

International Telecommunication Union

# ITU-T G.7718.1/Y.1709.1

TELECOMMUNICATION  
STANDARDIZATION SECTOR  
OF ITU

(12/2006)

SERIES G: TRANSMISSION SYSTEMS AND MEDIA,  
DIGITAL SYSTEMS AND NETWORKS

Data over Transport – Generic aspects – Transport  
network control aspects

SERIES Y: GLOBAL INFORMATION  
INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS  
AND NEXT-GENERATION NETWORKS

Internet protocol aspects – Operation, administration and  
maintenance

---

## Protocol-neutral management information model for the control plane view

ITU-T Recommendation G.7718.1/Y.1709.1



ITU-T G-SERIES RECOMMENDATIONS  
TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TRANSMISSION MEDIA CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
QUALITY OF SERVICE AND PERFORMANCE – GENERIC AND USER-RELATED ASPECTS	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DATA OVER TRANSPORT – GENERIC ASPECTS	G.7000–G.7999
General	G.7000–G.7099
<b>Transport network control aspects</b>	<b>G.7700–G.7799</b>
PACKET OVER TRANSPORT ASPECTS	G.8000–G.8999
ACCESS NETWORKS	G.9000–G.9999

*For further details, please refer to the list of ITU-T Recommendations.*

# **ITU-T Recommendation G.7718.1/Y.1709.1**

## **Protocol-neutral management information model for the control plane view**

### **Summary**

ITU-T Recommendation G.7718.1/Y.1709.1 provides a protocol-neutral management information model for managing the ASON control plane. It includes an electronic attachment containing the UML description in navigable html format and in the original Rational Rose format.

### **Source**

ITU-T Recommendation G.7718.1/Y.1709.1 was approved on 14 December 2006 by ITU-T Study Group 15 (2005-2008) under the ITU-T Recommendation A.8 procedure.

### **Keywords**

ASON, control plane, protocol-neutral information model, UML.

## FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

## NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

## INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2007

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

## CONTENTS

	<b>Page</b>
1 Scope .....	1
2 References.....	1
3 Definitions .....	2
3.1 Terms defined elsewhere .....	2
3.2 Terms defined in this Recommendation.....	2
4 Abbreviations.....	2
5 Conventions .....	4
5.1 Object creation and deletion .....	4
5.2 Textual conventions.....	4
6 Control plane management requirements .....	8
7 Analysis .....	8
8 Design of the protocol-neutral information model .....	9
8.1 Class diagram – Call/connection class diagram .....	9
8.2 Class diagram – Common attributes diagram (InheritanceTree) .....	10
8.3 Class diagram – Discovery service class diagram.....	11
8.4 Class diagram – Notification class diagram .....	11
8.5 Class diagram – Routing service class diagram .....	12
8.6 Class diagram – Signalling service class diagram.....	13
9 Class definitions.....	13
9.1 AlarmSeverityAssignmentProfile.....	13
9.2 CR-LDPProtocol .....	13
9.3 Call .....	14
9.4 CallCreateData .....	19
9.5 CallDetailRecord .....	21
9.6 CallService .....	22
9.7 CommonResourceInformation .....	26
9.8 Connection.....	27
9.9 ConnectionCreateData.....	29
9.10 ConnectionService.....	31
9.11 ConnectionTerminationPoint .....	32
9.12 ControlPlane .....	32
9.13 ControlPlaneService .....	34
9.14 DiscoveryService.....	35
9.15 ISISArea .....	36
9.16 ISISNeighbour .....	38
9.17 ISISProtocol .....	39
9.18 LayerAdjacencyDiscovery .....	40
9.19 ManagedElement.....	42

	<b>Page</b>	
9.20	OSPFArea.....	42
9.21	OSPFNeighbour .....	44
9.22	OSPFProtocol.....	45
9.23	PNNIArea.....	46
9.24	PNNINeighbour.....	48
9.25	PNNIRoutingProtocol .....	49
9.26	PNNISignallingProtocol.....	50
9.27	RSVPProtocol.....	51
9.28	RoutingArea .....	51
9.29	RoutingTable .....	55
9.30	SNP.....	55
9.31	SNPLinkConnection.....	56
9.32	SNPP.....	57
9.33	SNPPLink.....	58
9.34	SignallingProtocol .....	62
9.35	TerminationPoint.....	63
9.36	TrailTerminationPoint .....	63
9.37	TransportEntityCapabilityExchange .....	63
10	Notification definitions.....	65
10.1	Alarm.....	65
10.2	AttributeValueChangeNotification.....	66
10.3	Notification.....	66
10.4	ObjectCreationNotification .....	68
10.5	ObjectDeletionNotification .....	68
10.6	StateChangeNotification.....	68
11	Data Types.....	68
11.1	Enumerations.....	68
11.2	Structures.....	75
11.3	TypeDefinitions.....	76
	Appendix I – G.7718.1 and TMF 608v3.0.....	79
	Appendix II – Alternate representations .....	80

Electronic attachment:

- Layer rate supporting document.
- UML description in navigable html format.
- UML description in original Rational Rose format.

# ITU-T Recommendation G.7718.1/Y.1709.1

## Protocol-neutral management information model for the control plane view<sup>1</sup>

### 1 Scope

This Recommendation provides a protocol-neutral management information model for managing the ASON control plane. It identifies the telecommunication management network (TMN) managed entities required. These entities are relevant to information exchanged across standardized interfaces defined in [ITU-T M.3010] TMN architecture. The protocol-neutral management information model should be used as the base for defining protocol-specific management information models, for example, CMISE information model and CORBA IDL interfaces.

The specific mapping of the protocol-neutral entities into protocol-specific managed object classes is the decision of the protocol-specific modelling design. Protocol-specific information models and their mapping from the protocol-neutral model will be described in other documents.

This Recommendation applies to network elements performing ASON control plane functions, and those systems in the TMN that manage them. The architecture or ASON control plane and its components are given in [ITU-T G.8080] and the management framework and requirements are given in [ITU-T G.7718].

The object entities defined in this model apply to configuration and fault management.

There are several different perspectives from which management information may be defined for management purposes. The control plane fragment contains those object entities and information having to do with managing the control plane functions, irrespective of the platform performing those functions. That platform may be a traditional network element or a stand-alone element with no transport functionality.

### 2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- [ITU-T G.7710] ITU-T Recommendation G.7710/Y.1701 (2001), *Common equipment management function requirements*.
- [ITU-T G.7713] ITU-T Recommendation G.7713/Y.1704 (2006), *Distributed call and connection management (DCM)*.
- [ITU-T G.7714] ITU-T Recommendation G.7714/Y.1705 (2005), *Generalized automatic discovery for transport entities*.
- [ITU-T G.7718] ITU-T Recommendation G.7718/Y.1709 (2005), *Framework for ASON management*.
- [ITU-T G.8080] ITU-T Recommendation G.8080/Y.1304 (2006), *Architecture for the automatically switched optical network (ASON)*.

---

<sup>1</sup> This Recommendation includes an electronic attachment containing the layer rate supporting document, the UML description in navigable html format and in the original Rational Rose format.

- [ITU-T M.3010] ITU-T Recommendation M.3010 (2005), *Principles for a telecommunications management network*.
- [ITU-T M.3100] ITU-T Recommendation M.3100 (2005), *Generic network information model*.
- [ITU-T X.733] ITU-T Recommendation X.733 (1992), *Information technology – Open Systems Interconnection – Systems management: Alarm reporting function*.
- [AF-PNNI-0055] MFA Forum AF-PNNI-0055.002, *Private Network-Network Interface Specification Version 1.1 (PNNI 1.1)*, April 2002.
- [IETF RFC 2328] IETF RFC 2328 (1998), *OSPF Version 2*.
- [IETF RFC 3473] IETF RFC 3473 (2003), *Generalized Multi-Protocol Label Switching (GMPLS) Signalling Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions*.
- [ISO/IEC 10589] ISO/IEC 10589:2002, *Information technology – Telecommunications and information exchange between systems – Intermediate system to intermediate system intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)*.
- [TMF MTNM] Tele Management Forum MTNM SD1-17\_LayerRates.

### 3 Definitions

#### 3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

- 3.1.1 alarm severity assignment profile (ASAP):** [ITU-T M.3100].
- 3.1.2 connection termination point (CTP):** [ITU-T M.3100].
- 3.1.3 termination point (TP):** [ITU-T M.3100].
- 3.1.4 trail termination point (TTP):** [ITU-T M.3100].
- 3.1.5 alarm reporting control (ARC):** [ITU-T G.7710].
- 3.1.6 subnetwork point (SNP):** [ITU-T G.8080].
- 3.1.7 subnetwork point pool (SNPP):** [ITU-T G.8080].
- 3.1.8 subnetwork point pool link (SNPP link):** [ITU-T G.8080].

#### 3.2 Terms defined in this Recommendation

*None.*

### 4 Abbreviations

This Recommendation uses the following abbreviations:

ASON	Automatically Switched Optical Network
AVC	Attribute Value Change Notification
AVP	Attribute name Value Pair
CMISE	Common Management Information Service Element
CORBA	Common Object Request Broker Architecture



CP	Control Plane
CR-LDP	Constraint-based Label Distribution Protocol
DCM	Distributed Call and Connection Management
EMS	Element Management System
ENNI	External Network-Network Interface
GDMO	Guidelines for Definition of Managed Objects
HTML	HyperText Markup Language
Id, ID	Identifier
IDL	Interface Definition Language
I-NNI	Internal Network-Network Interface
IP	Internet Protocol
ISIS	Intermediate System – Intermediate System routing protocol
LCAS	Link Capacity Adjustment Scheme
LRM	Link Resource Manager
LSA	Link State Advertisement
MIB	Management Information Base
MTNM	Multi-Technology Network Management
MTU	Maximum Transmission Unit
NA	Not Applicable
NE	Network Element
NMS	Network Management System
OSPF	Open Shortest Path First
PNNI	Private Network-to-Network Interface
RSVP	Resource Reservation Protocol
RSVP-TE	Resource Reservation Protocol – Traffic Engineering
SCN	Signalling Communication Network
SRG	Shared Risk Group
TCE	Transport Capability Exchange
TMF	TeleManagement Forum
TMN	Telecommunication Management Network
TRI	Transport Resource Identifier
TTP	Trail Termination Point
UML	Unified Modelling Language
UNI	User-to-Network Interface
UNI-C	User-to-Network Interface – Customer side
UNI-N	User-to-Network Interface – Network side

## 5 Conventions

This Recommendation uses the UML notation to specify a protocol-neutral information model. UML itself does not specify an execution environment, so there are no pre-existing object methods provided. This is in contrast to the X.721 (GDMO/CMISE) notation which provides many methods in its assumed environment.

This Recommendation has therefore specified some general methods, and has specified them as operations against the ControlPlane object.

### 5.1 Object creation and deletion

An object can be Created or Deleted by using the relevant operation on the ControlPlane object.

However, some objects are created as the result of very significant activity in the control plane, and the generic create operation cannot have any knowledge of the very special behaviour of (for example) creating a call object.

Some objects are also expected to be automatically created when the platform supporting those objects is initialized. It is not expected to create such objects by independent management action.

This specification takes care of these significant differences by providing special operations to create these special objects, and it is a convention that these special operations are used in place of the generic ones.

### 5.2 Textual conventions

Several important properties of network management applications need to be specified, and UML does not provide any formal support. This Recommendation uses some English text conventions to provide for this specification.

In all cases, these specifications are inherited via the normal inheritance relationship, though the UML notation does not provide any formal support for this inheritance.

For the purpose of this Recommendation, these pieces of specification achieved by the English text are referred to as "properties".

#### 5.2.1 Attribute properties

In network management applications, controlling access to attributes is critical; however, UML does not provide any formal control, and so we resort to textual conventions in the object descriptions. The three Booleans Readable, Writable and Invariant appear in the object descriptions as below. The values are "Yes" or "No."

- Readable by EMS? Yes
- Writable by EMS? Yes
- Invariant? No

When Writable = yes and Invariant = yes together, this signifies an attribute that can be written once and that the EMS can provide a one time configuration value.

When Writable = No and Invariant = Yes, this signifies an attribute which receives a value at object creation time, or as a result of a side effect of creation, and that it is not expected to change over time.

When Writable = No and Invariant = No signifies an attribute whose value reflects a changing value in the managed object, and whose value cannot be affected by the EMS.

In network management applications, it is common that an object will autonomously report an attribute value change. Mentioning the attribute value change (AVC) notification in the Notification Type property description specifies this behaviour.

Attributes whose value should be set at object creation to reflect the current value in the system are shown as having a default value of "NA" (not applicable).

Note that the method of ensuring compliance with this access rights specification is left to protocol specific implementations.

### **5.2.2 Complex attribute types**

Complex attributes, which are used in more than one object, are listed in clause 11 as if they were objects. The fields of the complex type become the attributes of the type object.

### **5.2.3 Object properties**

Several properties of objects are interesting to network management applications, and these are added by textual conventions.

Object Lifecycle: indicates when the object is created, and by whom and when the object is deleted.

Notifications specify autonomous behaviour of the object, perhaps in response to a private internal method. In particular, Creation and Deletion are frequently notified to the management system. This behaviour is specified by the Booleans "Object Creation Notification Issued" and "Object Deletion Notification Issued".

Inheritance is indicated by the "Derived from" note.

### **5.2.4 Operation properties**

Operations have several aspects to their specification that are subject to conventions.

#### **Pre-condition**

The pre-condition in this Recommendation specifies the conditions that must be true before the operation is invoked. In case that no pre-condition is specified, "TRUE" is assumed. That is, the operation user need not take any particular steps to check any state before the operation is issued.

#### **Post-condition**

The post-condition is a condition or predicate that is always true after an operation has been performed successfully.

#### **Exceptions**

This describes various reasons that the operations may have failed, and conventionally includes a restatement of the pre-condition.

#### **Idempotence**

Operations are all expected to be idempotent, so setting a value will not result in an error if the operation does not actually cause a value change. In computing, idempotence is the quality of something that has the same effect if used multiple times as it does if used only once, similar to the idempotence notion in mathematics.

#### **Exceptions**

Operations support the following default exceptions.

notImplemented, invalidInput, and error. These exceptions have their usual meaning and the error exception is unspecified. Operations requiring a more detailed error exception specify this separately. Operations not requiring special exception behaviour have the operation exception description "Standard".

