

INTERNATIONAL STANDARD

ISO/IEC 10747

First edition
1994-10-01

Information technology — Telecommunications and information exchange between systems — Protocol for exchange of inter-domain routing information among intermediate systems to support forwarding of ISO 8473 PDUs

*Technologies de l'information — Télécommunications et échange
d'information entre systèmes — Protocole pour échange d'information
inter-domaine de routage parmi les systèmes intermédiaires supportant la
transmission de PDUs de l'ISO 8473*



Reference number
ISO/IEC 10747:1994(E)

AVA Documents
Document Sharing Hub

Contents

1 Scope	1	5.10 Routeing domain identifiers	9
2 Normative references	1	5.11 Formats of RDIs, NETs, and NSAP addresses	10
3 Definitions	2	5.12 Design objectives	10
3.1 Reference model definitions	2	5.12.1 Within the scope of the protocol	10
3.2 Network layer architecture definitions	3	5.12.2 Outside the scope of the protocol	10
3.3 Network layer addressing definitions	3	6 Structure of BISPDU	11
3.4 Routeing framework definitions	3	6.1 Header of BISPDU	11
3.5 Intra-domain routeing definitions	3	6.2 OPEN PDU	12
3.6 Additional definitions	3	6.3 UPDATE PDU	14
4 Symbols and abbreviations	4	6.3.1 Path attribute encoding	14
4.1 Data unit abbreviations	4	6.3.2 Network layer reachability information	19
4.2 Addressing abbreviations	4	6.4 IDRP ERROR PDU	20
4.3 Other abbreviations	5	6.5 KEEPALIVE PDU	21
5 General protocol information	5	6.6 CEASE PDU	21
5.1 Inter-RD topology	5	6.7 RIB REFRESH PDU	21
5.2 Routeing policy	6	7 Elements of procedure	21
5.3 Types of systems	7	7.1 Naming and addressing conventions	22
5.4 Types of routeing domains	7	7.1.1 Interpretation of address information	22
5.5 Routeing domain confederations	7	7.1.2 NSAP address prefixes	22
5.6 Routes: advertisement and storage	7	7.2 Deployment guidelines	22
5.7 Distinguishing path attributes and RIB-Atts	8	7.2.1 Minimum configuration of an RD	22
5.8 Selecting the information bases	8	7.2.2 Deployment of ISs and ESs	22
5.9 Routeing information exchange	8	7.3 Domain configuration information	23
5.9.1 Internal neighbor BIS	8	7.4 Advertising NLRI	23
5.9.2 External neighbor BIS	9	7.5 Receive process	24
		7.6 BIS-BIS connection management	24
		7.6.1 BIS finite state machines	24

© ISO/IEC 1994

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case Postale 56 • CH-1211 Genève 20 • Switzerland

Printed in Switzerland

7.6.2 Closing a connection	29	7.20.4 IDRP ERROR PDU error handling	59
7.7 Validation of BISPDUs	29	7.20.5 Hold timer expired error handling	59
7.7.1 Authentication type 1	29	7.20.6 KEEPALIVE PDU error handling	59
7.7.2 Authentication type 2	29	7.20.7 CEASE PDU error handling	59
7.7.3 Authentication type 3	29	7.20.8 RIB REFRESH PDU error handling	59
7.7.4 Sequence numbers	31		
7.7.5 Flow control	31	8 Forwarding process for CLNS	59
7.8 Version negotiation	33	8.1 Forwarding to internal destinations	60
7.9 Checksum algorithm	33	8.2 Determining the NPDU-derived distinguishing attributes	60
7.10 Routeing information bases	33	8.3 Matching RIB-Att to NPDU-derived distinguishing attributes	60
7.10.1 Identifying an information base	33	8.4 Forwarding to external destinations	61
7.10.2 Validation of RIBs	34		
7.10.3 Use of the RIB REFRESH PDU	35	9 Interface to ISO 8473	62
7.11 Path attributes	35	9.1 Use of network layer security protocol over ISO 8473.	62
7.11.1 Categories of path attributes	35		
7.11.2 Handling of distinguishing attributes	37	10 Constants	63
7.11.3 Equivalent distinguishing attributes	37		
7.12 Path attribute usage	37	11 System management and GDMO definitions	63
7.12.1 ROUTE_SEPARATOR	37	11.1 Name binding	63
7.12.2 EXT_INFO	38	11.2 Managed objects for IDRP	63
7.12.3 RD_PATH	38	11.3 Packages for IDRP	63
7.12.4 NEXT_HOP	40	11.4 Attribute definitions	67
7.12.5 DIST_LIST_INCL	41	11.5 Parameter definitions	71
7.12.6 DIST_LIST_EXCL	42	11.6 Behaviour	72
7.12.7 MULTI-EXIT_DISC	43	11.7 ASN.1 modules	72
7.12.8 TRANSIT DELAY	43		
7.12.9 RESIDUAL ERROR	43	12 Conformance	74
7.12.10 EXPENSE	44	12.1 Static conformance for all BISs	75
7.12.11 LOCALLY DEFINED QOS	44	12.2 Conformance to optional functions	75
7.12.12 HIERARCHICAL RECORDING	44	12.2.1 Generation of information in reduced form	75
7.12.13 RD_HOP_COUNT	45	12.2.2 Generation of well-known discretionary attributes	75
7.12.14 SECURITY	45	12.2.3 Propagation of well-known discretionary attributes	75
7.12.15 CAPACITY	45	12.2.4 Peer entity authentication	76
7.12.16 PRIORITY	46		
7.13 Routeing domain confederations	46	Annex A PICS proforma	77
7.13.1 RDC policies	46	A.1 Introduction	77
7.13.2 RDC configuration information	46	A.2 Abbreviations and special symbols	77
7.13.3 Detecting confederation boundaries	46	A.2.1 Status symbols	77
7.14 Update-Receive process	46	A.3 Instructions for completing the PICS proforma	77
7.15 Information consistency	47	A.3.1 General structure of the PICS proforma	77
7.15.1 Detecting inconsistencies	47	A.3.2 Additional information	78
7.16 Decision process	47	A.3.3 Exception information	78
7.16.1 Phase 1: calculation of degree of preference	48	A.3.4 Conditional status	78
7.16.2 Phase 2: route selection	48	A.4 Identification	79
7.16.3 Phase 3: route dissemination	49	A.4.1 PICS proforma: IDRP implementation identification	79
7.16.4 Interaction with update process	50	A.4.2 PICS proforma: IDRP protocol summary	80
7.17 Update-Send process	51	A.4.3 PICS proforma: IDRP general	80
7.17.1 Internal updates	51	A.4.4 PICS proforma: IDRP update send process	81
7.17.2 External updates	52		
7.17.3 Controlling routeing traffic overhead	52		
7.18 Efficient organization of routeing information	53		
7.18.1 Information reduction	53		
7.18.2 Aggregating routeing information	53		
7.19 Maintenance of the forwarding information bases	56		
7.20 Error handling for BISPDUs	56		
7.20.1 BISPDUs header error handling	56		
7.20.2 OPEN PDU error handling	57		
7.20.3 UPDATE PDU error handling	57		

A.4.5 PICS proforma: IDRP update receive process	81	Annex H Processor overload	94
A.4.6 PICS proforma: IDRP decision process	81	Annex J Formation of RDCs	95
A.4.7 PICS proforma: IDRP receive process	81	J.1 Forming a new lower level confederation	95
A.4.8 PICS proforma: IDRP CLNS forwarding	82	J.2 Forming a higher level confederation	95
A.4.9 PICS proforma: IDRP authentication	82	J.3 Deleting a lowest level confederation	96
A.4.10 PICS proforma: IDRP optional transitive attributes	82	J.4 Deleting a higher level confederation	96
A.4.11 PICS proforma: Generating IDRP well-known discretionary attributes	83	Annex K Example usage of MULTI-EXIT_DISC attribute	97
A.4.12 PICS proforma: Propagating IDRP well-known discretionary attributes	84	Annex L Syntax and semantics for policy	99
A.4.13 PICS proforma: Receiving IDRP well-known discretionary attributes	85	L.1 Overview	99
Annex B IDRP checksum generation algorithm	86	L.1.1 Preference statement	99
B.1 Mathematical notation	86	L.1.2 Aggregation statement	100
B.2 Algorithm description	86	L.1.3 Distribution statement	101
Annex C Bibliography	88	L.2 Policy configuration language BNF	102
Annex D Example of authentication type 2	89	L.2.1 PREF statement BNF	102
D.1 Authentication mechanism	89	L.2.2 AGGR statement BNF	102
Annex E Jitter algorithm	91	L.2.3 DIST statement BNF	102
Annex F Computing a checksum for an Adj-RIB	92	L.2.4 Common BNF symbols	103
Annex G RIB overload	93	L.3 Simple example	104
		L.3.1 Transit domain 3	104
		L.3.2 Policy configuration example	105
		L.3.3 Discussion	106
		Index	110

Figures

1.	Field of Application	2
2.	Intermediate Routeing Domains and End Routeing Domains	4
3.	Position of IDRP within Network Layer	6
4.	Inter-domain Routeing Components	7
5.	Structure of the UPDATE PDU	15
6.	Illustration of Authentication Types 1 and 3	30
7.	Routeing Information Base	34
8.	A Transitive Fully Connected Subnetwork	41
9.	IDRP Naming and Containment Hierarchy	65
D.1	An Example of the Authentication Type 2	90
K.1	Example 1 Configuration	98
K.2	Example 2 Configuration	98
L.1	A Portion of an Internet	105

Tables

1.	The IDRP Information Bases	9
2.	BIS Finite State Machine	26
3.	Path Attribute Characteristics	36
4.	NPDU-Derived Attribute Set	61
5.	IDRP-CL Primitives	63
6.	Architectural Constants of IDRP	64

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 10747 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems*.

Annexes A and B form an integral part of this International Standard. Annexes C, D, E, F, G, H, J, K and L are for information only.

Introduction

This Protocol is one of a set of International Standards which facilitate the interconnection of open systems. They cover the services and protocols required to achieve such interconnection.

This Protocol is positioned with respect to other related standards by the layered structure defined in ISO 7498, and by the Network layer organization defined in ISO 8648. It is located at the top of the Network layer and relies on the services of ISO 8473. This protocol permits a routing domain to exchange information with other routing domains to facilitate the operation of the routing and relaying functions of the Network Layer. It applies to the following categories of routing, which are described in ISO/IEC TR 9575, making no distinction between them:

- Intra-Administrative Domain routing between routing domains
- Inter-Administrative Domain routing between routing domains.

Within the hierarchical relations between routing protocols, as described in ISO/IEC TR 9575, this protocol is situated above the intra-domain routing protocols. That is, this Inter-domain IS-IS protocol:

- maintains information about the interconnections between routing domains, but does not require detailed information about their internal structures
- calculates path segments on a hop-by-hop basis

This protocol calculates path segments which consist of *Boundary Intermediate systems* and the links that interconnect them. An NPDU destined for an End system in another routing domain will be routed via Intra-domain routing to a Boundary Intermediate system (BIS) in the source routing domain. Then,

the BIS, using the methods of this inter-domain routing protocol, will calculate a path to a Boundary Intermediate system in an adjacent routing domain lying on a path to the destination. After arriving at the next routing domain, the NPDU may also travel within that domain on its way towards a BIS located in the next domain along its path. This process will continue on a hop-by-hop basis until the NPDU arrives at a BIS in the routing domain which contains the destination End system. The Boundary IS in this routing domain will hand the incoming NPDU over to the domain's intra-domain routing protocol, which will construct a path to the destination End system.

This inter-domain IS-IS routing protocol places requirements on the type of information that a routing domain must provide and on the methods by which this information will be distributed to other routing domains. These requirements are intended to be minimal, addressing only the interactions between Boundary ISs; all other internal operations of each routing domain are outside the scope of this protocol. That is, this Inter-domain routing protocol does not mandate that a routing domain run a particular intra-domain routing protocol: for example, it would be a local choice as to whether a domain implements a standard intra-domain protocol (such as ISO/IEC 10589) or a private protocol.

The methods of this protocol differ from those generally adopted for an intra-domain routing protocol because they emphasize the interdependencies between efficient route calculation and the preservation of legal, contractual, and administrative concerns. This protocol calculates routes which will be efficient, loop-free, and in compliance with the domain's local routing policies. IDRP may be used when routing domains do not fully trust each other; it imposes no upper limit on the number of routing domains that can participate in this protocol; and it provides isolation between its operations and the internal operations of each routing domain.

Information technology - Telecommunications and information exchange between systems - Protocol for exchange of inter-domain routing information among intermediate systems to support forwarding of ISO 8473 PDUs

1 Scope

This International Standard specifies a protocol to be used by Boundary Intermediate systems (defined in 3.6) to acquire and maintain information for the purpose of routing NPDUs between different routing domains. Figure 1 illustrates the field of application of this International Standard.

This International Standard specifies:

- the procedures for the exchange of inter-domain reachability and path information between BISs
- the procedures for maintaining inter-domain routing information bases within a BIS
- the encoding of protocol data units used to distribute inter-domain routing information between BISs
- the functional requirements for implementations that claim conformance to this International Standard

The procedures are defined in terms of:

- interactions between Boundary Intermediate systems through the exchange of protocol data units
- interactions between this protocol and the underlying Network Service through the exchange of service primitives
- constraints on policy feasibility and enforcement which must be observed by each Boundary Intermediate system in a routing domain

The boundaries of Administrative Domains are realized as artifacts of the placement of policy constraints and the aggregation of network layer reachability information; they are not manifested explicitly in the protocol. The protocol described in this International Standard operates at the level of individual routing domains. The establishment of administrative domains is outside the scope of this International Standard.

2 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/IEC 7498-4:1989, *Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 4: Management framework*.

ISO/IEC 8208:1990, *Information technology - Data communications - X.25 Packet Layer Protocol for Data Terminal Equipment*.

ISO/IEC 8348:1993, *Information technology - Network Service Definition*.

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

ISO 8648: 1988, *Information processing systems - Telecommunications and information exchange between systems - Internal organization of the Network Layer*.

