be forwarded matches any one of the area addresses of this IS, then consult the level 1 forwarding database to determine the adjacency which is the next hop on the path to the NPDU's destination. Forward the NPDU on this adjacency.

2) Otherwise, consult the level 2 forwarding database to determine the adjacency which is the next hop on the path to the destination area, and forward the NPDU on this adjacency.

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 field 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 the Forward procedure, described below.

7.4.3.3 The Procedure Forward

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".[6]

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. Generating Redirect PDUs

[6] 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.

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 NLPID of Intra-Domain routeing,
- 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 SEL = "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 an 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 setable 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	01-80-C2-00-00-14	The multi-destination address “All Level 1 Intermediate Systems”
AllL2ISs	01-80-C2-00-00-15	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	10000011	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.
MaximumArea-Addresses	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 (e.g. 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 managed object. The mappingType attribute 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 both point to point links and Static 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 adjacencyStateChange (Up)” notification 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 adjacencyStateChange (Down) notification, and the adjacency can be reused 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 neighbourSystemType “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 adjacencyStateChange (Down) notification;

   2) delete the adjacency; and

3) create a new adjacency with:

   i) state set to “Initialising”, and

   ii) neighbourSystemType set to “Unknown”.

4) perform the following actions..

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

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

d) The neighbourSystemType 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

An IS shall send Point-to-Point IIH PDUs on those Point-to-Point circuits whose externalDomain attribute is set “False”. The IIH shall be constructed and transmitted 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[7] where *maxsize* is the maximum of

   1) dataLinkBlocksize
   2) originatingL1LSPBufferSize
   3) 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.

d) If the value of the circuitTransmitPassword for the circuit is non-null, then the IS shall include the Authentication Information field in the transmitted IIH PDU, indicating an Authentication Type of “Password” and containing the circuitTransmitPassword as the authentication value.

8.2.4 Receiving Point to Point IIH PDUs

8.2.4.1 PDU Acceptance Tests

On receipt of a Point-to-Point IIH PDU, perform the following PDU acceptance tests:

a) If the IIH PDU was received over a circuit whose externalDomain attribute is set “True”, the IS shall discard the PDU.

b) If the ID Length field of the PDU is not equal to the value of the IS’s routeingDomainIDLength, the PDU shall be discarded and an iDFieldLengthMismatch notification generated.

c) If the set of circuitReceivePasswords for this circuit is non-null, then perform the following tests:

   1) If the PDU does not contain the Authentication Information field then the PDU shall be discarded and an authenticationFailure notification generated.

   2) If the PDU contains the Authentication Information field, but the Authentication Type is not equal to “Password”, then the PDU shall be accepted unless the IS implements the authenticatiion procedure indicated by the Authentication

Table 3 - Setting the value of the Circuit Type field

iSType	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)

[7] 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).

Type. In this case whether the IS accepts or ignores the PDU is outside the scope of this International Standard.

3) Otherwise, the IS shall compare the password in the received PDU with the passwords in the set of circuitReceivePasswords for the circuit on which the PDU was received. If the value in the PDU matches any of these passwords, the IS shall accept the PDU for further processing. If the value in the PDU does not match any of the circuitReceivePasswords, then the IS shall ignore the PDU and generate an authenticationFailure notification.

8.2.4.2 IIH PDU Processing

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 attribute.

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

1) If the local system is of iSType “L1Intermediate-System” the IS shall perform the action indicated by Table 4.

2) If the local system is of iSType “L2Intermediate-System” 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 iSType “L2Intermediate-System” 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 iSType “L1Intermediate-System” and the adjacency is not in state “Up”, the IS shall delete the adjacency (if any) and generate an initialisationFailure (Area Mismatch)” notification.

2) If the local system is of iSType “L1Intermediate-System” and the adjacency is in state “Up”, the IS shall delete the adjacency and generate an adjacencyStateChange (Down – Area Mismatch)” notification .

3) If the local system is of iSType “L2Intermediate-System” 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.

Table 4 - Level 1 State table for matching areas

Circuit Type[1]	Adjacency Usage	
	none[2]	Level 1[3]
Level 1 only	Up[4] L1[5]	Accept
Level 2 only	Reject[7] (Wrong system)	Down[6] (Wrong system)
Level 1 & 2	Up[4] L1[5]	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 adjacencyUsage is “Level 1”.

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

[5] The adjacencyUsage status is set to “Level 1”.

[6] An adjacencyStateChange (Down)” notification is generated, with the specified reason, and the adjacency deleted.

[7] An initialisationFailure notification is generated with the specified reason.

Table 5 - Level 2 State table for matching areas

Circuit Type[1]	Adjacency Usage			
	none[2]	Level 1[3]	Level 1&2[4]	Level 2[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 adjacencyUsage is “Level 1”.

[4] The adjacency is in state “Up” and the adjacencyUsage is "Level 1&2".

[5] The adjacency is in state “Up” and the adjacencyUsage is "Level 2".

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

[7] The adjacencyUsage status is set to “Level 1”.

[8] An adjacencyStateChange (Down)” notification is generated, with the specified reason, and the adjacency deleted.

[9] The adjacencyUsageis set to “Level 2”.

[10] The adjacencyUsage is set to “Level 1&2”.

Table 6 - Level 2 Only State table for matching areas

Circuit Type[1]	Adjacency Usage		
	none[2]	Level 1&2[3]	Level 2[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 adjacencyUsage is "Level 1&2".

[4] The adjacency is in state “Up” and the adjacencyUsage is "Level 2".

[5] An initialisationFailure notification is generated with the specified reason.

[6] An adjacencyStateChange (Down) notification is generated, with the specified reason, and the adjacency deleted.

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

[8] The adjacencyUsage is set to “Level 2”.

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

Circuit Type[1]	Adjacency Usage			
	none[2]	Level 1[3]	Level 1&2[4]	Level 2[5]
Level 1 only	Reject[6] (Area Mismatch)	Down[7] (Area Mismatch)	Down[7] (Wrong system)	Down[7] (Wrong system)
Level 2 only	Up[8] L2O[9]	Down[7] (Wrong system)	Down[7] (Wrong system)	Accept
Level 1 & 2	Up[8] L2O[9]	Down[7] (Wrong system)	Down[7] (Area Mismatch)	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 adjacencyUsage is “Level 1”.

[4] The adjacency is in state “Up” and the adjacencyUsage is "Level 1&2".

[5] The adjacency is in state “Up” and the adjacencyUsage is "Level 2"

[6] An initialisationFailure ” notification is generated with the specified reason.

[7] An adjacencyStateChange (Down)” notification is generated, with the specified reason, and the adjacency deleted.

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

[9] The adjacencyUsage is set to “Level 2”.

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 adjacencyStateChange(Down) notification, 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.

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 holdingTimer shall be set to the Holding Time as reported in the Holding Timer field of the Pt-Pt IIH PDU. If a neighbour is not heard from in that time, the IS shall

a) purge it from the database; and

b) generate an adjacencyStateChange (Down) notification.

8.3 ISO 8208 Subnetworks

8.3.1 Network Layer Protocols

The way in which the underlying service assumed by ISO 8473 is provided for ISO 8208 subnetworks is described in clause 8 of ISO 8473. This defines a set of Subnetwork Dependent Convergence Functions (SNDCFs) that relate the service provided by specific individual ISO-standard subnetworks to the abstract “underlying service” defined in clause 5.5 of ISO 8473. In particular 8.4.3 describes the Subnetwork Dependent Convergence Functions used with ISO 8208 Subnetworks.

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 Userdata.

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 DA 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 STATIC circuit. If this is not the case, the DA circuit may be used to establish a call to the remote SNPA supporting the STATIC circuit, which would then (wrongly) assume it was the 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 attribute (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).

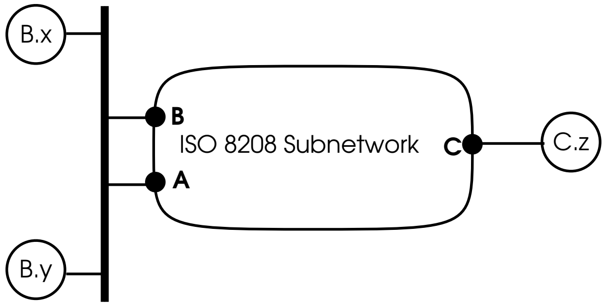


Figure 5 - Example for reverse path forwarding

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, if implemented, 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 DA 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 DA 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. Also, each dynamically assigned circuit has a specified call establishment cost measured by callEstablishmentMetric$_k$ (where k indexes the four defined metrics). The call establishment cost is always an internal metric, and is therefore directly comparable with the reachable address metric only if the reachable address metric is also internal.

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 object, together with the Reachable address metric$_k$ + Delta$_k$ increment. If reachable address metric$_k$ is internal, then Delta$_k$ = callEstablishmentMetric$_k$. If reachable address metric$_k$ is external, then Delta$_k$ = 0.

This causes this information to be included in subsequently generated LSPs as described in 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*[8] 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 metric$_k$ + Delta$_k$ increment (where Delta$_k$ is the same as defined above). 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 adjacencyStateChange" notification 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 Delta$_k$. If (based on the new value of callEstablishmentMetric$_k$Increment) 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 Delta$_k$ (based on the new value of callEstlishmentMetric$_k$Increment).

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 for 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$_k$ + Delta$_k$ (based on the new value of callEstablishmentMetric$_k$Increment). In addition, for all Adjacencies of the Circuit

[8] 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

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.

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 8.3.5.4 above.

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 neighbourSystemType "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[9] where *maxsize* for Level 1 IIH PDUs is the maximum of

- dataLinkBlocksize
- originatingL1LSPBufferSize

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 mis-configuration 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 neighbourSystemType "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 neighbourSystemType "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 IIH PDU Acceptance Tests

On receipt of a Broadcast IIH PDU, perform the following PDU acceptance tests:

a) If the IIH PDU was received over a circuit whose externalDomain attribute is "True", the IS shall discard the PDU.

b) If the ID Length field of the PDU is not equal to the value of the IS's routeingDomainIDLength, the PDU shall be discarded and an iDFieldLengthMismatch notification generated.

c) If the set of circuitReceivePasswords for this circuit is non-null, then perform the following tests:

   1) If the PDU does not contain the Authentication Information field then the PDU shall be discarded

[9] 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 an authenticationFailure notification generated.

2) If the PDU contains the Authentication Information field, but the Authentication Type is not equal to “Password”, then the PDU shall be accepted unless the IS implements the authenticatiion procedure indicated by the Authentication Type. In this case whether the IS accepts or ignores the PDU is outside the scope of this International Standard.

3) Otherwise, the IS shall compare the password in the received PDU with the passwords in the set of circuitReceivePasswords for the circuit on which the PDU was received. If the value in the PDU matches any of these passwords, the IS shall accept the PDU for further processing. If the value in the PDU does not match any of the circuitReceivePasswords, then the IS shall ignore the PDU and generate an authenticationFailure notification.

8.4.1.2 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 initialisationFailure (area mismatch) notification. Otherwise (a match is found) the IS shall accept the adjacency and set the Adjacency neighbourSystemType to “L1 Intermediate System”.

8.4.1.3 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 neighbourSystemType to “L2 Intermediate System”.

8.4.1.4 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 neighbourSystemType equal to “L*n* Intermediate System”,

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

8.4.1.5 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) neighbourSystemType 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 neighbourSystemType status to “L*n* Intermediate System” (where *n* is the level of the IIH PDU),

b) set the holding timer, LAN Priority, neighbourID 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 adjacencyStateChange (Up)” notification.

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 adjacencyStateChange (Down) notification.

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 adjacencyStateChange (Down) notification.

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 neighbourSystemType "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 rejectedAdjacency notification generated.

If an old adjacency must be purged, the IS shall generate an adjacencyStateChange (Down) notification 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

A Level 1 IS shall transmit a Level 1 LAN IIH PDU immediately when any circuit whose externalDomain attribute is "False" has been enabled. A Level 2 Intermediate System shall transmit a Level 2 LAN IIH PDU. A Level 2 Intermediate System shall also transmit a Level 1 LAN IIH PDU unless the circuit is marked as manualL2OnlyMode "True".

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[10] 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[11] 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.

If the value of the circuitTransmitPassword for the circuit is non-null, then the IS shall include the Authentication Information field in the transmitted IIH PDU, indicating an Authentication Type of "Password" and containing the circuitTransmitPassword as the authentication value.

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 7.1.5 for how to compare 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.

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 lanLevel1DesignatedIntermediateSystemChange notification. 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.

[10] Jitter is applied as described in 10.1.

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

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 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 lanLevel2DesignatedIntermediateSystemChange notification. 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 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 field 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 system 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 iSISHelloTimer 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 adjacencyStateChange (Up) notification on that adjacency.

The IS shall set a timer for the value of the Holding Time field 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 adjacencyStateChange (Down) notification, 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.

9.1 General encoding Rules

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 (or simply R) 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.

9.2 Encoding of Network Layer Addresses

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

9.3 Encoding of SNPA Addresses

SNPA addresses (e.g. lANAddress) shall be encoded according to the rules specified for the particular type of subnetwork being used. In the case of an ISO 8802 subnetwork, the SNPA address is the MAC address defined in ISO 10039, which is encoded according to the “binary representation of MAC addresses” specified in ISO 10039.

9.4 PDU Types

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

These are described in the following subclauses.

9.5 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 Extension	1
ID Length	1
R R R PDU Type	1
Version	1
ECO	1
User ECO	1
Reserved/Circuit Type	1
Source ID	ID Length
Holding Time	2
PDU Length	2
R Priority	1
LAN ID	ID Length + 1
VARIABLE LENGTH FIELDS	VARIABLE

- Intradomain Routeing Protocol Discriminator — architectural constant
- Length Indicator – Length of the fixed header in octets
- Version/Protocol ID Extension – 1
- ID Length — Length of the ID field of NSAP addresses and NETs used in this routeing domain. This field shall take on one of the following values:
  - An integer between 1 and 8, inclusive, indicating an ID field of the corresponding length
  - The value zero, which indicates a 6 octet ID field length
  - The value 255, whhich means a null ID field (i.e. zero length)

  All other values are illegal and shall not be used.
- PDU 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 – the system ID of transmitting Intermediate system
- Holding Time – Holding Timer to be used for this Intermediate system
- PDU Length – Entire length of this PDU, in octets, including header
- Reserved/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 – a field composed the system ID (1–8 octets) of the LAN Level 1 Designated Intermediate System, plus a 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.

  x CODE – 1

  x LENGTH – total length of the value field.

  x VALUE –

	No. of Octets
Address Length	1
Area Address	Address Length
Address Length	1
Area Address	Address Length

  · Address Length – Length of Area Address in octets.

  · Area Address – Area address.

- Intermediate System Neighbours – This option field can occur multiple times. The set of Intermediate systems on this LAN to which adjacencies of neighbourSystemType "L1 Intermediate System" exist in state "Up" or "Initialising" (i.e. those from which Level 1 IIH PDUs have been heard).

  x CODE – 6

  x LENGTH – total length of the value field.

  x VALUE –

	No. of Octets
LAN Address	6
LAN Address	6

  · 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.

  x CODE – 8.

  x LENGTH – total length of the value field (may be zero).

  x VALUE – LENGTH octets of arbitrary value.

- Authentication Information — information for performing authentication of the originator of the PDU.

  x CODE — 10.

  x LENGTH — variable from 1–254 octets

  x VALUE —

	No. of Octets
Authentication Type	1
Authentication Value	VARIABLE

  · Authentication Type — a one octet identifier for the type of authentication to be carried out. The following values are defined:

    0 — RESERVED

    1 — Cleartext Password

    2–254 — RESERVED

    255 — Routeing Domain private authentication method

  · Authentication Value — determined by the value of the authentication type. If Cleartext Password as defined in this International Standard is used, then the authentication value is an octet string.

9.6 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 8.4.1).

	No. of Octets
Intradomain Routeing Protocol Discriminator	1
Length Indicator	1
Version/Protocol ID Extension	1
ID Length	1
R R R PDU Type	1
Version	1
ECO	1
User ECO	1
Reserved/Circuit Type	1
Source ID	ID Length
Holding Time	2
PDU Length	2
R Priority	1
LAN ID	ID Length + 1
VARIABLE LENGTH FIELDS	VARIABLE

- Intradomain Routeing Protocol Discriminator – architectural constant
- Length Indicator – Length of fixed header in octets
- Version/Protocol ID Extension – 1
- ID Length — Length of the ID field of NSAP addresses and NETs used in this routeing domain. This field shall take on one of the following values:
  - An integer between 1 and 8, inclusive, indicating an ID field of the corresponding length
  - The value zero, which indicates a 6 octet ID field length
  - The value 255, whhich means a null ID field (i.e. zero length)

  All other values are illegal and shall not be used.
- PDU 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 – the system ID of transmitting Intermediate System
- Holding Time – Holding Timer to be used for this Intermediate System
- PDU Length – Entire length of this PDU, in octets, including header
- Reserved/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 – a field composed the system ID (1–8 octets) of the LAN Level 1 Designated Intermediate System, plus a 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.

  x CODE – 1

  x LENGTH – total length of the value field.

  x VALUE –

	No. of Octets
Address Length	1
Area Address	Address Length
Address Length	1
Area Address	Address Length

  · Address Length – Length of area address in octets.

  · Area Address – Area address.

- Intermediate System Neighbours – This option can occur multiple times. The set of Intermediate systems on this LAN to which adjacencies of neighbourSystemType “L2 Intermediate System” exist in state “Up” or “Initialising” (i.e. those from which Level 2 IIH PDUs have been heard).

  x CODE – 6

  x LENGTH – total length of the value field.

  x VALUE –

	No. of Octets
LAN Address	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.

  x CODE – 8.

  x LENGTH – total length of the value field (may be zero).

  x VALUE – LENGTH octets of arbitrary value.

- Authentication Information — information for performing authentication of the originator of the PDU.

  x CODE — 10.

  x LENGTH — variable from 1–254 octets

  x VALUE —

	No. of Octets
Authentication Type	1
Authentication Value	VARIABLE

  · Authentication Type — a one octet identifier for the type of authentication to be carried out. The following values are defined:

    0 — RESERVED

    1 — Cleartext Password

    2–254 — RESERVED

    255 — Routeing Domain private authentication method

  · Authentication Value — determined by the value of the authentication type. If Cleartext Password as defined in this International Standard is used, then the authentication value is an octet string.

9.7 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 8.2.3).

	No. of Octets
Intradomain Routeing Protocol Discriminator	1
Length Indicator	1
Version/Protocol ID Extension	1
ID Length	1
R R R PDU Type	1
Version	1
ECO	1
User ECO	1
Reserved/Circuit Type	1
Source ID	ID Length
Holding Time	2
PDU Length	2
Local Circuit ID	1
VARIABLE LENGTH FIELDS	VARIABLE

- Intradomain Routeing Protocol Discriminator — architectural constant
- Length Indicator – Length of fixed header in octets
- Version/Protocol ID Extension – 1
- ID Length — Length of the ID field of NSAP addresses and NETs used in this routeing domain. This field shall take on one of the following values:
  - An integer between 1 and 8, inclusive, indicating an ID field of the corresponding length
  - The value zero, which indicates a 6 octet ID field length
  - The value 255, whhich means a null ID field (i.e. zero length)

  All other values are illegal and shall not be used.
- PDU 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 – the system ID of transmitting Intermediate system
- Holding Time – 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

  x CODE – 1

  x LENGTH – total length of the value field.

x VALUE –

	No. of Octets
Address Length	1
Area Address	Address Length
Address Length	1
Area Address	Address Length

- Address Length – Length of area address in octets.
- Area Address – Area address.

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

  x CODE – 8.

  x LENGTH – total length of the value field (may be zero).

  x VALUE – LENGTH octets of arbitrary value.

- Authentication Information — information for performing authentication of the originator of the PDU.

  x CODE — 10.

  x LENGTH — variable from 1–254 octets

  x VALUE —

	No. of Octets
Authentication Type	1
Authentication Value	VARIABLE

- Authentication Type — a one octet identifier for the type of authentication to be carried out. The following values are defined:

  0 — RESERVED

  1 — Cleartext Password

  2–254 — RESERVED

  255 — Routeing Domain private authentication method

- Authentication Value — determined by the value of the authentication type. If Cleartext Password as defined in this International Standard is used, then the authentication value is an octet string.

9.8 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 Extension	1
ID Length	1
R R R PDU Type	1
Version	1
ECO	1
User ECO	1
PDU Length	2
Remaining Lifetime	2
LSP ID	ID Length + 2
Sequence Number	4
Checksum	2
P ATT LSPDBOL IS Type	1
VARIABLE LENGTH FIELDS	VARIABLE

- Intradomain Routeing Protocol Discriminator – architectural constant

- Length Indicator – Length if fixed header in octets

- Version/Protocol ID Extension — 1

- ID Length — Length of the ID field of NSAP addresses and NETs used in this routeing domain. This field shall take on one of the following values:

  - An integer between 1 and 8, inclusive, indicating an ID field of the corresponding length

  - The value zero, which indicates a 6 octet ID field length

  - The value 255, whhich means a null ID field (i.e. zero length)

  All other values are illegal and shall not be used.