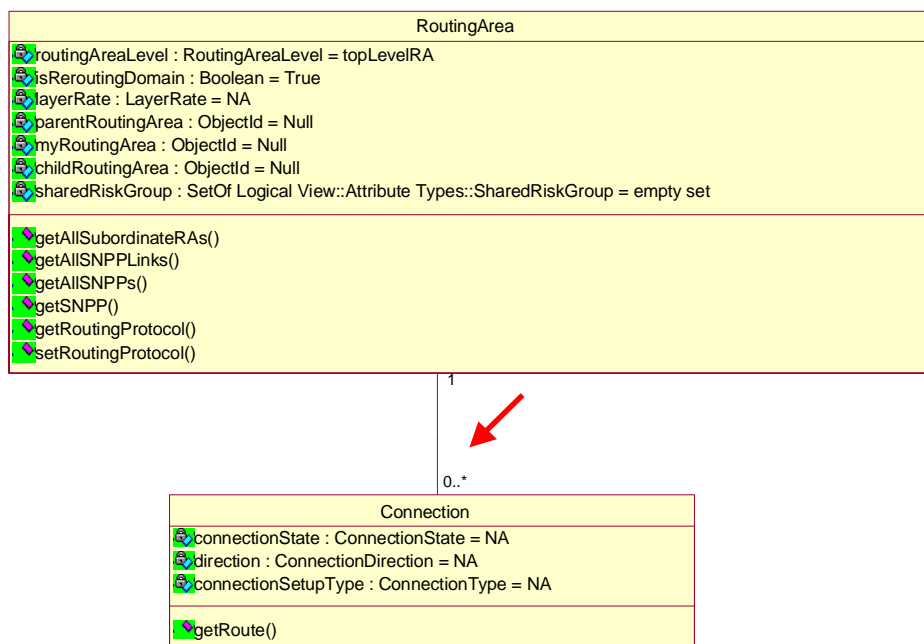
## 5.2.5 Relationships

Relationships among objects are indicated by UML associations; however, not every association is intended to be navigable in a protocol specific implementation. Navigability requires that the protocol specific design must include attributes (and possible operations) that model the relation. The means by which these specifications are implemented is left to the protocol specific design. Rather than confuse the UML information model with relation navigation details, which are not automatically checked for correctness, this Recommendation uses some conventions to indicate which relationships should be navigable, and which are explanatory.

The conventions are as follows:

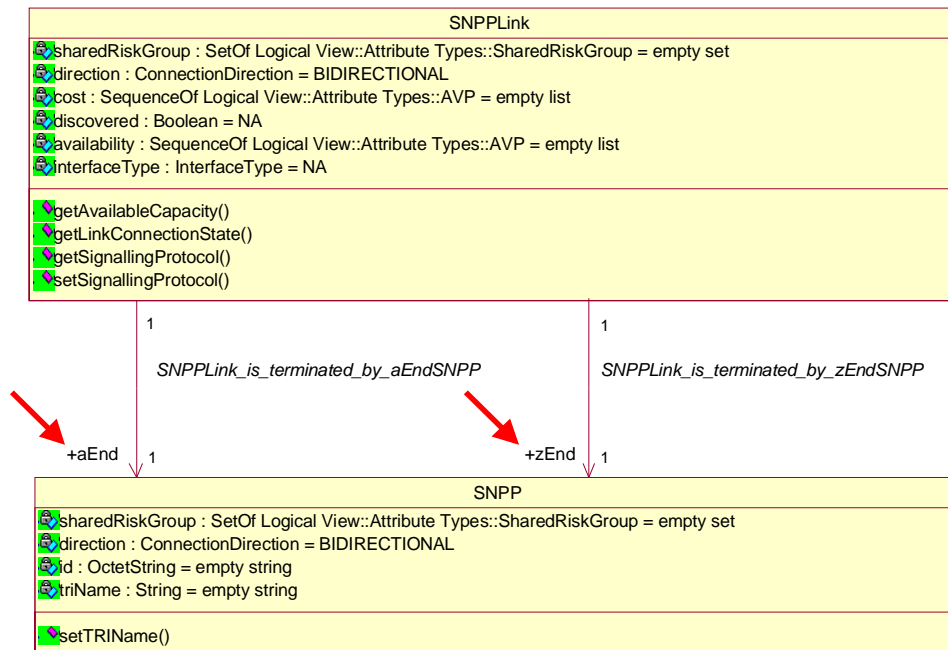
- 1) All Associations and Relations without any role names are just for explanation to the reader of the model and will not be implemented unless a specific operation or attribute requires it.

Example:



- 2) A public role name (+) indicates that the association will be navigable in the protocol specific design. The method of implementing the navigation (attribute or operation) is left to the specific design.

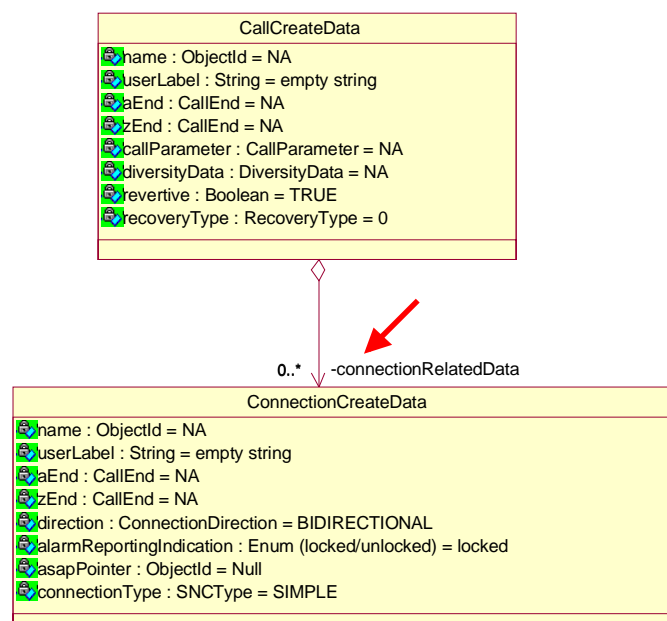
Example:



IDL: aEnd and zEnd will become attributes of the SNPPLink struct containing the name of the a-End and z-End SNPP.

- 3) If class A has a Containment or Composition relationship with class B and class B plays a private (-) role in this relationship, then the contained information of class B shall be part of the information when class A is retrieved via the interface (i.e., attributes of class B shall be retrieved together with attributes of class A).

Example:



## **6 Control plane management requirements**

This Recommendation models the control functions that are relevant. These functions are defined in [ITU-T G.7718]. General details of the management functions that need to be modelled are provided in [ITU-T G.7710].

## **7 Analysis**

In this Recommendation, managed resources and management support resources are modelled as objects in the information model. The management view of a resource is a managed object. This Recommendation specifies the properties of the resources visible for management. Objects with similar properties are grouped into object classes. An object instance is an instantiation of an object class. The properties of an object include the behaviour, attributes, and operations that can be applied on the object. An object instance is characterized by its object class and may possess multiple attribute types and associated values. In the protocol-neutral model, object classes are represented as unified modelling language (UML) classes.

Object classes, attribute types, and operations are defined for the purpose of communicating network management messages between systems, and need not be related to the structure of data within those systems.

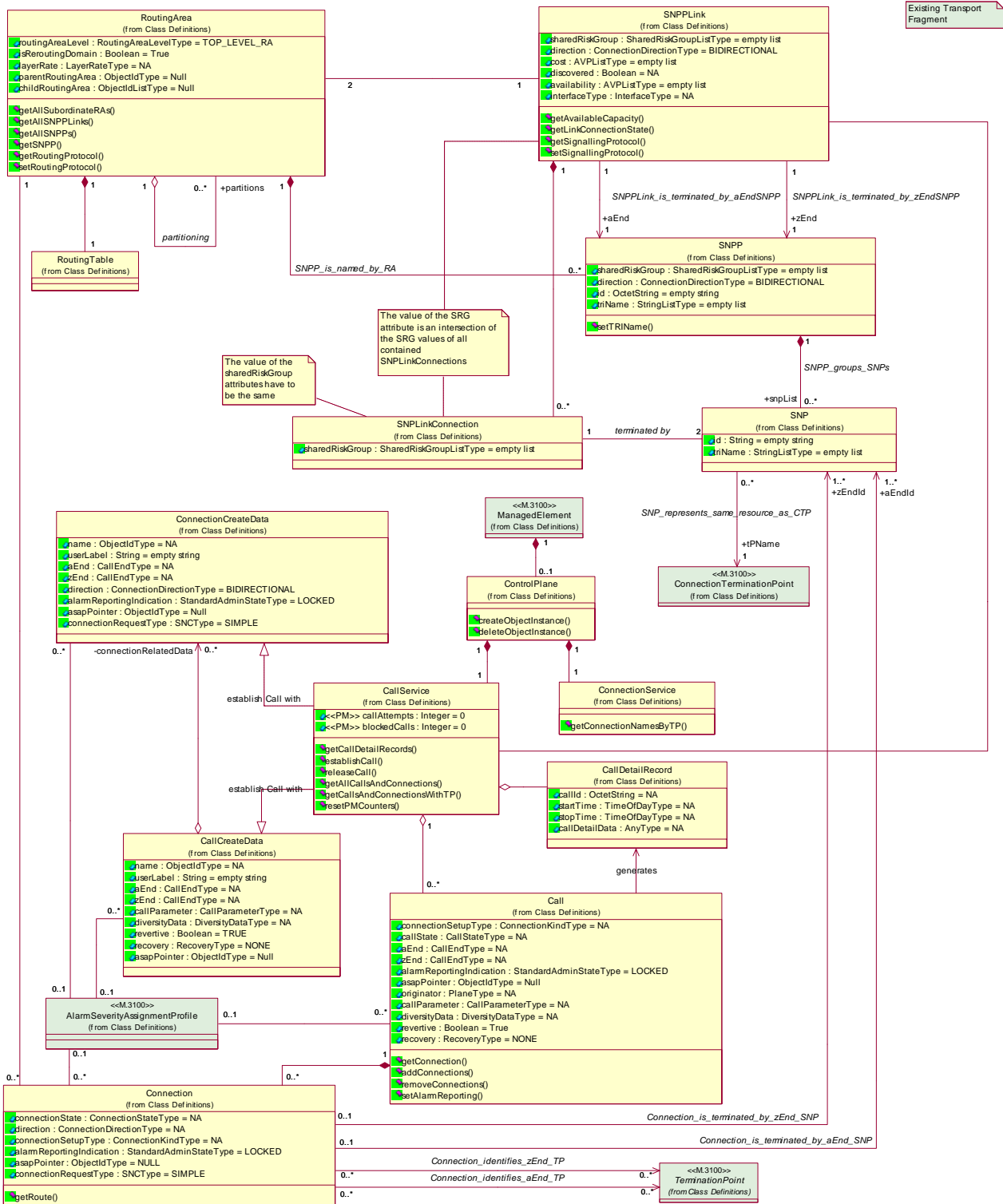
An object class may be a subclass of another class. A subclass inherits properties of its superclass, in addition to possessing its own specific attributes and properties. In this Recommendation, the control plane specific object classes are defined. In the future, when defining protocol-specific object classes, the protocol-specific object classes could be mapped from the protocol-neutral object classes and also inherited from the protocol-specific generic transport object classes for additional properties.

At the protocol specific design stage, designers have the freedom to change the structure of information specified in this Recommendation as long as the specified semantics is maintained. This allows for protocol specific design optimizations to be performed. For example, this specification returns object identifiers, while a possible design optimization could be to return the whole object and so avoid a second transaction.

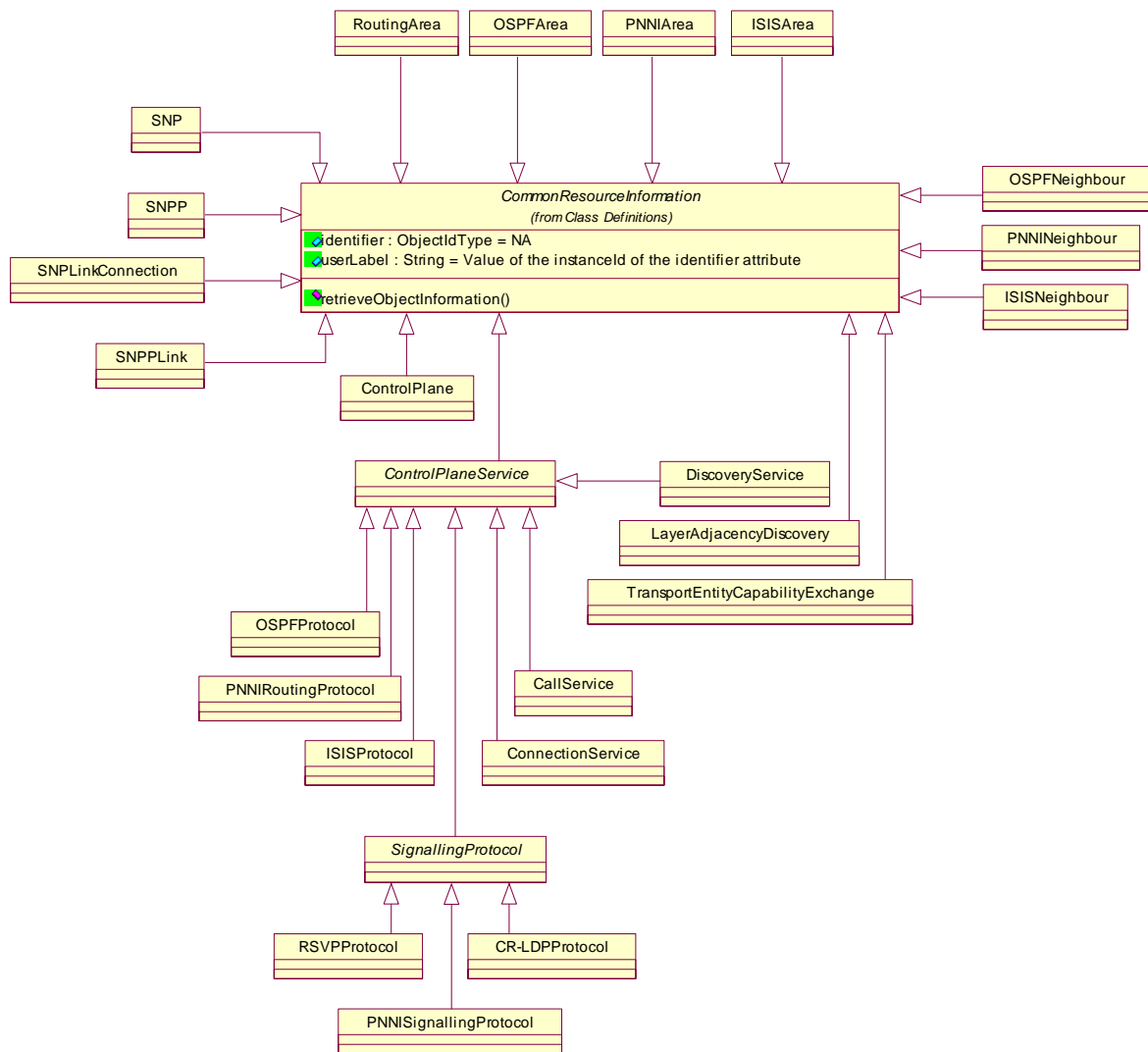
In addition, to the control plane resources, the model also refers to object classes for management support functions such as alarm reporting control and alarm severity assignment.