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

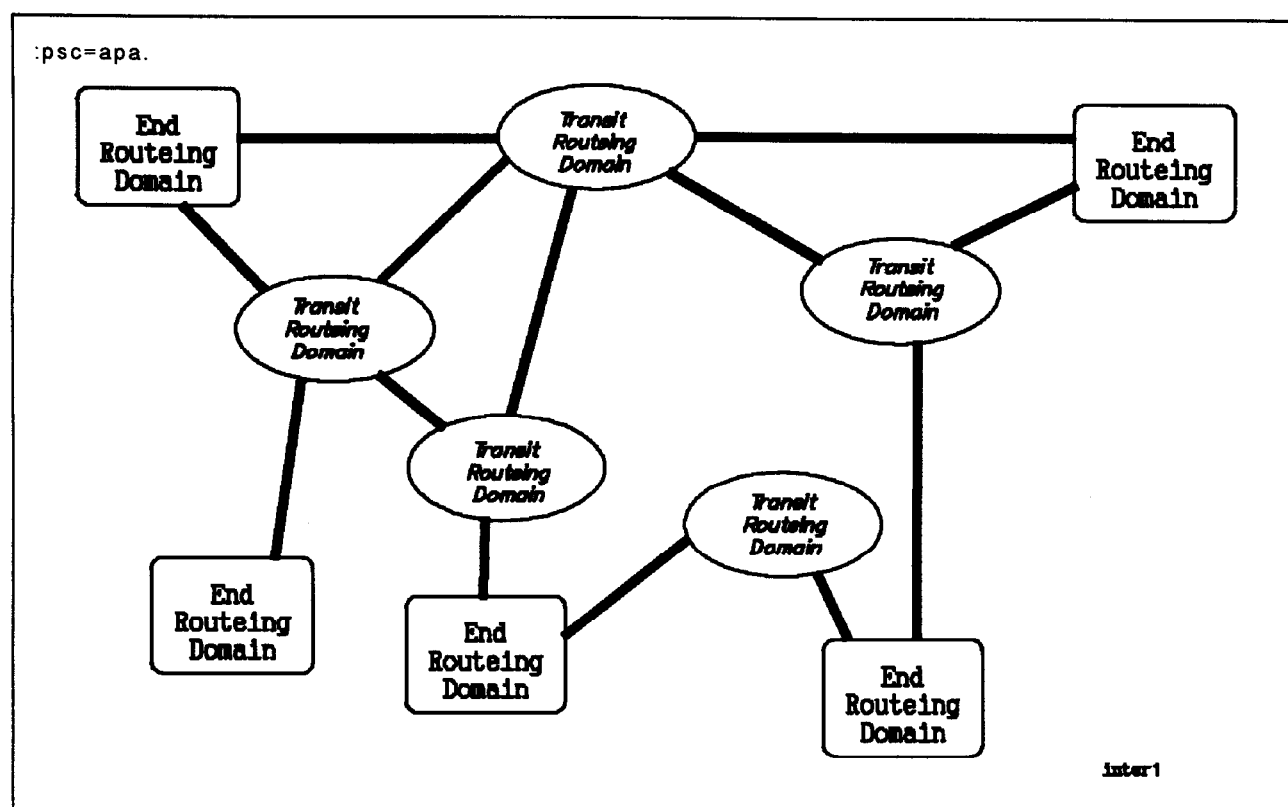


Figure 1 – Field of Application: The Inter-domain Routing Protocol operates between routing domains; intra-domain routing is not within its scope.

ISO/IEC TR 9575:1990, *Information technology - Telecommunications and information exchange between systems - OSI Routing Framework*.

ISO/IEC TR 9577:1993, *Information technology - Telecommunications and information exchange between systems - Protocol identification in the Network Layer*.

ISO/IEC 10030:1990, *Information technology - Telecommunications and information exchange between systems - End System Routing Information Exchange Protocol for use in conjunction with ISO 8878*.

ISO/IEC 10589:1992, *Information technology - Telecommunications and information exchange between systems - Intermediate system to intermediate system intra-domain routing routine information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)*.

ISO/IEC 10165-4:1992, *Information technology - Open Systems Interconnection - Structure of management information: Guidelines for the definition of managed objects*.

ISO/IEC 10165-2:1992, *Information technology - Open Systems Interconnection - Structure of management information: Definition of management information*.

3 Definitions

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

3.1 Reference model definitions

This International Standard uses the following terms defined in ISO 7498:

- a) Network entity
- b) Network Layer
- c) Network Protocol
- d) Network Protocol Data Unit
- e) Network relay
- f) Network Service Access Point
- g) Network Service Access Point Address
- h) Real system
- i) Routing

This International Standard uses the following term defined in ISO 7498-3:

- a) (N)-entity title

3.2 Network layer architecture definitions

This International Standard uses the following terms defined in ISO 8648:

- a) End system
- b) Intermediate System
- c) Subnetwork

3.3 Network layer addressing definitions

This International Standard uses the following term defined in ISO/IEC 8348:

- a) Subnetwork point of attachment

3.4 Routeing framework definitions

This International Standard uses the following terms defined in ISO 9575:

- a) Administrative Domain
- b) Common Domain
- c) Fire wall
- d) Routeing Domain

3.5 Intra-domain routeing definitions

This International Standard uses the following terms defined in ISO 10589:

- a) Adjacency
- b) Link

3.6 Additional definitions

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

3.6.1 Intra-domain IS-IS routeing protocol: A routeing protocol that is run between Intermediate systems in a single routeing domain to determine routes that pass through only systems and links wholly contained within the domain.

NOTE 1: Unless reference is made to a specific protocol, this term is used as a general designator, encompassing both private and internationally standardized protocols.

3.6.2 Inter-domain link: A real (physical) or virtual (logical) link between two or more Boundary Intermediate systems (see Figure 2). A link between two BISs in the same routeing domain carry both intra-domain traffic and inter-domain traffic; a link between two BISs located in adjacent routeing domains can carry inter-domain traffic, but not intra-domain traffic.

3.6.3 Boundary Intermediate system: An intermediate system that runs the protocol specified in this International Standard, has at least one inter-domain link attached to it, and may optionally have intra-domain links attached to it.

3.6.4 End Routeing Domain: A routeing domain whose local policies permit its BISs to calculate inter-domain path segments only for PDUs whose source is located within that routeing domain. There are two varieties of End routeing domains: stub and multi-homed. A stub ERD has inter-domain links to only one adjacent routeing domain, while a multi-homed ERD has inter-domain links to several adjacent routeing domains.

For example, the domains labelled as multi-homed ERDs in Figure 2 have policies which prohibit them from providing relaying functions; it is these policies, not the topology of their interconnections, that make them ERDs.

3.6.5 Transit Routeing Domain: A routeing domain whose policies permit its BISs to calculate inter-domain path segments for PDUs whose source is located either in the local routeing domain or in a different routeing domain. That is, it can provide a relaying service for such PDUs. See Figure 2 for an illustration of TRDs.

3.6.6 Adjacent RDs: Two RDs ("A" and "B") are adjacent to one another if there is at least one pair of BISs, one located in "A" and the other in "B", that are attached to each other by means of a real subnetwork.

3.6.7 RD Path: A list of the RDIs of the routeing domains and routeing domain confederations through which a given UPDATE PDU has travelled.

3.6.8 Routeing Domain Confederation: A set of routeing domains which have agreed to join together and to conform to the rules in 7.13 of this International Standard. To the outside world, a confederation is indistinguishable from a routeing domain.

3.6.9 Nested RDCs: A routeing domain confederation "A" (RDC-A) is nested within RDC-B when all of the following conditions are satisfied simultaneously:

- a) all members of RDC-A are also members of RDC-B
- b) there are some members of RDC-B that are not members of RDC-A

3.6.10 Overlapping RDCs: A routeing domain confederation (RDC-A) overlaps RDC-B when all the following conditions are satisfied simultaneously:

- a) there are some members of RDC-A that are also members of RDC-B, and
- b) there are some members of RDC-A that are not members of RDC-B, and
- c) there are some members of RDC-B that are not members of RDC-A.

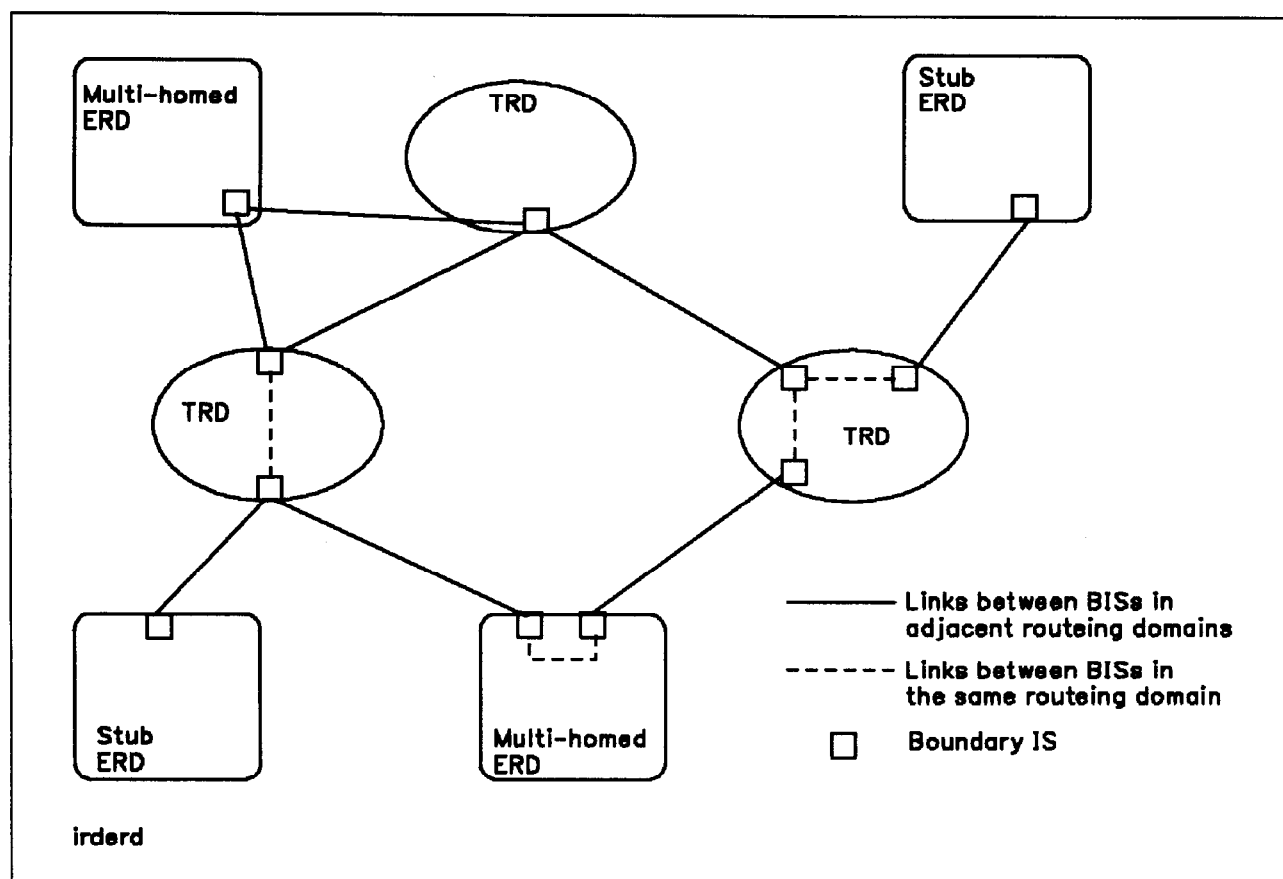


Figure 2 – Intermediate Routing Domains and End Routing Domains: The classification of a routing domain as an TRD or an ERD depends upon its relaying policies.

3.6.11 Disjoint RDCs: Two routing domain confederations, RDC-A and RDC-B, are disjoint from one another when there are no routing domains which are simultaneously members of both RDC-A and RDC-B.

3.6.12 Policy Information Base: The collection of routing policies that a BIS will apply to the routing information that it learns using this International standard. It is not required that all routing domains use the same syntax and semantics to express policy; that is, the format of the Policy Information Base is left as a local option.

3.6.13 Route Origin: Each route or component of an aggregated route has a single unique origin. This is the RD or RDC in which the route's destinations are located.

4 Symbols and abbreviations

The symbols, acronyms, and abbreviations listed in the following clauses are used in this International Standard.

4.1 Data unit abbreviations

BIS PDU	Boundary Intermediate System PDU
DT PDU	ISO 8473 Data Protocol Data Unit
ER PDU	ISO 8473 Error Protocol Data Unit
NPDU	Network Protocol Data Unit
NSDU	Network Service Data Unit
PDU	Protocol Data Unit

4.2 Addressing abbreviations

AFI	Authority and Format Identifier
DSP	Domain Specific Part
IDI	Initial Domain Identifier
IDP	Initial Domain Part
LSAP	Link Service Access Point
NET	Network Entity Title
NPAI	Network Protocol Address Information

NSAP	Network Service Access Point
SNPA	Subnetwork Point of Attachment

4.3 Other abbreviations

BIS	Boundary Intermediate System
CL	Connectionless Mode
CLNS	Connectionless Mode Network Service
CM	Confederation Member
ERD	End Routeing Domain
ES	End System
FIB	Forwarding Information Base
FSM	Finite State Machine
IDRP	Inter-domain Routeing Protocol (an acronym for the protocol described in this International Standard)
IPI	Initial Protocol Identifier
MIB	Management Information Base
NLRI	Network layer reachability information
NLSP	Network layer security protocol
OSIE	OSI Environment
PCI	Protocol Control Information
PIB	Policy Information Base
QOS	Quality of Service
RDC	Routeing Domain Confederation
RDI	Routeing Domain Identifier
RIB	Routeing Information Base
SPI	Subsequent Protocol Identifier
SNICP	Subnetwork independent convergence protocol
TRD	Transit Routeing Domain

5 General protocol information

IDRP is a routeing information exchange protocol which is located within the Network layer and interfaces to ISO 8473, which serves as a SNICP (see Figure 3). In particular, BISPDU are encapsulated as the data portion of ISO 8473 NPDUs. IDRP is a connection-oriented protocol which is implemented only in Intermediate systems. Routeing and control information is carried in BISPDU (as in clause 6), which flow on connections between pairs of BISs. Each BISPDU is packaged within one or more NPDUs for transmission by the underlying Network service.

IDRP relies on the underlying Network service to provide for fragmentation and reassembly of BISPDU. IDRP queues Outbound BISPDU as input to the underlying Network Layer service, retaining a copy of each BISPDU until an acknowledgement is received. Similarly, inbound BISPDU are queued as input to the BISPDU-Receive process.

IDRP exchanges BISPDU in a reliable fashion. It provides mechanisms for the ordered delivery of BISPDU and for the detection and retransmission of lost or corrupted BISPDU. The mechanisms for achieving reliable delivery of BISPDU are described in 7.7; methods for establishing BIS-BIS connections are described in 7.6.