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

	No. of Octets
Source ID	ID Length
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/LSPDBOL/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:

    x Bit 4 - the Default Metric

    x Bit 5 - the Delay Metric

    x Bit 6 - the Expense Metric

    x Bit 7 - the Error Metric.
  - LSPDBOL – Bit 3 – A value of 0 indicates no LSP Database Overload, and a value of 1 indicates that the LSP Database is Overloaded. An LSP with this bit set will not be used by any decision process to calculate routes to another IS through the originating system.
  - IS Type – Bits 1 and 2 indicate the type of Intermediate System – One of the following values:

    x 0 – Unused value

    x 1 – (i.e. bit 1 set) Level 1 Intermediate system

    x 2 – Unused value

    x 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.

  x CODE – 1

  x LENGTH – total length of the value field.

  x VALUE –

	No. of Octets
Address Length	1
Area Address	Address Length
Address Length	1
Area Address	Address Length

  · Address Length – Length of area address in octets.

  · 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.

  x CODE – 2.

  x LENGTH – 1. plus a multiple of 11.

x VALUE –

			No. of Octets
Virtual Flag			1
0	I/E	Default Metric	1
S	I/E	Delay Metric	1
S	I/E	Expense Metric	1
S	I/E	Error Metric	1
Neighbour ID			ID Length + 1
0	I/E	Default Metric	1
S	I/E	Delay Metric	1
S	I/E	Expense Metric	1
S	I/E	Error Metric	1
Neighbour ID			ID Length + 1

- 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).
- Default Metric is the value of the default metric for the link to the listed neighbour. Bit 8 of this field is reserved. Bit 7 of this field (marked I/E) indicates the metric type, and shall contain the value "0", indicating an Internal metric.
- 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 (marked I/E) indicates the metric type, and shall contain the value "0", indicating an Internal metric.
- 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 (marked I/E) indicates the metric type, and shall contain the value "0", indicating an Internal metric.
- 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 (marked I/E) indicates the metric type, and shall contain the value "0", indicating an Internal metric.
- Neighbour ID. For Intermediate System neighbours, the first ID Length octets are the neighbour's system ID, and the last octet is 0. For pseudonode neighbours, the first ID Length octets is the LAN Level 1 Designated Intermediate System's ID, and the last 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.

  x CODE – 3.

  x LENGTH – 4. plus a multiple of 6.

  x VALUE –

			No. of Octets
0	I/E	Default Metric	1
S	I/E	Delay Metric	1
S	I/E	Expense Metric	1
S	I/E	Error Metric	1
Neighbour ID			ID Length
Neighbour ID			ID Length

- Default Metric is the value of the default metric for the link to each of the listed neighbours. Bit 8 of this field is reserved. Bit 7 of this field (marked I/E) indicates the metric type, and shall contain the value "0", indicating an Internal metric.
- 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 (marked I/E) indicates the metric type, and shall contain the value "0", indicating an Internal metric.
- 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 (marked I/E) indicates the metric type, and shall contain the value "0", indicating an Internal metric.
- 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 (marked I/E) indicates the metric type, and shall contain the value "0", indicating an Internal metric.
- Neighbour ID – system ID of End system neighbour.

- Authentication Information — information for performing authentication of the originator of the PDU.

  x CODE — 10.

  x LENGTH — variable from 1–254 octets

  x VALUE —

	No. of Octets
Authentication Type	1
Authentication Value	VARIABLE

  · Authentication Type — a one octet identifier for the type of authentication to be carried out. The following values are defined:

    0 — RESERVED

    1 — Cleartext Password

    2–254 — RESERVED

    255 — Routeing Domain private authentication method

  · Authentication Value — determined by the value of the authentication type. If Cleartext Password as defined in this International Standard is used, then the authentication value is an octet string.

9.9 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.

	No. of Octets
Intradomain Routeing Protocol Discriminator	1
Length Indicator	1
Version/Protocol ID Extension	1
ID Length	1
R R R PDU Type	1
Version	1
ECO	1
User ECO	1
PDU Length	2
Remaining Lifetime	2
LSP ID	ID Length + 2
Sequence Number	4
Checksum	2
P ATT LSPDBOL IS Type	1
VARIABLE LENGTH FIELDS	VARIABLE

- Intradomain Routeing Protocol Discriminator – architectural constant

- Length Indicator – Length of fixed header in octets

- Version/Protocol ID Extension – 1

- ID Length — Length of the ID field of NSAP addresses and NETs used in this routeing domain. This field shall take on one of the following values:

  - An integer between 1 and 8, inclusive, indicating an ID field of the corresponding length

  - The value zero, which indicates a 6 octet ID field length

  - The value 255, whhich means a null ID field (i.e. zero length)

  All other values are illegal and shall not be used.

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

	No. of Octets
Source ID	ID Length
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/LSPDBOL/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:

    x Bit 4 - the Default Metric

    x Bit 5 - the Delay Metric

    x Bit 6 - the Expense Metric

    x Bit 7 - the Error Metric.

  - LSPDBOL – Bit 3 – A value of 0 indicates no LSP Database Overload, and a value of 1 indicates that the LSP Database is Overloaded. An LSP with this bit set will not be used by any decision process to calculate routes to another IS through the originating system.

  - IS Type – Bits 1 and 2 indicate the type of Intermediate System – One of the following values:

    x 0 – Unused value

    x 1 – (i.e. bit 1 set) Level 1 Intermediate system

    x 2 – Unused value

    x 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.

  x CODE – 1

  x LENGTH – total length of the value field.

  x VALUE –

	No. of Octets
Address Length	1
Area Address	Address Length
Address Length	1
Area Address	Address Length

  · Address Length – Length of area address in octets.

  · 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.

  x CODE – 4.

  x LENGTH – 6

  x 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.

x CODE – 2.

x LENGTH – 1. plus a multiple of 11.

x VALUE –

			No. of Octets
Virtual Flag			1
0	I/E	Default Metric	1
S	I/E	Delay Metric	1
S	I/E	Expense Metric	1
S	I/E	Error Metric	1
Neighbour ID			ID Length + 1
0	I/E	Default Metric	1
S	I/E	Delay Metric	1
S	I/E	Expense Metric	1
S	I/E	Error Metric	1
Neighbour ID			ID Length + 1

- 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).
- Default Metric is the value of the default metric for the link to the listed neighbour. Bit 8 of this field is reserved. Bit 7 of this field (marked I/E) indicates the metric type, and shall contain the value "0", indicating an Internal metric.
- 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 (marked I/E) indicates the metric type, and shall contain the value "0", indicating an Internal metric.
- 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 (marked I/E) indicates the metric type, and shall contain the value "0", indicating an Internal metric.
- 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 (marked I/E) indicates the metric type, and shall contain the value "0", indicating an Internal metric.
- Neighbour ID. For Intermediate System neighbours, the first ID Length octets are the neighbour's system ID, and the last octet is 0. For pseudonode neighbours, the first ID Length octets is the LAN Level 1 Designated Intermediate System's ID, and the last octet is a non-zero quantity defined by the LAN Level 1 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.

x CODE – 5.

x LENGTH – Total length of the VALUE field.

x VALUE –

			No. of Octets
0	I/E	Default Metric	1
S	I/E	Delay Metric	1
S	I/E	Expense Metric	1
S	I/E	Error Metric	1
Address Prefix Length			1
Address Prefix			⌈Address Prefix Length /2⌉
Address Prefix Length			1
Address Prefix			⌈Address Prefix Length /2⌉

- Default Metric is the value of the default metric for the link to each of the listed neighbours. Bit 8 of this field is reserved. Bit 7 (marked I/E) indicates the metric type, and may be set to zero indicating an internal metric, or may be set to 1 indicating an external metric.
- 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 (marked I/E) indicates the metric type, and may be set to zero indicating an internal metric, or may be set to 1 indicating an external metric.
- 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 (marked I/E) indicates the metric type, and may be set to zero indicating an internal metric, or may be set to 1 indicating an external metric.

- 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 (marked I/E) indicates the metric type, and may be set to zero indicating an internal metric, or may be set to 1 indicating an external metric.
- Address Prefix Length is the length in semi-octets of the following prefix. A length of zero indicates a prefix that matches all NSAPs.
- Address Prefix is a reachable address prefix encoded as described in 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.

- Authentication Information — information for performing authentication of the originator of the PDU.

x CODE — 10.

x LENGTH — variable from 1–254 octets

x VALUE —

	No. of Octets
Authentication Type	1
Authentication Value	VARIABLE

- Authentication Type — a one octet identifier for the type of authentication to be carried out. The following values are defined:

  0 — RESERVED

  1 — Cleartext Password

  2–254 — RESERVED

  255 — Routeing Domain private authentication method
- Authentication Value — determined by the value of the authentication type. If Cleartext Password as defined in this International Standard is used, then the authentication value is an octet string.

9.10 Level 1 Complete Sequence Numbers PDU

	No. of Octets
Intradomain Routeing Protocol Discriminator	1
Length Indicator	1
Version/Protocol ID Extension	1
ID Length	1
R R R PDU Type	1
Version	1
ECO	1
User ECO	1
PDU Length	2
Source ID	ID Length + 1
Start LSP ID	ID Length + 2
End LSP ID	ID Length + 2
VARIABLE LENGTH FIELDS	VARIABLE

- Intradomain Routeing Protocol Discriminator – architectural constant
- Length Indicator – Length of fixed header in octets
- Version/Protocol ID Extension – 1
- ID Length — Length of the ID field of NSAP addresses and NETs used in this routeing domain. This field shall take on one of the following values:
  - An integer between 1 and 8, inclusive, indicating an ID field of the corresponding length
  - The value zero, which indicates a 6 octet ID field length
  - The value 255, whhich means a null ID field (i.e. zero length)

  All other values are illegal and shall not be used.
- PDU 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 – the system ID of Intermediate System (with zero Circuit ID) generating this Sequence Numbers PDU.

- Start LSP ID – the system ID of first LSP in the range covered by this Complete Sequence Numbers PDU.

- End LSP ID – the system ID of last LSP in the range covered by this Complete Sequence Numbers PDU.

- VARIABLE LENGTH FIELDS – fields of the form:

	No. of Octets
CODE	1
LENGTH	1
VALUE	LENGTH

Any codes in a received CSNP that are not recognised are ignored.

Currently defined codes are:

- LSP Entries – This may appear multiple times. The option fields, if they appear more than once, shall appear sorted into ascending LSPID order.

  x CODE – 9

  x LENGTH – total length of the value field.

  x VALUE – a list of LSP entries of the form:

	No. of Octets
Remaining Lifetime	2
LSP ID	ID Length + 2
LSP Sequence Number	4
Checksum	2
Remaining Lifetime	2
LSP ID	ID Length + 2
LSP Sequence Number	4
Checksum	2

  · Remaining Lifetime – Remaining Lifetime of LSP.

  · LSP ID – system ID of the LSP to which this entry refers.

  · LSP Sequence 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).

- Authentication Information — information for performing authentication of the originator of the PDU.

  x CODE — 10.

  x LENGTH — variable from 1–254 octets

  x VALUE —

	No. of Octets
Authentication Type	1
Authentication Value	VARIABLE

  · Authentication Type — a one octet identifier for the type of authentication to be carried out. The following values are defined:

  0 — RESERVED

  1 — Cleartext Password

  2–254 — RESERVED

  255 — Routeing Domain private authentication method

  · Authentication Value — determined by the value of the authentication type. If Cleartext Password as defined in this International Standard is used, then the authentication value is an octet string.

9.11 Level 2 Complete Sequence Numbers PDU

	No. of Octets
Intradomain Routeing Protocol Discriminator	1
Length Indicator	1
Version/Protocol ID Extension	1
ID Length	1
R | R | R | PDU Type	1
Version	1
ECO	1
User ECO	1
PDU Length	2
Source ID	ID Length + 1
Start LSP ID	ID Length + 2
End LSP ID	ID Length + 2
VARIABLE LENGTH FIELDS	VARIABLE

- Intradomain Routeing Protocol Discriminator – architectural constant

- Length Indicator – Length of fixed header in octets

- Version/Protocol ID Extension – 1

- ID Length — Length of the ID field of NSAP addresses and NETs used in this routeing domain. This field shall take on one of the following values:

  - An integer between 1 and 8, inclusive, indicating an ID field of the corresponding length

  - The value zero, which indicates a 6 octet ID field length

  - The value 255, whhich means a null ID field (i.e. zero length)

  All other values are illegal and shall not be used.

- PDU 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 – the system ID of Intermediate System (with zero Circuit ID) generating this Sequence Numbers PDU.

- Start LSP ID – the system ID of first LSP in the range covered by this Complete Sequence Numbers PDU.

- End LSP ID – the system ID of last LSP in the range covered by this Complete Sequence Numbers PDU.

- VARIABLE LENGTH FIELDS – fields of the form:

	No. of Octets
CODE	1
LENGTH	1
VALUE	LENGTH

Any codes in a received CSNP that are not recognised are ignored.

Currently defined codes are:

- LSP Entries – this may appear multiple times. The option fields, if they appear more than once, shall appear sorted into ascending LSPID order.

  x CODE – 9

  x LENGTH – total length of the value field.

  x VALUE – a list of LSP entries of the form:

	No. of Octets
Remaining Lifetime	2
LSP ID	ID Length + 2
LSP Sequence Number	4
Checksum	2
Remaining Lifetime	2
LSP ID	ID Length + 2
LSP Sequence Number	4
Checksum	2

  · Remaining Lifetime – Remaining Lifetime of LSP.

  · LSP ID – the system ID of the LSP to which this entry refers.

  · LSP Sequence 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).

- Authentication Information — information for performing authentication of the originator of the PDU.

x CODE — 10.

x LENGTH — variable from 1–254 octets

x VALUE —

	No. of Octets
Authentication Type	1
Authentication Value	VARIABLE

- Authentication Type — a one octet identifier for the type of authentication to be carried out. The following values are defined:

  0 — RESERVED

  1 — Cleartext Password

  2–254 — RESERVED

  255 — Routeing Domain private authentication method

- Authentication Value — determined by the value of the authentication type. If Cleartext Password as defined in this International Standard is used, then the authentication value is an octet string.

9.12 Level 1 Partial Sequence Numbers PDU

	No. of Octets
Intradomain Routeing Protocol Discriminator	1
Length Indicator	1
Version/Protocol ID Extension	1
ID Length	1
R R R PDU Type	1
Version	1
ECO	1
User ECO	1
PDU Length	2
Source ID	ID Length + 1
VARIABLE LENGTH FIELDS	VARIABLE

- Intradomain Routeing Protocol Discriminator – architectural constant

- Length Indicator – Length of fixed header in octets

- Version/Protocol ID Extension – 1

- ID Length — Length of the ID field of NSAP addresses and NETs used in this routeing domain. This field shall take on one of the following values:

  - An integer between 1 and 8, inclusive, indicating an ID field of the corresponding length

  - The value zero, which indicates a 6 octet ID field length

  - The value 255, whhich means a null ID field (i.e. zero length)

  All other values are illegal and shall not be used.

- PDU 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 – the system ID of Intermediate system (with zero Circuit ID) generating this Sequence Numbers PDU.

- VARIABLE LENGTH FIELDS – fields of the form:

	No. of Octets
CODE	1
LENGTH	1
VALUE	LENGTH

Any codes in a received PSNP that are not recognised are ignored.

Currently defined codes are:

- LSP Entries – this may appear multiple times. The option fields, if they appear more than once, shall appear sorted into ascending LSPID order.

  x CODE – 9

  x LENGTH – total length of the value field.

  x VALUE – a list of LSP entries of the form:

	No. of Octets
Remaining Lifetime	2
LSP ID	ID Length + 2
LSP Sequence Number	4
Checksum	2
Remaining Lifetime	2
LSP ID	ID Length + 2
LSP Sequence Number	4
Checksum	2

  · Remaining Lifetime – Remaining Lifetime of LSP.

  · LSP ID – the system ID of the LSP to which this entry refers.

  · LSP Sequence 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).

- Authentication Information — information for performing authentication of the originator of the PDU.

  x CODE — 10.

  x LENGTH — variable from 1–254 octets

  x VALUE —

	No. of Octets
Authentication Type	1
Authentication Value	VARIABLE

  · Authentication Type — a one octet identifier for the type of authentication to be carried out. The following values are defined:

    0 — RESERVED

    1 — Cleartext Password

    2–254 — RESERVED

    255 — Routeing Domain private authentication method

  · Authentication Value — determined by the value of the authentication type. If Cleartext Password as defined in this International Standard is used, then the authentication value is an octet string.

9.13 Level 2 Partial Sequence Numbers PDU

	No. of Octets
Intradomain Routeing Protocol Discriminator	1
Length Indicator	1
Version/Protocol ID Extension	1
ID Length	1
R | R | R | PDU Type	1
Version	1
ECO	1
User ECO	1
PDU Length	2
Source ID	ID Length + 1
VARIABLE LENGTH FIELDS	VARIABLE

- Intradomain Routeing Protocol Discriminator – architectural constant

- Length Indicator – Length of fixed header in octets

- Version/Protocol ID Extension – 1

- ID Length — Length of the ID field of NSAP addresses and NETs used in this routeing domain. This field shall take on one of the following values:

  - An integer between 1 and 8, inclusive, indicating an ID field of the corresponding length

  - The value zero, which indicates a 6 octet ID field length

  - The value 255, whhich means a null ID field (i.e. zero length)

  All other values are illegal and shall not be used.

- PDU 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 – the system ID of Intermediate system (with zero Circuit ID) generating this Sequence Numbers PDU.

- VARIABLE LENGTH FIELDS – fields of the form:

	No. of Octets
CODE	1
LENGTH	1
VALUE	LENGTH

Any codes in a received PSNP that are not recognised are ignored.

Currently defined codes are:

- LSP Entries – this may appear multiple times. The option fields, if they appear more than once, shall appear sorted into ascending LSPID order.

  x CODE – 9

  x LENGTH – total length of the value field.

  x VALUE – a list of LSP entries of the form:

	No. of Octets
Remaining Lifetime	2
LSP ID	ID Length + 2
LSP Sequence Number	4
Checksum	2
Remaining Lifetime	2
LSP ID	ID Length + 2
LSP Sequence Number	4
Checksum	2

  · Remaining Lifetime – Remaining Lifetime of LSP.

  · LSP ID – the system ID of the LSP to which this entry refers.

  · LSP Sequence 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).

- Authentication Information — information for performing authentication of the originator of the PDU.

  x CODE — 10.

  x LENGTH — variable from 1–254 octets

x VALUE —

	No. of Octets
Authentication Type	1
Authentication Value	VARIABLE

- Authentication Type — a one octet identifier for the type of authentication to be carried out. The following values are defined:

  0 — RESERVED

  1 — Cleartext Password

  2–254 — RESERVED

  255 — Routeing Domain private authentication method

- Authentication Value — determined by the value of the authentication type. If Cleartext Password as defined in this International Standard is used, then the authentication value is an octet string.

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 ISO 10589 in the GDMO notation as defined in ISO 10165-4.

11.1.1 Naming Hierarchy

The containment hierarchy for ISO 10589 is illustrated below in figure 8.

11.1.2 Resetting of Timers

Many of the attributes defined herein 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 applicable behaviour clause of the GDMO. See 11.2.16

11.1.3 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 managed object is enabled and it may be impossible to change the allocation without first disabling and re-enabling the corresponding Network entity. Therefore this International Standard only requires that system management shall be able to change these attributes 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 behaviour clause of the GDMO. See 11.2.16.