# 8 Design of the protocol-neutral information model

## 8.1 Class diagram – Call/connection class diagram

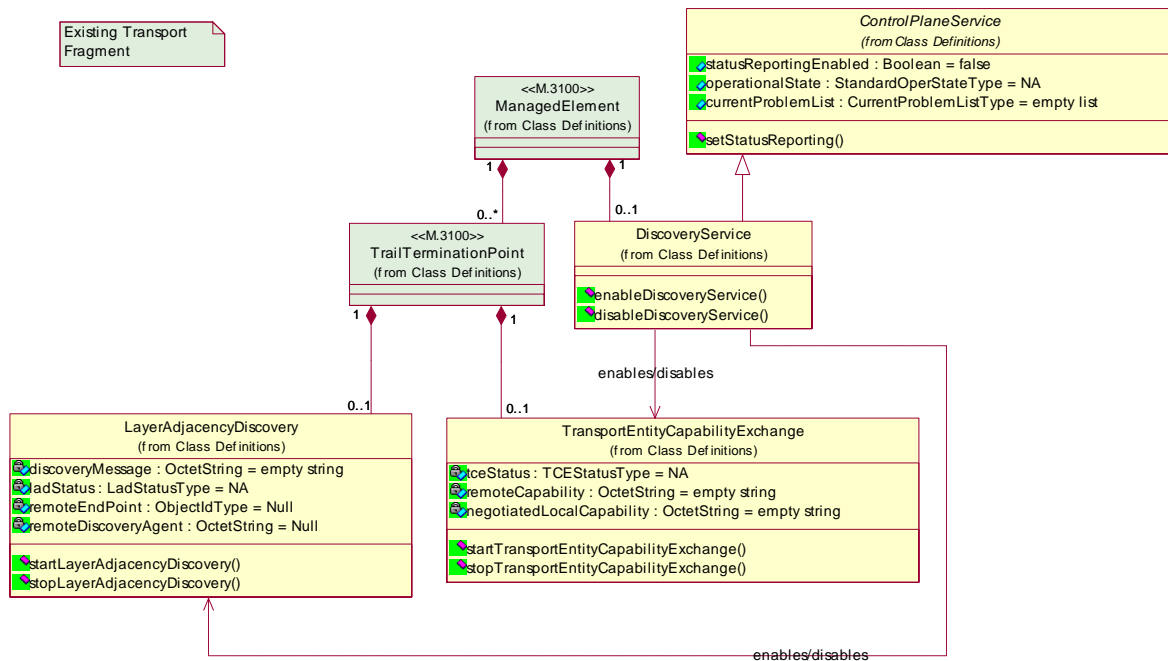


## 8.2 Class diagram – Common attributes diagram (InheritanceTree)

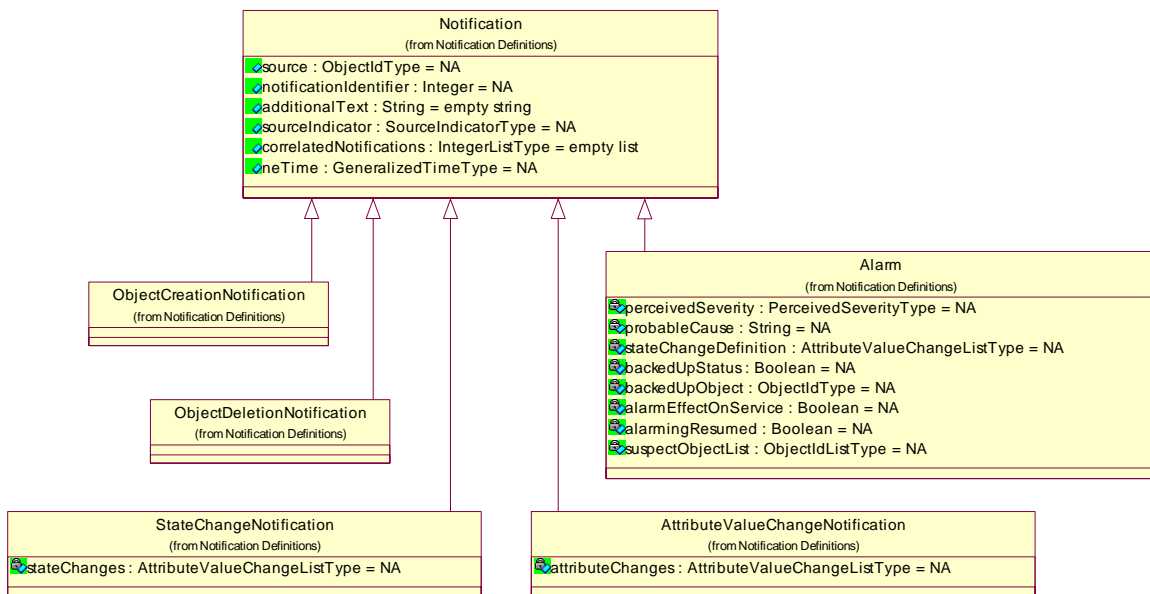




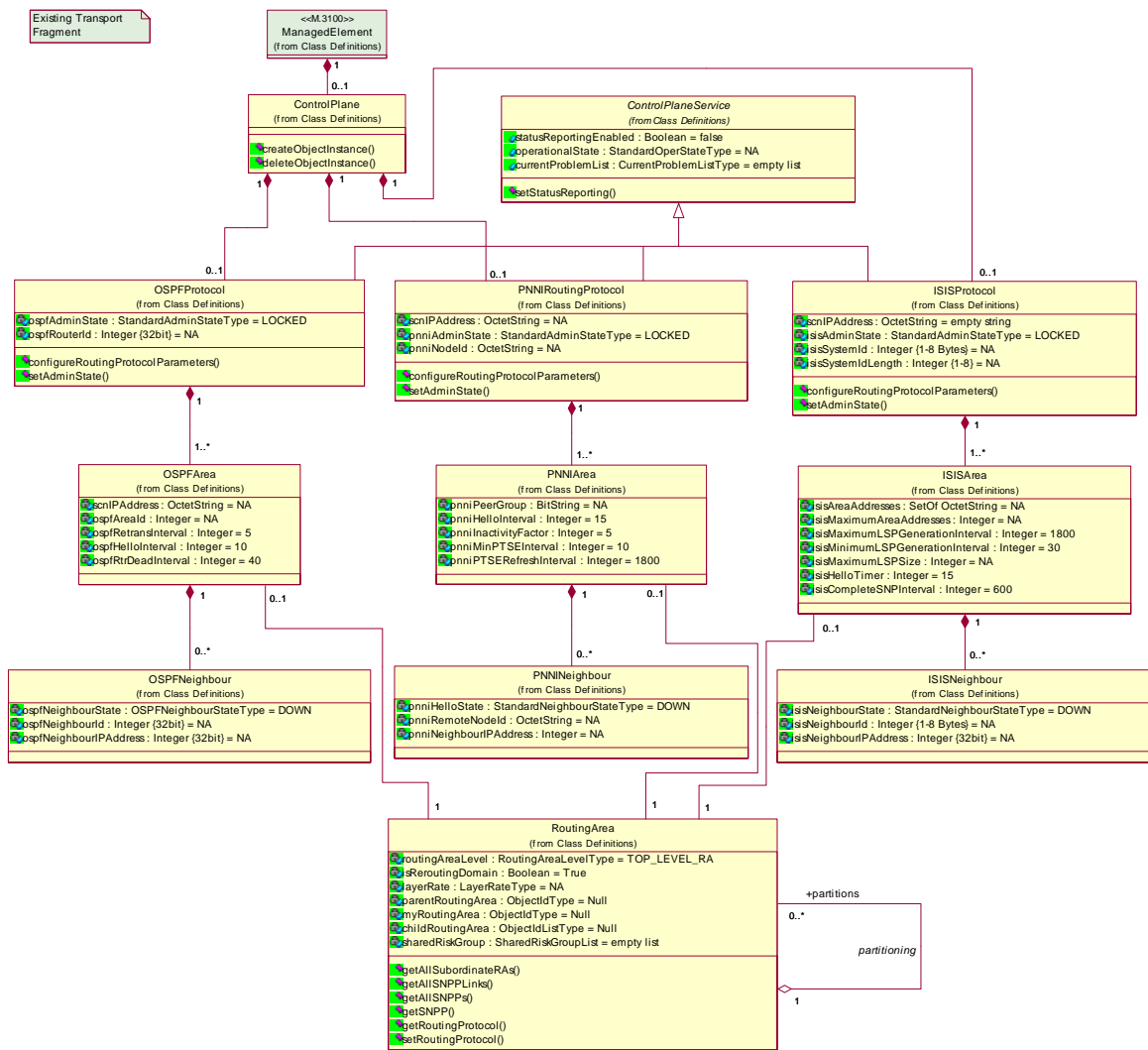
### 8.3 Class diagram – Discovery service class diagram



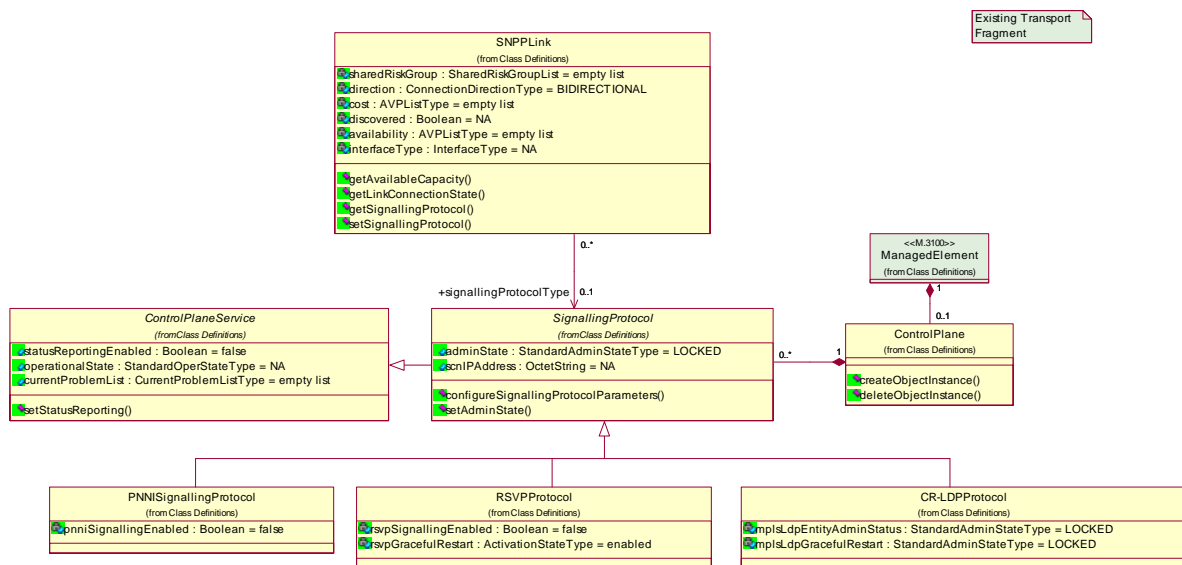
### 8.4 Class diagram – Notification class diagram



## 8.5 Class diagram – Routing service class diagram



## 8.6 Class diagram – Signalling service class diagram



## 9 Class definitions

### 9.1 AlarmSeverityAssignmentProfile

The AlarmSeverityAssignmentProfile object specifies the alarm severity assignment for managed objects. Instances of this object are referenced by the asapPointer attribute in the managed objects.

#### 9.1.1 Associations

9.1.1.1 (Unnamed Target Role) [0..\*]: Unnamed Association (Unnamed Supplier Role) to Connection

9.1.1.2 (Unnamed Target Role) [0..\*]: Unnamed Association (Unnamed Supplier Role) to ConnectionCreateData

9.1.1.3 (Unnamed Target Role) [0..\*]: Unnamed Association (Unnamed Supplier Role) to Call

9.1.1.4 (Unnamed Target Role) [0..\*]: Unnamed Association (Unnamed Supplier Role) to CallCreateData

### 9.2 CR-LDPProtocol

The CR-LDP protocol object provides a management view of the specific attributes of a CR-LDP protocol instance.

Object Lifecycle: Could be created and deleted by Management Action

Object Creation Notification Issued? Yes

Object Deletion Notification Issued? Yes

Derived from: SignallingProtocol

## 9.2.1 Attributes

### 9.2.1.1 mplsLdpEntityAdminStatus: StandardAdminStateType = LOCKED

If set to 'unlocked', then Entity will attempt to establish a new session with the Peer.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

### 9.2.1.2 mplsLdpGracefulRestart: StandardAdminStateType = LOCKED

Flag to enable/disable the graceful restart procedure.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

## 9.3 Call

The Call Object represents the service between two end points in the network. Calls are identified globally.

Object Lifecycle: The Call object is the result of an establishCall operation. The Call object exists for the lifetime of the Call.

Object Creation Notification Issued? No; it is expected that Calls are available for inspection when required, but due to potential high call volume, individual calls do not issue notifications.

Object Deletion Notification Issued? No

Derived from: CommonResourceInformation

## 9.3.1 Attributes

### 9.3.1.1 connectionSetupType: ConnectionKindType = NA

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

### 9.3.1.2 callState: CallStateType = NA

This attribute represents the state of the Call. The Call state depends on the state of the associated Connections.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? None

### 9.3.1.3 **aEnd: CallEndType = NA**

This attribute represents the source end point of the Call. The value of this attribute shall be in one of the following name types:

- SNPP;
- SNP;
- TRI Name, which identifies either a TRI or a Group-TRI Name.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

### 9.3.1.4 **zEnd: CallEndType = NA**

This attribute represents the sink end point of the Call. The value of this attribute shall be in one of the following name types:

- SNPP;
- SNP;
- TRI Name, which identifies either a TRI or a Group-TRI Name.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

### 9.3.1.5 **alarmReportingIndication: StandardAdminStateType = LOCKED**

Provides an indication of whether alarm reporting for this Call is administratively activated or de-activated.

Unlocked = alarm reporting is activated; locked = alarm reporting is de-activated.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? AVC

### 9.3.1.6 **asapPointer: ObjectIdType = Null**

This attribute indicates the assignment of an Alarm Severity Assignment Profile to the Call.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? AVC

### 9.3.1.7 **originator: PlaneType = NA**

This attribute indicates the origin of the Call create request.

NOTE – The attribute has the scope of an individual NE.

Allowed values are ControlPlane or ManagementPlane.

Readable by EMS? Yes

Writable by EMS? No

Invariant? Yes

Notification Type? None

#### **9.3.1.8 callParameter: CallParameterType = NA**

This attribute identifies the requested thresholds (wrt. the number of Connections) of the Call:

- Degraded Threshold;
- Severely Degraded Threshold.