IDRP is consistent with the routeing model presented in ISO TR 9575. To emphasize its policy-based nature, the IDRP routeing model includes a Policy Information Base, as shown in Figure 4. IDRP can be described in terms of four major components:

- a) **BISPDU-Receive Process:** responsible for accepting and processing control and routeing information from the local environment and from BISPDU of other BISs. This information is used for a variety of purposes, such as receiving error reports and guaranteeing reliable reception of BISPDU from neighboring BISs. (For example, the Update-Receive process (see 7.14) is the part of the BISPDU-Receive process that deals with the reception of routeing information after a BIS-BIS connection has been established.)
- b) **BISPDU-Send Process:** responsible for constructing BISPDU which contain control and routeing information. BISPDU are used by the local BIS for a variety of purposes, such as advertising routeing information to other BISs, initiating BIS-BIS communication, and validating BIS routeing information bases.
- c) **Decision Process:** responsible for calculating routes which will be consistent with local routeing policies. It operates on information in both the PIB and the Adj-RIBs, using it to create the Local RIBs (Loc-RIBs) and the local Forwarding Information Bases (see 7.10).
- d) **Forwarding Process:** responsible for supplying resources to accomplish relaying of NPDUs to their destinations. It uses the FIB(s) created by the Decision Process.

5.1 Inter-RD topology

This protocol views the overall global OSIE as an arbitrary interconnection of Transit Routeing Domains and End Routeing Domains which are connected by real inter-domain links placed between BISs located in the respective routeing domains. This International

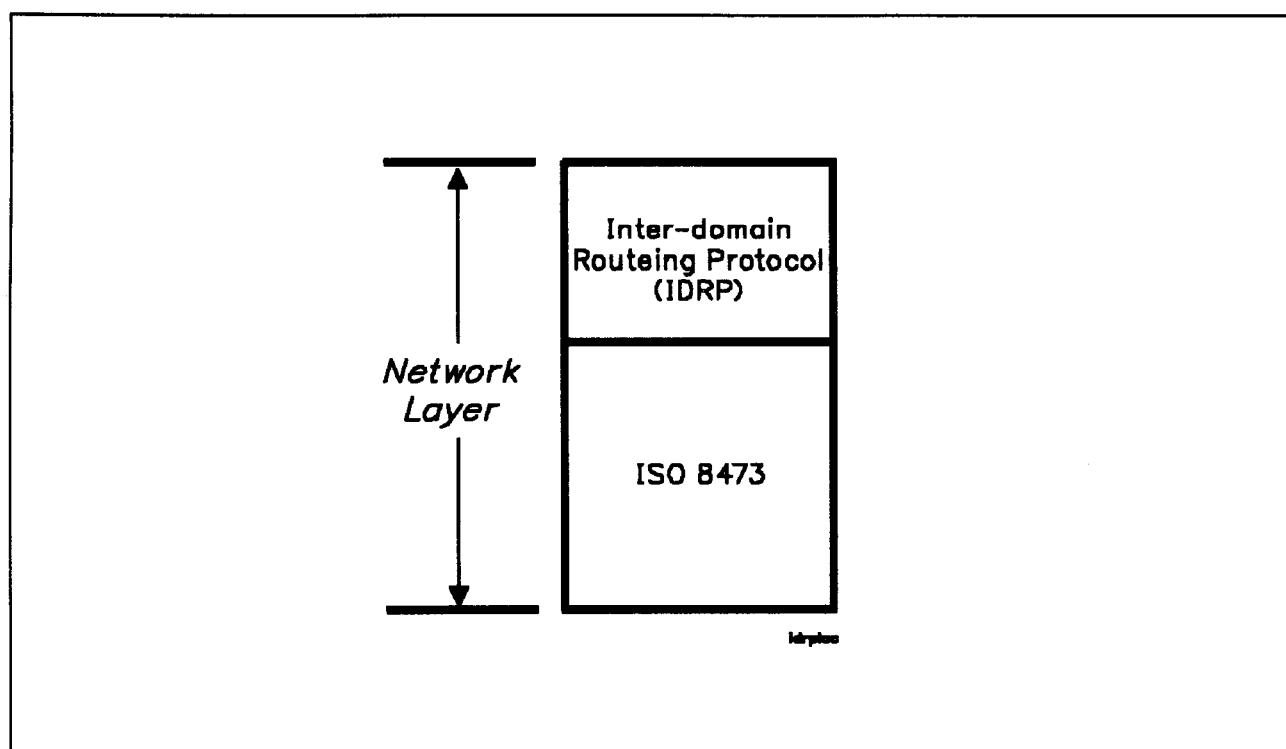


Figure 3 – Position of IDR within Network Layer

Standard provides for the direct exchange of routing information between BISs, which may be located either in the same routing domain or in adjacent routing domains.

5.2 Routing policy

The direct exchange of policy information is outside the scope of IDR. Instead, IDR communicates policy information indirectly in its UPDATE PDUs which reflect the effects of the local policies of RDs on the path to the destination. Since all BISs within a routing domain must enforce consistent active routing policies, IDR provides methods for detecting the existence of active inconsistent policies within a routing domain. However, the semantics of routing policies and the methods for establishing them are outside the scope of this International Standard.

NOTE 2: Annex L illustrates a policy description method and its associated semantics as one example of how policies might be expressed.

Each routing domain chooses its routing policies independently, and insures that all its BISs calculate inter-domain paths which satisfy those policies. Local routing policies are applied to information in the Routing Information Base (RIB) to determine a degree of preference for potential paths (see 7.16). From those paths which are not rejected by the routing policy, a BIS selects the paths which it will use locally; from the locally selected paths, the BIS

will then select the paths that it will advertise externally.

To enforce routing policies and to insure that policies are both feasible and consistent, this protocol:

- carries path information, expressed in terms of Routing Domain Identifiers (RDIs) and various path attributes, in its UPDATE PDUs
- permits a routing domain to selectively propagate its reachability information to a limited set of other routing domains
- provides a method to detect policy inconsistencies within the set of BISs located in a single routing domain
- permits each routing domain to set its policies individually: that is, global coordination of policy is not required.

The set of rules that comprises the routing policy enforced by a BIS are held in a Policy Information Base (PIB), which is separate from the RIB. Depending on local Security and QOS requirements, the PIB may also contain:

- a) rules for the aggregation of routes that include the SECURITY and LOCALLY DEFINED QOS path attributes (see 7.18.2)
- b) rules for enforcing local QOS Maintenance Policies and the effective Security Policy, during NPDU forwarding

- ### 5.3 Types of systems

5.4 Types of routeing domains

5.5 Routing domain confederations

An RDC is formed by means outside the scope of this protocol, and composed of a set of *confederation members*. Confederation members (CMs) are either individual routing domains or routing domain confederations. Thus, the definition of an RDC is recursive: a confederation member may be a single routing domain or another confederation.

5.6 Routes: advertisement and storage

- *Routes* are advertised between a pair of BISs in UPDATE PDUS: the *destinations* are the systems whose NSAP prefixes are reported in the NLRI field, and the *path* is the information reported in the path attributes fields of the same UPDATE PDU.



- *Routes* are stored in the Routeing Information Bases: namely, the Adj-RIBs-In, the Loc-RIBs, and the Adj-RIBs-Out. Routes that will be advertised to other BISs must be present in the Adj-RIBs-Out; routes that will be used by the local BIS must be present in the Loc-RIBs, and the next hop for each of these routes must present in the local BIS's Forwarding Information Bases; and routes that are received from other BISs are present in the Adj-RIBs-In.
- A *Route-ID* is assigned to each route that is advertised by a BIS. This identifier is unambiguous in the context of the BIS-BIS connection between the advertising BIS and the receiving BIS.

A BIS can support multiple routes to the same destination by maintaining multiple RIBs and the corresponding multiple FIBs. Each Loc-RIB will be identified by a different RIB-Att (see 5.7 and 5.8); an Adj-RIB-Out shall contain at most one route to a particular destination.

If the BIS chooses to advertise the route, it may add to or modify the path attributes of the route before advertising it to adjacent BISs. For example, it is possible under certain circumstances to aggregate path attributes, NLRI, or entire routes, as described more fully in 7.18.2; or, as another example, the further distribution of a route may be restricted through the use of the DIST_LIST_EXCL attribute, as described in 7.12.6.

IDRP also provides mechanisms by which a BIS can inform its neighbor that a previously advertised route is no longer available for use. There are three methods by which a given BIS can indicate that a route has been withdrawn from service:

- a) the Route-ID for a previously advertised route can be advertised in the WITHDRAWN ROUTES field of an UPDATE PDU, thus marking the associated route as being no longer available for use
- b) a replacement route (with the same distinguishing attributes and NLRI) can be advertised, or
- c) the BIS-BIS connection can be closed, which implicitly removes from service all routes which the pair of BISs had advertised to each other.

5.7 Distinguishing path attributes and RIB-Atts

Certain path attributes are classified as *Distinguishing Attributes*. Each distinct combination of such attributes identifies a particular information base which will be used to store information about the route. Each combination of distinguishing attributes is called a *RIB-Att* (RIB attribute); the RIB-Att is a common identifier for the Adj-RIB-In, Loc-RIB, Adj-RIB-Out, and FIB with which the route information is associated.

The number of RIB-Atts is limited by the number of distinct sets of permissible distinguishing attributes (see 7.11.2); this in turn limits the number of RIBs and FIBs that a BIS can support. The number of RIBs and FIBs can be further constrained by local decisions—a BIS may choose to support only a limited number of distinct routeing information bases (that is, a limited number of RIB-Atts, as described in 7.10.1).

5.8 Selecting the information bases

Each RIB is identified by a RIB-Att (RIB attribute), and the same RIB-Att also uniquely identifies the associated FIB.

For an UPDATE PDU, the BIS determines the ROUTE_ID, LOCAL_PREF, and the set of distinguishing path attributes associated with each route that is advertised. The set of distinguishing path attributes contained between a pair of consecutively occurring ROUTE_SEPARATORS or between the last ROUTE_SEPARATOR and the end of the BISPDU unambiguously determine the RIB-Att for that route.

For an NPDU, the BIS unambiguously determines the FIB that should be used for forwarding this NPDU. It maps certain fields in NPDU's header into a RIB-Att, which then unambiguously identifies a particular FIB (see 8.2 and 8.3).

A summary of IDRP's information bases is presented in Table 1.

5.9 Routeing information exchange

This International Standard provides several rules governing the distribution and exchange of routeing information:

- rules for distributing routeing information internally (to BISs within a routeing domain)
- rules for distributing routeing information externally (to BISs in adjacent routeing domains)

Routeing information is carried in the protocol's BISPDU's, which are generated on an event-driven basis whenever a BIS receives information which causes it to advertise new paths.

5.9.1 Internal neighbor BIS

Each BIS establishes and maintains communications with all other BISs located in its routeing domain. The identity of all BISs within a routeing domain is contained in managed object **INTERNAL_BIS** described in 7.3.

Table 1 – The IDRP Information Bases: The indexing variables and contents of the RIBs and FIBs are shown.		
<i>Information Base</i>	<i>Indexed by...</i>	<i>Contains...</i>
<i>Adj-RIB-In</i>	— NET of adjacent BIS — RIB-Atts — Route-ID	— Path attributes — NLRI
<i>Loc-RIB</i>	— RIB-Atts	— Path attributes — NLRI
<i>Adj-RIB-Out</i>	— NET of adjacent BIS — RIB-Atts — Route-ID	— Path attributes — NLRI
<i>FIB</i>	— RIB-Atts — NLRI	— NET of next hop BIS — Output SNPA of local BIS — Input SNPA of next hop BIS — minimum priority associated with this subnetwork, when the RIB-Att contains the PRIORITY attribute — security related information associated with this subnetwork, when the RIB-Att contains the SECURITY attribute — QOS metric value, when the RIB-Att contains a RESIDUAL ERROR, TRANSIT DELAY, EXPENSE, or LOCALLY DEFINED QOS attribute
NOTES: a) As a local option, a BIS may elect to apply information reduction techniques to path attributes and NLRI information. b) For each adjacent BIS, a given BIS maintains an Adj-RIB-In for each RIB-Att (including the Empty RIB-Att) that it supports. c) A BIS maintains a separate Loc-RIB for each RIB-Att (including the Empty RIB-Att) that it supports. d) For each adjacent BIS, a given BIS maintains an Adj-RIB-Out for each set of RIB-Atts (including the Empty RIB-Att) that it advertises to that neighbor. e) A given BIS maintains a separate FIB for each set of RIB-Atts (including the empty RIB-Att) that it supports— that is, each FIB corresponds to a Loc-RIB. To facilitate the forwarding process, a BIS can organize each of its FIBs into two conceptual parts: one containing information for NLRI located within its own RD, and another for NLRI located in other RDs (as in clause 8). For external NLRI, a BIS can further organize the FIB information based on whether the next-hop-BIS is located within its own RD or in another RD (see 8.4, items "a" and "b"). And finally, for those next-hop BISs located in its own RD, the local BIS can organize the information according to a specific forwarding mechanism (see 8.4, items "b1" and "b2").		

5.9.2 External neighbor BIS

Each BIS may establish and maintain communications with other BISs in adjacent routeing domains. A BIS has no direct communications link with any BIS in another routeing domain unless that RD is adjacent to it, as defined in 3.6: that is, a BIS does not communicate directly with a BIS located in a different routeing domain unless the pair of BISs are attached to at least one common subnetwork. The identity of neighbor BISs in adjacent routeing domains is contained in managed object **EXTERNAL-BIS-NEIGHBORS** described in 7.3.

NOTE 3: In the absence of an implementation specific method for ascertaining that a neighbor BIS listed

in managed object **EXTERNAL-BIS-NEIGHBORS** is located on a common subnetwork with itself, a local BIS can include the ISO 8473 Complete Route Record parameter so that the recipient of the BISPDUs can determine whether the sending BIS is located on the same subnetwork as itself.

5.10 Routeing domain identifiers

Each routeing domain (RD) and routeing domain confederations (RDC) whose BIS(s) implement the protocol described in this International Standard shall have an unambiguous routeing domain identifier (RDI), which is a generic Network entity title, as described in ISO 7498.

An RDI is assigned statically in accordance with ISO/IEC 8348, and does not change based on the operational status of the routing domain. An RDI identifies a routing domain or a confederation uniquely, but does not necessarily convey any information about its policies or the identities of its members.

5.11 Formats of RDIs, NETs, and NSAP addresses

Within routing domains whose BISs implement the protocol defined in this International Standard, RDIs, NETs and NSAP addresses shall be structured as described in 7.1.