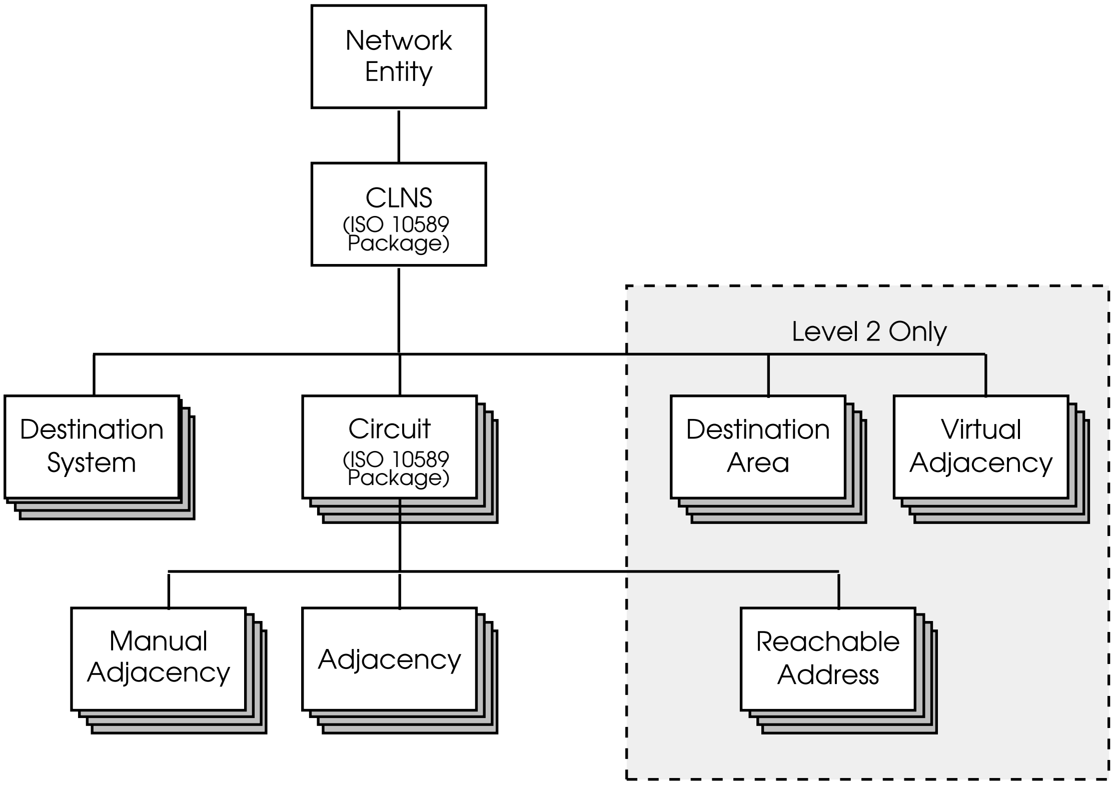


Figure 8 - Containment and Naming Hierarchy

11.2 GDMO Definition

11.2.1 Name Bindings

```
iSO10589-NB NAME BINDING
  SUBORDINATE OBJECT CLASS cLNS;
  NAMED BY
    SUPERIOR OBJECT CLASS
        "ISO/IEC xxxxx":networkEntity;
    WITH ATTRIBUTE
        "ISO/IEC xxxxx":cLNS-MO-Name;
  CREATE with-automatic-instance-naming
        iSO10589-NB-p1;
  DELETE only-if-no-contained-objects;
REGISTERED AS {ISO10589-ISIS.nboi iSO10589-NB
        (1)};
```

```
level1ISO10589Circuit-NB NAME BINDING
  SUBORDINATE OBJECT CLASS circuit;
  NAMED BY
    SUPERIOR OBJECT CLASS cLNS;
    WITH ATTRIBUTE
        "ISO/IEC xxxxx":circuit-MO-Name;
  CREATE with-reference-object
        iSO10589Circuit-MO-p1;
  DELETE only-if-no-contained-objects;
REGISTERED AS {ISO10589-ISIS.nboi
        level1ISO10589Circuit-NB (2)};
```

```
destinationSystem-NB NAME BINDING
  SUBORDINATE OBJECT CLASS destinationSystem;
  NAMED BY
    SUPERIOR OBJECT CLASS cLNS;
    WITH ATTRIBUTE networkEntityTitle;
REGISTERED AS {ISO10589-ISIS.nboi
        destinationSystem-NB (3)};
```

```
destinationArea-NB NAME BINDING
  SUBORDINATE OBJECT CLASS destinationArea;
  NAMED BY
    SUPERIOR OBJECT CLASS cLNS;
    WITH ATTRIBUTE addressPrefix;
  BEHAVIOUR destinationArea-NB-B BEHAVIOUR
    DEFINED AS This name binding is only applicable
        where the superior object has an iSType of Level2;;
REGISTERED AS {ISO10589-ISIS.nboi
        destinationArea-NB (4)};
```

```
virtualAdjacency-NB NAME BINDING
  SUBORDINATE OBJECT CLASS virtualAdjacency;
  NAMED BY
    SUPERIOR OBJECT CLASS cLNS;
    WITH ATTRIBUTE networkEntityTitle;
  BEHAVIOUR virtualAdjacency-NB-B BEHAVIOUR
    DEFINED AS This name binding is only applicable
        where the superior  object has an iSType of Level2;;
REGISTERED AS {ISO10589-ISIS.nboi
        virtualAdjacency-NB (5)};
```

```
reachableAddress-NB NAME BINDING
  SUBORDINATE OBJECT CLASS reachableAddress;
  NAMED BY
    SUPERIOR OBJECT CLASS circuit;
    WITH ATTRIBUTE addressPrefix;
  BEHAVIOUR reachableAddress-NB-B BEHAVIOUR
    DEFINED AS This name binding is only applicable
        where the superior object of the Circuit instance is
        an object  with iSType level2IS;;
  CREATE with-reference-object  reachableAddressP1
        reachableAddressP2;
  DELETE only-if-no-contained-objects;
REGISTERED AS {ISO10589-ISIS.nboi
        reachableAddress-NB (6)};
```

```
adjacency-NB NAME BINDING
  SUBORDINATE OBJECT CLASS adjacency;
  NAMED BY
    SUPERIOR OBJECT CLASS circuit;
    WITH ATTRIBUTE adjacencyName;
REGISTERED AS {ISO10589-ISIS.nboi adjacency-NB
        (7)};
```

```
manualAdjacency-NB NAME BINDING
  SUBORDINATE OBJECT CLASS manualAdjacency;
  NAMED BY
    SUPERIOR OBJECT CLASS circuit;
    WITH ATTRIBUTE adjacencyName;
  BEHAVIOUR manualAdjacency-NB-B BEHAVIOUR
    DEFINED AS When an instance name is specified in
        the CREATE operation, that value shall be used for
        the adjacencyName, otherwise automatic instance
        naming shall be used;;
  CREATE with-reference-object,
        with-automatic-instance-naming
    manualAdjacencyP1  manualAdjacencyP2;
  DELETE only-if-no-contained-objects;
REGISTERED AS {ISO10589-ISIS.nboi
        manualAdjacency-NB (8)};
```

11.2.2 The CLNS Managed Object for ISO 10589

```
cLNS MANAGED OBJECT CLASS
  DERIVED FROM "ISO/IEC xxxx":cLNS;
        -- To be replaced by the number of the network layer
        MO definitions when assigned.
  CONDITIONAL PACKAGES
    level1ISO10589Package
      PRESENT IF The Intermediate System is a Level 1
        Intermediate System,
    level2ISO10589Package
      PRESENT IF The Intermediate System is a Level 2
        Intermediate System (i.e. the value of iSType is
        Level2),
    partitionRepairPackage
      PRESENT IF The Intermediate System is a Level 2
        Intermediate System and the partition repair option
        is implemented,
    level1AuthenticationPackage
      PRESENT IF The authentication procedures are implemented,
    level2AuthenticationPackage
      PRESENT IF The Intermediate System is a Level 2
        Intermediate System and the authentication procedures are implemented;
REGISTERED AS {ISO10589-ISIS.moi cLNS (1)};


level1ISO10589Package PACKAGE
  ATTRIBUTES
    version GET,
    iSType GET,
        maximumPathSplits
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.maximumPathSplits-Default
      PERMITTED VALUES
        ISO10589-ISIS.MaximumPathSplits-Permitted
      GET-REPLACE,
    maximumBuffers
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.maximumBuffers-Default
      PERMITTED VALUES
        ISO10589-ISIS.MaximumBuffers-Permitted
      GET-REPLACE,
    minimumLSPTransmissionInterval
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.minimumLSPTransmissionInterval-Default
      PERMITTED VALUES
        ISO10589-ISIS.MinimumLSPTransmissionInterval-Permitted
      GET-REPLACE,
    maximumLSPGenerationInterval
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.maximumLSPGenerationInterval-Default
      PERMITTED VALUES
        ISO10589-ISIS.MaximumLSPGenerationInterval-Permitted
      GET-REPLACE,
    minimumBroadcastLSPTransmissionInterval
      REPLACE-WITH-DEFAULT
  DEFAULT VALUE
    ISO10589-ISIS.minimumBroadcastLSPTransmissionInterval-Default
  PERMITTED VALUES
    ISO10589-ISIS.MinimumBroadcastLSPTransmissionInterval-Permitted
  GET-REPLACE,
  -- Note this is defined for all Circuits, but would only
    be required if one of them were a broadcast Circuit
completeSNPInterval
  REPLACE-WITH-DEFAULT
  DEFAULT VALUE
    ISO10589-ISIS.completeSNPInterval-Default
  PERMITTED VALUES
    ISO10589-ISIS.CompleteSNPInterval-Permitted
  GET-REPLACE,
  -- Ditto
originatingL1LSPBufferSize
  REPLACE-WITH-DEFAULT
  DEFAULT VALUE
    ISO10589-ISIS.originatingL1LSPBufferSize-Default
  PERMITTED VALUES
    ISO10589-ISIS.OriginatingL1LSPBufferSize-Permitted
  GET-REPLACE,
  -- Note: redirectHoldingTime moved to
    ISO9542ISPackage
manualAreaAddresses
  REPLACE-WITH-DEFAULT
  DEFAULT VALUE
    ISO10589-ISIS.manualAreaAddresses-Default
  PERMITTED VALUES
    ISO10589-ISIS.ManualAreaAddresses-Permitted
  GET ADD-REMOVE,
minimumLSPGenerationInterval
  REPLACE-WITH-DEFAULT
  DEFAULT VALUE
    ISO10589-ISIS.minimumLSPGenerationInterval-Default
  PERMITTED VALUES
    ISO10589-ISIS.MinimumLSPGenerationInterval-Permitted
  GET-REPLACE,
defaultESHelloTimer
  REPLACE-WITH-DEFAULT
  DEFAULT VALUE
    ISO10589-ISIS.defaultESHelloTime-Default
  PERMITTED VALUES
    ISO10589-ISIS.DefaultESHelloTime-Permitted
  GET-REPLACE,
pollESHelloRate
  REPLACE-WITH-DEFAULT
  DEFAULT VALUE
    ISO10589-ISIS.pollESHelloRate-Default
  PERMITTED VALUES
    ISO10589-ISIS.PollESHelloRate-Permitted
  GET-REPLACE,
partialSNPInterval
  REPLACE-WITH-DEFAULT
  DEFAULT VALUE
    ISO10589-ISIS.partialSNPInterval-Default
  PERMITTED VALUES
    ISO10589-ISIS.PartialSNPInterval-Permitted
  GET-REPLACE,
waitingTime
  REPLACE-WITH-DEFAULT
```

```
      DEFAULT VALUE
        ISO10589-ISIS.waitingTime-Default
      PERMITTED VALUES
        ISO10589-ISIS.WaitingTime-Permitted
      GET-REPLACE,
    dRISISHelloTimer
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.dRISISHelloTimer-Default
      PERMITTED VALUES
        ISO10589-ISIS.DRISISHelloTimer-Permitted
          GET-REPLACE,
    l1State GET,
    areaAddresses GET,
    -- PDUFormatErrors now in network layer MO
    corruptedLSPsDetected GET,
    lSPL1DatabaseOverloads GET,
    manualAddressesDroppedFromArea GET,
    attemptsToExceedMaximumSequenceNumber GET,
    sequenceNumberSkips GET,
    ownLSPPurges GET,
    iDFieldLengthMismatches GET;
  ATTRIBUTE GROUPS
    counters
      -- PDUFormatErrors now in Network Layer MO
      corruptedLSPsDetected
      lSPL1DatabaseOverloads
      manualAddressesDroppedFromArea
      attemptsToExceedMaximumSequenceNumber
      sequenceNumberSkips
      ownLSPPurges
      iDFieldLengthMismatches;
  -- activate and deactivate actions now in Network Layer
        MO
  NOTIFICATIONS
    "ISO/IEC xxxxx":pduFormatError
        notificationReceivingAdjacency,
      -- extra parameter for ISO 10589
    corruptedLSPDetected,
    lSPL1DatabaseOverload,
    manualAddressDroppedFromArea,
    attemptToExceedMaximumSequenceNumber,
    sequenceNumberSkip,
    ownLSPPurge,
    iDFieldLengthMismatch;
REGISTERED AS {ISO10589-ISIS.poi
        level1ISO10589Package (1)};


level2ISO10589Package PACKAGE
  ATTRIBUTES
    originatingL2LSPBufferSize
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.originatingL2LSPBufferSize-Default
      PERMITTED VALUES
        ISO10589-ISIS.OriginatingL2LSPBufferSize-Permitted
      GET-REPLACE,
    l2State GET,
    lSPL2DatabaseOverloads GET;
  ATTRIBUTE GROUPS
    counters
      lSPL2DatabaseOverloads;
  NOTIFICATIONS
    lSPL2DatabaseOverload;
REGISTERED AS {ISO10589-ISIS.poi
        level2ISO10589Package (2)};


partitionRepairPackage PACKAGE
  BEHAVIOUR DEFINITIONS partitionRepairPackage-B
        BEHAVIOUR
    DEFINED AS Present when the partition repair option
        is implemented;;
  ATTRIBUTES
    maximumVirtualAdjacencies
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.maximumVirtualAdjacencies-Default
      PERMITTED VALUES
        ISO10589-ISIS.MaximumVirtualAdjacencies-Permitted
      GET-REPLACE,
    partitionAreaAddresses GET,
    partitionDesignatedL2IntermediateSystem GET,
    partitionVirtualLinkChanges GET;
  ATTRIBUTE GROUPS
    counters
      partitionVirtualLinkChanges;
  NOTIFICATIONS
    partitionVirtualLinkChange;
REGISTERED AS {ISO10589-ISIS.poi
        partitionRepairPackage (3)};


level1AuthenticationPackage PACKAGE
  BEHAVIOUR DEFINITIONS
        level1AuthenticationPackage-B BEHAVIOUR
    DEFINED AS Present when the authentication procedures option is implemented;;
  ATTRIBUTES
    areaTransmitPassword
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.password-Default
      GET-REPLACE,
    areaReceivePasswords
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.passwords-Default
      GET-REPLACE
      ADD-REMOVE,
    authenticationFailures
      GET;
  ATTRIBUTE GROUPS
    counters
      authenticationFailures;
  NOTIFICATIONS
    authenticationFailure;
REGISTERED AS {ISO10589-ISIS.poi
        level1AuthenticationPackage (4)};


level2AuthenticationPackage PACKAGE
  BEHAVIOUR DEFINITIONS
        level2AuthenticationPackage-B BEHAVIOUR
    DEFINED AS Present when the authentication procedures option is implemented and the value of the
        iSType attribute is Level2;;
  ATTRIBUTES
    domainTransmitPassword
      REPLACE-WITH-DEFAULT
    DEFAULT VALUE
      ISO10589-ISIS.password-Default
    GET-REPLACE,
  domainReceivePasswords
    REPLACE-WITH-DEFAULT
    DEFAULT VALUE
      ISO10589-ISIS.passwords-Default
      GET-REPLACE
    ADD-REMOVE;
REGISTERED AS {ISO10589-ISIS.poi
      level2AuthenticationPackage (5)};
```