Readable by EMS? Yes

Writable by EMS? Yes

Invariant? No

Notification Type? AVC

#### **9.3.1.9 diversityData: DiversityDataType = NA**

Readable by EMS? Yes

Writable by EMS? No

Invariant? Yes

Notification Type? None

#### **9.3.1.10 revertive: Boolean = True**

Readable by EMS? Yes

Writable by EMS? No

Invariant? Yes

Notification Type? None

#### **9.3.1.11 recovery: RecoveryType = NONE**

Allowed values are none (0), restorable (1), protected (2).

Readable by EMS? Yes

Writable by EMS? No

Invariant? Yes

Notification Type? None

### **9.3.2 Associations**

**9.3.2.1** (Unnamed Target Role) [0..\*]: Unnamed Association (Unnamed Supplier Role) to Connection

**9.3.2.2** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to CallService

### **9.3.3 Operations**

**9.3.3.1 getConnection (name: ObjectIdType = NA, connection: Connection = Null): void**

This operation retrieves the details of a Connection given a specified Connection name.

NOTE – (In a control plane environment) The managed element has to control the originating end of the Call otherwise the Connection information is not available.

Operation exceptions:

Standard

Parameter description:

*name*

*/\* in \*/* The name of the Connection to be retrieved.

*connection*

*/\* out \*/* The Connection object requested.

Pre-conditions:

None

Post-conditions:

The system remains unchanged.

**9.3.3.2 addConnections (connectionCreateDataList: SequenceOf ConnectionCreateData = NA, connectionRouteRearrangementAllowed: Boolean = False, createdConnections: SequenceOf Connection = empty list, errorReason: String = empty string): void**

This operation adds one or more Connections to an existing Call. If necessary, it is also possible to set any TP transmission parameters.

The operation will either set up the entire Connection, or will fail the entire Connection. There is no partial Connection state.

If routing constraints are specified for the Connections in the request, they should match the constraints based on the degree mandated by the appropriate effort statement. If the managed element cannot satisfy the routing criteria, within the boundaries of the effort level requested, an exception is raised.

Operation exceptions:

- 1) NOT\_IMPLEMENTED – Raised when the entire operation is not supported, or the operation with the specified input parameters is not supported.
- 2) INTERNAL\_ERROR – Raised when the operation has resulted in an internal error.
- 3) INVALID\_INPUT – Raised when the operation contains an input parameter that is syntactically incorrect or identifies an object of the wrong type or is out of range.
- 4) ENTITY\_NOT\_FOUND – Raised when any specified object instance does not exist.
- 5) USERLABEL\_IN\_USE – Raised when the user label uniqueness constraint is not met; the specified user label is currently being used.
- 6) UNABLE\_TO\_COMPLY – Raised when the operation cannot respond to the request.
- 7) UNSUPPORTED\_ROUTING\_CONSTRAINTS – Raised when the operation is unable to satisfy the requested routing constraints.
- 8) OBJECT\_IN\_USE – Raised when the object identified in the request is currently in use.
- 9) CAPACITY\_EXCEEDED: Raised when the request will result in resources being created or activated beyond the current capacity.

Parameter description:

*connectionCreateDataList*

/\* in \*/ A list of parameters defining the Connection(s) to be created in support of the Call.

*connectionRouteRearrangementAllowed*

/\* in \*/ Defines whether any existing Connections can be rearranged in order to meet the request.

False = rearrangement not allowed; True = rearrangement allowed.

*createdConnections*

/\* out \*/ A list of Connections.

*errorReason*

/\* out \*/ Specifies the creation error(s) if any.

Pre-conditions:

None

Post-conditions:

The operation has created the connections specified, or no connections have been created. Connections are not partially created.

### **9.3.3.3 removeConnections (connectionNameList: ObjectIdListType = empty list, modifiedCall: Call = NA): void**

This operation removes one or more supporting Connections from an existing Call. If necessary, it is also possible to set any TP transmission parameters.

Operation exceptions:

Standard

Parameter description:

*connectionNameList*

/\* in \*/ The list of Connections to be removed.

*modifiedCall*

/\* out \*/ The modified Call object.

Pre-conditions:

None

Post-conditions:

The connections specified are removed or an exception has been raised.

### **9.3.3.4 setAlarmReporting (activated: Boolean = NA): void**

This operation activates or de-activates alarm reporting on the Call.

Operation exceptions:

Standard

Parameter description:

*activated*

/\* in \*/ Defines whether alarm reporting shall be activated = "true" or de-activated = "false".



## 9.4 CallCreateData

This object class represents the read-create attributes that are required to create a Connection. This object will be passed as one of the parameters in the establishCall operation or in the addConnections operation.

Object Lifecycle: This is part of a Call creation operation. Lifecycle is limited to the duration of the operation.

Object Creation Notification Issued? No

Object Deletion Notification Issued? No

### 9.4.1 Attributes

#### 9.4.1.1 name: ObjectIdType = NA

Readable by EMS? No

Writable by EMS? Yes

Invariant? Yes

Notification Type? None

#### 9.4.1.2 userLabel: String = empty string

This attribute represents a provisionable, user-friendly name for the Call object to be created. It may be initialized to the value in the name attribute by the NE; however, the userLabel attribute is owned and may be set by the EMS.

This attribute may be unique amongst all instances of this object in the control plane.

Readable by EMS? No

Writable by EMS? Yes

Invariant? Yes

Notification Type? None

#### 9.4.1.3 aEnd: CallEndType = NA

This parameter identifies the source end point of the Call. The value of this parameter shall be in one of the following name types:

- SNPP;
- SNP;
- TRI Name, which identifies either a TRI or a Group-TRI Name.

If the new Call shall have multiple Connections, then only SNPP or Group-TRI is allowed.

Readable by EMS? No

Writable by EMS? Yes

Invariant? Yes

Notification Type? None

#### **9.4.1.4 zEnd: CallEndType = NA**

This parameter identifies the sink end point of the Call. The value of this parameter shall be in one of the following name types:

- SNPP;
- SNP;
- TRI Name, which identifies either a TRI or a Group-TRI Name.

If the new Call shall have multiple Connections, then only SNPP or Group-TRI is allowed.

Readable by EMS? No

Writeable by EMS? Yes

Invariant? Yes

Notification Type? None

#### **9.4.1.5 callParameter: CallParameterType = NA**

This parameter shall identify the requested thresholds (wrt. the number of Connections) of the new Call:

- Degraded Threshold;
- Severely Degraded Threshold.

Readable by EMS? No

Writeable by EMS? Yes

Invariant? Yes

Notification Type? None

#### **9.4.1.6 diversityData: DiversityDataType = NA**

Readable by EMS? No

Writeable by EMS? Yes

Invariant? Yes

Notification Type? None

#### **9.4.1.7 revertive: Boolean = TRUE**

Readable by EMS? No

Writeable by EMS? Yes

Invariant? Yes

Notification Type? None

#### **9.4.1.8 recovery: RecoveryType = NONE**

Allowed values are none (0), restorable (1), protected (2).

Readable by EMS? No

Writeable by EMS? Yes

Invariant? Yes

Notification Type? None

#### **9.4.1.9 asapPointer: ObjectIdType = Null**

This attribute indicates the assignment of an ASAP to the Call.

Readable by EMS? No

Writeable by EMS? Yes

Invariant? Yes

Notification Type? None

#### **9.4.2 Associations**

##### **9.4.2.1 connectionRelatedData [0..\*]: Unnamed Association (Unnamed Supplier Role) to ConnectionCreateData**

Readable by EMS? No

Writeable by EMS? Yes

Invariant? Yes

Notification Type? None

#### **9.5 CallDetailRecord**

The Call Detail Record provides billing information for a specific Call.

Object Lifecycle: Created when the Call is established and explicitly deleted by the EMS after the Call is ended.

Object Creation Notification Issued? No

Object Deletion Notification Issued? No

##### **9.5.1 Attributes**

###### **9.5.1.1 callId: OctetString = NA**

This attribute represents the unique identifier of the Call.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

###### **9.5.1.2 startTime: TimeOfDayType = NA**

Provides tracking of the duration of a Call sourced by the Call Controller.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

###### **9.5.1.3 stopTime: TimeOfDayType = NA**

Provides tracking of the duration of a Call sourced by the Call Controller.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes (written once when the Call ends)

Notification Type? None

#### **9.5.1.4 callDetailData: AnyType = NA**

Provides additional information about the Call.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? None

### **9.5.2 Associations**

**9.5.2.1** (Unnamed Target Role) []: Unnamed Association (Unnamed Supplier Role) to CallService

## **9.6 CallService**

The Call Service provides the point to create, query and delete Calls, and represents the management view of a Call Controller ASON Component. A control plane instance contains one CallService object, which is bound to the SNPP Links for which the Call Service is being provided.

Object Lifecycle: Lifecycle is associated with the lifecycle of the control plane.

Object Creation Notification Issued? Yes

Object Deletion Notification Issued? Yes

Derived from: ControlPlaneService

### **9.6.1 Attributes**

#### **9.6.1.1 callAttempts: Integer = 0**

Provides information about the number of call setup requests (or attempts) received by the Call Service.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? None

#### **9.6.1.2 blockedCalls: Integer = 0**

Provides information about the number of Call setup requests received by the Call Service that have been blocked.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? None

### **9.6.2 Associations**

**9.6.2.1** (Unnamed Target Role) []: Unnamed Association (Unnamed Supplier Role) to CallDetailRecord

**9.6.2.2** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to ControlPlane

**9.6.2.3** (Unnamed Target Role) []: Unnamed Association (Unnamed Supplier Role) to SNPPLink

**9.6.2.4** (Unnamed Target Role) [0..\*]: Unnamed Association (Unnamed Supplier Role) to Call

### **9.6.3 Operations**

**9.6.3.1** **getCallDetailRecords (callName: ObjectIdType = NA, detailList: ObjectIdListType = empty list): void**

This operation returns a list of object identifiers for all Call Detail Records associated with the specified Call.

Operation exceptions:

Standard

Parameter description:

*callName*

*/\* in \*/* This parameter provides the object identifier of the desired Call object.

*detailList*

*/\* out \*/* This parameter returns a list of object identifiers for all Call Detail Records assigned to the requested Call. In the event of no Call Detail Records being available, an empty list is returned.

Pre-conditions:

None

Post-conditions:

The system remains unchanged.

**9.6.3.2** **establishCall (callCreateData: CallCreateData = NA, connectionCreateDataList: SequenceOf ConnectionCreateData = NA, numberOfRouteGroups: Integer = NA, callAndConnections: CallAndConnectionsType = NA, errorReason: String = empty string): void**

This operation establishes a point-to-point Call with zero or more Connections across one or more Routing Areas. The operation is atomic; i.e., the control plane will either set up the entire Call that matches the parameters requested (including all requested Connections) or will fail the entire Call.

At least one end point of the Call has to be within the managed element domain. The other end points may be "off-network" (not directly visible to the local managed element).

If routing constraints are specified for the associated Connections, they should match the constraints based on the degree mandated by the appropriate effort statement.

Operation exceptions:

- 1) NOT\_IMPLEMENTED – Raised when the entire operation is not supported, or the operation with the specified input parameters is not supported.
- 2) INTERNAL\_ERROR – Raised when the operation has resulted in an internal error.
- 3) INVALID\_INPUT – Raised when the operation contains an input parameter that is syntactically incorrect or identifies an object of the wrong type or is out of range.
- 4) ENTITY\_NOT\_FOUND – Raised when any specified object instance does not exist.
- 5) USERLABEL\_IN\_USE – Raised when the user label uniqueness constraint is not met; the specified user label is currently being used.

- 6) UNABLE\_TO\_COMPLY – Raised when the operation cannot respond to the request.
- 7) UNSUPPORTED\_ROUTING\_CONSTRAINTS – Raised when the operation is unable to satisfy the requested routing constraints.
- 8) OBJECT\_IN\_USE – Raised when the object identified in the request is currently in use.

Parameter description:

*callCreateData*

*/\* in \*/* The parameters describing the Call to be established.

*connectionCreateDataList*

*/\* in \*/* A list of connectionCreateData objects that define the Connections to be created to support the Call.

*numberOfRouteGroups*

*/\* in \*/* This parameter identifies the number of requested Route Groups in case the Route Group names are not provided with the Connection-related data.

A non-zero value defines the number of requested Route Groups. In this case, there must not be any Route Group names provided with the Connection-related data.

Value "0" means that the NMS does not care about the number of Route Groups.

Value "NA" means that this parameter can be ignored.

*callAndConnections*

*/\* out \*/* The newly created Call object and the associated Connection objects.

*errorReason*

*/\* out \*/* Specifies the creation error(s) if any.

Pre-conditions:

None

Post-conditions:

The call is created according to the specifications, or an exception is raised.

Note that partial connections are not allowed and all the connections are created or the operation fails.

### 9.6.3.3 **releaseCall (callName: ObjectIdType = NA, remainingConnections: SequenceOf Connection = empty list): void**

This operation releases a point-to-point Call with zero or more Connections. The operation is directed against the CallService object, which contains the Call. As a result of this operation the Call and the supporting Connections will be deleted from the network.

If the managed element is able to determine that the Call is still carrying traffic, the operation is rejected.

Operation exceptions:

Standard

Parameter description:

*callName*

*/\* in \*/* The name of the Call to be released.

*remainingConnections*

*/\* out \*/* The list of Connections that could not be deleted.

Pre-conditions:

None

Post-conditions:

The specific call has been released or an exception has been raised.

#### **9.6.3.4 getAllCallsAndConnections (callsAndConnections: CallAndConnectionsListType = empty list): void**

This operation returns all the Calls originated by the managed element (with all the attributes for each Call) and the Connections for each Call (with all the attributes for each Connection).

The Connections are returned regardless of their state.

This operation will only return the Calls (and associated connections) that originate on this CallService object. Note that the reply can be an empty list, only the Calls and Connections that originate on this Managed Element are returned. By implication this informs the EMS which managed element controls each Call.

Operation exceptions:

Standard

Parameter description:

*callsAndConnections*

*/\* out \*/* A list of all Calls originated by the CallService object and their associated Connection objects, or an empty list.

Pre-conditions:

None

Post-conditions:

The system remains unchanged.

#### **9.6.3.5 getCallsAndConnectionsWithTP (tpName: ObjectIdType = NA, callsAndConnections: CallAndConnectionsListType = empty list): void**

This operation retrieves the details of all Calls (and their associated Connections) associated with the specified Termination Point (TTP, CTP, SNPP, or TRI).

A Call is using the specified point if any of its Connections, in any state, use this point.

Operation exceptions:

Standard

Parameter description:

*tpName*

*/\* in \*/* The termination point (TTP, CTP, SNPP or TRI) associated to the Calls to be retrieved.

*callsAndConnections*

*/\* out \*/* A list of all Calls associated to the provided termination point and their associated Connection objects, or an empty list.

Pre-conditions:

None

Post-conditions:

The system remains unchanged.