All Boundary Intermediate systems shall be able to generate and forward PDUs containing NSAP addresses or NETs whose abstract syntax is as described in ISO/IEC 8348.

5.12 Design objectives

The protocol described in this international standard was developed with a view towards satisfying certain design goals, while others specifically were not addressed by the mechanisms of this protocol.

5.12.1 Within the scope of the protocol

This International Standard supports the following design requirements:

Control Transit through an RD: It provides mechanisms to permit a given routing domain to control the ability of NPDUs from other routing domains to transit through itself.

Network Layer Protocol Compatibility: It does not require that any changes be made to the following existing Network layer protocols: ISO 8473, ISO 9542, ISO/IEC 10030, ISO/IEC 10589, and ISO/IEC 8208.

Autonomous Operation: It provides stable operation in the global OSIE where significant sections of the interworking environment will be controlled by disjoint entities.

Distributed Information Bases: It does not require a centralized global repository for either routing information or policy information.

QOS-based Routing: It provides the ability to select routes based on the QOS characteristics of the NPDUs.

Deliverability: It accepts and delivers NPDUs addressed to reachable routing domains and rejects NPDUs addressed to routing domains known to be unreachable.

Adaptability: It adapts to topological changes between routing domains, but not to traffic changes.

Promptness: It provides a period of adaptation to topological changes between domains that is a reasonable function of the maximum logical distance between any pair of routing domains participating in an instance of this protocol.

Efficiency: It is efficient in the use of both processing resources and memory resources; it does not create excessive routing traffic overhead.

Robustness: It recovers from transient errors such as lost or temporarily incorrect routing PDUs, and it tolerates imprecise parameter settings.

Stability: It stabilizes in finite time to "good routes", as long as there are no continuous topological changes or corruptions of the routing and policy information bases.

Heterogeneity: It is designed to operate correctly over a set of routing domains that may employ diverse intra-domain routing protocols. It is capable of running over a wide variety of subnetworks, including but not limited to: ISO/IEC 8208 and X.25 subnetworks, PSTN networks, and the OSI Data Link Service.

Availability: It will not result in inability to calculate acceptable inter-domain paths when a single point of failure happens for a pairing of topology and policy that have a *cut set* greater than one.

Fire walls: It will provide fire walls so that:

- Problems within one routing domain will not affect intra-domain routing in any other routing domain
- Problems within one routing domain will not affect inter-domain routing, unless they occur on internal inter-domain links
- Inter-domain routing will not adversely affect intra-domain routing.

Scaling: It imposes no upper limit on the number of routing domains that can participate in a single instance of this protocol.

Multiple Routing Administrations: It will accommodate inter-domain route calculations without regard to whether or not the participating routing domains are under control of one or several administrative authorities.

5.12.2 Outside the scope of the protocol

The following requirements are not within the design scope of this protocol:

User Data Security: The security of user data carried within an NPDU is outside the scope of this protocol. This protocol is not designed to serve as a substitute for conventional data security practices.

However, it can provide a security-related facility to the extent that route computation can be based upon the contents of the ISO 8473 security parameter.

Traffic Adaptation: It does not automatically modify routes based on the traffic load.

Guaranteed delivery: It does not guarantee delivery of all offered NPDUs.

Repair of Partitioned Routeing Domains: It does not use paths external to a partitioned routeing domain to repair the partition.

Repair of Partitioned Routeing Domain Confederations: It does not use paths external to a partitioned routeing domain confederation to repair the partition.

Shared Use of Inter-domain Links: It will not permit an external inter-domain link to carry intradomain traffic.

Suppression of Transient Loops: Although it provides mechanisms to detect and suppress looping of routeing information, it provides no mechanisms to detect or suppress transient looping of ISO 8473 NPDUs.

6 Structure of BISPDU

In this document, the term *BISPDU* (Boundary IS PDU) is used as a general term to refer to any of the PDUs defined in this clause.

Octets in a PDU are numbered starting from 1, in increasing order. Bits in an octet are numbered from 1 to 8, where bit 1 is the low-order 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. The more significant semi-octet of each pair of semi-octets in a given octet is encoded in the high order bit positions (8 through 5).

Values are given in decimal, and all numeric fields are unsigned, unless otherwise noted.

The types of PDUs used by this protocol are:

- OPEN PDU
- UPDATE PDU
- IDRP ERROR PDU
- KEEPALIVE PDU
- CEASE PDU
- RIB REFRESH PDU

6.1 Header of BISPDU

Each BISPDU has a fixed size header. There may or may not be a data portion following the header, depending on the PDU type.

The layout of the header fields and their meanings are shown below:

Inter-domain Routeing Protocol Identifier (1 octet)
BISPDU Length (2 octets)
BISPDU Type (1 octet)
Sequence (4 octets)
Acknowledgement (4 octets)
Credit Offered (1 octet)
Credit Available (1 octet)
Validation Pattern (16 octets)

The meaning and use of these fields are as follows:

Inter-domain Routeing Protocol Identifier: An architectural constant for use in identifying the protocol by the methods of ISO DTR 9577.

BISPDU Length: The BISPDU Length field is a 2 octet unsigned integer. It contains the total length in octets of this BISPDU, including both header and data portions. The value of the BISPDU Length field shall be at least **MinBISPDULength** octets, and no greater than the value carried in the **Maximum_PDU_Size** field of the OPEN PDU received from the remote BIS.

Further, depending on the PDU type, there may be other constraints on the value of the Length field; for example, a KEEPALIVE PDU must have a length of exactly 30 octets. No padding after the end of BISPDU is allowed, so the value of the Length field must be the smallest value required given the rest of the BISPDU.

BISPDU Type: The BISPDU Type field contains a one octet type code which identifies the specific type of the PDU. The supported type codes are:

BISPDU Type	Code
OPEN PDU	1
UPDATE PDU	2
IDRP ERROR PDU	3
KEEPALIVE PDU	4
CEASE PDU	5
RIB REFRESH PDU	6

All other BISPDU type codes are reserved for future extensions.

Sequence: The Sequence field contains a 4 octet unsigned integer that is the sequence number of this PDU. The procedures for generating sequence numbers for the various types of BISPDU are described in 7.7.4.

Acknowledgement: The Acknowledgment field is a 4 octet unsigned integer that contains the sequence number of the PDU that the sender last received correctly and in sequence number order.

Credit Offered: The contents of this field indicate the number of additional BISPDU's that the sender is willing to accept from the remote BIS; it is used by the flow control process described in 7.7.5.

Credit Available: The contents of this field indicate the number of additional BISPDU's that the sender is able to send to the remote BIS; it is used by the flow control process described in 7.7.5.

Validation Pattern: This 16-octet field is used to provide a validation function for the BISPDU. Depending upon the contents of the field "Authentication Code" of the OPEN PDU, this field can provide:

- data integrity for the contents of the BISPDU (see 7.7.1 and 7.7.3), or
- data integrity for the contents of the BISPDU plus authentication of the peer BIS (see 7.7.2).

6.2 OPEN PDU

The OPEN PDU is used by a BIS for starting a BIS-BIS connection. The first PDU sent by either side is an OPEN PDU.

The OPEN PDU contains a fixed header and the additional fields shown below:

Fixed Header
Version (1 octet)
Hold Time (2 octets)
Maximum PDU Size (2 octets)
Source RDI Length Indicator (1 octet)
Source RDI (variable)
RIB-AttsSet (variable)
Confed-IDs (variable)
Authentication Code (1 octet)
Authentication Data (variable)

The meaning and use of these fields are as follows:

Version: This one octet field contains the version number of the protocol. Its value is currently 1.

Hold Time: This field contains a 2 octet unsigned integer that is the maximum number of seconds that this BIS will allow the FSM for a given BIS-BIS connection to remain in the ESTABLISHED state without receipt of a KEEPALIVE, UPDATE, or RIB REFRESH PDU from the peer BIS. (See 7.6.1.4 and 7.20.5.)

Maximum PDU Size: This field contains a 2 octet unsigned integer that is the maximum number of octets that this BIS will accept in an incoming UPDATE PDU, IDRP ERROR PDU, or RIB REFRESH PDU.

Independent of this value, every BIS shall accept KEEPALIVE PDUs or CEASE PDUs of length 30 octets; every BIS shall also accept any OPEN PDU with length less than or equal to 3000 octets.

As a minimum, every BIS is required to support reception of all BISPDU's whose size is greater than or equal to **MinBISPDULength** octets and less than or equal to 1024 octets: that is, the minimum acceptable value for this field is 1024.

Source RDI Length Indicator: This one octet field contains the length in octets of the Source RDI field.

Source RDI: This variable length field contains the RDI of the routing domain in which the BIS that is sending this BISPDU is located.

RIB-AttsSet: This variable length field contains a list of all *RIB-Atts* that the local BIS is willing to support when communicating with the neighbor BIS (that is, the BIS to which this OPEN PDU is being sent). It contains an encoding of all or part of the information contained in managed object **RIBAttsSet** (See 7.3 and 7.10.1).

A BIS is not required to list all of its supported RIB-Atts in an OPEN PDU that is sent to a neighbor BIS located in an adjacent routing domain. It must include only those RIB-Atts that correspond to Adj-RIBs-Out that the local BIS will use to communicate with the neighbor BIS, and those that correspond to the RIB-Atts of the Adj-RIBs-In that the local BIS supports for storing UPDATE PDUs received from that neighbor BIS.

However, a BIS shall include all of the RIB-Atts listed in managed object **RIBAttsSet** in an OPEN PDU that is sent to another BIS located in its own routing domain. Failure to do so will result in an OPEN PDU error, as described in 7.20.2.

The encoding of this field is as follows:

Number of Non-empty RIB-Atts
Number of Distinguishing Attributes in First RIB-Att
First attribute, first RIB-Att
....
Last Attribute, first RIB-Att
Number of Distinguishing Attributes in Second RIB-Att
First attribute, second RIB-Att
....
Last Attribute, second RIB-Att
...
Number of Distinguishing Attributes in Nth RIB-Att
First attribute, Nth RIB-Att
....
Last Attribute, Nth RIB-Att
...
Number of Distinguishing Attributes in Last RIB-Att
First attribute, last RIB-Att
....
Last Attribute, last RIB-Att

The field *Number of RIB-Atts* is one octet long. It contains the total number of non-empty RIB-Atts supported by this BIS. Since every BIS supports an empty RIB-Att (see 7.10.1), the empty RIB-Att shall not be listed in the OPEN PDU.

If a BIS supports no RIB-Atts other than the empty RIB-Att, then the field *Number of Non-empty RIB-Atts* shall contain 0.

If the *Number of Non-Empty RIB-Atts* is non-zero, then the BIS supports all of the listed RIB-Atts plus the empty RIB-Att.

Each field *Number of Distinguishing Attributes in the Nth Set* is one octet long, and it contains the number of distinguishing attributes that are contained in the *Nth* RIB-Att. Since a RIB-Att consists of a set of distinguishing attributes, there is no significance to the order in which the distinguishing attributes of a given RIB-Att are listed.

Each of the individual attributes in a set is encoded as follows:

- a type specific distinguishing attribute (see 7.11.2) is encoded as a single octet, using the type code specified in 6.3 for the corresponding UPDATE PDU path attribute.

- a type-value-specific distinguishing attribute (see 7.11.2) is encoded as a triple <type, length, value>, using the type code for the attribute, the length in octets of the subsequent value, and the value itself. The value is encoded as follows for the following distinguishing attributes:

SECURITY: The value shall contain the Security Registration Identifier that identifies the Security Authority for which security related information is supported by this RIB-Att, encoded identically to the encoding of the SECURITY attribute specified in 6.3.1.14 including length fields, but with a zero length security information.

LOCALLY DEFINED QOS: The value shall comprise the NSAP Address Prefix of the QoS Authority and the QoS value that identifies the QoS measurement, encoded identically to the encoding of the LOCALLY DEFINED QOS attribute specified in 6.3.1.11 including length fields, but with a zero length metric.

Confed-IDs: This is a variable length field which reports the RDIs of all RDCs that this BIS is a member of. The encoding of this field is as follows:

Number of RDCs (1 octet)
Length of First RDI (1 octet)
RDI of first RDC
....
Length of Last RDI (1 octet)
RDI of last confederation

The 1 octet field *Number of RDCs* gives the number of RDCs of which this BIS is a member. A value of zero indicates that this BIS participates in no RDCs.

For each such confederation, the following fields give the length and RDI for each confederation, where each RDI is encoded according to the preferred binary encoding of ISO/IEC 8348.

Authentication Code: This 1-octet unsigned integer indicates the authentication mechanism being used:

- a) Code 1 indicates that the Validation Pattern field in the header of each BISPDUs contains an unencrypted checksum that provides data integrity for the contents of that BISPDUs. Its use is described in 7.7.1 Its use is described in 7.7.1.
- b) Code 2 indicates that the Validation Pattern field in the header of each BISPDUs provides both peer-BIS authentication and data integrity for the contents of the BISPDUs. The specific mechanism used to generate the validation

pattern is mutually agreed to by the pair of BISs, but is not specified by this International Standard. Its use is described in 7.7.2.

- c) Code 3 indicates that the Validation Pattern field in the header of each BISPDU contains an unencrypted checksum covering the concatenation of the contents of the BISPDU with untransmitted password string(s). Its use is defined in 7.7.3.

Authentication Data: The form and meaning of this field is a variable-length field that depends on the Authentication Code, as described in clause D.1. The length of the authentication data field can be determined from the Length field of the BISPDU header.

6.3 UPDATE PDU

An UPDATE PDU is used to advertise feasible routes to a neighbor BIS, or to withdraw multiple unfeasible routes from service (see 5.6). An UPDATE PDU may simultaneously advertise multiple feasible routes and withdraw multiple unfeasible routes from service. The UPDATE PDU always includes the fixed header, Unfeasible Route Count field, and Total Path Length Attributes field; it can optionally contain the other fields shown in Figure 5:

- if routes are being explicitly withdrawn from service, then the UNFEASIBLE ROUTE COUNT field will be non-zero, and the WITHDRAWN ROUTES fields will be present
- if feasible routes are being advertised, then the TOTAL PATH ATTRIBUTES LENGTH field will be non-zero, and the PATH ATTRIBUTES and NLRI fields will be present.

An UPDATE PDU can advertise multiple routes; a route is described by several path attributes, each of which is encoded as a 4-tuple. All path attributes contained in a given UPDATE PDU apply to the destinations carried in the Network Layer Reachability Information field of the UPDATE PDU.

An UPDATE PDU can list multiple routes to be withdrawn from service. Each such route is identified by its Route-ID, which unambiguously identifies the route in the context of the BIS-BIS connection in which it had been previously been advertised.