11.2.3 The Circuit Managed Object for ISO 10589

```
circuit MANAGED OBJECT CLASS
  DERIVED FROM "ISO/IEC xxxx":circuit;
  -- xxxx to be replaced with the number of the network
      layer managed object definitions when one is
      assigned
  CONDITIONAL PACKAGES
  level1ISO10589CircuitPackage
    PRESENT IF the Circuit is a level 1 ISO 10589 Cir-
      cuit,
  level1ISO10589BroadcastCircuitPackage
    PRESENT IF the Circuit is a level 1 ISO 10589
      broadcast Circuit,
  level1ISO10589PtToPtCircuitPackage
    PRESENT IF the Circuit is a level 1 ISO 10589 Point
      to Point Circuit,
  level2ISO10589DACircuitPackage
    PRESENT IF the Circuit is a level 2 ISO 10589 X.25
      DA Circuit,
  level1ISO10589StaticCircuitPackage
    PRESENT IF the Circuit is a level 1 ISO10589 X.25
      STATIC Circuit (IN or OUT),
  level1ISO10589StaticOutCircuitPackage
    PRESENT IF the Circuit is a level1 ISO 10589 X.25
      STATIC OUT SNAP,
  level2ISO10589CircuitPackage
    PRESENT IF the IS is a Level2 ISO 10589 IS,
  level2ISO10589BroadcastCircuitPackage
    PRESENT IF the Circuit is a level 1 ISO 10589
      broadcast Circuit and the IS is a L2 IS,
  dACircuitCallEstablishmentMetricIncrementPackage
    PRESENT IF the Circuit is an X.25 DA circuit and
      support is implemented for call establishement met-
      ric increment values greater than zero,
  circuitAuthenticationPackage
    PRESENT IF the authentication procedures are im-
      plemented on this IS;
REGISTERED AS {ISO10589-ISIS.moi circuit (2)};
```

```
level1ISO10589CircuitPackage PACKAGE
  ATTRIBUTES
  type GET,
  helloTimer
    REPLACE-WITH-DEFAULT
    DEFAULT VALUE
      ISO10589-ISIS.helloTimer-Default
    PERMITTED VALUES
      ISO10589-ISIS.HelloTimer-Permitted
    GET-REPLACE,
  l1DefaultMetric
    REPLACE-WITH-DEFAULT
    DEFAULT VALUE
      ISO10589-ISIS.defaultMetric-Default
    PERMITTED VALUES
      ISO10589-ISIS.DefaultMetric-Permitted
    GET-REPLACE,
  l1DelayMetric
    REPLACE-WITH-DEFAULT
    DEFAULT VALUE
      ISO10589-ISIS.optionalMetric-Default
    PERMITTED VALUES
      ISO10589-ISIS.OptionalMetric-Permitted
    GET-REPLACE,
  l1ExpenseMetric
    REPLACE-WITH-DEFAULT
    DEFAULT VALUE
      ISO10589-ISIS.optionalMetric-Default
    PERMITTED VALUES
      ISO10589-ISIS.OptionalMetric-Permitted
    GET-REPLACE,
  l1ErrorMetric
    REPLACE-WITH-DEFAULT
    DEFAULT VALUE
      ISO10589-ISIS.optionalMetric-Default
    PERMITTED VALUES
      ISO10589-ISIS.OptionalMetric-Permitted
    GET-REPLACE,
  externalDomain
    REPLACE-WITH-DEFAULT
    DEFAULT VALUE
      ISO10589-ISIS.externalDomain-Default
    GET-REPLACE,
  circuitChanges GET,
  changesInAdjacencyState GET,
  initializationFailures GET,
  rejectedAdjacencies GET,
  controlPDUsSent GET,
  controlPDUsReceived GET,
  iDFieldLengthMismatches GET;
  ATTRIBUTE GROUPS
  counters
    circuitChanges
    changesInAdjacencyState
    initializationFailures
    rejectedAdjacencies
    controlPDUsSent
    controlPDUsReceived
    iDFieldLengthMismatches;
  -- Note: activate and deactivate are now imported from
      the network layer definition of circuit MO
  NOTIFICATIONS
  circuitChange,
  adjacencyStateChange,
  initializationFailure,
  rejectedAdjacency,
  iDFieldLengthMismatch;
REGISTERED AS {ISO10589-ISIS.poi
      level1ISO10589CircuitPackage (6)};
```

```
level1ISO10589BroadcastCircuitPackage PACKAGE
  BEHAVIOUR DEFINITIONS
      level1BroadcastCircuitPackage-B BEHAVIOUR
    DEFINED AS Present when the Circuit is of type
      Broadcast;;
  ATTRIBUTES
  iSISHelloTimer
    REPLACE-WITH-DEFAULT
```

```
        DEFAULT VALUE
          ISO10589-ISIS.iSISHelloTimer-Default
        PERMITTED VALUES
          ISO10589-ISIS.ISISHelloTimer-Permitted
        GET-REPLACE,
      l1IntermediateSystemPriority
        REPLACE-WITH-DEFAULT
        DEFAULT VALUE
          ISO10589-ISIS.l1IntermediateSystemPriority-Default
        PERMITTED VALUES
          ISO10589-ISIS.L1IntermediateSystemPriority-Permitted
        GET-REPLACE,
      l1CircuitID GET,
      l1DesignatedIntermediateSystem GET,
      lanL1DesignatedIntermediateSystemChanges GET;
    ATTRIBUTE GROUPS
      counters
        lanL1DesignatedIntermediateSystemChanges;
    NOTIFICATIONS
      lanL1DesignatedIntermediateSystemChange;
REGISTERED AS {ISO10589-ISIS.poi
          level1ISO10589BroadcastCircuitPackage (7)};


level1ISO10589PtToPtCircuitPackage PACKAGE
    BEHAVIOUR DEFINITIONS
          level1PtToPtCircuitPackage-B BEHAVIOUR
      DEFINED AS Present when the Circuit is of type PtToPt;;
    ATTRIBUTES
      ptPtCircuitID GET;
REGISTERED AS {ISO10589-ISIS.poi
          level1ISO10589PtToPtCircuitPackage (8)};


dACircuitCallEstablishmentMetricIncrementPackage
          PACKAGE
    BEHAVIOUR DEFINITIONS
          dACircuitCallEstablishmentMetricIncrementPackage-B BEHAVIOUR
      DEFINED AS Present when values of call establishment metric increment greater than zero are supported and the parent iS MO has iSType Level2;;
    ATTRIBUTES
      callEstablishmentDefaultMetricIncrement
        REPLACE-WITH-DEFAULT
        DEFAULT VALUE
          ISO10589-ISIS.callEstablishmentMetricIncrement-Default
        PERMITTED VALUES
          ISO10589-ISIS.CallEstablishmentMetricIncrement-Permitted
        GET-REPLACE,
      callEstablishmentDelayMetricIncrement
        REPLACE-WITH-DEFAULT
        DEFAULT VALUE
          ISO10589-ISIS.callEstablishmentMetricIncrement-Default
        PERMITTED VALUES
          ISO10589-ISIS.CallEstablishmentMetricIncrement-Permitted
        GET-REPLACE,
      callEstablishmentExpenseMetricIncrement
        REPLACE-WITH-DEFAULT
        DEFAULT VALUE
          ISO10589-ISIS.callEstablishmentMetricIncrement-Default
        PERMITTED VALUES
          ISO10589-ISIS.CallEstablishmentMetricIncrement-Permitted
        GET-REPLACE,
      callEstablishmentErrorMetricIncrement
        REPLACE-WITH-DEFAULT
        DEFAULT VALUE
          ISO10589-ISIS.callEstablishmentMetricIncrement-Default
        PERMITTED VALUES
          ISO10589-ISIS.CallEstablishmentMetricIncrement-Permitted
        GET-REPLACE;
REGISTERED AS {ISO10589-ISIS.poi
          dACircuitCallEstablishmentMetricIncrementPackage (9)};


level2ISO10589DACircuitPackage PACKAGE
    BEHAVIOUR DEFINITIONS
          level2ISO10589DACircuitPackage-B
          BEHAVIOUR
      DEFINED AS Present when the Circuit is of type DA, and the IS is operating as a L2 IS;;
        -- Note: a DA Circuit is only permitted on an L2 IS
    ATTRIBUTES
      recallTimer
        REPLACE-WITH-DEFAULT
        DEFAULT VALUE
          ISO10589-ISIS.recallTimer-Default
        PERMITTED VALUES
          ISO10589-ISIS.RecallTimer-Permitted
        GET-REPLACE,
      idleTimer
        REPLACE-WITH-DEFAULT
        DEFAULT VALUE
          ISO10589-ISIS.idleTimer-Default
        PERMITTED VALUES
          ISO10589-ISIS.IdleTimer-Permitted
        GET-REPLACE,
      initialMinimumTimer
        REPLACE-WITH-DEFAULT
        DEFAULT VALUE
          ISO10589-ISIS.initialMinimumTimer-Default
        PERMITTED VALUES
          ISO10589-ISIS.InitialMinimumTimer-Permitted
        GET-REPLACE,
      reserveTimer
        REPLACE-WITH-DEFAULT
        DEFAULT VALUE
          ISO10589-ISIS.reserveTimer-Default
        PERMITTED VALUES
          ISO10589-ISIS.ReserveTimer-Permitted
        GET-REPLACE,
      maximumSVCAdjacencies
        REPLACE-WITH-DEFAULT
        DEFAULT VALUE
          ISO10589-ISIS.maximumSVCAdjacencies-Default
        PERMITTED VALUES
          ISO10589-ISIS.MaximumSVCAdjacencies-Permitted
        GET-REPLACE,
      reservedAdjacency
        REPLACE-WITH-DEFAULT
        DEFAULT VALUE
          ISO10589-ISIS.reservedAdjacency-Default
      GET-REPLACE,
      -- Note: it is not clear that this attribute is required
    callsPlaced GET,
    callsFailed GET,
    timesExceededMaximumSVCAdjacencies GET;
  ATTRIBUTE GROUPS
    counters
      callsPlaced
      callsFailed
      timesExceededMaximumSVCAdjacencies;
  NOTIFICATIONS
    exceededMaximumSVCAdjacencies;
REGISTERED AS {ISO10589-ISIS.poi
        level2ISO10589DACircuitPackage (10)};


level1ISO10589StaticCircuitPackage PACKAGE
  BEHAVIOUR DEFINITIONS
        level1StaticCircuitPackage-B BEHAVIOUR
    DEFINED AS Present when the Circuit is of type
        Static;;
  ATTRIBUTES
    neighbourSNPAAddress
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.neighbourSNPAAddress-Default
      GET-REPLACE,
      -- Note: should this be handled by an X.25 IVMO?
    ptPtCircuitID GET;
REGISTERED AS {ISO10589-ISIS.poi
        level1ISO10589StaticCircuitPackage (11)};


level1ISO10589StaticOutCircuitPackage PACKAGE
  BEHAVIOUR DEFINITIONS
        level1StsticOutCircuitPackage-B BEHAVIOUR
    DEFINED AS Present when the Circuit is of type Static
        Out;;
  ATTRIBUTES
    recallTimer
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.recallTimer-Default
      PERMITTED VALUES
        ISO10589-ISIS.RecallTimer-Permitted
      GET-REPLACE,
    maximumCallAttempts
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.maximumCallAttempts-Default
      PERMITTED VALUES
        ISO10589-ISIS.MaximumCallAttempts-Permitted
      GET-REPLACE,
    callsPlaced GET,
    callsFailed GET,
    timesExceededMaximumCallAttempts GET;
  ATTRIBUTE GROUPS
    counters
      callsPlaced
      callsFailed
      timesExceededMaximumCallAttempts;
  NOTIFICATIONS
    exceededMaximumCallAttempts ;
REGISTERED AS {ISO10589-ISIS.poi
        level1ISO10589StaticOutCircuitPackage (12)};
```

```
level2ISO10589CircuitPackage PACKAGE
  BEHAVIOUR DEFINITIONS level2CircuitPackage-B
        BEHAVIOUR
    DEFINED AS Present when IS is an L2 IS;;
  ATTRIBUTES
    l2DefaultMetric
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.defaultMetric-Default
      PERMITTED VALUES
        ISO10589-ISIS.DefaultMetric-Permitted
      GET-REPLACE,
    l2DelayMetric
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.optionalMetric-Default
      PERMITTED VALUES
        ISO10589-ISIS.OptionalMetric-Permitted
      GET-REPLACE,
    l2ExpenseMetric
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.optionalMetric-Default
      PERMITTED VALUES
        ISO10589-ISIS.OptionalMetric-Permitted
      GET-REPLACE,
    l2ErrorMetric
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.optionalMetric-Default
      PERMITTED VALUES
        ISO10589-ISIS.OptionalMetric-Permitted
      GET-REPLACE,
    manualL2OnlyMode
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.manualL2OnlyMode-Default
      GET-REPLACE;
REGISTERED AS {ISO10589-ISIS.poi
        level2ISO10589CircuitPackage (13)};


level2ISO10589BroadcastCircuitPackage PACKAGE
  BEHAVIOUR DEFINITIONS
        level2BroadcastCircuitPackage-B BEHAVIOUR
    DEFINED AS Present when the Circuit is of type
        Broadcast and the IS is an L2 IS;;
  ATTRIBUTES
    l2IntermediateSystemPriority
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.l2IntermediateSystemPriority-Default
      PERMITTED VALUES
        ISO10589-ISIS.L2IntermediateSystemPriority-Permitted
      GET-REPLACE,
    l2CircuitID GET,
    l2DesignatedIntermediateSystem GET,
    lanL2DesignatedIntermediateSystemChanges GET;
  ATTRIBUTE GROUPS
    counters
      lanL2DesignatedIntermediateSystemChanges;
  NOTIFICATIONS
    lanL2DesignatedIntermediateSystemChange;
REGISTERED AS {ISO10589-ISIS.poi
        level2ISO10589BroadcastCircuitPackage (14)};
```

```
circuitAuthenticationPackage PACKAGE
  BEHAVIOUR DEFINITIONS
        circuitAuthenticationPackage-B BEHAVIOUR
    DEFINED AS Present when the authentication proce-
        dures option is implemented;;
  ATTRIBUTES
    circuitTransmitPassword
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.password-Default
      GET-REPLACE,
    circuitReceivePasswords
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.passwords-Default
      GET-REPLACE
      ADD-REMOVE,
    authenticationFailures GET;
  ATTRIBUTE GROUPS
    counters
      authenticationFailures;
  NOTIFICATIONS
    authenticationFailure;
REGISTERED AS {ISO10589-ISIS.poi
        circuitAuthenticationPackage (15)};
```

11.2.4 The Adjacency managed Object

```
adjacency MANAGED OBJECT CLASS
  DERIVED FROM "ISO/IEC 10165-2":top;
  CHARACTERIZED BY adjacencyPackage PACKAGE
    ATTRIBUTES
      adjacencyName GET,
      adjacencyState GET;
        -- Note: this is NOT operational state
  ;;
  CONDITIONAL PACKAGES
    broadcastAdjacencyPackage
      PRESENT IF the parent Circuit is of type broadcast,
    dAAdjacencyPackage
      PRESENT IF the parent Circuit is of type DA,
    ptToPtAdjacencyPackage
      PRESENT IF the parent Circuit is of type PtToPt or
        STATIC,
    iSAdjacencyPackage
      PRESENT IF the adjacency is to an IS (i.e the
        neighbourSystemType is Intermediate System L1
        Intermediate System or L2 Intermediate System),
    broadcastISAdjacencyPackage
      PRESENT IF the parent Circuit is of type broadcast
        and is to an IS as above,
    eSAdjacencyPackage
      PRESENT IF the adjacency is to an ES (i.e. the
        neighbourSystemType is EndSystem;
REGISTERED AS {ISO10589-ISIS.moi adjacency (3)};
```

```
broadcastAdjacencyPackage PACKAGE
  BEHAVIOUR DEFINITIONS
        broadcastAdjacencyPackage-B BEHAVIOUR
    DEFINED AS present if the parent Circuit is of type
        broadcast;;
  ATTRIBUTES
    neighbourLANAddress GET,
    neighbourSystemType GET;
REGISTERED AS {ISO10589-ISIS.poi
        broadcastAdjacencyPackage (16)};
```

```
dAAdjacencyPackage PACKAGE
  BEHAVIOUR DEFINITIONS dAAdjacencyPackage-B
        BEHAVIOUR
    DEFINED AS present if the parent Circuit is of type
        DA;;
  ATTRIBUTES
    sNPAAddress GET;
REGISTERED AS {ISO10589-ISIS.poi
        dAAdjacencyPackage (17)};
```

```
ptToPtAdjacencyPackage PACKAGE
  BEHAVIOUR DEFINITIONS
        ptToPtAdjacencyPackage-B BEHAVIOUR
    DEFINED AS present if the parent Circuit is of type
        PtToPt;;
  ATTRIBUTES
    neighbourSystemType GET;
REGISTERED AS {ISO10589-ISIS.poi
        ptToPtAdjacencyPackage (18)};
```

```
iSAdjacencyPackage PACKAGE
  BEHAVIOUR DEFINITIONS iSAdjacencyPackage-B
        BEHAVIOUR
    DEFINED AS present if the adjacency is to an IS;;
  ATTRIBUTES
    adjacencyUsageType GET,
    neighbourSystemID GET,
    neighbourAreas GET,
    holdingTimer GET;
REGISTERED AS {ISO10589-ISIS.poi
        iSAdjacencyPackage (19)};
```

```
broadcastISAdjacencyPackage PACKAGE
  BEHAVIOUR DEFINITIONS
        broadcastISAdjacencyPackage-B BEHAVIOUR
    DEFINED AS present if the parent Circuit is of type
        broadcast and the adjacency is to an IS;;
  ATTRIBUTES
    lANPriority GET;
REGISTERED AS {ISO10589-ISIS.poi
        broadcastISAdjacencyPackage (20)};
```

```
eSAdjacencyPackage PACKAGE
  BEHAVIOUR DEFINITIONS eSAdjacencyPackage-B
        BEHAVIOUR
    DEFINED AS present if the adjacency is to an ES;;
  ATTRIBUTES
    endSystemIDs GET;
REGISTERED AS {ISO10589-ISIS.poi
        eSAdjacencyPackage (21)};
```

11.2.5 The Manual Adjacency Managed Object

```
manualAdjacency MANAGED OBJECT CLASS
  DERIVED FROM "ISO/IEC 10165-2":top;
  CHARACTERIZED BY manualAdjacencyPackage
        PACKAGE
    ATTRIBUTES
      adjacencyName GET,
      neighbourLANAddress GET,
      endSystemIDs GET;
    ;;
REGISTERED AS {ISO10589-ISIS.moi
        manualAdjacency (4)};
```

11.2.6 The Virtual Adjacency managed Object

```
virtualAdjacency MANAGED OBJECT CLASS
  DERIVED FROM "ISO/IEC 10165-2":top;
  CHARACTERIZED BY virtualAdjacencyPackage
        PACKAGE
    ATTRIBUTES
      networkEntityTitle GET,
      metric GET;
    ;;
REGISTERED AS {ISO10589-ISIS.moi virtualAdjacency
        (5)};
```

11.2.7 The Destination Managed Object

```
-- The destination MO class is never instantiated. It exists
        only to allow the destinationSystem and
        destinationArea MO classes to be derived from it.
destination MANAGED OBJECT CLASS
  DERIVED FROM "ISO/IEC 10165-2":top;
  CHARACTERIZED BY destinationPackage
        PACKAGE
    ATTRIBUTES
      defaultMetricPathCost GET,
      defaultMetricOutputAdjacencies GET,
      delayMetricPathCost GET,
      delayMetricOutputAdjacencies GET,
      expenseMetricPathCost GET,
      expenseMetricOutputAdjacencies GET,
      errorMetricPathCost GET,
      errorMetricOutputAdjacencies GET;
    ;; -- no need for an object ID since it is never
        instantiated, but GDMO  needs one
REGISTERED AS {ISO10589-ISIS.moi destination (6)};
```

11.2.8 The Destination System Managed Object

```
destinationSystem MANAGED OBJECT CLASS
  DERIVED FROM destination;
  CHARACTERIZED BY destinationSystemPackage
        PACKAGE
    ATTRIBUTES
      networkEntityTitle GET;
    ;;
REGISTERED AS {ISO10589-ISIS.moi
        destinationSystem (7)};
```

11.2.9 The Destination Area Managed Object

```
destinationArea MANAGED OBJECT CLASS
  DERIVED FROM destination;
  CHARACTERIZED BY destinationAreaPackage
        PACKAGE
    ATTRIBUTES
      addressPrefix GET;
    ;;
REGISTERED AS {ISO10589-ISIS.moi destinationArea
        (8)};
```

11.2.10 The Reachable Address Managed Object

```
reachableAddress MANAGED OBJECT CLASS
  DERIVED FROM "ISO/IEC 10165-2":top;
  CHARACTERIZED BY reachableAddressPackage
        PACKAGE
    ATTRIBUTES
      addressPrefix GET,
      defaultMetric
        REPLACE-WITH-DEFAULT
        DEFAULT VALUE
        ISO10589-ISIS.defaultMetric-Default
        PERMITTED VALUES
        ISO10589-ISIS.DefaultMetric-Permitted
        GET-REPLACE,
      delayMetric
        REPLACE-WITH-DEFAULT
        DEFAULT VALUE
        ISO10589-ISIS.optionalMetric-Default
        PERMITTED VALUES
        ISO10589-ISIS.OptionalMetric-Permitted
        GET-REPLACE,
      expenseMetric
        REPLACE-WITH-DEFAULT
        DEFAULT VALUE
        ISO10589-ISIS.optionalMetric-Default
        PERMITTED VALUES
        ISO10589-ISIS.OptionalMetric-Permitted
        GET-REPLACE,
      errorMetric
        REPLACE-WITH-DEFAULT
        DEFAULT VALUE
        ISO10589-ISIS.optionalMetric-Default
        PERMITTED VALUES
        ISO10589-ISIS.OptionalMetric-Permitted
        GET-REPLACE,
      defaultMetricType
        REPLACE-WITH-DEFAULT
        DEFAULT VALUE
        ISO10589-ISIS.metricType-Default
        GET-REPLACE,
      delayMetricType
        REPLACE-WITH-DEFAULT
        DEFAULT VALUE
        ISO10589-ISIS.metricType-Default
        GET-REPLACE,
      expenseMetricType
        REPLACE-WITH-DEFAULT
        DEFAULT VALUE
        ISO10589-ISIS.metricType-Default
        GET-REPLACE,
      errorMetricType
```

```
            REPLACE-WITH-DEFAULT
            DEFAULT VALUE
            ISO10589-ISIS.metricType-Default
              GET-REPLACE,
        "ISO/IEC 10165-2":operationalState GET;
    ACTIONS
        activate,
        deactivate;
    ;;
  CONDITIONAL PACKAGES
    mappingRAPackage
      PRESENT IF the parent Circuit is of type broadcast or DA,
    broadcastRAPackage
      PRESENT IF the parent Circuit is of type broadcast and the value of mappingType is ‘manual’,
    dARAPackage
      PRESENT IF the parent Circuit is of type DA and the value of mappingType is ‘manual’;
REGISTERED AS {ISO10589-ISIS.moi
            reachableAddress (9)};


mappingRAPackage PACKAGE
  BEHAVIOUR DEFINITIONS mappingRAPackage-B
            BEHAVIOUR
    DEFINED AS When present, the NSAP to Circuit mapping is controlled by the value of the mappingType attribute;;
  ATTRIBUTES
    mappingType GET;
REGISTERED AS {ISO10589-ISIS.poi
            mappingRAPackage (22)};


broadcastRAPackage PACKAGE
  BEHAVIOUR DEFINITIONS broadcastRAPackage-B
            BEHAVIOUR
    DEFINED AS When present, the remote SNPA address is determined by the value of the lANAddress attribute;;
  ATTRIBUTES
    lANAddress
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.lANAddress-Default
      GET-REPLACE;
REGISTERED AS {ISO10589-ISIS.poi
            broadcastRAPackage (23)};


dARAPackage PACKAGE
  BEHAVIOUR DEFINITIONS dARAPackage-B
            BEHAVIOUR
    DEFINED AS When present, the remote SNPA address is determined by the value of the sNPAAddresses attribute;;
  ATTRIBUTES
    sNPAAddresses
      REPLACE-WITH-DEFAULT
      DEFAULT VALUE
        ISO10589-ISIS.sNPAAddresses-Default
      GET-REPLACE;
REGISTERED AS {ISO10589-ISIS.poi dARAPackage
            (24)};
```