#### **9.6.3.6 resetPMCounters (): void**

This operation resets the counter attributes for this CallService object instance to zero.

Operation exceptions:

Standard

Pre-conditions:

None

Post-conditions:

The CallService counters have been reset to zero.

### **9.7 CommonResourceInformation**

This object provides common information elements for all resource objects. It is used for inheritance purposes only.

Object Lifecycle: abstract object

Object Creation Notification Issued? No

Object Deletion Notification Issued? No

#### **9.7.1 Attributes**

##### **9.7.1.1 identifier: ObjectIdType = NA**

The attribute is assigned by the EMS. It represents a unique identifier of the object within the context of the NE.

Readable by EMS? Yes

Writable by EMS? No

Invariant? Yes

Notification Type? None

##### **9.7.1.2 userLabel: String = Value of the instanceId of the identifier attribute**

This attribute represents a provisionable, user-friendly name for the object. It is initialized to the value of the instanceId of the identifier attribute by the NE; however, the userLabel attribute is owned and may be set by the EMS.

The difference between the userLabel and the instanceId of the identifier attribute is that the userLabel is an attribute of the object that may be "set" by the EMS.

This attribute may be unique amongst all instances of this object in the control plane.

Readable by EMS? Yes

Writable by EMS? Yes

Invariant? No

Notification Type? AVC



## 9.7.2 Operations

### 9.7.2.1 retrieveObjectInformation (objectData: AVPListType = empty list): void

This operation retrieves the data of the own object, and causes no change in system state.

Operation exceptions:

Standard

Parameter description:

*objectData*

*/\* out \*/* This parameter returns the data for the own object as a set of attribute names and values.

Pre-conditions:

None

Post-conditions:

There is no change in system state as a result of this operation.

## 9.8 Connection

The Connection object provides a view of a single Connection that supports a particular Call.

Object Lifecycle: Lifecycle is from setup to teardown of the connection.

Object Creation Notification Issued? No

Object Deletion Notification Issued? No

Derived from: CommonResourceInformation

### 9.8.1 Attributes

#### 9.8.1.1 connectionState: ConnectionStateType = NA

Provides information about the state of a particular Connection.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? None

#### 9.8.1.2 direction: ConnectionDirectionType = NA

Specified by EMS by object creation request.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

#### 9.8.1.3 connectionSetupType: ConnectionKindType = NA

This attribute identifies the setup type of the Connection. The type shall be one of the following:

- PC (permanent connection);
- SC (switched connection); or
- SPC (soft permanent connection).

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

#### **9.8.1.4 alarmReportingIndication: StandardAdminStateType = LOCKED**

Provides an indication of whether alarm reporting for this Connection is administratively activated or de-activated.

Unlocked = alarm reporting is activated; locked = alarm reporting is de-activated.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? AVC

#### **9.8.1.5 asapPointer: ObjectIdType = NULL**

This attribute indicates the assignment of an Alarm Severity Assignment Profile to the Connection.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

#### **9.8.1.6 connectionRequestType: SNCType = SIMPLE**

This version only supports SIMPLE connection requests.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

### **9.8.2 Associations**

**9.8.2.1** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to Call

**9.8.2.2** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to RoutingArea

**9.8.2.3** zEndId [1..\*]: Connection\_is\_terminated\_by\_zEnd\_SNP (Unnamed Supplier Role) to SNP

This association models the Z end of a Connection.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

**9.8.2.4** aEndId [1..\*]: Connection\_is\_terminated\_by\_aEnd\_SNP (Unnamed Supplier Role) to SNP

This association models the A end of a Connection.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

**9.8.2.5** (Unnamed Target Role) [0..\*]: Connection\_identifies\_aEnd\_TP (Unnamed Supplier Role) to TerminationPoint

**9.8.2.6** (Unnamed Target Role) [0..\*]: Connection\_identifies\_zEnd\_TP (Unnamed Supplier Role) to TerminationPoint

### **9.8.3 Operations**

**9.8.3.1 getRoute (routeList: SNPPLinkOrSNPPListType = empty list): void**

This operation retrieves the route followed by the Connection. When the node has performed source routing, then complete route details may be returned. In other cases, only the ingress and egress SNPP links may be returned and the manager must navigate further details. When a link is at the edge of a domain, then an SNPP is returned and further navigation is not possible.

Operation exceptions:

Standard

Parameter description:

*routeList*

*/\* out \*/* Contains the details of the Route as a sequence of SNPPLinks or SNPPs (when the node is at the end of a domain).

Pre-conditions:

None

Post-conditions:

The system remains unchanged.

## **9.9 ConnectionCreateData**

This object class represents the read-create attributes that are required to create a Connection. This object is passed as one of the parameters in the establishCall operation or in the addConnections operation.

Object Lifecycle: This is part of Connection creation operations, and the lifecycle is limited to the duration of the operation.

Object Creation Notification Issued? No

Object Deletion Notification Issued? No

### **9.9.1 Attributes**

**9.9.1.1 name: ObjectIdType = NA**

Assigned by EMS. Unique identifier of the object within the context of the EMS.

Readable by EMS? No

Writable by EMS? Yes – on create only

Invariant? Yes

Notification Type? None

#### **9.9.1.2 userLabel: String = empty string**

This attribute represents a provisionable, user-friendly name for the Call object to be created. It may be initialized to the value in the name attribute by the NE; however, the userLabel attribute is owned and may be set by the EMS.

This attribute may be unique amongst all instances of this object in the EMS.

Readable by EMS? No

Writable by EMS? Yes

Invariant? Yes

Notification Type? None

#### **9.9.1.3 aEnd: CallEndType = NA**

This parameter identifies the source end point of the Connection. The value of this parameter shall be in one of the following name types:

- SNPP;
- SNP;
- TRI Name, which identifies either a TRI or a Group-TRI Name.

Readable by EMS? No

Writable by EMS? Yes

Invariant? Yes

Notification Type? None

#### **9.9.1.4 zEnd: CallEndType = NA**

This parameter identifies the sink end point of the Connection. The value of this parameter shall be in one of the following name types:

- SNPP;
- SNP;
- TRI Name, which identifies either a TRI or a Group-TRI Name.

Readable by EMS? No

Writable by EMS? Yes

Invariant? Yes

Notification Type? None

#### **9.9.1.5 direction: ConnectionDirectionType = BIDIRECTIONAL**

Direction of the Connection.

Readable by EMS? No

Writable by EMS? Yes

Invariant? Yes

Notification Type? None

#### **9.9.1.6 alarmReportingIndication: StandardAdminStateType = LOCKED**

Provides an indication of whether alarm reporting for this Connection is administratively activated or de-activated.

Unlocked = alarm reporting is activated; locked = alarm reporting is de-activated.

Readable by EMS? No

Writeable by EMS? Yes

Invariant? Yes

Notification Type? None

#### **9.9.1.7 asapPointer: ObjectIdType = Null**

This attribute indicates the assignment of an Alarm Severity Assignment Profile to the Connection.

Readable by EMS? No

Writeable by EMS? Yes

Invariant? Yes

Notification Type? None

#### **9.9.1.8 connectionRequestType: SNCType = SIMPLE**

This version only supports SIMPLE connection requests.

Readable by EMS? No

Writeable by EMS? Yes

Invariant? Yes

Notification Type? None

### **9.9.2 Associations**

#### **9.9.2.1 (Unnamed Target Role) []: Unnamed Association (connectionRelatedData Role) to CallCreateData**

Readable by EMS? No

Writeable by EMS? Yes

Invariant? Yes

Notification Type? None

### **9.10 ConnectionService**

The ConnectionService represents the management view of a Connection Controller ASON Component. A Network Element may contain 0 or more ConnectionService objects, which are bound to the SNPP links for which the Connection Service is being provided.

Object Lifecycle: Lifecycle is from startup of the Connection Controller Component to termination of the component.

Object Creation Notification Issued? Yes

Object Deletion Notification Issued? Yes

Derived from: ControlPlaneService

## 9.10.1 Associations

**9.10.1.1** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to ControlPlane

## 9.10.2 Operations

**9.10.2.1** **getConnectionNamesByTP (ttpName: ObjectIdType = NA, connectionNameList: ObjectIdListType = empty list): void**

This operation returns the list of Connections that are directly supported by the trail that is terminated by this TTP. If the managed element is unable to identify any Connections, an exception is raised.

Operation exceptions:

Standard

Parameter description:

*ttpName*

*/\* in \*/* This parameter specifies the object identity of the Trail Termination Point being queried.

*connectionNameList*

*/\* out \*/* This parameter contains the names of all Connections supported by the requested TTP. This may be an empty list in the case that either the TTP is not supporting any Connections.

Pre-conditions:

None

Post-conditions:

The system remains unchanged.

## 9.11 ConnectionTerminationPoint

This object class is outside the scope of this Recommendation. It has been added here to show the relationship between the SNP and its implementing CTP.

## 9.12 ControlPlane

The ControlPlane object provides containment for all the control plane management objects, allowing for the Control Plane to be managed on non-NE platforms.

The control plane is logically independent from any Transport Plane entities (although it could physically reside in a Transport Plane NE).

Object Lifecycle: Lifecycle is the lifecycle of the Network Element (or other) platform that supports control plane components.

Object Creation Notification Issued? No

Object Deletion Notification Issued? No

Derived from: CommonResourceInformation

### 9.12.1 Associations

**9.12.1.1** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to ManagedElement

**9.12.1.2** (Unnamed Target Role) [0..1]: Unnamed Association (Unnamed Supplier Role) to OSPFProtocol

- 9.12.1.3 (Unnamed Target Role) [0..1]: Unnamed Association (Unnamed Supplier Role) to PNNIRoutingProtocol
- 9.12.1.4 (Unnamed Target Role) [0..1]: Unnamed Association (Unnamed Supplier Role) to ISISProtocol
- 9.12.1.5 (Unnamed Target Role) [0..\*]: Unnamed Association (Unnamed Supplier Role) to SignallingProtocol
- 9.12.1.6 (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to CallService
- 9.12.1.7 (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to ConnectionService

## 9.12.2 Operations

### 9.12.2.1 **createObjectInstance (className: String = NA, instanceData: AVPListType = NA, object: ObjectIdType = Null): void**

This operation will create an instance of the specified object class. Note that only writeable or "set by create" attributes can be specified as part of instanceData.

Note that not all object instances can be created by this operation (e.g., Call).

Some objects are created as the result of very significant activity in the control plane, and this generic create operation cannot have any knowledge of the very special behaviour of (for example) creating a Call object.

Some objects are also expected to be automatically created when the platform supporting those objects is initialized. It is not expected to create such objects by independent management action.

This specification takes care of these significant differences by providing special operations to create these special objects, and it is a convention that these special operations are used in place of this generic operation.

Operation exceptions:

Standard

Parameter description:

*className*

*/\* in \*/* This parameter provides the identifier of the object class to be created.

*instanceData*

*/\* in \*/* This parameter provides a sequence of attribute value pairs which are to be used as initial data for object creation of the specified object class.

*object*

*/\* out \*/* This parameter returns the identifier of the object instance created.

Pre-conditions:

None

Post-conditions:

The Object instance requested has been created, or an exception has been raised.

### 9.12.2.2 **deleteObjectInstance (objectInstance: ObjectIdType = NA): void**

Note that not all object instances can be deleted by this operation.

When some objects are deleted, very significant activity in the control plane can occur and this generic delete operation cannot have any knowledge of the very special behaviour of (for example) deleting a Call object.

This specification takes care of these significant differences by providing special operations to delete these special objects, and it is a convention that these special operations are used in place of this generic operation.

Operation exceptions:

Standard

Parameter description:

*objectInstance*

*/\* in \*/* This parameter provides the object identifier of the object to be deleted.

Pre-conditions:

None

Post-conditions:

The specified object is deleted unless that object has dependant objects, in which case the delete will fail.

### **9.13 ControlPlaneService**

The ControlPlaneService object provides inheritance of common attributes for all control plane Services (e.g., Routing, Signalling and Discovery).

Object Lifecycle: Lifecycle is from instantiation of the service to termination of the service.

Object Creation Notification Issued? Yes

Object Deletion Notification Issued? Yes

Derived from: CommonResourceInformation

#### **9.13.1 Attributes**

##### **9.13.1.1 statusReportingEnabled: Boolean = false**

Defines whether status reporting is enabled or disabled for the identified component.

Readable by EMS? Yes

Writable by EMS? Yes

Invariant? No

Notification Type? None

##### **9.13.1.2 operationalState: StandardOperStateType = NA**

Operational State of a ControlPlaneService component (e.g., CallService, ConnectionService, DiscoveryService, SignallingProtocol).

Readable by EMS? Yes

Writable by EMS? No

Invariant? No

Notification Type? StateChange



### 9.13.1.3 **currentProblemList: CurrentProblemListType = empty list**

Autonomously provides detailed information about the health of control plane components.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? None

### 9.13.2 **Operations**

#### 9.13.2.1 **setStatusReporting (statusReporting: ActivationStateType = NA): void**

Allows the management plane to specify whether status reporting is enabled or disabled for the identified components. The operation is idempotent and no error will occur in the event that the operation required no change in state.

Operation exceptions:

Standard

Parameter description:

*statusReporting*

*/\* in \*/* This parameter selects whether status reporting is to be enabled or disabled.

Pre-conditions:

None

Post-conditions:

Status Reporting has been set as requested.

### 9.14 **DiscoveryService**

This object represents the capability of the NE to support Discovery Services.

Object Lifecycle: Inherited

Object Creation Notification Issued? Inherited

Object Deletion Notification Issued? Inherited

Derived from: ControlPlaneService

#### 9.14.1 **Associations**

9.14.1.1 (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to ManagedElement

#### 9.14.2 **Operations**

##### 9.14.2.1 **enableDiscoveryService (ttpList: ObjectIdListType = NA): void**

This operation enables the Discovery Service on a specified list of TTPs in the NE. This operation is idempotent and no error will occur in the event that the operation required no change in state.

Operation exceptions:

Standard

Parameter description:

*ttpList*

/\* in \*/ This parameter provides the set of Termination Points for which the discovery service is to be enabled.

Pre-conditions:

None

Post-conditions:

Discovery Service is enabled for the specified TTPs.

#### **9.14.2.2 disableDiscoveryService (ttpList: ObjectIdListType = NA): void**

This operation disables the Discovery Service on specified TTPs in the NE. This operation is idempotent, and no error will occur in the event that the operation required no change in state.

Operation exceptions:

Standard

Parameter description:

*ttpList*

/\* in \*/ This parameter provides the set of Termination Points for which the discovery service is to be disabled.

Pre-conditions:

None

Post-conditions:

Discovery Service is disabled for the specified TTPs.

### **9.15 ISISArea**

The ISISArea object provides a container for information specific to a particular IS-IS Routing Area. The IS-IS Routing area defines a sub-set of IS-IS Routing instances that exchange information on the SCN links connecting the IS-IS instances.