An UPDATE PDU that is used only to withdraw routes from service (but not to advertise any feasible routes) will not include Path Attributes or NLRI. Conversely, if an UPDATE PDU does not withdraw any routes from service, the UNFEASIBLE ROUTE COUNT field will contain the value 0, and WITHDRAWN ROUTES field will not be present.

The components of the UPDATE PDU are described below:

Fixed Header
Unfeasible Route Count (2 octets)
Withdrawn Routes (variable)
Total Path Attributes Length (2 octets)
Path Attributes (variable)
Network Layer Reachability Information (variable)

The use of these fields is as follows:

Unfeasible Route Count: This is a 2-octet long field that contains an integer whose value is equal to the number of ROUTE-IDs that are included in the subsequent WITHDRAWN ROUTES field. A value of 0 indicates that no routes are being withdrawn from service, and that the WITHDRAWN ROUTES field is not present in this UPDATE PDU.

Withdrawn Routes: This is a variable length field that contains a list of Route-IDs for the routes that are being withdrawn from service. Each Route-ID is 4 octets in length.

Total Path Attribute Length: The length of this field is 2 octets. It is the total length of all Path Attributes in the UPDATE PDU in octets.

Path Attributes: A variable length sequence of path attributes is present in every UPDATE PDU that is used to advertise a feasible route.

Network Layer Reachability Information: A variable length field that lists the destinations for the feasible routes that are being advertised in this UPDATE PDU.

6.3.1 Path attribute encoding

Each path attribute is a 4-tuple of variable length—<flag, type, length, value>. The elements are used as follows:

- **Flag:**

The first element of each attribute is a one octet field:

 - The high-order bit (bit 8) of the attribute flags octet is the Optional bit. If it is set to 1, the attribute is optional; if set to 0, the attribute is well-known.
 - The second high-order bit (bit 7) of the attribute flags octet is the Transitive bit. It defines whether an optional attribute is transitive (if set to 1) or non-transitive (if set to 0). For well-known attributes, the Transitive bit shall be set to 1.

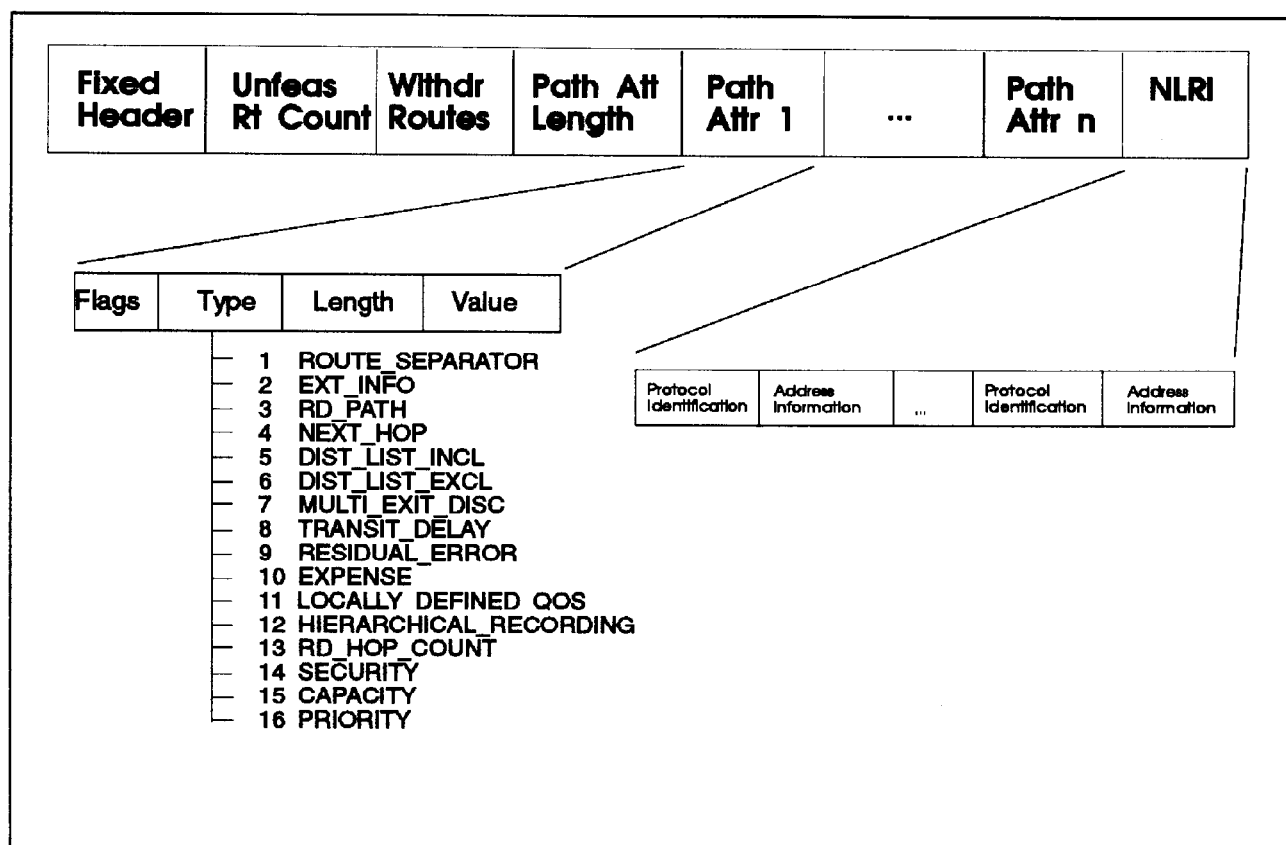


Figure 5 – Structure of the UPDATE PDU

- The third high-order bit (bit 6) of the attribute flags octet is the Partial bit. It defines whether the optional transitive attribute is partial (if set to 1) or complete (if set to 0). For well-known attributes and for optional non-transitive attributes the Partial bit shall be set to 0.
- The lower order five bits (1 through 5) of the attribute flags octet are reserved. They shall be transmitted as 0 and shall be ignored when received.

— **Type:**

The second element of each attribute is a one octet field which contains the type code for the attribute. Currently defined attribute type codes are discussed in clause 7.11.

NOTE 4: It is not the intention of this International Standard to define globally understood path attributes for type codes greater than value 128. Such codes are reserved for local use.

— **Length:**

The third field of each path attribute is 2 octets in length; it contains the length in octets of the immediately following Value field.

— **Value:**

The remaining octets of each path attribute field contain the value of the attribute, which is interpreted according to the attribute flags and the attribute type code. The supported attribute values and their encodings are defined below.

6.3.1.1 ROUTE_SEPARATOR (Type Code 1)

ROUTE_SEPARATOR is a well-known mandatory attribute whose length is 5 octets. One ROUTE_SEPARATOR attribute shall be present for each feasible route that is advertised in an UPDATE PDU. Multiple ROUTE_SEPARATORS may appear in a single UPDATE PDU. Each one provides a 2-tuple <ROUTE_ID, LOCAL_PREF> for the route whose distinguishing attributes immediately follow. It is encoded as shown below:

ROUTE-ID 1 (4 octets)
LOCAL-PREF 1 (1 octet)

a) ROUTE-ID:

A four octet field that contains the route identifier for the route associated with the RIB-Attr composed of the set of distinguishing path attributes that appear between itself and either the next

ROUTE_SEPARATOR attribute or the end of the UPDATE PDU.

- b) LOCAL_PREF: A one octet field that contains the local preference value for route.

The usage of this attribute is defined in 7.12.1.

6.3.1.2 EXT_INFO (Type Code 2)

EXT_INFO is a well-known discretionary attribute that has a length of zero octets; its presence indicates that some (or all) of the path attributes or Network Layer Reachability Information contained in this UPDATE PDU have been obtained by methods not specified by IDRP. Conversely, its absence indicates that all path attributes and NLRI have been learned by methods defined within IDRP. Its usage is defined in 7.12.2

6.3.1.3 RD_PATH (Type Code 3)

The RD_PATH attribute is a well-known mandatory attribute composed of a series of RD path segments. Each path segment is represented by a triple <path segment type, path segment length, path segment value>.

The path segment type is a 1-octet long field, with the following values defined:

Segment Type	Value
RD_SET	1
RD_SEQ	2
ENTRY_SEQ	3
ENTRY_SET	4

An RD_SEQ and a ENTRY_SEQ provide a list of RDIs, for routing domains or for confederations respectively, in the order that the routing information has travelled through them. An RD_SET and an ENTRY_SET provide an unordered list of RDIs, for routing domains or for confederations respectively; the routing information has not necessarily travelled through all of the listed domains or confederations.

The path segment length is a two octet field containing the length in octets of the path segment value field.

The path segment value field contains one or more 2-tuples <length, RDI>. *Length* is a one octet long field that contains the length of the RDI in octets; *RDI* itself is encoded according to the preferred binary encoding of ISO/IEC 8348.

Usage of this attribute is defined in 7.12.3.

6.3.1.4 NEXT_HOP (Type Code 4)

This is a well-known discretionary attribute that can be used for two principal purposes:

- to permit a BIS to advertise a different BIS's NET in the "NET of Next Hop" field
- to allow a given BIS to report some or all of the SNPAs that exist within the local system

It is encoded as shown below:

IDRP_Server_Allowed (1 octet)

followed by one or more of the following:

Proto_type (1 octet)
Proto_length (1 octet)
Protocol (variable)
Length of NET (1 octet)
NET of Next Hop (variable)
Number of SNPAs (1 octet)
Length of first SNPA(1 octet)
First SNPA (variable)
Length of second SNPA (1 octet)
Second SNPA (variable)
...
Length of Last SNPA (1 octet)
Last SNPA (variable)

The use and meaning of these fields are as follows:

IDRP_Server_Allowed: The value X'FF' indicates the recipient of this UPDATE PDU has the option of advertising (in its own outbound UPDATE PDUs) the NET and SNPA information learned from this UPDATE PDU. If the value is not X'FF', then the recipient of this UPDATE PDU shall not advertise the NET and SNPA information learned from this UPDATE PDU.

Proto_type: This field indicates the type of protocol identifier that follows:

- ISO/IEC TR 9577 IPI/SPI
- ISO 8802 LSAP

For use with ISO 8473, this field shall contain the value 1.

Proto_length: This field indicates the length in octets of the protocol identifier that follows. For use with ISO 8473, this field shall contain the value 1.

Protocol: This field carries the identity of the protocol associated with the address information that follows. For use with ISO 8473, this field shall contain the value X'81'.

Length of NET: A 1 octet field whose value expresses the length of the "NET of Next Hop" field as measured in octets

NET of Next Hop: A variable length field that contains the NET of the next BIS on the path to the destination system

Number of SNPA: A 1 octet field which contains the number of distinct SNPAs to be listed in the following fields. The value 0 may be used to indicate that no SNPAs are listed in this attribute.

Length of Nth SNPA: A 1 octet field whose value expresses the length of the "Nth SNPA of Next Hop" field as measured in semi-octets

Nth SNPA of Next Hop: A variable length field that contains an SNPA of the BIS whose NET is contained in the "NET of Next Hop" field. The field length is an integral number of octets in length, namely the rounded-up integer value of one half the SNPA length expressed in semi-octets; if the SNPA has an contains an odd number of semi-octets, a value in this field will be padded with a trailing all-zero semi-octet.

Its usage is defined in 7.12.4. A conforming IDRP implementation may ignore next hop information for all protocols other than ISO 8473. The Decision and Forwarding processes of IDRP for use with protocols other than ISO 8473 are outside the scope of this international standard.

NOTE 5: The ISO 8802 LSAP format is intended for use with protocols that are identified directly by the use of a particular LSAP value; in such cases, the value of the Proto_length field should be 1.

The ISO/IEC TR 9577 IPI/SPI format may include additional information beyond the IPI/SPI value in the Protocol field; for example, if the IPI/SPI value is X'80', an IEEE 802.1a SNAP header might be included.

6.3.1.5 DIST_LIST_INCL (Type Code 5)

The DIST_LIST_INCL is a well-known discretionary attribute that contains a list of the RDIs of routeing domains and confederations to which the Network Layer Reachability Information contained in that UPDATE PDU may be distributed. Its usage is described in 7.12.5.

Each RDI in the list is encoded as a <length, value> pair, using the following fields:

Count (1 octet)
Length (1 octet)
Value (variable)
...
Length (1 octet)
Value (variable)

The use and meaning of these fields are as follows:

- Count:**
The count field is an integer that gives the number of RDIs in the list, where each RDI is represented by a <length, value> pair as described below.
- Length:**
The length field indicates the length in octets of the following RDI.
- Value:**
The value field contains the RDI, encoded according to the preferred binary encoding of ISO/IEC 8348.

The DIST_LIST_INCL attribute shall not be present together with the DIST_LIST_EXCL attribute in the same UPDATE PDU.

6.3.1.6 DIST_LIST_EXCL (Type Code 6)

The DIST_LIST_EXCL is a well-known discretionary attribute that contains a list of the RDIs of routeing domains and confederations to which the Network Layer Reachability Information contained in that UPDATE PDU shall not be distributed. Its usage is described in 7.12.6.

Each RDI in the list is encoded as a <length, value> pair, using the following fields:

Count (1 octet)
Length (1 octet)
Value (variable)
...
Length (1 octet)
Value (variable)

The use and meaning of these fields are as follows:

- Count:**
The count field is an integer that gives the number of RDIs in the list, where each RDI is

represented by a <length, value> pair as described below.

b) Length:

The length field indicates the length in octets of the following RDI.

c) Value:

The value field contains the RDI, encoded according to the preferred binary encoding of ISO/IEC 8348.

The DIST_LIST_EXCL attribute shall not be present together with the DIST_LIST_INCL attribute in the same UPDATE PDU.

6.3.1.7 MULTI-EXIT_DISC (Type Code 7)

MULTI-EXIT_DISC (Multi-exit Discriminator) is an optional non-transitive attribute that is a 1 octet non-negative integer. The value of this attribute may be used by a BIS's decision process to discriminate between multiple exit points to an adjacent routing domain. Its usage is defined in 7.12.7.

6.3.1.8 TRANSIT DELAY (Type Code 8)

The TRANSIT DELAY is a well-known discretionary attribute that has a length of 2 octets, and is used to notify a BIS that the path to destination has transit delay determined by the value of the attribute. Its usage is defined in 7.12.8.

6.3.1.9 RESIDUAL ERROR (Type Code 9)

The RESIDUAL ERROR is a well-known discretionary attribute that has a length of 4 octets, and is used to notify a BIS that the path to destination has residual error probability determined by the value of the attribute. Its usage is defined in 7.12.9.