11.2.11 Attribute Definitions

```
version ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ISO10589-ISIS.Version;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR version-B BEHAVIOUR
    DEFINED AS The version number of this International Standard to which the implementation conforms;;
REGISTERED AS {ISO10589-ISIS.aoi version (1)};


iSType ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ISO10589-ISIS.ISType;
  MATCHES FOR Equality;
  BEHAVIOUR iSType-B BEHAVIOUR
    DEFINED AS The type of this Intermediate System. The value of this attribute is only settable via the create parameter;;
REGISTERED AS {ISO10589-ISIS.aoi iSType (2)};


maximumPathSplits ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.MaximumPathSplits;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR maximumPathSplits-B BEHAVIOUR
    DEFINED AS Maximum number of paths with equal routeing metric value which it is permitted to split between;,
    replaceOnlyWhileDisabled-B;
  PARAMETERS constraintViolation;
REGISTERED AS {ISO10589-ISIS.aoi
        maximumPathSplits (3)};


maximumBuffers ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.MaximumBuffers;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR maximumBuffers-B BEHAVIOUR
    DEFINED AS 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 clause D.1.1);,
    resourceLimiting-B;
  PARAMETERS constraintViolation;
REGISTERED AS {ISO10589-ISIS.aoi maximumBuffers
        (4)};


minimumLSPTransmissionInterval ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.MinimumLSPTransmissionInterval;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR minimumLSPTransmissionInterval-B
        BEHAVIOUR
    DEFINED AS Minimum interval, in seconds, between re- transmissions of an LSP;,
    resettingTimer-B;
REGISTERED AS {ISO10589-ISIS.aoi
        minimumLSPTransmissionInterval (5)};
```

```
maximumLSPGenerationInterval ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.MaximumLSPGenerationInterval;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR maximumLSPGenerationInterval-B
        BEHAVIOUR
    DEFINED AS Maximum interval, in seconds, between
        generated LSPs by this system;,
    resettingTimer-B;
REGISTERED AS {ISO10589-ISIS.aoi
        maximumLSPGenerationInterval (6)};
```

```
minimumBroadcastLSPTransmissionInterval ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.MinimumBroadcastLSPTransmissio
        nInterval;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR
        minimumBroadcastLSPTransmissionInterval-B
        BEHAVIOUR
    DEFINED AS Minimum interval, in milliseconds, between
        transmission of LSPs on a broadcast circuit
        (See clause 7.3.15.6);,
    resettingTimer-B;
REGISTERED AS {ISO10589-ISIS.aoi
        minimumBroadcastLSPTransmissionInterval (7)};
```

```
completeSNPInterval ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.CompleteSNPInterval;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR completeSNPInterval-B BEHAVIOUR
    DEFINED AS Interval, in seconds, between generation
        of Complete Sequence Numbers PDUs by a Designated
        Intermediate System on a broadcast circuit;,
    resettingTimer-B;
REGISTERED AS {ISO10589-ISIS.aoi
        completeSNPInterval (8)};
```

```
originatingL1LSPBufferSize ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.OriginatingLSPBufferSize;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR originatingL1LSPBufferSize-B
        BEHAVIOUR
    DEFINED AS The maximum size of Level 1 LSPs and
        SNPs originated by this system;,
    replaceOnlyWhileDisabled-B;
  PARAMETERS constraintViolation;
REGISTERED AS {ISO10589-ISIS.aoi
        originatingL1LSPBufferSize (9)};
```

```
manualAreaAddresses ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.AreaAddresses;
  MATCHES FOR Equality, Set Comparison, Set
        Intersection;
  BEHAVIOUR manualAreaAddresses-B BEHAVIOUR
    DEFINED AS 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;;
REGISTERED AS {ISO10589-ISIS.aoi
        manualAreaAddresses (10)};
```

```
minimumLSPGenerationInterval ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.MinimumLSPGenerationInterval;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR minimumLSPGenerationInterval-B
        BEHAVIOUR
    DEFINED AS Maximum interval in seconds between
        successive generation of LSPs with the same LSPID
        by this IS;,
    resettingTimer-B;
REGISTERED AS {ISO10589-ISIS.aoi
        minimumLSPGenerationInterval (11)};
```

```
defaultESHelloTimer ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.DefaultESHelloTimer;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR defaultESHelloTimer-B BEHAVIOUR
    DEFINED AS The value to be used for the suggested
        ES configuration timer in ISH PDUs when not soliciting
        the ES configuration;;
REGISTERED AS {ISO10589-ISIS.aoi
        defaultESHelloTimer (12)};
```

```
pollESHelloRate ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.PollESHelloRate;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR pollESHelloRate-B BEHAVIOUR
    DEFINED AS The value to be used for the suggested
        ES configuration timer in ISH PDUs when soliciting
        the ES configuration;;
REGISTERED AS {ISO10589-ISIS.aoi pollESHelloRate
        (13)};
```

```
partialSNPInterval ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.PartialSNPInterval;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR partialSNPInterval-B BEHAVIOUR
    DEFINED AS Minimum interval between sending Partial
        Sequence Number PDUs;,
    resettingTimer-B;
REGISTERED AS {ISO10589-ISIS.aoi
        partialSNPInterval (14)};
```

```
waitingTime ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.WaitingTime;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR waitingTime-B BEHAVIOUR
    DEFINED AS Number of seconds to delay in waiting
        state before entering On state;,
    resettingTimer-B;
REGISTERED AS {ISO10589-ISIS.aoi waitingTime
        (15)};
```

```
dRISISHelloTimer ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.DRISISHelloTimer;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR dRISISHelloTimer-B BEHAVIOUR
    DEFINED AS The interval in seconds between the
        generation of IIH PDUs by the designated IS on a
        LAN;,
    resettingTimer-B;
REGISTERED AS {ISO10589-ISIS.aoi
        dRISISHelloTimer (16)};
```

```
l1State ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.DatabaseState;
  MATCHES FOR Equality;
  BEHAVIOUR l1State-B BEHAVIOUR
    DEFINED AS The state of the Level 1 database;;
REGISTERED AS {ISO10589-ISIS.aoi l1State (17)};
```

```
areaAddresses ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.AreaAddresses;
  MATCHES FOR Equality, Set Comparison, Set
        Intersection;
  BEHAVIOUR areaAddresses-B BEHAVIOUR
    DEFINED AS The union of the sets of manualAreaAddresses
        reported in all Level 1 Link State PDUs received
        by this Intermediate System;;
REGISTERED AS {ISO10589-ISIS.aoi areaAddresses
        (18)};
```

```
corruptedLSPsDetected ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR corruptedLSPsDetected-B BEHAVIOUR
    DEFINED AS Number of Corrupted LSP Detected
        events generated;;
REGISTERED AS {ISO10589-ISIS.aoi
        corruptedLSPsDetected (19)};
```

```
lSPL1DatabaseOverloads ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR lSPL1DatabaseOverloads-B
        BEHAVIOUR
    DEFINED AS Number of times the LSP L1 Database
        Overload event has been generated;;
REGISTERED AS {ISO10589-ISIS.aoi
        lSPL1DatabaseOverloads (20)};
```

```
manualAddressesDroppedFromArea ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR manualAddressesDroppedFromArea-B
        BEHAVIOUR
    DEFINED AS Number of times the Manual Addresses
        Dropped From Area event has been generated;;
REGISTERED AS {ISO10589-ISIS.aoi
        manualAddressesDroppedFromArea (21)};
```

```
attemptsToExceedMaximumSequenceNumber
        ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR
        attemptsToExceedMaximumSequenceNumber-B
        BEHAVIOUR
    DEFINED AS Number of times the Attempt To Exceed
        Maximum Sequence Number event has been
        generated;;
REGISTERED AS {ISO10589-ISIS.aoi
        attemptsToExceedMaximumSequenceNumber
        (22)};
```

```
sequenceNumberSkips ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR sequenceNumberSkips-B BEHAVIOUR
    DEFINED AS Number of times the Sequence Number
        Skipped event has been generated;;
REGISTERED AS {ISO10589-ISIS.aoi
        sequenceNumberSkips (23)};
```

```
ownLSPPurges ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR ownLSPPurges-B BEHAVIOUR
    DEFINED AS Number of times the Own LSP Purged
        event has been generated;;
REGISTERED AS {ISO10589-ISIS.aoi ownLSPPurges
        (24)};
```

```
iDFieldLengthMismatches ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR iDFieldLengthMismatches-B
        BEHAVIOUR
    DEFINED AS Number of times the iDFieldLengthMismatch
        event has been generated;;
REGISTERED AS {ISO10589-ISIS.aoi
        iDFieldLengthMismatches (25)};
```

```
originatingL2LSPBufferSize ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.OriginatingLSPBufferSize;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR originatingL2LSPBufferSize-B
        BEHAVIOUR
    DEFINED AS The maximum size of Level 2 LSPs and
        SNPs originated by this system;,
    replaceOnlyWhileDisabled-B;
  PARAMETERS constraintViolation;
REGISTERED AS {ISO10589-ISIS.aoi
        originatingL2LSPBufferSize (26)};
```

```
maximumVirtualAdjacencies ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.MaximumVirtualAdjacencies;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR maximumVirtualAdjacencies-B
        BEHAVIOUR
    DEFINED AS Maximum number of Virtual Adjacencies
        which may be created to repair partitioned
        Level 1 domains;;
REGISTERED AS {ISO10589-ISIS.aoi
        maximumVirtualAdjacencies (27)};
```

```
l2State ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.DatabaseState;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR l2State-B BEHAVIOUR
    DEFINED AS The state of the Level 2 database;;
REGISTERED AS {ISO10589-ISIS.aoi l2State (28)};

partitionAreaAddresses ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.AreaAddresses;
  MATCHES FOR Equality, Set Comparison, Set Intersection;
  BEHAVIOUR partitionAreaAddresses-B BEHAVIOUR
    DEFINED AS The set union of all manualAreaAddresses of all Intermediate systems in the partition reachable by non-virtual links (calculated from their Level 1 LSPs);;
REGISTERED AS {ISO10589-ISIS.aoi
        partitionAreaAddresses (29)};

partitionDesignatedL2IntermediateSystem ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.SystemID;
  MATCHES FOR Equality;
  BEHAVIOUR
        partitionDesignatedL2IntermediateSystem-B
        BEHAVIOUR
    DEFINED AS The ID of the Partition Designated Level 2 Intermediate System for this system;;
REGISTERED AS {ISO10589-ISIS.aoi
        partitionDesignatedL2IntermediateSystem (30)};

partitionVirtualLinkChanges ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR partitionVirtualLinkChanges-B
        BEHAVIOUR
    DEFINED AS Number of times the Partition Virtual Link Change Notification has been generated;;
REGISTERED AS {ISO10589-ISIS.aoi
        partitionVirtualLinkChanges (31)};

lSPL2DatabaseOverloads ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR lSPL2DatabaseOverloads-B
        BEHAVIOUR
    DEFINED AS Number of times the LSP L2 Database Overload event has been generated;;
REGISTERED AS {ISO10589-ISIS.aoi
        lSPL2DatabaseOverloads (32)};

type ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.CircuitType;
  MATCHES FOR Equality;
  BEHAVIOUR type-B BEHAVIOUR
    DEFINED AS The type of the circuit. This attribute may only be set when the Circuit is created. Subsequently it is read-only;;
REGISTERED AS {ISO10589-ISIS.aoi type (33)};

helloTimer ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.HelloTimer;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR helloTimer-B BEHAVIOUR
    DEFINED AS The period, in seconds, between ISH
        PDUs;,
    resettingTimer-B;
REGISTERED AS {ISO10589-ISIS.aoi helloTimer (34)};

l1DefaultMetric ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.HopMetric;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR l1defaultMetric-B BEHAVIOUR
    DEFINED AS The default metric value of this circuit for Level 1 traffic. The value of zero is reserved to indicate that this metric is not supported;;
REGISTERED AS {ISO10589-ISIS.aoi l1DefaultMetric
        (35)};

l1DelayMetric ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.HopMetric;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR l1DelayMetric-B BEHAVIOUR
    DEFINED AS The delay metric value of this circuit for Level 1 traffic. The value of zero is reserved to indicate that this metric is not supported;;
REGISTERED AS {ISO10589-ISIS.aoi l1DelayMetric
        (36)};

l1ExpenseMetric ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.HopMetric;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR l1ExpenseMetric-B BEHAVIOUR
    DEFINED AS The expense metric value of this circuit for Level 1 traffic. The value of zero is reserved to indicate that this metric is not supported;;
REGISTERED AS {ISO10589-ISIS.aoi l1ExpenseMetric
        (37)};

l1ErrorMetric ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.HopMetric;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR l1ErrorMetric-B BEHAVIOUR
    DEFINED AS The error metric value of this circuit for Level 1 traffic. The value of zero is reserved to indicate that this metric is not supported;;
REGISTERED AS {ISO10589-ISIS.aoi l1ErrorMetric
        (38)};

circuitChanges ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR circuitChanges-B BEHAVIOUR
    DEFINED AS Number of times this Circuit state changed between On and Off and vice versa;;
REGISTERED AS {ISO10589-ISIS.aoi circuitChanges
        (39)};
```

```
changesInAdjacencyState ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR changesInAdjacencyState-B
        BEHAVIOUR
    DEFINED AS Number of Adjacency State Change
        events generated;;
REGISTERED AS {ISO10589-ISIS.aoi
        changesInAdjacencyState (40)};

initializationFailures ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR initializationFailures-B BEHAVIOUR
    DEFINED AS Number of Initialization Failure events
        generated;;
REGISTERED AS {ISO10589-ISIS.aoi
        initializationFailures (41)};

rejectedAdjacencies ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR rejectedAdjacencies-B BEHAVIOUR
    DEFINED AS Number of Rejected Adjacency events
        generated;;
REGISTERED AS {ISO10589-ISIS.aoi
        rejectedAdjacencies (42)};

controlPDUsSent ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR controlPDUsSent-B BEHAVIOUR
    DEFINED AS Number of control PDUs sent on this
        circuit;;
REGISTERED AS {ISO10589-ISIS.aoi controlPDUsSent
        (43)};

controlPDUsReceived ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR controlPDUsReceived-B BEHAVIOUR
    DEFINED AS Number of control PDUs received on
        this circuit;;
REGISTERED AS {ISO10589-ISIS.aoi
        controlPDUsReceived (44)};

iSISHelloTimer ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.ISISHelloTimer;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR iSISHelloTimer-B BEHAVIOUR
    DEFINED AS 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 con-
        figuration;,
    resettingTimer-B;
REGISTERED AS {ISO10589-ISIS.aoi iSISHelloTimer
        (45)};

externalDomain ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ISO10589-ISIS.Boolean;
  MATCHES FOR Equality;
  BEHAVIOUR externalDomain-B BEHAVIOUR
    DEFINED AS If TRUE, suppress notmal transmission
        of and interpretation of Intra-domain ISIS PDUs on
        this circuit.;;
REGISTERED AS {ISO10589-ISIS.aoi externalDomain
        (46)};
```

```
l1IntermediateSystemPriority ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.IntermediateSystemPriority;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR l1IntermediateSystemPriority-B
        BEHAVIOUR
    DEFINED AS Priority for becoming LAN Level 1
        Designated Intermediate System;;
REGISTERED AS {ISO10589-ISIS.aoi
        l1IntermediateSystemPriority (47)};

l1CircuitID ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.CircuitID;
  MATCHES FOR Equality;
  BEHAVIOUR l1CircuitID-B BEHAVIOUR
    DEFINED AS The  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 concate-
        nation of the local system ID and the one octet local
        Circuit ID for this circuit.;;
REGISTERED AS {ISO10589-ISIS.aoi l1CircuitID
        (48)};

l1DesignatedIntermediateSystem ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.SystemID;
  MATCHES FOR Equality;
  BEHAVIOUR l1DesignatedIntermediateSystem-B
        BEHAVIOUR
    DEFINED AS The ID  of the LAN Level 1 Designated
        Intermediate System on this circuit. If, for any rea-
        son this system is not partaking in the relevant Des-
        ignated Intermediate System election process, then
        the value returned is zero;;
REGISTERED AS {ISO10589-ISIS.aoi
        l1DesignatedIntermediateSystem (49)};

lanL1DesignatedIntermediateSystemChanges
        ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR
        lanL1DesignatedIntermediateSystemChanges-B
        BEHAVIOUR
    DEFINED AS Number of LAN L1 Designated Inter-
        mediate System Change events generated;;
REGISTERED AS {ISO10589-ISIS.aoi
        lanL1DesignatedIntermediateSystemChanges (50)};
```

ptPtCircuitID **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.CircuitID;
MATCHES FOR Equality;
BEHAVIOUR ptPtCircuitID-B **BEHAVIOUR**
DEFINED AS The 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.;;
REGISTERED AS {ISO10589-ISIS.aoi ptPtCircuitID (51)};

callEstablishmentDefaultMetricIncrement **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.MetricIncrement;
MATCHES FOR Equality, Ordering;
BEHAVIOUR
callEstablishmentDefaultMetricIncrement-B **BEHAVIOUR**
DEFINED AS Additional value to be reported for the default metric value of unestablished DA adjacencies;;
REGISTERED AS {ISO10589-ISIS.aoi callEstablishmentDefaultMetricIncrement (52)};

callEstablishmentDelayMetricIncrement **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.MetricIncrement;
MATCHES FOR Equality, Ordering;
BEHAVIOUR
callEstablishmentDelayMetricIncrement-B **BEHAVIOUR**
DEFINED AS Additional value to be reported for the delay metric value of unestablished DA adjacencies;;
REGISTERED AS {ISO10589-ISIS.aoi callEstablishmentDelayMetricIncrement (53)};

callEstablishmentExpenseMetricIncrement **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.MetricIncrement;
MATCHES FOR Equality, Ordering;
BEHAVIOUR
callEstablishmentExpenseMetricIncrement-B **BEHAVIOUR**
DEFINED AS Additional value to be reported for the Expense metric value of unestablished DA adjacencies;;
REGISTERED AS {ISO10589-ISIS.aoi callEstablishmentExpenseMetricIncrement (54)};

callEstablishmentErrorMetricIncrement **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.MetricIncrement;
MATCHES FOR Equality, Ordering;
BEHAVIOUR callEstablishmentErrorMetricIncrement-B **BEHAVIOUR**
DEFINED AS Additional value to be reported for the Error metric value of unestablished DA adjacencies;;
REGISTERED AS {ISO10589-ISIS.aoi callEstablishmentErrorMetricIncrement (55)};

recallTimer **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.RecallTimer;
MATCHES FOR Equality, Ordering;
BEHAVIOUR recallTimer-B **BEHAVIOUR**
DEFINED AS Number of seconds that must elapse between a call failure on a DED circuit and a recall;,
resettingTimer-B;
REGISTERED AS {ISO10589-ISIS.aoi recallTimer (56)};

idleTimer **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.IdleTimer;
MATCHES FOR Equality, Ordering;
BEHAVIOUR idleTimer-B **BEHAVIOUR**
DEFINED AS Number of seconds of idle time before call is cleared;,
resettingTimer-B;
REGISTERED AS {ISO10589-ISIS.aoi idleTimer (57)};

initialMinimumTimer **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.InitialMinimumTimer;
MATCHES FOR Equality, Ordering;
BEHAVIOUR initialMinimumTimer-B **BEHAVIOUR**
DEFINED AS 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.);,
resettingTimer-B;
REGISTERED AS {ISO10589-ISIS.aoi initialMinimumTimer (58)};

reserveTimer **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.ReserveTimer;
MATCHES FOR Equality, Ordering;
BEHAVIOUR reserveTimer-B **BEHAVIOUR**
DEFINED AS Number of seconds, after call is cleared due to lack of traffic, during which the SVC remains reserved for the previous SNPA address;,
resettingTimer-B;
REGISTERED AS {ISO10589-ISIS.aoi reserveTimer (59)};

maximumSVCAdjacencies **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.MaximumSVCAdjacencies;
MATCHES FOR Equality, Ordering;
BEHAVIOUR maximumSVCAdjacencies-B **BEHAVIOUR**
DEFINED AS Number of Adjacencies to reserve for SVCs for this circuit. This is the maximum number of simultaneous calls which are possible on this circuit;,
resourceLimiting-B;
PARAMETERS constraintViolation;
REGISTERED AS {ISO10589-ISIS.aoi maximumSVCAdjacencies (60)};

```
reservedAdjacency ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ISO10589-ISIS.Boolean;
  MATCHES FOR Equality;
  BEHAVIOUR reservedAdjacency-B BEHAVIOUR
    DEFINED AS When True, indicates that one SVC
        must be reserved for a connection to an Intermediate
        System;,
    replaceOnlyWhileDisabled-B;
  PARAMETERS constraintViolation;
REGISTERED AS {ISO10589-ISIS.aoi
        reservedAdjacency (61)};

callsPlaced ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR callsPlaced-B BEHAVIOUR
    DEFINED AS Number of Call attempts (successful or
        unsuccessful);;
REGISTERED AS {ISO10589-ISIS.aoi callsPlaced (62)};

callsFailed ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR callsFailed-B BEHAVIOUR
    DEFINED AS Number of Unsuccessful Call attempts;;
REGISTERED AS {ISO10589-ISIS.aoi callsFailed (63)};

timesExceededMaximumSVCAdjacencies ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR
        timesExceededMaximumSVCAdjacencies-B
        BEHAVIOUR
    DEFINED AS Number of Exceeded Maximum SVC
        Adjacencies events generated;;
REGISTERED AS {ISO10589-ISIS.aoi
        timesExceededMaximumSVCAdjacencies (64)};

neighbourSNPAAddress ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.SNPAAddress;
  MATCHES FOR Equality;
  BEHAVIOUR neighbourSNPAAddress-B
        BEHAVIOUR
    DEFINED AS SNPA Address to call, or SNPA Address
        from which to accept call;;
REGISTERED AS {ISO10589-ISIS.aoi
        neighbourSNPAAddress (65)};

maximumCallAttempts ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.MaximumCallAttempts;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR maximumCallAttempts-B BEHAVIOUR
    DEFINED AS Maximum number of successive call
        failures before halting. (A value of zero means infinite
        retries.;;
REGISTERED AS {ISO10589-ISIS.aoi
        maximumCallAttempts (66)};

timesExceededMaximumCallAttempts ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR timesExceededMaximumCallAttempts-B
        BEHAVIOUR
    DEFINED AS Number of Exceeded Maximum Call
        Attempts events generated;;
REGISTERED AS {ISO10589-ISIS.aoi
        timesExceededMaximumCallAttempts (67)};

l2DefaultMetric ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.HopMetric;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR l2defaultMetric-B BEHAVIOUR
    DEFINED AS The default metric value of this circuit
        for Level 2 traffic. The value of zero is reserved to
        indicate that this metric is not supported;;
REGISTERED AS {ISO10589-ISIS.aoi l2DefaultMetric
        (68)};

l2DelayMetric ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.HopMetric;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR l2DelayMetric-B BEHAVIOUR
    DEFINED AS The delay metric value of this circuit for
        Level 2 traffic. The value of zero is reserved to indicate
        that this metric is not supported;;
REGISTERED AS {ISO10589-ISIS.aoi l2DelayMetric
        (69)};

l2ExpenseMetric ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.HopMetric;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR l2ExpenseMetric-B BEHAVIOUR
    DEFINED AS The expense metric value of this circuit
        for Level 2 traffic. The value of zero is reserved to
        indicate that this metric is not supported;;
REGISTERED AS {ISO10589-ISIS.aoi l2ExpenseMetric
        (70)};

l2ErrorMetric ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.HopMetric;
  MATCHES FOR Equality, Ordering;
  BEHAVIOUR l2ErrorMetric-B BEHAVIOUR
    DEFINED AS The error metric value of this circuit for
        Level 2 traffic. The value of zero is reserved to indicate
        that this metric is not supported;;
REGISTERED AS {ISO10589-ISIS.aoi l2ErrorMetric
        (71)};

manualL2OnlyMode ATTRIBUTE
  WITH ATTRIBUTE SYNTAX ISO10589-ISIS.Boolean;
  MATCHES FOR Equality;
  BEHAVIOUR manualL2OnlyMode-B BEHAVIOUR
    DEFINED AS When True, indicates that this Circuit is
        to be used only for Level 2;,
    replaceOnlyWhileDisabled-B;
  PARAMETERS constraintViolation;
REGISTERED AS {ISO10589-ISIS.aoi
        manualL2OnlyMode (72)};
```