Object Lifecycle: Lifecycle is from instantiation of the ISIS Routing Service to termination of that service.

Object Creation Notification Issued? Yes

Object Deletion Notification Issued? Yes

Derived from: CommonResourceInformation

#### **9.15.1 Attributes**

##### **9.15.1.1 isisAreaAddresses: SetOf OctetString = NA**

Each AreaAddress is an Octet String of variable length with a minimum length of 4 octets and a maximum length of 18 octets.

At least one Area Address has to be set.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? AVC

**9.15.1.2 isisMaximumAreaAddresses: Integer = NA**

The attribute identifies the maximum number of Area Addresses.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? Yes

Notification Type? None

**9.15.1.3 isisMaximumLSPGenerationInterval: Integer = 1800**

Time in seconds.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

**9.15.1.4 isisMinimumLSPGenerationInterval: Integer = 30**

Time in seconds.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

**9.15.1.5 isisMaximumLSPSize: Integer = NA**

Cannot be larger than the smallest MTU size in the area.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

**9.15.1.6 isisHelloTimer: Integer = 15**

Time in seconds.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

**9.15.1.7 isisCompleteSNPIInterval: Integer = 600**

Time in seconds.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

## 9.15.2 Associations

**9.15.2.1** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to RoutingArea

**9.15.2.2** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to ISISProtocol

**9.15.2.3** (Unnamed Target Role) [0..\*]: Unnamed Association (Unnamed Supplier Role) to ISISNeighbour

## 9.16 ISISNeighbour

The ISISNeighbour object provides a container used by the local IS-IS instance for information specific to a single IS-IS neighbour. The IS-IS neighbour relationship exists in the control plane independent of the existence of SNPP Links. As a result, this object is limited to information specific to the control plane.

Object Lifecycle: Lifecycle is from the appearance of a neighbour to removal of the neighbour.

Object Creation Notification Issued? Yes

Object Deletion Notification Issued? Yes

Derived from: CommonResourceInformation

### 9.16.1 Attributes

#### 9.16.1.1 isisNeighbourState: StandardNeighbourStateType = DOWN

[ISO/IEC 10589] (2002), page 119: The functional level of the neighbour conversation.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? None

#### 9.16.1.2 isisNeighbourId: Integer {1-8 Bytes} = NA

[ISO/IEC 10589] (2002), page 120: The ISIS System ID of the neighbouring router. The Neighbour ID is learned when Hello packets are received from the neighbour.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

#### 9.16.1.3 isisNeighbourIPAddress: Integer {32bit} = NA

The IP address of the neighbouring routing controller protocol controller.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? AVC

## 9.16.2 Associations

### 9.16.2.1 (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to ISISArea

## 9.17 ISISProtocol

The ISISProtocol object provides a management view of the specific attributes of an ISIS protocol instance.

Object Lifecycle: Inherited

Object Creation Notification Issued? Inherited

Object Deletion Notification Issued? Inherited

Derived from: ControlPlaneService

### 9.17.1 Attributes

#### 9.17.1.1 scnIPAddress: OctetString = empty string

This attribute represents the control plane address of the protocol instance. Its actual value and length depends on the protocol specific implementation, and on the specific control plane in use.

At create time, its value should reflect the address of the NE platform.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? AVC

#### 9.17.1.2 isisAdminState: StandardAdminStateType = LOCKED

This attribute represents the administratively settable state of the ISIS protocol.

At create, the protocol instance is locked, and remains locked until unlocked by a management operation.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

#### 9.17.1.3 isisSystemId: Integer {1-8 Bytes} = NA

6 bytes typically.

Readable by EMS? Yes

Writeable by EMS? No; set by create

Invariant? Yes

Notification Type? None

#### 9.17.1.4 isisSystemIdLength: Integer {1-8} = NA

Readable by EMS? Yes

Writeable by EMS? No; set by create

Invariant? Yes

Notification Type? None

### 9.17.2 Associations

**9.17.2.1** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to ControlPlane

**9.17.2.2** (Unnamed Target Role) [1..\*]: Unnamed Association (Unnamed Supplier Role) to ISISArea

### 9.17.3 Operations

#### 9.17.3.1 **configureRoutingProtocolParameters (paramList: AVPListType = NA): void**

This operation provides a set of values for the named attributes. Only attributes which are writeable and not set by create may be configured.

Operation exceptions:

None additional

Parameter description:

*paramList*

*/\* in \*/* This parameter provides the values for the writable attributes of the selected protocol object.

Pre-conditions:

None

Post-conditions:

The specified attributes are configured or NO values are changed.

#### 9.17.3.2 **setAdminState (adminState: AdminStateSetType = NA): void**

Allows the management plane to specify whether the object's Administrative state is enabled or disabled. The operation is idempotent and no error will occur in the event that the operation required no change in state.

Operation exceptions:

Standard

Parameter description:

*adminState*

*/\* in \*/* This parameter selects whether the administrative state is to be enabled (unlocked) or disabled (locked).

Pre-conditions:

None

Post-conditions:

Admin state has been set as requested or an exception has been raised.

## 9.18 LayerAdjacencyDiscovery

This object represents the capability of the TTP to support Layer Adjacency Discovery Services.

Object Lifecycle: Objects exist as long as the supporting TTP is able to support the service.

Object Creation Notification Issued? Yes

Object Deletion Notification Issued? Yes

Derived from: CommonResourceInformation

## **9.18.1 Attributes**

### **9.18.1.1 discoveryMessage: OctetString = empty string**

This attribute contains the message (see [ITU-T G.7714]) to be sent by the discovery process. It is kept up to date by the NE.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? None

### **9.18.1.2 ladStatus: LadStatusType = NA**

This attribute indicates the current status of the discovery process. It is maintained by the NE.

Allowed values are: idle, A-Z\_Known, Z-A\_Known, A-Z\_Z-A\_Known.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? None

### **9.18.1.3 remoteEndPoint: ObjectIdType = Null**

This attribute represents the name of the remote Trail Termination Point that is connected to the containing TTP. It is set to NULL by Create and is autonomously maintained by the Discovery Process.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? None

### **9.18.1.4 remoteDiscoveryAgent: OctetString = Null**

This attribute represents the control plane address of the remote Discovery Agent. It is set to NULL by create, and is autonomously maintained by the Discovery Process. The attribute length and value depends on the control plane used.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? AVC

## **9.18.2 Associations**

**9.18.2.1** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to TrailTerminationPoint

## **9.18.3 Operations**

### **9.18.3.1 startLayerAdjacencyDiscovery (): void**

This operation starts the Layer Adjacency Discovery process on the containing TTP. This operation is idempotent.

Operation exceptions:

The Discovery Service is disabled for the containing TTP.

Pre-conditions:

None

Post-conditions:

The Layer Adjacency Discovery process is started.

### **9.18.3.2 stopLayerAdjacencyDiscovery (): void**

This operation stops the Layer Adjacency Discovery process on the containing TTP. This operation is idempotent.

Operation exceptions:

Standard

Pre-conditions:

None

Post-conditions:

The Layer Adjacency Discovery process is stopped.

## **9.19 ManagedElement**

This object class is outside the scope of this Recommendation. It has been added here to show the relationship to ControlPlane and DiscoveryService.

### **9.19.1 Associations**

**9.19.1.1** (Unnamed Target Role) [0..1]: Unnamed Association (Unnamed Supplier Role) to ControlPlane

**9.19.1.2** (Unnamed Target Role) [0..1]: Unnamed Association (Unnamed Supplier Role) to DiscoveryService

**9.19.1.3** (Unnamed Target Role) [0..\*]: Unnamed Association (Unnamed Supplier Role) to TrailTerminationPoint

## **9.20 OSPFArea**

The OSPFArea object provides a container for information specific to a particular OSPF Routing Area. The OSPF Routing area defines a sub-set of OSPF Routing instances that exchange information on the SCN links connecting the OSPF instances.

Object Lifecycle: Exists as long as the OSPF area exists.

Object Creation Notification Issued? Yes

Object Deletion Notification Issued? Yes

Derived from: CommonResourceInformation



## 9.20.1 Attributes

### 9.20.1.1 scnIPAddress: OctetString = NA

This attribute represents the control plane address of the protocol instance. Its actual value and length depends on the protocol specific implementation, and on the specific control plane in use.

At create time, its value should reflect the address of the NE platform.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? AVC

### 9.20.1.2 ospfAreaId: Integer = NA

[IETF RFC 2328]: This is a 32-bit number that identifies the OSPF Area. The Area ID of 0.0.0.0 is reserved for the backbone. If the OSPF Area represents a subnetted network, the IP network number of the subnetted network may be used for the Area ID.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? Yes

Notification Type? None

### 9.20.1.3 ospfRetransInterval: Integer = 5

The number of seconds between link-state advertisement retransmissions, for adjacencies belonging to this interface. This value is also used when retransmitting database description and link-state request packets.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

### 9.20.1.4 ospfHelloInterval: Integer = 10

The length of time, in seconds, between the Hello packets that the router sends on the interface. This value must be the same for all routers attached to a common network.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

### 9.20.1.5 ospfRtrDeadInterval: Integer = 40

The number of seconds during which the router's Hello has not been seen before its neighbours declare the router down. This should be some multiple of the Hello interval. This value must be the same for all routers attached to a common network.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

## 9.20.2 Associations

**9.20.2.1** (Unnamed Target Role) [0..\*]: Unnamed Association (Unnamed Supplier Role) to OSPFNeighbour

**9.20.2.2** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to OSPFProtocol

**9.20.2.3** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to RoutingArea

## 9.21 OSPFNeighbour

The OSPFNeighbour object provides a container used by the local OSPF instance for information specific to a single OSPF neighbour. The OSPF neighbour relationship exists in the control plane independent of the existence of SNPP Links. As a result, this object is limited to information specific to the control plane.

Object Lifecycle: Exists as long as the Neighbour relationship exists.

Object Creation Notification Issued? Yes

Object Deletion Notification Issued? Yes

Derived from: CommonResourceInformation

### 9.21.1 Attributes

#### 9.21.1.1 ospfNeighbourState: OSPFNeighbourStateType = DOWN

[IETF RFC 2328]: The functional level of the neighbour conversation. This is described in more detail in section 10.1 of [IETF RFC 2328].

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

#### 9.21.1.2 ospfNeighbourId: Integer {32bit} = NA

[IETF RFC 2328]: The OSPF Router ID of the neighbouring router. The Neighbour ID is learned when Hello packets are received from the neighbour, or is configured if this is a virtual adjacency (see section C.4 of [IETF RFC 2328]).

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? None

#### 9.21.1.3 ospfNeighbourIPAddress: Integer {32bit} = NA

[IETF RFC 2328]: The IP address of the neighbouring routing controller's attachment to the SCN. Used as the Destination IP address when protocol packets are sent as unicasts along this adjacency.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? AVC

## 9.21.2 Associations

**9.21.2.1** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to OSPFArea

## 9.22 OSPFProtocol

The OSPFProtocol object provides a management view of the specific attributes of an OSPF Protocol instance.

Object Lifecycle: Inherited

Object Creation Notification Issued? Inherited

Object Deletion Notification Issued? Inherited

Derived from: ControlPlaneService

### 9.22.1 Attributes

#### 9.22.1.1 ospfAdminState: StandardAdminStateType = LOCKED

This attribute reflects the semantics of the X.721:1992 administrativeState attribute. When set to "unlocked" OSPF is enabled.

At create, its value is disabled, and remains disabled until enabled by management action.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

#### 9.22.1.2 ospfRouterId: Integer {32bit} = NA

[IETF RFC 2328]: This is a 32-bit number that uniquely identifies the router in the Autonomous System. One algorithm for Router ID assignment is to choose the largest or smallest IP address assigned to the router. If a router's OSPF Router ID is changed, the router's OSPF software should be restarted before the new Router ID takes effect. Before restarting in order to change its Router ID, the router should flush its self-originated LSAs from the routing domain (see section 14.1 of [IETF RFC 2328]), or they will persist for up to MaxAge minutes.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? AVC

### 9.22.2 Associations

**9.22.2.1** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to ControlPlane

**9.22.2.2** (Unnamed Target Role) [1..\*]: Unnamed Association (Unnamed Supplier Role) to OSPFArea

### 9.22.3 Operations

#### 9.22.3.1 **configureRoutingProtocolParameters (paramList: AVPListType = NA): void**

This operation provides a set of values for the named attributes. Only attributes which are writeable and not set by create may be configured.

Operation exceptions:

Standard

Parameter description:

*paramList*

*/\* in \*/* This parameter provides the values for the writable attributes of the selected protocol object.

Pre-conditions:

None

Post-conditions:

The specified attributes are configured or NO values are changed.

#### 9.22.3.2 **setAdminState (adminState: AdminStateSetType = NA): void**

Allows the management plane to specify whether the object Administrative state is enabled or disabled. The operation is idempotent and no error will occur in the event that the operation required no change in state.

Operation exceptions:

Standard

Parameter description:

*adminState*

*/\* in \*/* This parameter selects whether the administrative state is to be enabled (unlocked) or disabled (locked).

Pre-conditions:

None

Post-conditions:

Admin state has been set as requested.

### 9.23 PNNIArea

The PNNIArea object provides a container for information specific to a particular PNNI Routing Area. The PNNI Routing Area defines a sub-set of PNNI Routing instances that exchange information on the SCN links connecting the PNNI instances.

Object Lifecycle: Exists as long as the PNNI Area exists.

Object Creation Notification Issued? Yes

Object Deletion Notification Issued? Yes

Derived from: CommonResourceInformation

#### 9.23.1 Attributes

##### 9.23.1.1 **pnniPeerGroup: BitString = NA**

This attribute reflects the semantics of the [AF-PNNI-0055] PeerGroupId type.

[AF-PNNI-0055], Annex E: The Peer Group Id is a Bit String of variable length with a minimum length of 0 bits and a maximum length of 104 bits.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification? AVC

#### **9.23.1.2 pnniHelloInterval: Integer = 15**

[AF-PNNI-0055], Annex E: The amount of time in seconds between Hellos that a node sends on a link.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

#### **9.23.1.3 pnniInactivityFactor: Integer = 5**

[AF-PNNI-0055], Annex E: The number of consecutive Hello intervals allowed to pass without receiving a Hello.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

#### **9.23.1.4 pnniMinPTSEInterval: Integer = 10**

[AF-PNNI-0055], Annex E: Minimum time interval (in tenth of a second) between updates of any given PTSE.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

#### **9.23.1.5 pnniPTSERefreshInterval: Integer = 1800**

[AF-PNNI-0055], Annex E: Time in seconds between reorigination of a PTSE in absence of triggered updates.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

### **9.23.2 Associations**

**9.23.2.1** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to RoutingArea

**9.23.2.2** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to PNNIRoutingProtocol

**9.23.2.3** (Unnamed Target Role) [0..\*]: Unnamed Association (Unnamed Supplier Role) to PNNINeighbour

## **9.24 PNNINeighbour**

The PNNI Neighbour object provides a container used by the local PNNI routing instance for information specific to a single PNNI neighbour. The PNNI neighbour relationship exists in the control plane independent of the existence of SNPP Links. As a result, this object is limited to information specific to the control plane.