6.3.1.10 EXPENSE (Type Code 10)

The EXPENSE is a well-known discretionary attribute that has a length of 2 octets, and is used to notify a BIS that the path to destination has expense determined by the value of the attribute. Its usage is defined in 7.12.10.

6.3.1.11 LOCALLY DEFINED QOS (Type Code 11)

This is a well-known discretionary attribute whose variable length field contains the parameters associated with a QOS related path attribute defined by a QOS authority.

The parameters of this attribute identify the QOS authority, the name of the QOS measurement, and, optionally, a metric associated with that measurement. The syntax and semantics of the QOS meas-

urement and its metric are outside the scope of this International Standard, and are specified by the responsible QOS Authority.

It is encoded as follows:

NSAP prefix length (1 octet)
NSAP prefix (variable)
QOS length (1 octet)
QOS value (variable)
Metric length (1 octet)
Metric value (variable)

The use and meaning of the fields is as follows:

a) NSAP prefix length:

The length field indicates the length in bits of the NSAP address prefix (see 7.1.2.1). A length of zero indicates a prefix that matches all NSAPs.

b) NSAP prefix:

If the Length field specifies an integral number of octets, then the Prefix field shall contain only the NSAP address prefix encoded according to 7.1.2.1. If the Length field does not specify an integral number of octets, then the Prefix field shall contain the NSAP address prefix encoded according to 7.1.2.1, followed by enough trailing zeroes to make the end of the field fall on an octet boundary.

This field identifies the QoS Authority and comprises an NSAP Address Prefix when the QoS Authority is also an NSAP Addressing Authority (see ISO 8473 clauses 7.5.6.1 and 7.5.6.2); the NSAP Address Prefix is that of the QoS Authority's Addressing Subdomain. When the QoS Authority is not also an NSAP Addressing Authority, then this field contains a NET that uniquely identifies the QoS Authority.

c) QOS length:

Gives the length in octets of the QOS value field.

d) QOS value:

Contains the value of the QOS parameter.

e) Metric length:

Gives the length in octets of the metric value field. A length of zero is a permitted value.

f) Metric value:

Contains a positive integer that is the value of the metric associated with the path being advertised (that is, its "cost")

Its usage is defined in 7.12.11.

6.3.1.12 HIERARCHICAL RECORDING (Type Code 12)

This is a well-known discretionary attribute. It contains a 1-octet field used by members of a routing domain confederation to control the transitivity of NPDUs through the confederation. It is encoded as follows:

Flag (1 octet)

Its usage is defined in 7.12.12.

6.3.1.13 RD_HOP_COUNT (Type Code 13)

The RD_HOP_COUNT is a well-known mandatory attribute that contains a 1 octet long field. It contains an unsigned integer that is the upper bound on the number of routing domains through which this UPDATE PDU has travelled. Its usage is defined in 7.12.13.

6.3.1.14 SECURITY (Type Code 14)

This is a well-known discretionary attribute whose variable length field contains the parameters associated with a security related path attribute defined by a Security Authority.

The parameters of this attribute identify the Security Authority, and can also contain additional security related information, which may identify the protection provided by the route, a set of Security Labels associated with the route, or both. The syntax and semantics of the security related information are outside the scope of this International Standard, and are specified by the responsible Security Authority.

The encoding of the attribute is as follows:

Security Registration ID Length (1 octet)
Security Registration ID (variable)
Security Information Length (1 octet)
Security Information (variable)

The use and meaning of the fields is as follows:

a) Security Registration ID Length:

This field contains the length in octets of the Security Authority's Security Registration Identifier.

b) Security Registration ID:

This variable length field contains the Security Registration Identifier of the Security Authority responsible for defining the following security information.

c) Security Information Length:

This field contains the length in octets of the Security Information, as a non-zero unsigned integer. A value of zero is not permitted.

d) Security Information:

This variable length field contains an integral number of octets comprising security related information specified by the Security Authority identified above. The syntax and semantics of this information are outside of the scope of this International Standard.

6.3.1.15 CAPACITY (Type code 15)

CAPACITY is a well-known mandatory attribute that has a length of 1 octet, and is used to denote the relative capacity of the RD_PATH for handling traffic. High values indicate a lower traffic handling capacity than do low values. Its usage is defined in 7.12.15.

6.3.1.16 PRIORITY (Type Code 16)

PRIORITY is a well-known discretionary attribute that is 1 octet in length. The content of the field is an integer in the range from 0 to 14. It enables paths to be distinguished on the basis of which values of the ISO 8473 priority parameter (see ISO 8473, sub-clause 7.5.7). As in ISO 8473, the value 0 indicates the normal (default) priority, and the values 1 through 14 indicate higher priorities. A particular value of this parameter, *m*, means that the BIS will not forward any ISO 8473 PDUs whose priority parameter has a value less than *m*. Detailed usage rules are presented in 7.12.16.

6.3.2 Network layer reachability information

This variable length field contains a list of reachable destinations encoded as zero or more 5-tuples of the form <Proto_type, Proto_length, Protocol, Addr_length, Addr_info>, whose fields are described below:

Proto_type (1 octet)
Proto_length (1 octet)
Protocol (variable)
Addr_length (2 octets)
Addr_info (variable)

The use and meaning of these fields are as follows:

Proto_type: This field indicates the type of protocol identifier that follows:

- 1 ISO/IEC TR 9577 IPI/SPI
- 2 ISO 8802 LSAP

For use with ISO 8473, this field shall contain the value 1.

Proto_length: This field indicates the length in octets of the protocol identifier that follows. For use with ISO 8473, this field shall contain the value 1.

Protocol: This field carries the identity of the protocol associated with the address information that follows. For use with ISO 8473, this field shall contain the value X'81'.

Addr_length: This field specifies the total length in octets of the address information that follows.

Addr_info: This field contains a list of reachable address prefixes. The encoding of this field is specific to each protocol supported.

For use with ISO 8473, this field shall be encoded as one or more 2-tuples of the form <length, prefix>, whose fields are described below:

- a) The Length field is one octet long, and it indicates the length in bits of the address prefix. A length of zero indicates a prefix that matches all addresses.
- b) The Prefix field has a variable length, and it contains an address prefix. If the Length field does not specify an integral number of octets, then the Prefix field shall be padded with trailing zero bits to the next octet boundary. For purposes of use with ISO 8473, this field shall contain an NSAP address prefix encoded according to 7.1.2.1.

A conforming IDRP implementation may ignore NLRI for all protocols other than ISO 8473. The Decision and Forwarding processes of IDRP for use with protocols other than ISO 8473 are outside the scope of this international standard.

NOTE 6: The ISO 8802 LSAP format is intended for use with protocols that are identified directly by the use of a particular LSAP value; in such cases, the value of the proto_length field should be 1.

The ISO/IEC TR 9577 IPI/SPI format may include additional information beyond the IPI/SPI value in the Protocol field; for example, if the IPI/SPI value is X'80', an IEEE 802.1a SNAP header might be included.

6.4 IDRP ERROR PDU

IDRP ERROR PDUs report error conditions which have been detected by the local BIS. In addition to its fixed header, the IDRP ERROR PDU contains the following fields:

Fixed Header
Error Code (1 octet)

Error Subcode (1 octet)
Data (variable)

The use of these fields is as follows:

Error code: The Error code field is 1 octet long, and shall be present in every IDRP ERROR PDU. It describes the type of error. The following error codes are defined:

Error Code	Value
OPEN_PDU_Error	1
UPDATE_PDU_Error	2
Hold_Timer_Expired	3
FSM_Error	4
RIB_REFRESH_PDU_Error	5

All errors are fatal to the BIS connection.

Error subcode: The Error subcode is one octet long, and shall be present in every IDRP ERROR PDU. The error subcode provides more specific information about the nature of the reported error. A given IDRP ERROR PDU may report only one error subcode for the indicated error code. The supported error subcodes are as follows:

- a) OPEN PDU Error subcodes:

Subcode	Value
Unsupported_Version_Number	1
Bad_Max_PDU_Size	2
Bad_Peer_RD	3
Unsupported_Authentication_code	4
Authentication_Failure	5
Bad_RIB_AttsSet	6
RDC_Mismatch	7

- b) UPDATE PDU Error subcodes:

Subcode	Value
Malformed_Attribute_List	1
Unrecognized_Well-known_Attribute	2
Missing_Well-known_Attribute	3
Attribute_Flags_Error	4
Attribute_Length_Error	5
RD_Routeing_Loop	6
Invalid_NEXT_HOP_Attribute	7
Optional_Attribute_error	8
Invalid_Reachability_Information	9
Misconfigured_RDCs	10
Malformed_NLRI	11
Duplicated_Attributes	12
Illegal_RD_Path_Segment	13

- c) Hold_Timer_Expired Error Subcodes:

Subcode	Value
NULL	0

d) FSM_Error Error Subcodes:

When an FSM Error (see 7.6.1) has occurred, the first semi-octet of the error subcode carries the type number of the BISPDU that should not have been received and the second semi-octet encodes the state that the FSM was in when the reception took place. The BISPDU type numbers are defined in 6.1; the FSM states are encoded as follows:

FSM State	Encoded Value
CLOSED	1
OPEN-RCVD	2
OPEN-SENT	3
CLOSE-WAIT	4
ESTABLISHED	5

e) RIB_REFRESH_PDU_Error Subcodes:

Subcode	Value
Invalid_OpCode	1
Unsupported_RIB-Atts	2

Data: This variable length field contains zero or more octets of data to be used in diagnosing the reason for the IDRP ERROR PDU. The contents of the Data field depends upon the error code and error subcode.

Note that the length of the Data field can be determined from the Length field of the BISPDU header. The minimum length of the IDRP ERROR PDU is 32 octets, including BISPDU header.

6.5 KEEPALIVE PDU

A KEEPALIVE PDU consists of only a PDU header and has a length of 30 octets.

A BIS can use the periodic exchange of KEEPALIVE PDUs with an adjacent BIS to verify that the peer BIS is reachable and active. KEEPALIVE PDUs are exchanged often enough as not to cause the hold time advertised in the OPEN PDU to expire. The maximum time between KEEPALIVE PDUs shall be one third of the Hold Time interval as advertised in the Hold Time field of the OPEN PDU.

A KEEPALIVE PDU may be sent asynchronously to acknowledge the receipt of other BISPDU. Sending a

KEEPALIVE PDU does not cause the sender's sequence number to be incremented.

6.6 CEASE PDU

A CEASE PDU consists of only a PDU header and has length of 30 octets.

A CEASE PDU is used by the originating BIS to instruct the peer BIS to close the BIS-BIS connection.

Receipt of a CEASE PDU will cause the BIS to close down the connection with the BIS that issued it, as described in 7.6.2.

6.7 RIB REFRESH PDU

The RIB REFRESH PDU is used to allow a BIS to send a refresh of the routing information in an Adj-RIB-Out to a neighbor BIS, or to solicit a neighbor BIS to send a refresh of its Adj-RIB-Out to the local BIS. The RIB REFRESH PDU contains a fixed header and also the additional fields shown below:

Fixed Header
OpCode (1 octet)
RIB-Atts (variable)

The use and meaning of these fields is as follows:

There are three OpCode values defined:

Code	Operation
1	RIB Refresh Request
2	RIB Refresh Start
3	RIB Refresh End

The RIB-Atts field contains the RIB-Atts of the Adj-RIB-In for which a refresh is being requested. This field is encoded in the same way that the RIB-AttsSet field of the OPEN PDU is encoded.

Its usage is defined in 7.10.3.

7 Elements of procedure

This clause explains the elements of procedure used by the protocol specified in this International Standard; it also describes the naming conventions and system deployment practices assumed by this protocol.

7.1 Naming and addressing conventions

Correct operation of this protocol requires that all NSAP-addresses, NETs, and RDIs shall be encoded according to the preferred binary encoding method of ISO/IEC 8348.

7.1.1 Interpretation of address information

IDRP does not assume or require any particular internal structure for the address. That is, as long as the routing domain administrator assigns values within this field that are consistent with the deployment constraints of 7.2.2, the protocol specified in this International Standard will operate correctly.

7.1.2 NSAP address prefixes

NSAP address prefixes provide a compact method for identifying groups of systems that reside in a given routing domain or confederation. A prefix may have a length that is either smaller than, or the same size as, the base NSAP address.

NOTE 7: At one extreme, for example, the largest NSAP address prefix will be identical with the full NSAP address—in this case, it would identify a single system rather than a group of systems. At another extreme, the NSAP prefix may be based on only a portion of NSAP address's IDP—in this case, it will identify a large group of systems.

7.1.2.1 Encoding of prefixes

The encoded form of an NSAP address prefix is obtained by encoding the prefix (expressed in its abstract syntax) according to the preferred binary encoding of ISO 8348, unless the end of the prefix falls within the IDP. In this case, each decimal digit in the prefix shall be encoded as the corresponding semi-octet in the range 0000-1001 and no padding characters shall be inserted.

The length of an encoded prefix is denominated in bits. The prefix shall end on a boundary that is legal in the abstract syntax of the address family from which it is derived. For example, the encoding of a prefix whose DSP is expressed in decimal syntax must end on a semi-octet boundary, while the encoding of a prefix whose DSP is expressed in binary syntax can end on an arbitrary bit boundary.

7.1.2.2 Prefix matching

As part of the forwarding process, a BIS must match the destination NSAP address from an ISO 8473 NPDU against the NSAP address prefixes that are used to identify the Forwarding Information Bases. An NSAP address prefix that extends into the DSP shall be compared directly against the encoded NSAP address, including any padding characters that may be

present. An NSAP address prefix which does not extend into the DSP shall be compared against the derived quantity NSAP', which is obtained from the encoded NSAP address by removing all padding characters that were inserted by the binary encoding process of ISO 8348.

The existence of a match shall be determined as follows:

- a) If the encoded NSAP address (or NSAP') contains fewer bits than the NSAP address prefix, then there is no match.
- b) If the encoded NSAP address (or NSAP') contains at least as many bits as the NSAP address prefix, and all bits of the NSAP address prefix are identical to the corresponding leading bits of the encoded NSAP address (or NSAP'), there is a match. Otherwise, there is no match.

In cases where a given NSAP address matches several address prefixes, the match with the longest prefix shall take precedence. For purposes of prefix matching, the length of a prefix is defined to be the number of bits that it contains.

NOTE 8: Any implementation of a matching process that satisfies the requirements listed above may be used. The key point is that matching process must be aware of whether or not the NSAP address prefix extends into the DSP, and must then either include or exclude padding characters from the encoded NSAP address, as defined above.