l2IntermediateSystemPriority **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO10589-ISIS.IntermediateSystemPriority;
MATCHES FOR Equality, Ordering;
BEHAVIOUR l2IntermediateSystemPriority-B **BEHAVIOUR**
DEFINED AS Priority for becoming LAN Level 2 Designated Intermediate System;;
REGISTERED AS {ISO10589-ISIS.aoi l2IntermediateSystemPriority (73)};

l2CircuitID **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO10589-ISIS.CircuitID;
MATCHES FOR Equality;
BEHAVIOUR l2CircuitID-B **BEHAVIOUR**
DEFINED AS The 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.;;
REGISTERED AS {ISO10589-ISIS.aoi l2CircuitID (74)};

l2DesignatedIntermediateSystem **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO10589-ISIS.SystemID;
MATCHES FOR Equality;
BEHAVIOUR l2DesignatedIntermediateSystem-B **BEHAVIOUR**
DEFINED AS The ID of the LAN Level 2 Designated Intermediate System on this circuit. If, for any reason this system is not partaking in the relevant Designated Intermediate System election process, then the value returned is ;;
REGISTERED AS {ISO10589-ISIS.aoi l2DesignatedIntermediateSystem (75)};

lanL2DesignatedIntermediateSystemChanges **ATTRIBUTE**
DERIVED FROM nonWrappingCounter;
BEHAVIOUR lanL2DesignatedIntermediateSystemChanges-B **BEHAVIOUR**
DEFINED AS Number of LAN L2 Designated Intermediate System Change events generated;;
REGISTERED AS {ISO10589-ISIS.aoi lanL2DesignatedIntermediateSystemChanges (76)};

adjacencyName **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO10589-ISIS.GraphicString;
MATCHES FOR Equality, Substrings;
BEHAVIOUR adjacencyName-B **BEHAVIOUR**
DEFINED AS A string which is the Identifier for the Adjacency and which is unique amongst the set of Adjacencies maintained 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 "Auto" are reserved for Automatic Adjacencies. An attempt to create a Manual Adjacency with such an identifier will cause an exception to be raised;;
REGISTERED AS {ISO10589-ISIS.aoi adjacencyName (77)};

adjacencyState **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO10589-ISIS.AdjacencyState;
MATCHES FOR Equality;
BEHAVIOUR adjacencyState-B **BEHAVIOUR**
DEFINED AS The state of the adjacency;;
REGISTERED AS {ISO10589-ISIS.aoi adjacencyState (78)};

neighbourLANAddress **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO10589-ISIS.LANAddress;
MATCHES FOR Equality;
BEHAVIOUR neighbourLANAddress-B **BEHAVIOUR**
DEFINED AS The MAC address of the neighbour system on a broadcast circuit;,
replaceOnlyWhileDisabled-B;
PARAMETERS constraintViolation;
REGISTERED AS {ISO10589-ISIS.aoi neighbourLANAddress (79)};

neighbourSystemType **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO10589-ISIS.NeighbourSystemType;
MATCHES FOR Equality;
BEHAVIOUR neighbourSystemType-B **BEHAVIOUR**
DEFINED AS The type of the neighbour system one of: Unknown End system Intermediate system L1 Intermediate system L2 Intermediate system;;
REGISTERED AS {ISO10589-ISIS.aoi neighbourSystemType (80)};

sNPAAddress **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO10589-ISIS.SNPAAddress;
MATCHES FOR Equality;
BEHAVIOUR sNPAAddress-B **BEHAVIOUR**
DEFINED AS The SNPA Address of the neighbour system on an X.25 circuit;,
replaceOnlyWhileDisabled-B;
PARAMETERS constraintViolation;
REGISTERED AS {ISO10589-ISIS.aoi sNPAAddress (81)};

adjacencyUsageType **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.AdjacencyUsageType;
MATCHES FOR Equality;
BEHAVIOUR level-B **BEHAVIOUR**
DEFINED AS The usage of the Adjacency. An Adjacency of type Level 1" will be used for Level 1 traffic only. An adjacency of type Level 2" will be used for Level 2 traffic only. An adjacency of type Level 1 and 2" will be used for both Level 1 and Level 2 traffic. There may be two adjacencies (of types Level 1" and Level 2" between the same pair of Intermediate Systems.;;
REGISTERED AS {ISO10589-ISIS.aoi adjacencyUsageType (82)};

neighbourSystemID **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.SystemID;
MATCHES FOR Equality;
BEHAVIOUR neighbourSystemID-B **BEHAVIOUR**
DEFINED AS 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.;;
REGISTERED AS {ISO10589-ISIS.aoi neighbourSystemID (83)};

neighbourAreas **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.AreaAddresses;
MATCHES FOR Equality, Set Comparison, Set Intersection;
BEHAVIOUR neighbourAreas-B **BEHAVIOUR**
DEFINED AS This contains the Area Addresses of a neighbour Intermediate System from the IIH PDU.;;
REGISTERED AS {ISO10589-ISIS.aoi neighbourAreas (84)};

holdingTimer **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.HoldingTimer;
MATCHES FOR Equality, Ordering;
BEHAVIOUR holdingTimer-B **BEHAVIOUR**
DEFINED AS Holding time for this adjacency updated from the IIH PDUs;;
REGISTERED AS {ISO10589-ISIS.aoi holdingTimer (85)};

lANPriority **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.IntermediateSystemPriority;
MATCHES FOR Equality, Ordering;
BEHAVIOUR lANPriority-B **BEHAVIOUR**
DEFINED AS 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;;
REGISTERED AS {ISO10589-ISIS.aoi lANPriority (86)};

endSystemIDs **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.EndSystemIDs;
MATCHES FOR Equality, Set Comparison, Set Intersection;
BEHAVIOUR endSystemIDs-B **BEHAVIOUR**
DEFINED AS 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.;;
REGISTERED AS {ISO10589-ISIS.aoi endSystemIDs (87)};

networkEntityTitle **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.NetworkEntityTitle;
MATCHES FOR Equality, Ordering;
BEHAVIOUR networkEntityTitle-B **BEHAVIOUR**
DEFINED AS The Network entity Title which is the destination of a Virtual link being used to repair a partitioned Level 1 area (see clause 7.2.10);;
REGISTERED AS {ISO10589-ISIS.aoi networkEntityTitle (88)};

metric **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.PathMetric;
MATCHES FOR Equality, Ordering;
BEHAVIOUR metric-B **BEHAVIOUR**
DEFINED AS Cost of least cost L2 path(s) to destination area based on the default metric;;
REGISTERED AS {ISO10589-ISIS.aoi metric (89)};

defaultMetricPathCost **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.PathMetric;
MATCHES FOR Equality, Ordering;
BEHAVIOUR defaultMetricPathCost-B **BEHAVIOUR**
DEFINED AS Cost of least cost path(s) using the default metric to destination;;
REGISTERED AS {ISO10589-ISIS.aoi defaultMetricPathCost (90)};

defaultMetricOutputAdjacencies **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.OutputAdjacencies;
MATCHES FOR Equality, Set Comparison, Set Intersection;
BEHAVIOUR defaultMetricOutputAdjacencies-B **BEHAVIOUR**
DEFINED AS The set of Adjacency (or Reachable Address) managed object identifiers representing the forwarding decisions based upon the default metric for the destination;;
REGISTERED AS {ISO10589-ISIS.aoi defaultMetricOutputAdjacencies (91)};

delayMetricPathCost **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.PathMetric;
MATCHES FOR Equality, Ordering;

BEHAVIOUR delayMetricPathCost-B **BEHAVIOUR**
DEFINED AS Cost of least cost path(s) using the delay metric to destination;;
REGISTERED AS {ISO10589-ISIS.aoi delayMetricPathCost (92)};

delayMetricOutputAdjacencies **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO10589-ISIS.OutputAdjacencies;
MATCHES FOR Equality, Set Comparison, Set Intersection;
BEHAVIOUR delayMetricOutputAdjacencies-B **BEHAVIOUR**
DEFINED AS The set of Adjacency (or Reachable Address) managed object identifiers representing the forwarding decisions based upon the delay metric for the destination;;
REGISTERED AS {ISO10589-ISIS.aoi delayMetricOutputAdjacencies (93)};

expenseMetricPathCost **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO10589-ISIS.PathMetric;
MATCHES FOR Equality, Ordering;
BEHAVIOUR expenseMetricPathCost-B **BEHAVIOUR**
DEFINED AS Cost of least cost path(s) using the expense metric to destination;;
REGISTERED AS {ISO10589-ISIS.aoi expenseMetricPathCost (94)};

expenseMetricOutputAdjacencies **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO10589-ISIS.OutputAdjacencies;
MATCHES FOR Equality, Set Comparison, Set Intersection;
BEHAVIOUR expenseMetricOutputAdjacencies-B **BEHAVIOUR**
DEFINED AS The set of Adjacency (or Reachable Address) managed object identifiers representing the forwarding decisions based upon the expense metric for the destination;;
REGISTERED AS {ISO10589-ISIS.aoi expenseMetricOutputAdjacencies (95)};

errorMetricPathCost **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO10589-ISIS.PathMetric;
MATCHES FOR Equality, Ordering;
BEHAVIOUR errorMetricPathCost-B **BEHAVIOUR**
DEFINED AS Cost of least cost path(s) using the error metric to destination;;
REGISTERED AS {ISO10589-ISIS.aoi errorMetricPathCost (96)};

errorMetricOutputAdjacencies **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO10589-ISIS.OutputAdjacencies;
MATCHES FOR Equality, Set Comparison, Set Intersection;
BEHAVIOUR errorMetricOutputAdjacencies-B **BEHAVIOUR**
DEFINED AS The set of Adjacency (or Reachable Address) managed object identifiers representing the forwarding decisions based upon the error metric for the destination;;
REGISTERED AS {ISO10589-ISIS.aoi errorMetricOutputAdjacencies (97)};

addressPrefix **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO10589-ISIS.AddressPrefix;
MATCHES FOR Equality, Substrings;
BEHAVIOUR addressPrefix-B **BEHAVIOUR**
DEFINED AS An Area Address (or prefix) of a destination area;;
REGISTERED AS {ISO10589-ISIS.aoi addressPrefix (98)};

defaultMetric **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO10589-ISIS.HopMetric;
MATCHES FOR Equality, Ordering;
BEHAVIOUR defaultMetric-B **BEHAVIOUR**
DEFINED AS 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 clause 8.3.5.4 must be taken. The value of zero is reserved to indicate that this metric is not supported;;
REGISTERED AS {ISO10589-ISIS.aoi defaultMetric (99)};

delayMetric **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO10589-ISIS.HopMetric;
MATCHES FOR Equality, Ordering;
BEHAVIOUR delayMetric-B **BEHAVIOUR**
DEFINED AS The delay metric value for reaching the specified prefix over this Circuit.**BEHAVIOUR**If this attribute is changed while both the Reachable Address and the Circuit are Enabled (i.e. state On), the actions described in clause 8.3.5.4 must be taken. The value of zero is reserved to indicate that this metric is not supported;;
REGISTERED AS {ISO10589-ISIS.aoi delayMetric (100)};

expenseMetric **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO10589-ISIS.HopMetric;
MATCHES FOR Equality, Ordering;
BEHAVIOUR expenseMetric-B **BEHAVIOUR**
DEFINED AS 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 clause 8.3.5.4 must be taken. The value of zero is reserved to indicate that this metric is not supported;;
REGISTERED AS {ISO10589-ISIS.aoi expenseMetric (101)};

errorMetric **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.HopMetric;
MATCHES FOR Equality, Ordering;
BEHAVIOUR errorMetric-B **BEHAVIOUR**
DEFINED AS 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 clause 8.3.5.4 must be taken. The value of zero is reserved to indicate that this metric is not supported;;
REGISTERED AS {ISO10589-ISIS.aoi errorMetric (102)};

defaultMetricType **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.MetricType;
MATCHES FOR Equality;
BEHAVIOUR defaultMetricType-B **BEHAVIOUR**
DEFINED AS Indicates whether the default metric is internal or external;;
REGISTERED AS {ISO10589-ISIS.aoi defaultMetricType (103)};

delayMetricType **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.MetricType;
MATCHES FOR Equality;
BEHAVIOUR delayMetricType-B **BEHAVIOUR**
DEFINED AS Indicates whether the delay metric is internal or external;;
REGISTERED AS {ISO10589-ISIS.aoi delayMetricType (104)};

expenseMetricType **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.MetricType;
MATCHES FOR Equality;
BEHAVIOUR expenseMetricType-B **BEHAVIOUR**
DEFINED AS Indicates whether the expense metric is internal or external;;
REGISTERED AS {ISO10589-ISIS.aoi expenseMetricType (105)};

errorMetricType **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.MetricType;
MATCHES FOR Equality;
BEHAVIOUR errorMetricType-B **BEHAVIOUR**
DEFINED AS Indicates whether the error metric is internal or extternal;;
REGISTERED AS {ISO10589-ISIS.aoi errorMetricType (106)};

mappingType **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.MappingType;
MATCHES FOR Equality;
BEHAVIOUR mappingType-B **BEHAVIOUR**
DEFINED AS 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;;
REGISTERED AS {ISO10589-ISIS.aoi mappingType (107)};

lANAddress **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.LANAddress;
MATCHES FOR Equality;
BEHAVIOUR lANAddress-B **BEHAVIOUR**
DEFINED AS 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;;
REGISTERED AS {ISO10589-ISIS.aoi lANAddress (108)};

sNPAAddresses **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.SNPAAddresses;
MATCHES FOR Equality;
BEHAVIOUR sNPAAddresses-B **BEHAVIOUR**
DEFINED AS 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;;
REGISTERED AS {ISO10589-ISIS.aoi sNPAAddresses (109)};

nonWrappingCounter **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.NonWrappingCounter;
MATCHES FOR Equality, Ordering;
BEHAVIOUR nonWrappingCounter-B **BEHAVIOUR**
DEFINED AS Non-replaceable, non-wrapping counter;;
-- This attibute is only defined in order to allow other counter attributes to be derived from it.
REGISTERED AS {ISO10589-ISIS.aoi nonWrappingCounter (110)};

areaTransmitPassword **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX
ISO10589-ISIS.Password;
MATCHES FOR Equality;
BEHAVIOUR areaTransmitPassword-B **BEHAVIOUR**
DEFINED AS The value to be used as a transmit password in Level 1 LSP, and SNP PDUs transmitted by this Intermediate System;;
REGISTERED AS {ISO10589-ISIS.aoi areaTransmitPassword (111)};

```
areaReceivePasswords ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.Passwords;
  MATCHES FOR Equality;
  BEHAVIOUR areaReceivePasswords-B BEHAVIOUR
    DEFINED AS The values to be used as receive passwords to check the receipt of Level 1 LSP, and SNP PDUs;;
REGISTERED AS {ISO10589-ISIS.aoi
        areaReceivePasswords (112)};


domainTransmitPassword ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.Password;
  MATCHES FOR Equality;
  BEHAVIOUR domainTransmitPassword-B
        BEHAVIOUR
    DEFINED AS The value to be used as a transmit password in Level 2 LSP, and SNP PDUs transmitted by this Intermediate System;;
REGISTERED AS {ISO10589-ISIS.aoi
        domainTransmitPassword (113)};


domainReceivePasswords ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.Passwords;
  MATCHES FOR Equality;
  BEHAVIOUR domainReceivePasswords-B
        BEHAVIOUR
    DEFINED AS The values to be used as receive passwords to check the receipt of Level 2 LSP, and SNP PDUs;;
REGISTERED AS {ISO10589-ISIS.aoi
        domainReceivePasswords (114)};


circuitTransmitPassword ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.Password;
  MATCHES FOR Equality;
  BEHAVIOUR circuitTransmitPassword-B
        BEHAVIOUR
    DEFINED AS The value to be used as a transmit password in IIH PDUs transmitted by this Intermediate System;;
REGISTERED AS {ISO10589-ISIS.aoi
        circuitTransmitPassword (115)};


circuitReceivePasswords ATTRIBUTE
  WITH ATTRIBUTE SYNTAX
        ISO10589-ISIS.Passwords;
  MATCHES FOR Equality;
  BEHAVIOUR circuitReceivePasswords-B
        BEHAVIOUR
    DEFINED AS The values to be used as receive passwords to check the receipt of IIH PDUs;;
REGISTERED AS {ISO10589-ISIS.aoi
        circuitReceivePasswords (116)};


authenticationFailures ATTRIBUTE
  DERIVED FROM nonWrappingCounter;
  BEHAVIOUR authenticationFailures-B BEHAVIOUR
    DEFINED AS Count of authentication Failure notifications generated;;
REGISTERED AS {ISO10589-ISIS.aoi
        authenticationFailures (117)};
```

11.2.12 Notification Definitions

```
-- Note pduFormatError notification now included in
        Network layer definitions
corruptedLSPDetected NOTIFICATION
  BEHAVIOUR corruptedLSPDetected-B BEHAVIOUR
    DEFINED AS The Corrupted LSP Detected Notification is generated when a corrupted Link State PDU is detected in memory.  The occurance of this event is counted by the corruptedLSPsDetected counter.;;
  MODE NON-CONFIRMED;
  WITH INFORMATION SYNTAX
        ISO10589-ISIS.NotificationInfo;
REGISTERED AS {ISO10589-ISIS.noi
        corruptedLSPDetected (1)};


lSPL1DatabaseOverload NOTIFICATION
  BEHAVIOUR lSPL1DatabaseOverload-B
        BEHAVIOUR
    DEFINED AS The LSP L1 Database Overload Notification 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.  The occurance of this event is counted by the lSPL1DatabaseOverloads counter.;;
  MODE NON-CONFIRMED;
  PARAMETERS
    notificationOverloadStateChange,
    notificationSourceID;
  WITH INFORMATION SYNTAX
        ISO10589-ISIS.NotificationInfo;
REGISTERED AS {ISO10589-ISIS.noi
        lSPL1DatabaseOverload (2)};


manualAddressDroppedFromArea NOTIFICATION
  BEHAVIOUR manualAddressDroppedFromArea-B
        BEHAVIOUR
    DEFINED AS The Manual Address Dropped From Area Notification 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). The occurance of this event is counted by the manualAddressDroppedFromAreas counter.;;
  MODE NON-CONFIRMED;
  PARAMETERS
    notificationAreaAddress;
  WITH INFORMATION SYNTAX
```