Object Lifecycle: Exists as long as the neighbour relationship exists.

Object Creation Notification Issued? Yes

Object Deletion Notification Issued? Yes

Derived from: CommonResourceInformation

### **9.24.1 Attributes**

#### **9.24.1.1 pnniHelloState: StandardNeighbourStateType = DOWN**

[AF-PNNI-0055], Annex H, page 287: The functional level of the neighbour conversation.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? None

#### **9.24.1.2 pnniRemoteNodeId: OctetString = NA**

[AF-PNNI-0055], Annex H, page 287: The PNNI Remote Node Id of the neighbouring node. The PNNI Remote Node Id is learned when Hello packets are received from the neighbour.

Length: 6-19 octets.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

#### **9.24.1.3 pnniNeighbourIPAddress: Integer = NA**

The IP address of the neighbouring routing controller protocol controller.

Length: 32 bits.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? AVC

## 9.24.2 Associations

**9.24.2.1** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to PNNIArea

## 9.25 PNNIRoutingProtocol

The PNNI Routing Protocol object provides a management view of the specific attributes of a PNNI Routing Protocol instance.

Object Lifecycle: Inherited

Object Creation Notification Issued? Inherited

Object Deletion Notification Issued? Inherited

Derived from: ControlPlaneService

### 9.25.1 Attributes

#### 9.25.1.1 **scnIPAddress: OctetString = NA**

This attribute represents the control plane address of the protocol instance. Its actual value and length depends on the protocol specific implementation, and on the specific control plane in use.

At create time, its value should reflect the address of the NE platform.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? AVC

#### 9.25.1.2 **pnniAdminState: StandardAdminStateType = LOCKED**

This attribute reflects the semantics of the X.721:1992 administrativeState attribute.

When set to "unlocked" PNNI is enabled.

On create the attribute is set to "locked".

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

#### 9.25.1.3 **pnniNodeId: OctetString = NA**

[AF-PNNI-0055], Annex H, page 266: 6-19 octets

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

### 9.25.2 Associations

**9.25.2.1** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to ControlPlane

**9.25.2.2** (Unnamed Target Role) [1..\*]: Unnamed Association (Unnamed Supplier Role) to PNNIArea

### **9.25.3 Operations**

#### **9.25.3.1 configureRoutingProtocolParameters (paramList: AVPListType = NA): void**

This operation provides a set of values for the named attributes. Only attributes which are writeable and not set by create may be configured.

Operation exceptions:

Standard

Parameter description:

*paramList*

*/\* in \*/* This parameter provides the values for the writable attributes of the selected protocol object.

Pre-conditions:

None

Post-conditions:

The specified attributes are configured or NO values are changed.

#### **9.25.3.2 setAdminState (adminState: AdminStateSetType = NA): void**

Allows the management plane to specify whether the object's Administrative state is enabled or disabled. The operation is idempotent and no error will occur in the event that the operation required no change in state.

Operation exceptions:

Standard

Parameter description:

*adminState*

*/\* in \*/* This parameter selects whether the administrative state is to be enabled (unlocked) or disabled (locked).

Pre-conditions:

None

Post-conditions:

Admin state has been set as requested or an exception has been raised.

### **9.26 PNNISignallingProtocol**

The PNNISignallingProtocol object provides a management view of the specific attributes of a PNNI Signalling Protocol instance.

Object Lifecycle: Inherited

Object Creation Notification Issued? Inherited

Object Deletion Notification Issued? Inherited

Derived from: SignallingProtocol



## 9.26.1 Attributes

### 9.26.1.1 pnniSignallingEnabled: Boolean = false

When set to "true" PNNI signalling is enabled.

Any PNNI message received will be processed. When set to "false" any PNNI message received will be dropped and no PNNI messages will be transmitted.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

## 9.27 RSVPProtocol

The RSVPProtocol object provides a management view of the specific attributes of an RSVP Protocol instance.

Object Lifecycle: Inherited

Object Creation Notification Issued? Inherited

Object Deletion Notification Issued? Inherited

Derived from: SignallingProtocol

### 9.27.1 Attributes

#### 9.27.1.1 rsvpSignallingEnabled: Boolean = false

When set to "true", RSVP signalling is enabled.

Any RSVP message received will be processed. When set to "false", any RSVP message received will be dropped and no RSVP messages will be transmitted.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

#### 9.27.1.2 rsvpGracefulRestart: ActivationStateType = enabled

This flag controls the sending of the Restart\_Cap Object. This object must not be sent when there is no mechanism to detect data channel failures independent of control channel failures. When created, the value reflects the ability of the NE to send the Restart\_Cap Object. The EMS may override this as long as the override does not conflict with [IETF RFC 3473].

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

## 9.28 RoutingArea

The RoutingArea is an object that supports the nodes (contained Routing Areas) and SNPP Links that describe the network topology from the control plane's point of view. It allows ASON topology to be constructed by binding objects to the correct level of the routing hierarchy.

Object Lifecycle: Exists as long as the Routing Area exists.

Object Creation Notification Issued? Yes

Object Deletion Notification Issued? Yes

Derived from: CommonResourceInformation

### **9.28.1 Attributes**

#### **9.28.1.1 routingAreaLevel: RoutingAreaLevelType = TOP\_LEVEL\_RA**

This attribute specifies whether the prescribed Routing Area (RA) is

- a top level RA;
- an intermediate level RA; or
- a Routing Node.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

#### **9.28.1.2 isReroutingDomain: Boolean = True**

This attribute specifies whether the prescribed Routing Area is a rerouting domain.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

#### **9.28.1.3 layerRate: LayerRateType = NA**

This attribute defines the layer rate of the Connection.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

#### **9.28.1.4 parentRoutingArea: ObjectIdType = Null**

This attribute specifies any containing parent Routing Area.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

#### **9.28.1.5 childRoutingArea: ObjectIdListType = Null**

This attribute specifies any contained child Routing Area.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

## 9.28.2 Associations

**9.28.2.1** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to SNPPLink

**9.28.2.2** (Unnamed Target Role) [1]: partitioning (partitions Role) to RoutingArea

This association models the partitioning of this Routing Area into one or more child Routing Areas.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

**9.28.2.3** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to RoutingTable

**9.28.2.4** (Unnamed Target Role) [0..\*]: Unnamed Association (Unnamed Supplier Role) to Connection

**9.28.2.5** (Unnamed Target Role) [0..\*]: SNPP\_is\_named\_by\_RA (Unnamed Supplier Role) to SNPP

**9.28.2.6** (Unnamed Target Role) [0..1]: Unnamed Association (Unnamed Supplier Role) to OSPFArea

**9.28.2.7** (Unnamed Target Role) [0..1]: Unnamed Association (Unnamed Supplier Role) to PNNIArea

**9.28.2.8** (Unnamed Target Role) [0..1]: Unnamed Association (Unnamed Supplier Role) to ISISArea

## 9.28.3 Operations

**9.28.3.1** **getAllSubordinateRAs (routingAreas: SetOf RoutingArea = empty list): void**

This operation returns all the RoutingArea objects that are "contained" within the specified Routing Area. This operation returns all the Routing Areas down to and including those that represent Routing Nodes.

Operation exceptions:

Standard

Parameter description:

*routingAreas*

*/\* out \*/* A list of the RoutingArea objects.

Pre-conditions:

None

Post-conditions:

The system remains unchanged.

**9.28.3.2** **getAllSNPPLinks (snppLinkList: SetOf SNPPLink = empty list): void**

This operation returns all the SNPPLink objects that are considered within the targeted Routing Area. (The links between the child Routing Areas).

Operation exceptions:

Standard

Parameter description:

*snppLinkList*

*/\* out \*/* This parameter contains the list of SNPPLink objects to be returned.

Pre-conditions:

None

Post-conditions:

The system remains unchanged.

### **9.28.3.3 getAllSNPPs (snppList: SetOf SNPP = empty list): void**

This operation returns all the SNPP objects that are on the boundary of this Routing Area.

Operation exceptions:

Standard

Parameter description:

*snppList*

*/\* out \*/* This parameter contains the list of SNPP objects to be returned.

Pre-conditions:

None

Post-conditions:

The system remains unchanged.

### **9.28.3.4 getSNPP (snppName: ObjectIdType = NA, snpp: SNPP = Null): void**

This operation returns the details of the SNPP object requested.

Operation exceptions:

Standard

Parameter description:

*snppName*

*/\* in \*/* This parameter contains the name of the SNPP to be retrieved.

*snpp*

*/\* out \*/* This parameter contains the returned SNPP object.

Pre-conditions:

None

Post-conditions:

The system remains unchanged.

### **9.28.3.5 getRoutingProtocol (routingProtocols: ObjectIdListType = Null): void**

This operation returns the identifiers of the Routing Protocol instances currently supporting this Routing Area.

Note that more than one Routing Protocol instance can be bound to a Routing Area (see operation `setRoutingProtocol`, clause 9.28.3.6).

Operation exceptions:

Standard

Parameter description:

*routingProtocols*

*/\* out \*/* This parameter returns the object identifiers of the supporting routing protocol instances.

Pre-conditions:

None

Post-conditions:

The system remains unchanged.

### **9.28.3.6 setRoutingProtocol (routingProtocol: ObjectIdType = NA): void**

This operation binds a Routing Protocol instance to this Routing Area.

Note that more than one instance can be bound.

Operation exceptions:

Standard

Parameter description:

*routingProtocol*

*/\* in \*/* This parameter specifies the Routing Protocol instance to be used to support this Routing Area.

Pre-conditions:

None

Post-conditions:

The routing protocol instance is bound to the selected routing protocol.

## **9.29 RoutingTable**

The Routing Table describes the internal topology as a graph of nodes (contained Routing Areas) and Links. Operations are provided to enable the Routing Table to be inspected and configured for those cases that automatic routing is not supported.

Object Lifecycle: Exists as long as a Routing Protocol instance exists.

Object Creation Notification Issued? No

Object Deletion Notification Issued? No

### **9.29.1 Associations**

**9.29.1.1** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to RoutingArea

## **9.30 SNP**

The Sub-Network Point object provides the control plane alias for the Transport Plane CTP and so provides the bridge between Control, Transport and Management operations.

Object Lifecycle: Exists as long as the corresponding CTP is assigned to the control plane.

Object Creation Notification Issued? No

Object Deletion Notification Issued? No

Derived from: CommonResourceInformation

### **9.30.1 Attributes**

#### **9.30.1.1 id: String = empty string**

This attribute provides a name for this SNP instance. It is automatically generated on SNP creation.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

#### **9.30.1.2 triName: StringListType = empty list**

Note that the attribute can be set by the EMS once if it is created empty. Multiple TRI Names can be assigned to provide any required aliases.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? Yes

Notification Type? None

### **9.30.2 Associations**

**9.30.2.1** (Unnamed Target Role) [1]: terminated by (Unnamed Supplier Role) to SNPLinkConnection

**9.30.2.2** tPName [1]: SNP\_represents\_same\_resource\_as\_CTP (Unnamed Supplier Role) to ConnectionTerminationPoint

This association models the SNP being an alias for a CTP.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

### **9.31 SNPLinkConnection**

Object Lifecycle: Exists as long as the corresponding CTP-CTP link connection is assigned to the control plane.

Object Creation Notification Issued? No

Object Deletion Notification Issued? No

Derived from: CommonResourceInformation

### **9.31.1 Attributes**

#### **9.31.1.1 sharedRiskGroup: SharedRiskGroupListType = empty list**

NOTE – The semantics of SRG allows much richer expressions than can be currently handled by other specifications. The value is expected to be inherited from the containing SNPP Link.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? None

### **9.31.2 Associations**

**9.31.2.1** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to SNPPLink

**9.31.2.2** (Unnamed Target Role) [2]: terminated by (Unnamed Supplier Role) to SNP

### **9.32 SNPP**

Object Lifecycle: Exists as long as an SNPP Link exists between two Routing Areas.

Object Creation Notification Issued? Yes

Object Deletion Notification Issued? Yes

Derived from: CommonResourceInformation

#### **9.32.1 Attributes**

**9.32.1.1 sharedRiskGroup: SharedRiskGroupListType = empty list**

NOTE – The semantics of SRG allows much richer expressions than can be currently handled by other specifications.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? None

**9.32.1.2 direction: ConnectionDirectionType = BIDIRECTIONAL**

This attribute specifies the directionality of the associated SNPP Link.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

**9.32.1.3 id: OctetString = empty string**

Control plane name for the SNPP.

NOTE – SNPP component, not distinguished name. This name may be assigned by the EMS once.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? Yes

Notification Type? None

#### **9.32.1.4 triName: StringListType = empty list**

A TRI is a globally unique identifier assigned by the transport network to one or more data links connecting a UNI-N and a UNI-C. This attribute can be assigned by the EMS once. Multiple TRI Names can be assigned to provide any required aliases.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? Yes

Notification Type? None

#### **9.32.2 Associations**

##### **9.32.2.1 snpList [0..\*]: SNPP\_groups\_SNPs (Unnamed Supplier Role) to SNP**

This association models the SNP being a member of the SNPP Link.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

##### **9.32.2.2 (Unnamed Target Role) [1]: SNPP\_is\_named\_by\_RA (Unnamed Supplier Role) to RoutingArea**

#### **9.32.3 Operations**

##### **9.32.3.1 setTRIName (triName: String = NA): void**

This operation sets the triName attribute of the SNPP Link. Setting the current name does not result in an error.

Operation exceptions:

Standard

Parameter description:

*triName*

*/\* in \*/* This parameter provides the desired TRI Name.

Pre-conditions:

None

Post-conditions:

If the TRI Name is unique, then the value of the attribute is set.

#### **9.33 SNPPLink**

The SNPP Link supports the relation between two adjacent Routing Areas and contains the set of SNP Link Connections allocated to that link.

Object Lifecycle: Exists as long as an SNPP Link exists between two Routing Areas.