7.2 Deployment guidelines

7.2.1 Minimum configuration of an RD

To participate in this inter-domain routing protocol, a routing domain must, as a minimum:

- contain at least one Boundary Intermediate system
- contain at least one BIS capable of delivering NPDUs to the intra-domain routing function, if the RD contains End systems.

7.2.2 Deployment of ISs and ESs

End systems and intermediate systems may use any NSAP address or NET that has been assigned under ISO/IEC 8348 guidelines. However, correct and efficient operation of this protocol can only be guaranteed if the following additional guidelines are met:

- a) An NSAP prefix carried in the NLRI field of route originated by a BIS in a given routing domain should be associated with only that routing domain: that is, no system identified by the prefix should reside in a different routing domain. Ambiguous routing may result if several

routing domains originate routes whose NLRI fields contain identical NSAP address prefixes, since this would imply that the same system(s) are simultaneously located in several routing domains. (As noted in 7.1.2.2, prefixes may be discriminated by both content and length.)

- b) Several different NSAP prefixes may be associated with a single routing domain identifier (RDI). Thus, it is possible to construct a single routing domain which contains a mix of systems which use NSAP addresses assigned by several different naming authorities.

The protocol defined in this International Standard assumes that the guidelines listed above have been satisfied, but it contains no means to verify that this is so. Therefore, such verification will be the responsibility of the administrators of routing domains.

7.3 Domain configuration information

Correct operation of the protocol described in this international standard assumes that a minimum amount of information is available to both the inter-domain and the intra-domain routing protocols. The information is static in nature, and is not expected to change frequently. This protocol specifies the content of the information in terms of managed objects.

The information required by a BIS that implements the protocol described in this International Standard is:

- a) **Location and identity of adjacent intra-domain ISs:**

The managed object **intraIS** lists the NETs of systems to which the local BIS may deliver an inbound NPDU whose destination lies within the BIS's routing domain. The managed object contains the NETs of ISs that support the intra-domain routing protocol and are located on the same common subnetwork as the local BIS. In particular, if the BIS participates in the intra-domain routing protocol (that is, the protocol machines for both intra- and inter-domain routing are located in the same real system), then the NET of the local BIS will be listed in the managed object **intraIS**.

- b) **Location and identity of BISs in the domain:**

This information permits a BIS to identify all other BISs located within its routing domain. This information is contained in the managed object **internalBIS**, which contains a set of NETs which identify the BISs in the domain.

- c) **Location and identity of BISs in adjacent domain:**

Each BIS needs information to identify the NETs of each BIS located in an adjacent RD and reachable via a single subnetwork hop. This information

is contained in managed object **externalBISNeighbor**, which consists of a list of NETs.

- d) **NETs and NSAP addresses of all systems in the routing domain:**

This information is used by the BIS to construct its network layer reachability information. This information is contained within the managed object **internalSystems**, which lists the address prefixes of the systems contained within the routing domain.

- e) **Local RDI:**

This information is contained in managed object **localRDI**; it is the RDI of the routing domain in which the BIS is located.

- f) **RIBAttsSet:**

This managed object lists all of the RIB-Atts (RIB attributes) which are supported by BISs located in this routing domain. As noted in 7.10.1, a RIB-Att is a set of Distinguishing Attributes.

- g) **Routing Domain Configuration:**

Managed object **rdcConfig** identifies all the routing domain confederations (RDCs) to which the RD of the local BIS belongs, and it describes the nesting relationships that are in force between them. It is contained in managed object **rdcConfig**.

- h) **Local SNPs:**

Managed object **localSNPA** contains a list of the SNPs of the BIS.

7.4 Advertising NLRI

The NLRI field in an UPDATE PDU contains information about the NSAP addresses of systems that reside within a given routing domain or whose NSAP addresses are under control of the administrator of that routing domain; it should not be used to convey information about the operational status of these systems. The information in the NLRI field is intended to convey static administrative information rather than dynamic transient information: for example, it is not necessary to report that a given system has changed its status from online to offline.

The following guidelines for inclusion of NSAP address prefixes in the NLRI field of UPDATE PDUs originated within a given routing domain will provide efficient operation of this protocol:

- a) NSAP addresses that are within the control of the administrator of a given routing domain may be reported in the NLRI field for that routing domain. The NSAP prefixes can provide information about systems that are online, systems that

are offline, or unallocated NSAP addresses. The ability to include unallocated NSAP addresses and NSAP addresses of offline systems in the NLRI allows a routing domain administrator to advertise compact prefixes, thus minimizing the amount of data carried in the BISPDU.

- b) NSAP addresses that are known to correspond to systems that are not under control of the routing domain administrator should not be included in the NLRI field for that routing domain. By not listing NSAP address prefixes that identify systems that are not under his control, a routing domain administrator will comply with the deployment guidelines for ISs and ESs as described in 7.2.2, thus assuring correct operation of this protocol.
- c) For efficient operation of this protocol, the WITHDRAWN ROUTES field should not be used to report the NLRI of systems in the local routing domain that are offline. This field should be used only to advertise NSAP prefixes that are no longer under control of the administrator of the local routing domain, regardless of whether such systems are online or offline.

NOTE 9: Although the protocol in this International Standard will operate correctly if each system is reported individually as a maximum-length NSAP address prefix, this will result in a large amount of routing information, and hence can result in inefficient operation of this protocol.

This protocol provides no means to verify that the preceding guidelines are followed. However, it is within the prerogative of the administrator of any routing domain to implement policies that ignore UPDATE PDUs that contain an excessive amount of NLRI information or that can cause inefficient operation of this protocol.

7.5 Receive process

Within the Network layer, the IDRP protocol is located above the ISO 8473 protocol. Therefore, IDRP relies upon ISO 8473 to perform the initial processing of incoming PDUs. After processing the input NPDU, the ISO 8473 protocol machine that resides within the receiving BIS will deliver:

- a) BISPDU to the IDRP Receive Process, or
- b) ISO 8473 NPDU to the IDRP CLNS Forwarding Process.

The ISO 8473 protocol machine shall process inbound NPDU according to the appropriate ISO 8473 functions.

If the NPDU contains an SPI that identifies IDRP, and the NPDU's source address identifies any system listed in managed objects **internalBIS** or **externalBISNeighbor**, then the data part of the NPDU

contains a BISPDU. This BISPDU shall be passed to the IDRP finite state machine defined in 7.6.1.

However, if the source address of the NPDU is not an NET listed in the **internalBIS** or the **externalBISNeighbor** attributes of the idrpConfig Managed Object, then:

- a) If the NPDU is an OPEN BISPDU without errors, then the BIS shall send a notification ("connectRequestBISUnknown") to Systems Management,
- b) If the NPDU is any other BISPDU with or without errors, then the BIS shall send a notification ("PacketBomb") to Systems Management.

The NPDU shall then be discarded.

If the SPI identifies ISO 8473, decapsulate the inner PDU and pass it back to the ISO 8473 protocol machine. (This step permits iterations on multiply encapsulated NPDU, which may occur, for example, as described in 8.4, item b1.)

7.6 BIS-BIS connection management

The protocol described in this International Standard relies on the underlying Network layer service to establish a full-duplex communications channel between each pair of BISs.

7.6.1 BIS finite state machines

A BIS shall maintain one finite state machine (FSM) for each BIS-BIS connection that it supports, and each FSM in a given BIS shall run independently of one another. A BIS-BIS connection will progress through a series of states during its lifetime, which are summarized in the state table shown in Table 2.

BISPDU passed to this finite state machine are subject the flow control procedures of 7.7.5 if the FSM is in the ESTABLISHED state. When the FSM is in the ESTABLISHED state, only BISPDU that are not discarded by the flow control process are processed by the FSM. In all other states, all BISPDU are processed directly by the finite state machine without being subject to flow control procedures.

In describing the FSM transitions in response to receipt of BISPDU, the following shorthand notation is used:

- a) *Receive with no errors* means that the none of the error conditions defined in the appropriate subclause of 7.20 have been detected.
- b) *Receive with errors* means that an error condition defined in the appropriate subclause of 7.20 has been detected.

It is possible to receive a BISPDU which is properly formed, but which normally should not be received while the FSM is in the given state. Such an event constitutes an *FSM Error*. If an *FSM Error* can occur for a given state, it is shown in the description of that state.

7.6.1.1 CLOSED State

Initially the BIS Finite State Machine is in the CLOSED state. The CLOSED state exists when no BIS-BIS connection exists and there is no connection record allocated.

While in the CLOSED state, the BIS shall take the following actions:

- a) If the BIS receives a **deactivate**, no action shall be taken and the FSM shall remain in the CLOSED state.
- b) If the FSM receives an **activate**, the local BIS shall generate an initial sequence number (see 7.7.4), and shall send an OPEN PDU to the remote BIS. The sequence field of the OPEN PDU shall contain the Initial Sequence Number (ISN); the Acknowledgement and Credit Available fields shall contain the value 0; and the Credit Offered field shall contain the initial flow control credit. The FSM shall enter the OPEN-SENT state.
- c) If the managed object **ListenForOPEN** is TRUE, and the BIS receives an OPEN PDU with no errors, then the local BIS shall generate an initial sequence number (see 7.7.4) and shall send an OPEN PDU to the remote BIS. The sequence field of the OPEN PDU shall contain the Initial Sequence Number (ISN), the Acknowledgement field shall acknowledge the received OPEN PDU, the Credit Available field shall be set according to the procedures of 7.7.5b, and the Credit Offered field shall contain the initial flow control credit. The FSM shall then change its state to OPEN_RCVD.
- d) If the managed object **ListenForOPEN** is TRUE and the BIS receives any BISPDU other than an OPEN PDU with no errors, or if the managed object **ListenForOPEN** is FALSE and the BIS receives any BISPDU, with or without errors, the BIS shall ignore the BISPDU and the FSM shall remain in the CLOSED state.

7.6.1.2 OPEN-SENT State

While in the OPEN-SENT state, the BIS shall take the following actions:

- a) If the FSM receives an **activate**, the BIS shall ignore it, and the FSM shall remain in the OPEN-SENT state.

- b) If the FSM receives a **deactivate**, the BIS shall send a CEASE PDU to its peer, and the FSM shall enter the CLOSE-WAIT state.
- c) If the BIS receives a BISPDU with header errors (see 7.20.1), it shall log the error event, and the FSM shall remain in the OPEN-SENT state.
- d) If the BIS receives an OPEN PDU with errors (see 7.20.2), it shall send an IDRP ERROR PDU to the adjacent BIS, acknowledging the remote BIS's OPEN PDU. The FSM shall then enter the CLOSE-WAIT state.
- e) If the BIS receives an OPEN PDU with no errors that does not acknowledge its own previously sent OPEN PDU, then the local BIS shall resend its own OPEN PDU with the same sequence number and with an acknowledgement of the remote BIS's OPEN PDU. The value of the Credit Available field shall be set according to the procedures of 7.7.5b. The FSM shall then change its state to OPEN-RCVD.
- f) If the BIS receives an OPEN PDU with no errors that acknowledges its own previously sent OPEN PDU, the local BIS shall send a KEEPALIVE, RIB REFRESH, or UPDATE PDU that acknowledges the OPEN PDU received from the remote BIS. The FSM shall then enter the ESTABLISHED state.
- g) If the BIS receives an IDRP ERROR PDU, either with or without error, it shall send a CEASE PDU, and the FSM shall change its state to CLOSED.
- h) If the BIS receives a RIB REFRESH PDU or UPDATE PDU, either with or without errors, it shall issue an IDRP ERROR PDU, indicating "FSM Error". The FSM shall then enter the CLOSE-WAIT state.
- i) If the BIS receives a KEEPALIVE PDU, it shall issue an IDRP ERROR PDU, indicating "FSM Error". The FSM shall then enter the CLOSE-WAIT state.
- j) If the BIS receives a CEASE PDU, it shall issue a CEASE PDU in return, and then the FSM shall enter the CLOSED state.
- k) If the BIS does not receive an OPEN PDU within a period t_R after sending an OPEN PDU, the BIS shall resend the OPEN PDU. If the OPEN PDU is retransmitted n times, the local BIS shall issue a **deactivate** to close the BIS-BIS connection.

NOTE 10: The value t_R should be chosen to be large enough so that attempting to establish a connection to an unresponsive peer BIS does not consume significant network resources. The values of t_R and n must be chosen so that $< t_R > * n$ is greater than the architectural constant **CloseWaitDelay**.