ISO10589-ISIS.NotificationInfo;
REGISTERED AS {ISO10589-ISIS.noi manualAddressDroppedFromArea (3)};

attemptToExceedMaximumSequenceNumber **NOTIFICATION**
BEHAVIOUR attemptToExceedMaximumSequenceNumber-B **BEHAVIOUR**
DEFINED AS The Attempt To Exceed Maximum Sequence Number Notification 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 the operation of the Routeing state machine shall be disabled for at least (MaxAge + ZeroAgeLifetime) seconds. The occurance of this event is counted by the attemptsToExceedMaximumSequenceNumber counter.;;
MODE NON-CONFIRMED;
WITH INFORMATION SYNTAX ISO10589-ISIS.NotificationInfo;
REGISTERED AS {ISO10589-ISIS.noi attemptToExceedMaximumSequenceNumber (4)};

sequenceNumberSkip **NOTIFICATION**
BEHAVIOUR sequenceNumberSkip-B **BEHAVIOUR**
DEFINED AS The Sequence Number Skipped Notification is generated when the sequence number of an LSP is incremented by more than one. The occurance of this event is counted by the sequenceNumberSkips counter.;;
MODE NON-CONFIRMED;
WITH INFORMATION SYNTAX ISO10589-ISIS.NotificationInfo;
REGISTERED AS {ISO10589-ISIS.noi sequenceNumberSkip (5)};

ownLSPPurge **NOTIFICATION**
BEHAVIOUR ownLSPPurge-B **BEHAVIOUR**
DEFINED AS The Own LSP Purged Notification 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. The occurance of this event is counted by the ownLSPPurges counter.;;
MODE NON-CONFIRMED;
WITH INFORMATION SYNTAX ISO10589-ISIS.NotificationInfo;
REGISTERED AS {ISO10589-ISIS.noi ownLSPPurge (6)};

partitionVirtualLinkChange **NOTIFICATION**
BEHAVIOUR partitionVirtualLinkChange-B **BEHAVIOUR**
DEFINED AS The Partition Virtual Link Change Notification 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. The occurance of this event is counted by the partitionVirtualLinkChanges counter.;;
MODE NON-CONFIRMED;
PARAMETERS
notificationVirtualLinkChange,
notificationVirtualLinkAddress;
WITH INFORMATION SYNTAX ISO10589-ISIS.NotificationInfo;
REGISTERED AS {ISO10589-ISIS.noi partitionVirtualLinkChange (7)};

lSPL2DatabaseOverload **NOTIFICATION**
BEHAVIOUR lSPL2DatabaseOverload-B **BEHAVIOUR**
DEFINED AS The LSP L2 Database Overload Notification 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. The occurance of this event is counted by the lSPL2DatabaseOverloads counter.;;
MODE NON-CONFIRMED;
PARAMETERS
notificationOverloadStateChange,
notificationSourceID;
WITH INFORMATION SYNTAX ISO10589-ISIS.NotificationInfo;
REGISTERED AS {ISO10589-ISIS.noi lSPL2DatabaseOverload (8)};

iDFieldLengthMismatch **NOTIFICATION**
BEHAVIOUR iDFieldLengthMismatch-B **BEHAVIOUR**
DEFINED AS The iDFieldLengthMismatch Notification is generated when a PDU is received with a different value for ID field length to that of the receiving Intermediate system. The occurance of this event is counted by the iDFieldLengthMismatches counter.;;
MODE NON-CONFIRMED;
PARAMETERS
notificationIDLength,
notificationSourceID;
WITH INFORMATION SYNTAX ISO10589-ISIS.NotificationInfo;
REGISTERED AS {ISO10589-ISIS.noi iDFieldLengthMismatch (9)};

circuitChange **NOTIFICATION**
BEHAVIOUR circuitChange-B **BEHAVIOUR**
DEFINED AS The Circuit Change Notification 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. The occurance of this event is counted by the circuitChanges counter.;;
MODE NON-CONFIRMED;
PARAMETERS
notificationNewCircuitState;
WITH INFORMATION SYNTAX ISO10589-ISIS.NotificationInfo;
REGISTERED AS {ISO10589-ISIS.noi circuitChange (10)};

```
adjacencyStateChange NOTIFICATION
  BEHAVIOUR adjacencyStateChange-B BEHAVIOUR
    DEFINED AS The Adjacency State Change Notification 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.  The occurance of this event is counted by the adjacencyStateChanges counter.;;
  MODE NON-CONFIRMED;
  PARAMETERS
    notificationAdjacentSystem,
    notificationNewAdjacencyState,
    notificationReason,
    notificationPDUHeader,
    notificationCalledAddress,
    notificationVersion;
  WITH INFORMATION SYNTAX
        ISO10589-ISIS.NotificationInfo;
REGISTERED AS {ISO10589-ISIS.noi
        adjacencyStateChange (11)};


initializationFailure NOTIFICATION
  BEHAVIOUR initializationFailure-B BEHAVIOUR
    DEFINED AS The Initialisation Failure Notification 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.  The occurance of this event is counted by the initializationFailures counter.;;
  MODE NON-CONFIRMED;
  PARAMETERS
    notificationAdjacentSystem,
    notificationReason,
    notificationPDUHeader,
    notificationCalledAddress,
    notificationVersion;
  WITH INFORMATION SYNTAX
        ISO10589-ISIS.NotificationInfo;
REGISTERED AS {ISO10589-ISIS.noi
        initializationFailure (12)};


rejectedAdjacency NOTIFICATION
  BEHAVIOUR rejectedAdjacency-B BEHAVIOUR
    DEFINED AS The Rejected Adjacency Notification is generated when an attempt to create a new adjacency is rejected, because of a lack of resources. The occurance of this event is counted by the rejectedAdjacencies counter.;;
  MODE NON-CONFIRMED;
  PARAMETERS
    notificationAdjacentSystem,
    notificationReason,
    notificationPDUHeader,
    notificationCalledAddress,
    notificationVersion;
  WITH INFORMATION SYNTAX
        ISO10589-ISIS.NotificationInfo;
REGISTERED AS {ISO10589-ISIS.noi
        rejectedAdjacency (13)};


lanL1DesignatedIntermediateSystemChange
        NOTIFICATION
  BEHAVIOUR
        lanL1DesignatedIntermediateSystemChange-B
        BEHAVIOUR
    DEFINED AS The LAN L1 Designated Intermediate System Change Notification 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. The occurance of this event is counted by the lanL1DesignatedIntermediateSystemChanges counter.;;
  MODE NON-CONFIRMED;
  PARAMETERS
    notificationDesignatedIntermediateSystemChange;
  WITH INFORMATION SYNTAX
        ISO10589-ISIS.NotificationInfo;
REGISTERED AS {ISO10589-ISIS.noi
        lanL1DesignatedIntermediateSystemChange (14)};


exceededMaximumSVCAdjacencies NOTIFICATION
  BEHAVIOUR exceededMaximumSVCAdjacencies-B
        BEHAVIOUR
    DEFINED AS The Exceeded Maximum SVC Adjacencies Notification is generated when there is no free adjacency on which to establish an SVC for a new destination.(see clause 8.3.2.3)  The occurance of this event is counted by the timesExceededMaximumSVCAdjacencies counter.;;
  MODE NON-CONFIRMED;
  WITH INFORMATION SYNTAX
        ISO10589-ISIS.NotificationInfo;
REGISTERED AS {ISO10589-ISIS.noi
        exceededMaximumSVCAdjacencies (15)};


exceededMaximumCallAttempts NOTIFICATION
  BEHAVIOUR exceededMaximumCallAttempts-B
        BEHAVIOUR
    DEFINED AS The Exceeded Maximum Call Attempts Notification is generated when recallCount becomes equal to maximumCallAttempts.  The occurance of this event is counted by the timesExceededMaximumCallAttempts counter.;;
  MODE NON-CONFIRMED;
  WITH INFORMATION SYNTAX
        ISO10589-ISIS.NotificationInfo;
REGISTERED AS {ISO10589-ISIS.noi
        exceededMaximumCallAttempts (16)};


lanL2DesignatedIntermediateSystemChange
        NOTIFICATION
  BEHAVIOUR
        lanL2DesignatedIntermediateSystemChange-B
        BEHAVIOUR
    DEFINED AS The LAN L2 Designated Intermediate System Change Notification 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. The occurance of this event is counted by the lanL2DesignatedIntermediateSystemChanges counter.;;
  MODE NON-CONFIRMED;
```

PARAMETERS
notificationDesignatedIntermediateSystemChange;
WITH INFORMATION SYNTAX
ISO10589-ISIS.NotificationInfo;
REGISTERED AS {ISO10589-ISIS.noi lanL2DesignatedIntermediateSystemChange (17)};

authenticationFailure **NOTIFICATION**
BEHAVIOUR authenticationFailure-B **BEHAVIOUR**
DEFINED AS Generated when a PDU is received with an incorrect Authentication information field;;
MODE NON-CONFIRMED;
PARAMETERS
notificationAdjacentSystem;
WITH INFORMATION SYNTAX
ISO10589-ISIS.NotificationInfo;
REGISTERED AS {ISO10589-ISIS.noi authenticationFailure (18)};

11.2.13 Action Definitions

-- Note: The following actions have been proposed (in SC21 N4977) for inclusion in DMI. Until such time as this is completed, the definitions of these actions are given here.
--
activate **ACTION**
BEHAVIOUR activate-B **BEHAVIOUR**
DEFINED AS Sets OperationalState to ‘enabled’ and commences operation;;
MODE CONFIRMED;
PARAMETERS successResponse, failureResponse, failureReason;
WITH INFORMATION SYNTAX
ISO10589-ISIS.ActionInfo;
WITH REPLY SYNTAX ISO10589-ISIS.ActionReply;
REGISTERED AS {ISO10589-ISIS.acoi activate (1)};

deactivate **ACTION**
BEHAVIOUR deactivate-B **BEHAVIOUR**
DEFINED AS Sets OperationalState to ‘disabled’ and ceases operation;;
MODE CONFIRMED;
PARAMETERS successResponse, failureResponse, failureReason;
WITH INFORMATION SYNTAX
ISO10589-ISIS.ActionInfo;
WITH REPLY SYNTAX ISO10589-ISIS.ActionReply;
REGISTERED AS {ISO10589-ISIS.acoi deactivate (2)};

11.2.14 Parameter Definitions

iSO10589-NB-p1 **PARAMETER**
CONTEXT CREATE-INFO;
WITH SYNTAX ISO10589-ISIS.ISType;
BEHAVIOUR iSO10589-NB-p1-B **BEHAVIOUR**
DEFINED AS The value to be given to the iStype attribute on MO creation. This parameter is mandatory;;
REGISTERED AS {ISO10589-ISIS.proi iSO10589-NB-p1 (1)};

iSO10589Circuit-MO-p1 **PARAMETER**
CONTEXT CREATE-INFO;
WITH SYNTAX ISO10589-ISIS.CircuitType;
BEHAVIOUR iSO10589Circuit-MO-p1-B **BEHAVIOUR**
DEFINED AS The value to be given to the type attribute on MO creation. This parameter is mandatory;;
REGISTERED AS {ISO10589-ISIS.proi iSO10589Circuit-MO-p1 (2)};

reachableAddressP1 **PARAMETER**
CONTEXT CREATE-INFO;
WITH SYNTAX ISO10589-ISIS.AddressPrefix;
BEHAVIOUR reachableAddressp1-B **BEHAVIOUR**
DEFINED AS The value to be given to the addressPrefix attribute on MO creation. This parameter is mandatory;;
REGISTERED AS {ISO10589-ISIS.proi reachableAddressP1 (3)};

reachableAddressP2 **PARAMETER**
CONTEXT CREATE-INFO;
WITH SYNTAX ISO10589-ISIS.MappingType;
BEHAVIOUR reachableAddressp2-B **BEHAVIOUR**
DEFINED AS The value to be given to the mappingType attribute on MO creation. This parameter is only permitted when the ‘type’ of the parent circuit is either ‘broadcast’ or ‘DA’. In those cases the default value is ‘manual’;;
REGISTERED AS {ISO10589-ISIS.proi reachableAddressP2 (4)};

manualAdjacencyP1 **PARAMETER**
CONTEXT CREATE-INFO;
WITH SYNTAX ISO10589-ISIS.LANAddress;
BEHAVIOUR manualAdjacencyP1-B **BEHAVIOUR**
DEFINED AS The value to be given to the lANAddress attribute on MO creation;;
REGISTERED AS {ISO10589-ISIS.proi manualAdjacencyP1 (5)};

manualAdjacencyP2 **PARAMETER**
CONTEXT CREATE-INFO;
WITH SYNTAX ISO10589-ISIS.EndSystemIDs;
BEHAVIOUR manualAdjacencyP2-B **BEHAVIOUR**
DEFINED AS The value to be given to the endSystemIDs attribute on MO creation;;
REGISTERED AS {ISO10589-ISIS.proi manualAdjacencyP2 (6)};

successResponse **PARAMETER**
CONTEXT ACTION-REPLY;
WITH SYNTAX ISO10589-ISIS.ResponseCode;
BEHAVIOUR successResponse-B **BEHAVIOUR**
DEFINED AS Returned in the responseCode field of an ActionReply when the action has completed successfully.;;
REGISTERED AS {ISO10589-ISIS.proi successResponse (7)};

failureResponse **PARAMETER**
CONTEXT ACTION-REPLY;
WITH SYNTAX ISO10589-ISIS.ResponseCode;

BEHAVIOUR failureResponse-B **BEHAVIOUR**
DEFINED AS Returned in the responseCode field of an ActionReply when the action failed to complete. The failureReason parameter is returned with this responseCode, giving additional information;;
REGISTERED AS {ISO10589-ISIS.proi failureResponse (8)};

failureReason **PARAMETER**
CONTEXT ACTION-REPLY;
WITH SYNTAX ISO10589-ISIS.ActionFailureReason;
BEHAVIOUR failureReason-B **BEHAVIOUR**
DEFINED AS Gives the reason why an entity failed to activate or deactivate.;;
REGISTERED AS {ISO10589-ISIS.proi failureReason (9)};

constraintViolation **PARAMETER**
CONTEXT SPECIFIC-ERROR;
WITH SYNTAX ISO10589-ISIS.ConstraintViolationReason;
BEHAVIOUR constraintViolation-B **BEHAVIOUR**
DEFINED AS The specific error returned on failure of a REPLACE operation when the MO prohibits such operations under certain conditions, for example while the MO is in the disabled operational state.;;
REGISTERED AS {ISO10589-ISIS.proi constraintViolation (10)};

notificationReceivingAdjacency **PARAMETER**
CONTEXT EVENT-INFO;
WITH SYNTAX ISO10589-ISIS.LocalDistinguishedName;
BEHAVIOUR notificationReceivingAdjacency-B **BEHAVIOUR**
DEFINED AS The local managed object name of the adjacency upon which the NPDU was received;;
REGISTERED AS {ISO10589-ISIS.proi notificationReceivingAdjacency (11)};

notificationIDLength **PARAMETER**
CONTEXT EVENT-INFO;
WITH SYNTAX ISO10589-ISIS.IDLength;
BEHAVIOUR notificationIDLength-B **BEHAVIOUR**
DEFINED AS The IDLength specified in the ignored PDU;;
REGISTERED AS {ISO10589-ISIS.proi notificationIDLength (12)};

notificationAreaAddress **PARAMETER**
CONTEXT EVENT-INFO;
WITH SYNTAX ISO10589-ISIS.AreaAddress;
BEHAVIOUR notificationAreaAddress-B **BEHAVIOUR**
DEFINED AS The Area Address which caused MaximumAreaAddresses to be exceeded;;
REGISTERED AS {ISO10589-ISIS.proi notificationAreaAddress (13)};

notificationSourceID **PARAMETER**
CONTEXT EVENT-INFO;
WITH SYNTAX ISO10589-ISIS.SourceID;
BEHAVIOUR notificationSourceID-B **BEHAVIOUR**
DEFINED AS The source ID of the LSP;;
REGISTERED AS {ISO10589-ISIS.proi notificationSourceID (14)};

notificationVirtualLinkChange **PARAMETER**
CONTEXT EVENT-INFO;
WITH SYNTAX ISO10589-ISIS.VirtualLinkChange;
BEHAVIOUR notificationVirtualLinkChange-B **BEHAVIOUR**
DEFINED AS This indicates whether the event was genrated as a result of the creation or deletion of a Virtual Link between two Level 2 Intermediate Systems.;;
REGISTERED AS {ISO10589-ISIS.proi notificationVirtualLinkChange (15)};

notificationVirtualLinkAddress **PARAMETER**
CONTEXT EVENT-INFO;
WITH SYNTAX ISO10589-ISIS.NetworkEntityTitle;
BEHAVIOUR notificationVirtualLinkAddress-B **BEHAVIOUR**
DEFINED AS The Network Entity Title of the Level 2 Intermediate System at the remote end of the virtual link;;
REGISTERED AS {ISO10589-ISIS.proi notificationVirtualLinkAddress (16)};

notificationNewCircuitState **PARAMETER**
CONTEXT EVENT-INFO;
WITH SYNTAX ISO10589-ISIS.NewCircuitState;
BEHAVIOUR notificationNewCircuitState-B **BEHAVIOUR**
DEFINED AS The direction of the Circuit state change specified as the resulting state. i.e. a change from On to Off is specified as Off;;
REGISTERED AS {ISO10589-ISIS.proi notificationNewCircuitState (17)};

notificationNewAdjacencyState **PARAMETER**
CONTEXT EVENT-INFO;
WITH SYNTAX ISO10589-ISIS.NewAdjacencyState;
BEHAVIOUR notificationNewAdjacencyState-B **BEHAVIOUR**
DEFINED AS The direction of the Adjacency state change specified as the resulting state. i.e. a change from Up to Down is specified as Down. Any state other than Up is considered to be Down.;;
REGISTERED AS {ISO10589-ISIS.proi notificationNewAdjacencyState (18)};

notificationAdjacentSystem **PARAMETER**
CONTEXT EVENT-INFO;
WITH SYNTAX ISO10589-ISIS.SystemID;
BEHAVIOUR notificationAdjacentSystem-B **BEHAVIOUR**
DEFINED AS The system ID of the adjacent system;;
REGISTERED AS {ISO10589-ISIS.proi notificationAdjacentSystem (19)};

notificationReason **PARAMETER**
CONTEXT EVENT-INFO;
WITH SYNTAX ISO10589-ISIS.Reason;

BEHAVIOUR notificationReason-B **BEHAVIOUR**
DEFINED AS The associated Reason;;
REGISTERED AS {ISO10589-ISIS.proi
notificationReason (20)};

notificationPDUHeader **PARAMETER**
CONTEXT EVENT-INFO;
WITH SYNTAX ISO10589-ISIS.PDUHeader;
BEHAVIOUR notificationPDUHeader-B **BEHAVIOUR**
DEFINED AS The header of the PDU which caused the notification;;
REGISTERED AS {ISO10589-ISIS.proi
notificationPDUHeader (21)};

notificationCalledAddress **PARAMETER**
CONTEXT EVENT-INFO;
WITH SYNTAX ISO10589-ISIS.SNPAAddress;
BEHAVIOUR notificationCalledAddres-B
BEHAVIOUR
DEFINED AS The SNPA Address which was being called when the Adjacency was taken down as a result of a call reject;;
REGISTERED AS {ISO10589-ISIS.proi
notificationCalledAddress (22)};

notificationVersion **PARAMETER**
CONTEXT EVENT-INFO;
WITH SYNTAX ISO10589-ISIS.Version;
BEHAVIOUR notificationVersion-B **BEHAVIOUR**
DEFINED AS The version number reported by the other system;;
REGISTERED AS {ISO10589-ISIS.proi
notificationVersion (23)};

notificationDesignatedIntermediateSystemChange
PARAMETER
CONTEXT EVENT-INFO;
WITH SYNTAX ISO10589-ISIS.DesignatedISChange;
BEHAVIOUR
notificationDesignatedIntermediateSystemChange-B
BEHAVIOUR
DEFINED AS The direction of the change in Designated Intermediate System status of this system;;
REGISTERED AS {ISO10589-ISIS.proi
notificationDesignatedIntermediateSystemChange
(24)};

notificationOverloadStateChange **PARAMETER**
CONTEXT EVENT-INFO;
WITH SYNTAX ISO10589-ISIS.OverloadStateChange;
BEHAVIOUR notificationOverloadStateChange-B
BEHAVIOUR
DEFINED AS The direction of the change in Overload status;;
REGISTERED AS {ISO10589-ISIS.proi
notificationOverloadStateChange (25)};

11.2.15 Attribute Groups

counters **ATTRIBUTE GROUP**
DESCRIPTION The group of all counters;
REGISTERED AS {ISO10589-ISIS.agoi counters (1)};

11.2.16 Behaviour Definitions

resettingTimer-B **BEHAVIOUR**
DEFINED AS This attribute specifies the interval between certain events in the operation of the protocol state machine. If the value of this attribute is changed to a new value *t* while the protocol state machine is in operation, the implementation shall take the necessary steps to ensure that for any time interval which was in progress when the corresponding attribute was changed, the next expiration of the that interval takes place *t* seconds from the original start of that interval, or immediately, whichever is later. The precision with which this time shall be implemented shall be the same as that associated with the basic operation of the timer attribute;

replaceOnlyWhileDisabled-B **BEHAVIOUR**
DEFINED AS This attribute shall only permit the REPLACE operation to be performed on it while the MO is in the Disabled Operational State. An attempt to perform a REPLACE operation while the MO is in the Enabled Operation State shall fail with the generation of the constraintViolation specific error.;

resourceLimiting-B **BEHAVIOUR**
DEFINED AS This attribute places limits on some ”resource". In general implementations may allocate reources up to this limit when the managed object is enabled and it may be impossible to change the allocation without first disabling and re-enabling the managed object. Therefore this International Standard only requires that it shall be possible to perform a REPLACE operation on this attribute while the MO is disabled. However some implementations may be able to to change the allocation of resources without first disabling the MO. In this case it is permitted to *increase* the value of the atribute at any time, but it shall not be *decreased* below the currently ”used" value of the resource. Where an attempt to perform a REPLACE operation fails either because the MO is enabled, or because an attempt has been made to decrease the value, the REPLACE operation shall fail with the generation of the constraintViolation specific error.;