Table 2 (Page 1 of 2) – BIS Finite State Machine

STATE → INPUT ↓	CLOSED	OPEN-RCVD	OPEN-SENT	CLOSE-WAIT	ESTABLISHED
activate	S=OPEN-SENT A=send OPEN PDU	S=OPEN-RCVD A=none	S=OPEN-SENT A=none	S=CLOSE-WAIT A=none	S=ESTABLISHED A=none
deactivate	S=CLOSED A=none	S=CLOSE-WAIT A=send CEASE PDU	S=CLOSE-WAIT A=send CEASE PDU	S=CLOSE-WAIT A=none	S=CLOSE-WAIT A=send CEASE PDU
Expiration of CloseWaitDelay Timer	S=CLOSED A=none	S=OPEN-RCVD A=none	S=OPEN-SENT A=none	S=CLOSED A=7.6.2	S=ESTABLISHED A=none
Expiration of Hold Timer	S=CLOSED A=none	S=OPEN-RCVD A=none	S=OPEN-SENT A=none	S=CLOSE-WAIT A=none	S=CLOSE-WAIT A=Send IDRP ERROR PDU to report error
Receive BISPDU with Header Error	S=CLOSED A=none	S=OPEN-RCVD A=log error event	S=OPEN-SENT A=log error event	S=CLOSE-WAIT A=none	S=ESTABLISHED A=log error event
Receive KEEPALIVE PDU with no errors	S=CLOSED A=none	If ACK is correct, S=ESTABLISHED A=Restart Hold Timer If ACK is incorrect, S=CLOSE-WAIT A=Send IDRP ERROR PDU to peer BIS to report FSM Error	S=CLOSE-WAIT A=Send IDRP ERROR PDU to report FSM Error	S=CLOSE-WAIT A=send CEASE, restart CloseWaitDelay timer	S=ESTABLISHED A=Restart Hold Timer
Receive CEASE PDU with no errors	S=CLOSED A=none	S=CLOSED A=send CEASE, 7.6.2	S=CLOSED A=send CEASE, 7.6.2	S=CLOSED A=7.6.2	S=CLOSED A=send CEASE, 7.6.2
Receive OPEN PDU with errors	S=CLOSE A=none	S=CLOSE-WAIT A=Send IDRP ERROR PDU to peer BIS to report OPEN PDU error	S=CLOSE-WAIT A=Send IDRP ERROR PDU to peer BIS to report OPEN PDU error	S=CLOSE-WAIT A=none	S=CLOSE-WAIT A=Send IDRP ERROR PDU to peer BIS to report OPEN PDU error
Receive OPEN PDU with no errors	If ListenForOPEN is TRUE, S=OPEN-RCVD A=send OPEN PDU If ListenForOPEN is FALSE, S=CLOSED A=none	If ACK is correct, S=ESTABLISHED A=send KEEPALIVE, UPDATE, or RIB REFRESH PDU If ACK is incorrect, S=OPEN-RCVD, A=send OPEN PDU	If ACK is correct, S=ESTABLISHED A=send KEEPALIVE, UPDATE, or RIB REFRESH PDU If ACK is incorrect, S=OPEN-RCVD A=send OPEN PDU	S=CLOSE-WAIT A=send CEASE, restart CloseWaitDelay timer	S=CLOSE-WAIT A=Send IDRP ERROR PDU to peer BIS to report FSM error
Receive UPDATE PDU with errors	S=CLOSED A=none	S=CLOSE-WAIT A=Send IDRP ERROR PDU to peer BIS to report UPDATE PDU error	S=CLOSE-WAIT A=Send IDRP ERROR PDU to peer BIS to report FSM error	S=CLOSE-WAIT A=none	S=CLOSE-WAIT A=Send IDRP ERROR PDU to peer BIS to report UPDATE PDU error
Receive UPDATE PDU with no errors	S=CLOSED A=none	If ACK is correct, S=ESTABLISHED A=7.14, restart Hold Timer If ACK is incorrect, S=CLOSE-WAIT A=Send IDRP ERROR PDU to peer BIS to report FSM Error	S=CLOSE-WAIT A=Send IDRP ERROR PDU to peer BIS to report FSM error	S=CLOSE-WAIT A=send CEASE, restart CloseWaitDelay timer	S=ESTABLISHED A=7.14, restart Hold Timer

Table 2 (Page 2 of 2) – BIS Finite State Machine

STATE → INPUT ↓	CLOSED	OPEN-RCVD	OPEN-SENT	CLOSE-WAIT	ESTABLISHED
Receive IDRP ERROR PDU with errors	S=CLOSED A=none	S=CLOSED A=Send CEASE PDU, 7.6.2	S=CLOSED A=Send CEASE PDU, 7.6.2	S=CLOSED A=Send CEASE PDU, 7.6.2	S=CLOSED A=Send CEASE PDU, 7.6.2
Receive IDRP ERROR PDU with no errors	S=CLOSED A=none	S=CLOSED A=Send CEASE PDU, 7.6.2	S=CLOSED A=Send CEASE PDU, 7.6.2	S=CLOSED A=Send CEASE PDU, 7.6.2	S=CLOSED A=Send CEASE PDU, 7.6.2
Receive RIB REFRESH PDU with errors	S=CLOSED A=none	S=CLOSE-WAIT A=Send IDRP ERROR PDU to peer BIS to report RIB REFRESH PDU error	S=CLOSE-WAIT A=Send IDRP ERROR PDU to peer BIS to report FSM error	S=CLOSE-WAIT A=none	S=CLOSE-WAIT A=Send IDRP ERROR PDU to peer BIS to report RIB REFRESH PDU error
Receive RIB REFRESH PDU with no errors	S=CLOSED A=none	If ACK is correct, S=ESTABLISHED A=7.10.3, restart Hold Timer If ACK is incorrect, S=CLOSE-WAIT A=Send IDRP ERROR PDU to peer BIS to report FSM Error	S=CLOSE-WAIT A=Send IDRP ERROR PDU to report FSM Error	S=CLOSE-WAIT A=send CEASE, restart CloseWaitDelay timer	S=ESTABLISHED A=7.10.3, restart Hold Timer

NOTES:

- "S" indicates the state into which the FSM will make a transition after performing the indicated action.
- "A" indicates the action to be taken.
- "X.Y.Z" is shorthand notation for "do as specified in clause X.Y.Z".
- The phrase "no errors" for a given BISPDU type means that no condition described in the appropriate subclause of 7.20 has been detected.
- The phrase "with errors" for a given BISPDU type means that a condition described in the appropriate subclause of 7.20 has been detected.
- Since the KEEPALIVE PDU and the CEASE PDU consist of only a fixed BISPDU header, errors in these BISPDUs are handled as Header Errors. Hence, there are no explicit entries in the table for "KEEPALIVE with errors" or "CEASE with errors".

7.6.1.3 OPEN-RCVD State

While in the OPEN-RCVD state, the BIS shall take the following actions:

- If the BIS receives an **activate**, it shall ignore it and the FSM shall remain in the OPEN-RCVD state.
- If the BIS receives a **deactivate**, it shall send a CEASE PDU to the remote BIS, and the FSM shall enter the CLOSE-WAIT state.
- If the BIS receives a BISPDU with a header error, it shall log the error event, and the FSM shall remain in the OPEN-RCVD state.
- If the BIS receives a KEEPALIVE PDU that acknowledges its previously sent OPEN PDU, then the FSM shall enter the ESTABLISHED state.
- If the BIS receives a KEEPALIVE PDU that does not acknowledge its previously sent OPEN PDU, the BIS shall issue an IDRP ERROR PDU indi-

cating "FSM Error", and the FSM shall change its state to CLOSE-WAIT.

- If the BIS receives a CEASE PDU, it shall issue a CEASE PDU in return, and then the FSM shall enter the CLOSED state.
- If the BIS receives an OPEN PDU with no errors from the remote BIS that acknowledges the local BIS's previously sent OPEN PDU, the BIS shall send a KEEPALIVE, RIB REFRESH, or UPDATE PDU that acknowledges the OPEN PDU received from the remote BIS. The FSM shall then enter the ESTABLISHED state.
- If the BIS receives an OPEN PDU with no errors that does not acknowledge the local BIS's previously sent OPEN PDU, then the local BIS shall resend its own OPEN PDU with the same sequence number, and shall also include an acknowledgement of the remote BIS's OPEN PDU. The FSM shall remain in the OPEN-RCVD state.

- i) If the BIS receives an UPDATE PDU or RIB REFRESH PDU with errors, the BIS shall send an IDR P ERROR PDU to the remote BIS, and the FSM shall enter the CLOSE-WAIT state.
- j) If the BIS receives an UPDATE PDU or RIB REFRESH PDU with no errors that acknowledges the OPEN PDU previously sent by the local BIS, the FSM shall enter the ESTABLISHED state.
- k) If the BIS receives an UPDATE PDU or RIB REFRESH PDU with no errors that does not acknowledge the OPEN PDU previously sent by the local BIS, the BIS shall issue an IDR P ERROR PDU indicating "FSM Error", and the FSM shall change its state to CLOSE-WAIT.
- l) If the BIS receives an IDR P ERROR PDU, either with or without errors, it shall send a CEASE PDU to the remote BIS, and the FSM shall enter the CLOSED state.
- m) If the BIS does not exit the OPEN-RCVD state within a period t_R after sending an OPEN PDU, the BIS shall resend the OPEN PDU. If the OPEN PDU is transmitted n times, the local BIS shall issue a **deactivate**.

NOTE 11: The value t_R should be chosen to be large enough so that attempting to establish a connection to an unresponsive peer BIS does not consume significant network resources. The values of t_R and n must be chosen so that $<t_R> * n$ is greater than the architectural constant **CloseWaitDelay**.

7.6.1.4 ESTABLISHED State

The ESTABLISHED state is entered from either the OPEN-SENT or the OPEN-RCVD states. It is entered when a connection has been established by the successful exchange of state information between two sides of the connection. Each side has exchanged and received such data as initial sequence number, maximum PDU size, credit offered, protocol version number, hold time, and RDI of the other side. In addition, the remote BIS may also have been authenticated.

In ESTABLISHED state, both BISs that are involved in the connection may exchange UPDATE PDUs, KEEPALIVE PDUs, IDR P ERROR PDUs, RIB REFRESH PDUs, and CEASE PDUs.

While in the ESTABLISHED state, the local BIS shall take the following actions:

- a) If the FSM receives an **activate**, the FSM shall ignore it, and the FSM shall remain in the ESTABLISHED state.
- b) If the FSM receives a **deactivate**, the BIS shall send a CEASE PDU to the peer BIS. The FSM shall enter the CLOSE-WAIT state.

- c) If the Hold Timer expires, the BIS shall issue an IDR P ERROR PDU to the remote BIS, reporting a Hold_Timer error. The FSM shall enter the CLOSE-WAIT state.
- d) If the BIS receives a BISPDU with a header error, it shall log the error event, and the FSM shall remain in the ESTABLISHED state.
- e) If the BIS receives a KEEPALIVE PDU, it shall restart its Hold Timer, and the FSM shall remain in the ESTABLISHED state.
- f) If the BIS receives a CEASE PDU, it shall issue a CEASE PDU in return, and then the FSM shall enter the CLOSED state.
- g) If an OPEN PDU with no errors is received from the peer BIS, it shall issue an IDR P ERROR PDU, indicating FSM error. The FSM shall enter the CLOSE-WAIT state.
- h) If the BIS receives an UPDATE PDU with no errors, the BIS shall perform the actions provided in 7.14, and shall restart its Hold Timer. The FSM shall remain in the ESTABLISHED state.
- i) If the BIS receives a RIB REFRESH PDU with no errors, the BIS shall perform the actions provided in 7.10.3, and shall restart its Hold Timer. The FSM shall remain in the ESTABLISHED state.
- j) If the BIS receives an UPDATE PDU with errors, an OPEN PDU with errors, or a RIB REFRESH PDU with errors, it shall send an IDR P ERROR PDU to the remote BIS to report the error, and the FSM shall enter the CLOSE-WAIT state.
- k) If the BIS receives an IDR P ERROR PDU, either with or without errors, it shall send a CEASE PDU to the remote BIS. The FSM shall enter the CLOSED state.

7.6.1.5 CLOSE-WAIT State

When an FSM enters the CLOSE-WAIT state, the local BIS is preparing to close the connection with the remote BIS. Upon entering this state, the local BIS shall mark all entries in the Adj-RIB-In associated with the adjacent BIS as unreachable, and shall then re-run its Decision Process. The CloseWaitDelay timer shall be started.

While in the CLOSE-WAIT state, the BIS shall take the following actions:

- a) If the CloseWaitDelay timer expires, the connection ceases to exist. The FSM shall enter the CLOSED state.
- b) If the BIS receives a CEASE PDU, the FSM shall enter the CLOSED state.
- c) If the BIS receives an IDR P ERROR PDU, it shall send a CEASE PDU to the peer BIS. The FSM shall then enter the CLOSED state.

- d) If the BIS receives any other type of BISPDU, with or without errors, it shall issue a CEASE PDU. The FSM shall remain in the CLOSE-WAIT state, and the CloseWaitDelay timer shall be restarted.
- e) The BIS shall take no action for any of the following inputs, and the FSM shall remain in the CLOSE-WAIT state:
 - activate
 - deactivate
 - Expiration of Hold Timer

7.6.2 Closing a connection

The closing of a connection can be initiated by a **deactivate** generated by the local system, by receipt of an incorrect PDU, by receipt of a IDRP ERROR PDU, by expiration of the Hold Timer, or by receipt of a CEASE PDU. The actions taken in response to each of these stimuli are shown in Table 2

When the connection enters the CLOSED state, the sequence number last used by the local BIS is recorded in managed object **lastPriorSeqNo**, and all routes that had been exchanged between the pair of BISs are implicitly withdrawn from service; hence, the local BIS should rerun its Decision Process.

7.7 Validation of BISPDU

The protocol described in this International Standard is a connection oriented protocol in which the underlying Network Layer service is used to establish full-duplex communication channels between pairs of BISs, as described in 7.6. This International Standard supports the use of any of the following three mechanisms for validating BISPDU. Types 1, 2, and 3 provide data integrity for the contents of BISPDU; in addition, types 2 and 3 provide peer BIS authentication. Each mechanism is described below. Figure 6 illustrates the data integrity mechanisms used for authentication types 1 and 3; informative Annex D describes an example approach that might be used to provide both data integrity and peer BIS authentication for authentication type 2.

7.7.1 Authentication type 1

For all BISPDU that flow on a connection that was established in response to an OPEN PDU whose authentication code field was equal to 1, the validation field shall contain a 16-octet unencrypted checksum:

- a) Generating a Validation Pattern:

The contents of the Validation Pattern field that is included in an outbound BISPDU shall be generated by applying the algorithm of Annex B to the

input data stream that consists of the contents of the entire BISPDU with all bits of the Validation Pattern field initially set to 0. The output of this step is an unencrypted 16-octet long checksum, which shall be placed in the Validation Pattern field of the BISPDU.

- b) Checking the Validation Pattern of an Inbound BISPDU:

The contents of the Validation Pattern field of an inbound BISPDU shall be checked by applying the algorithm of Annex B to the contents of the inbound BISPDU with its Validation Pattern set to all zeros. Call this quantity the "reference pattern".

If the "reference pattern" matches the contents of the Validation Pattern field of the inbound BISPDU, then the BISPDU's checksum is correct; otherwise, it is incorrect.

7.7.2 Authentication type 2

When the authentication type code of the OPEN PDU is 2, the pattern carried in the 16-octet Validation Pattern field of the fixed header shall provide both peer-BIS authentication and data integrity for the contents of the BISPDU. The specific mechanisms used to provide these functions are not specified by this International Standard. However, they must be agreed to by the pair of communicating BISs as part of their security association.

An example of type 2 authentication that uses an encrypted version of MD4 checksum is described in Annex D.

NOTE 12: This International Standard includes as an optional function a mechanism that can be used for authentication of the source of a BISPDU. Other security-related facilities (for example, protection against replay of BISPDU or the ability to re-key during a BIS_BIS connection) are not intended to be provided by this protocol, and therefore are not specified in this International Standard.

All of the relevant security services identified in ISO 7498-2, including authentication, could be achieved by the use of ISO/IEC 11577, a security protocol specified for the provision of secure ISO 8473 communications (for example, see 9.1).

7.7.3 Authentication type 3

When the authentication type code of the OPEN PDU is 3, the Validation Pattern field shall contain a 16-octet checksum covering both the contents of the BISPDU and some additional Password Text, which is not transmitted to the peer BIS. The checksum provides data integrity and the untransmitted Password Text provides peer BIS authentication.