11.2.17 ASN1 Modules

ISO10589-ISIS{tbd1}
DEFINITIONS ::= **BEGIN**
-- object identifier definitions
sc6 **OBJECT IDENTIFIER** ::= {joint-iso-ccitt sc6(?)}
-- value to be assigned by SC21 secretariat
isisoi **OBJECT IDENTIFIER ::=** {sc6 iSO10589(?)}
-- value to be assigned by SC6 secretariat
moi **OBJECT IDENTIFIER** ::= {isisoi objectClass (3)}
poi **OBJECT IDENTIFIER** ::= {isisoi package (4)}
proi **OBJECT IDENTIFIER** ::= {isisoi parameter (5)}
nboi **OBJECT IDENTIFIER** ::= {isisoi nameBinding (6)}
aoi **OBJECT IDENTIFIER** ::= {isisoi attribute (7)}
agoi **OBJECT IDENTIFIER** ::= {isisoi attributeGroup (8)}
acoi **OBJECT IDENTIFIER** ::= {isisoi action (10)}
noi **OBJECT IDENTIFIER** ::= {isisoi notification (11)}

```
ActionFailureReason ::= ENUMERATED{
    reason1(0),
    reason2(1)}
  -- Note: actual reasons TBS
ActionInfo ::= SET OF Parameter
ActionReply ::= SEQUENCE{
    responseCode OBJECT IDENTIFIER,
    responseArgs SET OF Parameter OPTIONAL}
AddressPrefix ::=  OCTETSTRING(SIZE(0..20))
AdjacencyState ::= ENUMERATED{
    initializing(0),
    up(1),
    failed(2)}-- was 4 in N5821 , is it required at all?
AreaAddress ::=  OCTETSTRING(SIZE(1..20))
AreaAddresses ::= SET OF AreaAddress
Boolean ::= BOOLEAN
CircuitID ::=  OCTETSTRING(SIZE(1..10))
CompleteSNPInterval ::= INTEGER(1..600)
ConstraintViolationReason ::= OBJECT IDENTIFIER;
DRISISHelloTimer ::= INTEGER(1..65535)
DatabaseState ::= ENUMERATED{
    off(0),
    on(1),
    waiting(2)}
DesignatedISChange ::= ENUMERATED{
    resigned(0),
    elected(1)}
DefaultESHelloTimer ::= INTEGER(1..65535)
EndSystemIDs ::= SET OF SystemID
GraphicString ::=  GRAPHICSTRING
HelloTimer ::= INTEGER(1..65535)
HoldingTimer ::= INTEGER(1..65535)
HopMetric ::= INTEGER(0..63)
ISISHelloTimer ::= INTEGER(1..65535)
IDLength ::= INTEGER(0..9)
IdleTimer ::= INTEGER(1..65535)
InitialMinimumTimer ::= INTEGER(1..65535)
IntermediateSystemPriority ::= INTEGER(1..127)
ISType ::= ENUMERATED{
    level1IS(1),
    level2IS(2)}
LANAddress ::=  OCTETSTRING(SIZE(6))
AdjacencyUsageType::= ENUMERATED{
  undefined(0),
  level1(1),
  level2(2),
  level1and2(3)}
LocalDistinguishedName ::= CMIP-1.ObjectInstance
  -- A suitable free standing definition is requred
LSPID ::=  OCTETSTRING(SIZE(2..11))
MappingType ::= ENUMERATED{
    manual(0),
    x121(1)}
MaximumBuffers ::= INTEGER(1..65535)
MaximumCallAttempts ::= INTEGER(1..65535)
MaximumLSPGenerationInterval ::= INTEGER(1..65535)
MaximumPathSplits ::= INTEGER(1..32)
MaximumSVCAdjacencies ::= INTEGER(1..65535)
MaximumVirtualAdjacencies ::= INTEGER(0..32)
MetricIncrement ::= INTEGER(0..63)
MetricType ::= ENUMERATED{
    internal(0),
    external(1)}
MinimumBroadcastLSPTransmissionInterval ::=
        INTEGER(1..65535)
MinimumLSPGenerationInterval ::= INTEGER(1..65535)
MinimumLSPTransmissionInterval ::=
        INTEGER(1..65535)
NeighbourSystemType ::= ENUMERATED{
    unknown(0),
    endSystem(1),
    intermediateSystem(2),
    l1IntermediateSystem(3),
    l2IntermediateSystem(4)}
NetworkEntityTitle ::=  OCTETSTRING(SIZE(1..19))
NewAdjacencyState ::= ENUMERATED{
    down(0),
    up(1)}
NewCircuitState ::= ENUMERATED{
    off(0),
    on(1)}
NonWrappingCounter ::= INTEGER(0..264-1)
NotificationInfo ::= SET OF Parameter
NSAPAddress ::=  OCTETSTRING(SIZE(1..20))
OctetString ::=  OCTETSTRING
OriginatingLSPBufferSize ::= INTEGER(512..1492)
OutputAdjacencies ::= SET OF LocalDistinguishedName
OverloadStateChange ::= ENUMERATED{
  on(0),
  waiting(1)}
Parameter ::= SEQUENCE{
    paramIdOBJECT IDENTIFIER,
    paramInfoANY DEFINED BY paramID}
PartialSNPInterval ::= INTEGER(1..65535)
Password ::=  OCTETSTRING(SIZE(0..254)
Passwords ::= SET OF Password
PathMetric ::= INTEGER(0..1023)
PDUHeader ::=  OCTETSTRING(SIZE(0..255))
PollESHelloRate ::= INTEGER(1..65535)
Reason ::= ENUMERATED{
  holdingTimerExpired(0),
  checksumError(1),
  oneWayConnectivity(2),
  callRejected(3),
  reserveTimerExpired(4),
  circuitDisabled(5),
  versionSkew(6),
  areaMismatch(7),
  maximumBroadcastIntermediateSystemsExceeded(8),
  maximumBroadcastEndSystemsExceeded(9),
  wrongSystemType(10)}
ResponseCode ::= OBJECT IDENTIFIER
RecallTimer ::= INTEGER(1..65535)
ReserveTimer ::= INTEGER(1..65535)
SNPAAddress ::=
        NUMERICSTRING(FROM("0"|"1"|"2"|"3"|"4"|"5"|
        "6"|"7"|"8"|"9"))(SIZE(0..15))
  -- Up to 15 Digits 0..9
SNPAAddresses ::= SET OF SNPAAddress
CircuitType ::= ENUMERATED{
  broadcast(0),
  ptToPt(1),
  staticIN(2),
  staticOut(3),
  dA(4)}
SourceID ::=  OCTETSTRING(SIZE(1..10))
SystemID ::=  OCTETSTRING(SIZE(0..9))
VirtualLinkChange ::= ENUMERATED{
    deleted(0),
    created(1)}
Version ::=  GRAPHICSTRING
WaitingTime ::= INTEGER(1..65535)
maximumPathSplits-Default INTEGER ::= 2
MaximumPathSplits-Permitted ::= INTEGER(1..32)
```

```
maximumBuffers-Default INTEGER ::= ImpSpecific
MaximumBuffers-Permitted ::= INTEGER(1..ImpSpecific)
minimumLSPTransmissionInterval-Default INTEGER ::=
        5
MinimumLSPTransmissionInterval-Permitted ::=
        INTEGER(5..30)
maximumLSPGenerationInterval-Default INTEGER ::=
        900
MaximumLSPGenerationInterval-Permitted ::=
        INTEGER(60..900)
minimumBroadcastLSPTransmissionInterval-Default
        INTEGER ::=33
MinimumBroadcastLSPTransmissionInterval-Permitted ::=
        INTEGER(1..65535)
completeSNPInterval-Default INTEGER ::= 10
CompleteSNPInterval-Permitted ::= INTEGER(1..600)
originatingL1LSPBufferSize-Default INTEGER ::=
        receiveLSPBufferSize
OriginatingL1LSPBufferSize-Permitted ::=
        INTEGER(512..receiveLSPBufferSize)
manualAreaAddresses-Default AreaAddresses ::= {}
ManualAreaAddresses-Permitted ::= AreaAddresses
        (SIZE(0..MaximumAreaAddresses))
minimumLSPGenerationInterval-Default INTEGER ::= 30
MinimumLSPGenerationInterval-Permitted ::=
        INTEGER(5..300)
defaultESHelloTime-Default INTEGER ::= 600
DefaultESHelloTime-Permitted ::= INTEGER(1..65535)
pollESHelloRate-Default INTEGER ::= 50
PollESHelloRate-Permitted ::= INTEGER(1..65535)
partialSNPInterval-Default INTEGER ::= 2
PartialSNPInterval-Permitted ::= INTEGER(1..65535)
waitingTime-Default INTEGER ::= 60
WaitingTime-Permitted ::= INTEGER(1..65535)
dRISISHelloTimer-Default INTEGER ::= 1
DRISISHelloTimer-Permitted ::=  INTEGER(1..65535)
originatingL2LSPBufferSize-Default INTEGER ::=
        receiveLSPBufferSize
OriginatingL2LSPBufferSize-Permitted ::=
        INTEGER(512..receiveLSPBufferSize)
maximumVirtualAdjacencies-Default INTEGER ::= 2
MaximumVirtualAdjacencies-Permitted ::=
        INTEGER(0..32)
helloTimer-Default INTEGER ::= 10
HelloTimer-Permitted ::= INTEGER(1..21845)
defaultMetric-Default INTEGER ::= 20
DefaultMetric-Permitted ::= INTEGER(1..MaxLinkMetric)
optionalMetric-Default INTEGER ::= 0
OptionalMetric-Permitted ::=
        INTEGER(0..MaxLinkMetric)
metricType-Default MetricType ::= Internal
iSISHelloTimer-Default INTEGER ::= 3
ISISHelloTimer-Permitted ::= INTEGER(1..21845)
externalDomain-Default BOOLEAN ::= TRUE
l1IntermediateSystemPriority-Default INTEGER ::= 64
L1IntermediateSystemPriority-Permitted ::=
        INTEGER(1..127)
callEstablishmentMetricIncrement-Default INTEGER ::= 0
CallEstablishmentMetricIncrement-Permitted ::=
        INTEGER(0..MaxLinkMetric)
idleTimer-Default INTEGER ::= 30
IdleTimer-Permitted ::= INTEGER(0..65535)
initialMinimumTimer-Default INTEGER ::= 55
InitialMinimumTimer-Permitted ::= INTEGER(1..65535)
reserveTimer-Default INTEGER ::= 600
ReserveTimer-Permitted ::= INTEGER(1..65535)
maximumSVCAdjacencies-Default INTEGER ::= 1
MaximumSVCAdjacencies-Permitted ::=
        INTEGER(1..65535)
reservedAdjacency-Default BOOLEAN ::= FALSE
neighbourSNPAAddress-Default INTEGER ::= 0
recallTimer-Default INTEGER ::= 60
RecallTimer-Permitted ::= INTEGER(0..65535)
maximumCallAttempts-Default INTEGER ::= 10
MaximumCallAttempts-Permitted ::= INTEGER(0..255)
manualL2OnlyMode-Default BOOLEAN ::= FALSE
l2IntermediateSystemPriority-Default INTEGER ::= 64
L2IntermediateSystemPriority-Permitted ::=
        INTEGER(1..127)
lANAddress-Default LANAddress ::= 000000000000
sNPAAddresses-Default SNPAAddresses::= {}
password-Default Password ::= {}
passwords-Default Passwords ::= {} -- The empty set
END
```

12 Conformance

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 paths 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) performing on each supported Point-to-Point circuit (if any):

   1) forming and maintaining adjacencies 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 Reachable Addresses are supported, the operations specified in 8.3.2.2 – 8.3.5.6.

   3) If callEstablishmentMetricIncrement greater than zero are supported, the operations specified in 8.3.5.3.

   4) If the Reverse Path Cache is supported, the operations specified in 8.3.3

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. For all attributes referenced inthe normative text, the default value (if any) shall be supported. Other values shall be supported if referenced in a REQUIRED VALUES clause of the GDMO definition;

z) If authentication procedures are implemented:

   1) the authentication field processing functions of clauses 7.3.7–7.3.10, 7.3.15.1–7.3.15.4, 8.2.3–8.2.4, and 8.4.1.1;

   2) the Authentication Information field of the PDU in clauses 9.5–9.13.

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 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 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.

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.

12.2.5.1 Performance requirements on the Update process

The implementation shall guarantee the update process enough resources to process *N* LSPs per 30 seconds. (Resources = CPU, memory, buffers, etc.)

In a stable topology the arrival of a single new LSP on a circuit shall result in the propagation of that new LSP over the other circuits of the IS within one second, irrespective of the forwarding load for ISO 8473 data PDUs.

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:
   4000 IDs
   100 L1 Intermediate systems per area
   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)

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 implementation, 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 interworking can never be guaranteed, failure to interwork can often be predicted from incompatible 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 questionnaire divided into subclauses each containing a group of individual items. Answers to the questionnaire 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 items where two or more 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 provide 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 number): there are no other restrictions on its format and presentation.

A completed PICS proforma, including any Additional 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 large 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 questionnaire, 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 Exception 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 implementation.

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 questionnaire 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 its 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 this 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 labelled “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.1	□	M: Yes □ O: Yes □ No □

C.1 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	

Item		References	Status	N/A	Support
AllIS	Are all basic IS–IS routeing functions implemented?	12.1.2	M	□	M: Yes □
L1IS	Are Level 1 IS–IS routeing functions implemented?	12.1.3	M	□	M: Yes □
L2IS	Are Level 2 IS–IS routeing functions implemented?	12.1.4	O	□	O: Yes □ No □
PtPt	Are point-to-point circuits implemented?	12.1.2.t	O.1	□	O: Yes □ No □
8208	Are ISO 8208 circuits implemented?	12.1.2.u	O.1	□	O: Yes □ No □
LAN	Are broadcast circuits implemented?	12.1.2.v	O.1	□	O: Yes □ No □
EqualCost-Paths	Is computation of equal minimum cost paths implemented?	7.2.6	O	□	O: Yes □ No □
Downstream	Is computation of downstream routes implemented?	7.2.6	O	□	O: Yes □ No □
DelayMetric	Is path computation based on the delay metric implemented?	7.2.2	O	□	O: Yes □ No □
ExpenseMetric	Is path computation based on the Expense metric implemented?	7.2.2	O	□	O: Yes □ No □
ErrorMetric	Is path computation based on the Error metric implemented?	7.2.2	O	□	O: Yes □ No □
PartitionRepair	Is Level 1 Partition Repair implemented?	12.1.2.f	C.1	□	O: Yes □ No □
Prefixes	Are Reachable Address Prefixes implemented?	12.1.2.j	C.1	□	O: Yes □ No □
callEstablishmentMetricIncrement	Are non-zero values of the callEstablishmentMetricIncrement supported?	12.1.2.u.3	C.2	□	O: Yes □ No □
ReversePath-Cache	Is the 8208 Reverse Path Cache supported?	12.1.2.u.4	C.2	□	O: Yes □ No □
L1 ISCount	How many Level 1 ISs does the implementation support?	12.2.5.1.	M	□	N =
L2 ISCount	How many Level 2 ISs does the implementation support?	12.2.5.1.	C.1	□	N =
ForwardingRate	How many ISO 8473 PDUs can the implementation forward per second?	12.2.5.1.b	M	□	PDUs/sec

C.1 if L2IS then O else —

C.2 if 8208 then O else —

Item		References	Status	N/A	Support
PDU Authentication	Is PDU Authentication based on Passwords implemented?	12.1.2.z	O	□	O: Yes □ No □
ID field Length	What values of the routeingDomainIDLength are supported by this implementation?	7.1.1	M	□	Values = Is the value Setable by System Management? Yes □ No □

C.1 if L2IS then O else —

C.2 if 8208 then O else —

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 this International Standard. 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[1].

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

[1] 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.

space left to customise to the needs of inter-domain routing. The considerations for how addresses may 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

   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 shown in figure 9 below:

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 portion of the 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.

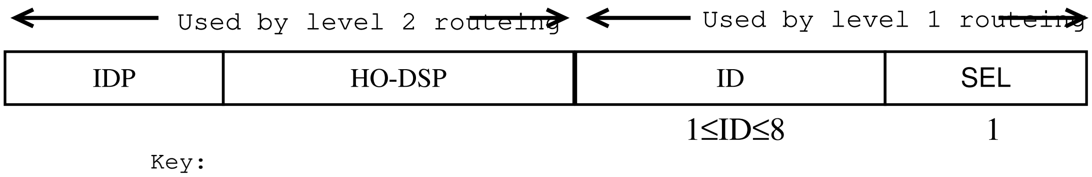
B.2.1.1 The IDP + HO-DSP

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.

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 this International Standard. In order to use a level 2 path as if it were a single subnetwork in a level 1 area, PDUs are encap-



Key:

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

Figure 9 - Preferred Address Format

sulated and addressed to an IS on the other side of the partition[1]. 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**). 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.[2] 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 permit 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;

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,permitting 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 may be specified as flat for the same reasons that the ID field may be flat.

B.3.1 Addressing considerations for Inter-domain Routeing

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

[1] This is a gross oversimplification for the purpose of illustrating the need for the SEL field. See 7.2.10.

[2] 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.

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[3]. 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 B.2 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[4]. 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 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[5] . 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.

[3] This International Standard 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.

[4] 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”.

[5] 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;
                          PDU 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 – 1];
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 – 1];
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 DA", 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).

A number of 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.

c) If only End system LSP information has changed, it is not necessary to re-compute the entire Dijkstra tree for the IS. If the proper data structures exist, End Systems may be attached and detached as leaves of the tree and their forwarding information base entries altered as appropriate

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 Use of Metrics in the SPF Calculation

Internal metrics are not comparable to external metrics. Therefore, the cost of the path from *N* to *S* for external routes (routes to destinations outside of the routing domain) may include both internal and external metrics. The cost of the path from *N* to *S* (called *d(N)* below in database PATHS*)* may therefore be maintained as a two-dimensioned vector quantity (specifying internal and external metric values). In incrementing *d(N)* by 1, if the internal metric value is less than the maximum value MaxPathMetric, then the internal metric value is incremented by one and the external metric value left unchanged; if the internal metric value is equal to the maximum value MaxPathMetric, then the internal metric value is set to 0 and the external metric value is incremented by 1. Note that this can be implemented in a straightforward manner by maintaining the external metric as the high order bits of the distance.

NOTE - In the code of the algorithm below, the current path length is held in a variable tentlength. This variable is a two-dimensional quantity tentlength=(internal,external) and is used for comparing the current path length with *d(N)* as described above.

C.2.3 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 $\langle M,*,*\rangle$ is in PATHS, this new path will be longer, and thus ignored.

If $\langle M,*,*\rangle$ 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, then the set of potential adjacencies *{adj(M)}* is set to the union of the old set (in TENT) and the new set *{adj(N)}*. If *M* is not in TENT, then the path is added to TENT.

Next the algorithm finds the triple $\langle N,x,\{Adj(N)\}\rangle$ 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.2.4 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 7.2.10.

Step 0: Initialise TENT and PATHS to empty. Initialise tentlength to (0,0).

(tentlength is the pathlength of elements in TENT we are examining.)

a) Add $\langle$SELF, *0, W*$\rangle$ 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 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 L_nCircuitID 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

$$d(N) = \text{metric}_k \text{ of the circuit.}$$

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 LSP Database Overload 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

$$dist(P,N) = d(P) + \text{metric}_k(P,N).$$

for each neighbour *N* (both Intermediate System and End system) of the system *P*. If the LSP Database Overload 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 $\text{metric}_k(P,N)$ 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 $Adj(\text{N}) \leftarrow \{Adj(\text{N})\} \cup Adj(P)$.

2) If there are now more adjacencies in $\{Adj(N)\}$ than maximumPathSplits, then remove excess adjacencies, as described in 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 $\langle *,\text{tentlength},*\rangle$ 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 $\langle P,\text{tentlength},\{Adj(P)\}\rangle$ 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 ⟨AREA.P, d(P), {adj(P)}⟩ 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[tem
        p]);
        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.

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).

Index