Object Creation Notification Issued? Yes

Object Deletion Notification Issued? Yes

Derived from: CommonResourceInformation



### 9.33.1 Attributes

#### 9.33.1.1 **sharedRiskGroup: SharedRiskGroupListType = empty list**

NOTE – The semantics of SRG allows much richer expressions than can be currently handled by other specifications.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

#### 9.33.1.2 **direction: ConnectionDirectionType = BIDIRECTIONAL**

This attribute can be written by the EMS once.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? Yes

Notification Type? None

#### 9.33.1.3 **cost: AVPListType = empty list**

This attribute specifies the cost of using this SNPP Link in a Connection.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

#### 9.33.1.4 **discovered: Boolean = NA**

This attribute indicates whether the SNPP Link was discovered or provisioned. It is set by create.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

#### 9.33.1.5 **availability: AVPListType = empty list**

This attribute specifies the availability to be expected for Connections provided on this SNPP Link.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

#### 9.33.1.6 **interfaceType: InterfaceType = NA**

This attribute represents the type of interface in which the SNPP Link participates. Note that the value "External E-NNI" means that the link is at the edge of the NMS domain.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? None

### 9.33.2 Associations

**9.33.2.1** (Unnamed Target Role) [0..\*]: Unnamed Association (Unnamed Supplier Role) to SNPLinkConnection

**9.33.2.2** aEnd [1]: SNPPLink\_is\_terminated\_by\_aEndSNPP (Unnamed Supplier Role) to SNPP  
This association models the A end of an SNPP Link.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

**9.33.2.3** (Unnamed Target Role) [2]: Unnamed Association (Unnamed Supplier Role) to RoutingArea

**9.33.2.4** zEnd [1]: SNPPLink\_is\_terminated\_by\_zEndSNPP (Unnamed Supplier Role) to SNPP  
This association models the Z end of an SNPP Link.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

Notification Type? None

**9.33.2.5** signallingProtocolType [0..1]: Unnamed Association (Unnamed Supplier Role) to SignallingProtocol

Examples of the types of protocol would be RSVP, PNNI, CR-LDP.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? AVC

**9.33.2.6** (Unnamed Target Role) []: Unnamed Association (Unnamed Supplier Role) to CallService

### 9.33.3 Operations

**9.33.3.1** **getAvailableCapacity (availableCapacity: Integer = 0): void**

This operation returns the number of free Connections in the SNPP Link that can be allocated.

Operation exceptions:

Standard

Parameter description:

*availableCapacity*

/\* out \*/ The number of free Connections that can be allocated.

Pre-conditions:

None

Post-conditions:

The system remains unchanged.

### **9.33.3.2 getLinkConnectionState (linkConnection: ObjectIdType = NA, linkConnectionState: LinkConnectionStateType = NA): void**

Provides information about the state of a particular SNP Link Connection within the SNPP Link.

Operation exceptions:

Standard

Parameter description:

*linkConnection*

*/\* in \*/* This parameter specifies the object identifier of the object for which the state is requested.

*linkConnectionState*

*/\* out \*/* This parameter returns the Link Connection state.

Pre-conditions:

The requested link connection must be allocated to this link.

Post-conditions:

The system remains unchanged.

### **9.33.3.3 getSignallingProtocol (signallingProtocol: ObjectIdType = Null): void**

This operation returns the signalling protocol instance that is supporting this SNPP Link. If there is no such instance, Null is returned.

Operation exceptions:

Standard

Parameter description:

*signallingProtocol*

*/\* out \*/* This parameter returns the required object identifier or Null.

Pre-conditions:

None

Post-conditions:

The system remains unchanged.

### **9.33.3.4 setSignallingProtocol (signallingProtocol: ObjectIdType = NA): void**

This operation sets the signalling protocol instance that will support this SNPP Link.

Operation exceptions:

Standard

Parameter description:

*signallingProtocol*

*/\* in \*/* This parameter specifies the signalling protocol instance that will support this SNPP Link.

Pre-conditions:

None

Post-conditions:

The signalling protocol instance is set, or an exception is raised.

### **9.34 SignallingProtocol**

The SignallingProtocol object provides common attributes and operations for all specific signalling protocols.

Object Lifecycle: Lifecycle is from instantiation of the protocol to termination of the protocol.

Object Creation Notification Issued? Yes

Object Deletion Notification Issued? Yes

Derived from: ControlPlaneService

#### **9.34.1 Attributes**

##### **9.34.1.1 adminState: StandardAdminStateType = LOCKED**

This attribute reflects the semantics of the X.721:1992 administrativeState attribute. When set to "unlocked", the protocol instance is administratively enabled.

On create, the attribute is set to "locked".

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? None

##### **9.34.1.2 scnIPAddress: OctetString = NA**

This attribute represents the control plane address of the protocol instance. Its actual value and length depends on the protocol specific implementation, and on the specific control plane in use.

At create time, its value should reflect the address of the NE platform.

Readable by EMS? Yes

Writeable by EMS? Yes

Invariant? No

Notification Type? AVC

#### **9.34.2 Associations**

**9.34.2.1** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to ControlPlane

#### **9.34.3 Operations**

##### **9.34.3.1 configureSignallingProtocolParameters (paramList: SetOf Logical View::Attribute Types::AVP = NA): void**

This operation provides a set of values for the named attributes. Only attributes which are writeable and not set by create may be configured.

Operation exceptions:

Standard

Parameter description:

*paramList*

*/\* in \*/* This parameter provides the values for the writable attributes of the SignalingProtocol object.

Pre-conditions:

None

Post-conditions:

The specified attributes are configured or NO values are changed.

### **9.34.3.2 setAdminState (adminState: AdminStateSetType = NA): void**

Allows the management plane to specify whether the object's Administrative state is enabled or disabled. The operation is idempotent and no error will occur in the event that the operation required no change in state.

Operation exceptions:

Standard

Parameter description:

*adminState*

*/\* in \*/* This parameter selects whether the administrative state is to be enabled (unlocked) or disabled (locked).

Pre-conditions:

None

Post-conditions:

Admin state has been set as requested or an exception has been raised.

## **9.35 TerminationPoint**

This object class is outside the scope of this Recommendation. It has been added here to show the relationship between the Connection and its end points.

## **9.36 TrailTerminationPoint**

This object class is outside the scope of this Recommendation. It has been added here to show the relationship to Layer Adjacency Discovery and Transport Entity Capability Exchange.

### **9.36.1 Associations**

**9.36.1.1** (Unnamed Target Role) [0..1]: Unnamed Association (Unnamed Supplier Role) to LayerAdjacencyDiscovery

**9.36.1.2** (Unnamed Target Role) [0..1]: Unnamed Association (Unnamed Supplier Role) to TransportEntityCapabilityExchange

**9.36.1.3** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to ManagedElement

## **9.37 TransportEntityCapabilityExchange**

This object represents the capability of the TTP to support Transport Entity Capability Exchange Services.

Object Lifecycle: Exists as long as the TTP can support TCE.

Object Creation Notification Issued? Yes  
Object Deletion Notification Issued? Yes  
Derived from: CommonResourceInformation

### **9.37.1 Attributes**

#### **9.37.1.1 tceStatus: TCEStatusType = NA**

This attribute indicates the current status of the Transport Entity Capability Exchange process. It is maintained by the NE.

Allowed values are: see Figure I.2, TCE state machine of [ITU-T G.7714].

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? None

#### **9.37.1.2 remoteCapability: OctetString = empty string**

After successful completion of the Transport Entity Capability Exchange process, this attribute contains the transmission capabilities of the remote TTP and its client layers.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? None

#### **9.37.1.3 negotiatedLocalCapability: OctetString = empty string**

After successful completion of the Transport Entity Capability Exchange process, this attribute contains the negotiated transmission capabilities between the local and remote TTP and their client layers.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? No

Notification Type? None

### **9.37.2 Associations**

**9.37.2.1** (Unnamed Target Role) [1]: Unnamed Association (Unnamed Supplier Role) to TrailTerminationPoint

### **9.37.3 Operations**

#### **9.37.3.1 startTransportEntityCapabilityExchange (): void**

This operation starts the Transport Entity Capability Exchange process on the containing TTP. This operation is idempotent.

Operation exceptions:

The Discovery Service is disabled for the containing TTP.

Pre-conditions:

None

Post-conditions:

The Transport Entity Capability Exchange process is started.

### **9.37.3.2 stopTransportEntityCapabilityExchange (): void**

This operation stops the Transport Entity Capability Exchange process on the containing TTP. This operation is idempotent.

Operation exceptions:

Standard

Pre-conditions:

None

Post-conditions:

The Transport Entity Capability Exchange process is stopped.

## **10 Notification definitions**

### **10.1 Alarm**

The alarm notification type is used to report abnormal conditions/fault conditions of a managed object. Alarm notifications can be subdivided into more specific alarm types such as communicationsAlarm, environmentalAlarm, EquipmentAlarm, processingErrorAlarm, qualityOfServiceAlarm, etc. (refer to [ITU-T X.733]).

Derived from: Notification

#### **10.1.1 Attributes**

##### **10.1.1.1 perceivedSeverity: PerceivedSeverityType = NA**

Specifies the criticality of the occurred alarm; can have the following values: cleared, warning, minor, major, critical, indetermined.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

##### **10.1.1.2 probableCause: String = NA**

Defines further qualification as to the probable cause of the alarm.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

##### **10.1.1.3 stateChangeDefinition: AttributeValueChangeListType = NA**

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

##### **10.1.1.4 backedUpStatus: Boolean = NA**

"True" if backed up.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

#### **10.1.1.5 backedUpObject: ObjectIdType = NA**

Will be null if backedUpStatus is "false".

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

#### **10.1.1.6 alarmEffectOnService: Boolean = NA**

True if alarm is service effecting.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

#### **10.1.1.7 alarmingResumed: Boolean = NA**

True if alarming was just resumed, possibly resulting in delayed reporting of an alarm.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

#### **10.1.1.8 suspectObjectList: ObjectIdListType = NA**

Objects possibly involved in failure.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

### **10.2 AttributeValueChangeNotification**

This notification type is used to report changes to the value of one or more attributes of a managed object (normally as a result of a management operation).

Derived from: Notification

#### **10.2.1 Attributes**

##### **10.2.1.1 attributeChanges: AttributeValueChangeListType = NA**

This attribute contains a list of attribute names and their old and new values.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

### **10.3 Notification**

Notifications are used to report extraordinary events related to a managed object to a management system (such as link is down, new connection was created, connection was deleted, attribute values of a connection were modified, etc.).



### **10.3.1 Attributes**

#### **10.3.1.1 source: ObjectIdType = NA**

Identifies the object class and object instance (via the object identifier) that caused the notification to be generated.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

#### **10.3.1.2 notificationIdentifier: Integer = NA**

Contains a unique identifier for the notification. (Can be present in the correlatedNotifications attribute of other notifications.)

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

#### **10.3.1.3 additionalText: String = empty string**

Contains textual information/description for that notification.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

#### **10.3.1.4 sourceIndicator: SourceIndicatorType = NA**

Source Indicator identifies whether the notification is a result of a management operation or of something that occurred on the managed system.

Possible values: resourceOperation, managementOperation, unknown.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

#### **10.3.1.5 correlatedNotifications: IntegerListType = empty list**

Contains a list of notifications that are considered to be correlated to this notification (notificationIdentifiers + source).

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

#### **10.3.1.6 neTime: GeneralizedTimeType = NA**

Identifies the time of generation of the notification in the Network Element. In case where the NE does not report time, zero time shall be reported.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

## **10.4 ObjectCreationNotification**

This notification type is used to report the creation of a managed object to a management system.

Derived from: Notification

## **10.5 ObjectDeletionNotification**

This notification type is used to report the deletion of a managed object to a management system.

Derived from: Notification

## **10.6 StateChangeNotification**

This notification type is used to report the change in the value of one or more state attributes of a managed object, that result through either internal operation of the managed object or via management operation.

Derived from: Notification

### **10.6.1 Attributes**

#### **10.6.1.1 stateChanges: AttributeValueChangeListType = NA**

This attribute contains a list of state attribute names and their old and new values.

Readable by EMS? Yes

Writeable by EMS? No

Invariant? Yes

## **11 Data Types**

NOTE – When defining an enumeration data type (by defining a class of stereotype <<Enumeration>>), a type for the attributes need not be specified.

Enumeration elements could be mapped to appropriate integer values (e.g., ENABLED = 0, DISABLED = 1, NOT\_SUPPORTED = 2) in an implementation, but this is out of scope of this Recommendation.

### **11.1 Enumerations**

#### **11.1.1 ActivationStateType**

Enumeration elements could be mapped to appropriate integer values (e.g., ENABLED = 0, DISABLED = 1, NOT\_SUPPORTED = 2) in an implementation, but this is out of scope of this Recommendation.

##### **11.1.1.1 Defined values**

- ENABLED
- DISABLED
- NOT\_SUPPORTED

#### **11.1.2 AdminStateSetType**

This data type is used to set the Administrative State of an object.

Enumeration elements could be mapped to appropriate integer values (e.g., LOCKED = 0, UNLOCKED = 1) in an implementation, but this is out of scope of this Recommendation.

### 11.1.2.1 Defined values

- LOCKED;
- UNLOCKED.

### 11.1.3 AlarmStatusType

Enumeration elements could be mapped to appropriate integer values (e.g., NO\_DEFECT = 0, WARNING = 1, MINOR = 2,...) in an implementation, but this is out of scope of this Recommendation.

#### 11.1.3.1 Defined values

- NO\_DEFECT;
- WARNING;
- MINOR;
- MAJOR;
- CRITICAL;
- UNDETERMINED;
- NOT\_SUPPORTED.

### 11.1.4 CallParameterType

Enumeration elements could be mapped to appropriate integer values (e.g., DEGRADED\_THRESHOLD = 0, SEVERELY\_DEGRADED\_THRESHOLD = 1, NOT\_SUPPORTED = 2) in an implementation, but this is out of scope of this Recommendation.

#### 11.1.4.1 Defined values

- DEGRADED\_THRESHOLD
- SEVERELY\_DEGRADED\_THRESHOLD
- NOT\_SUPPORTED

### 11.1.5 CallStateType

Enumeration elements could be mapped to appropriate integer values (e.g., IN\_PROGRESS = 0, ESTABLISHED\_IN\_SERVICE = 1, notSupported = 2, ESTABLISHED\_IN\_SERVICE\_SEARCHING = 3,...) in an implementation, but this is out of scope of this Recommendation.

NOTE – Values IN\_PROGRESS, ESTABLISHED\_IN\_SERVICE\_SEARCHING, ESTABLISHED\_OUT\_OF\_SERVICE\_SEARCHING, ESTABLISHED\_IN\_SERVICE\_DEGRADED\_SEARCHING, and ESTABLISHED\_IN\_SERVICE\_SEVERELY\_DEGRADED\_SEARCHING may not be applicable in a non control plane environment.

#### 11.1.5.1 Defined values

- IN\_PROGRESS  
The supporting Connections are currently being created.
- ESTABLISHED\_IN\_SERVICE  
All the supporting Connections have been created successfully.
- ESTABLISHED\_IN\_SERVICE\_SEARCHING  
A Call has been modified through the addition of Connections and not all new Connections have been successfully created.  
May not be valid for Non CPs.

- ESTABLISHED\_OUT\_OF\_SERVICE  
All Connections have failed and they are not being restored.
- ESTABLISHED\_OUT\_OF\_SERVICE\_SEARCHING  
All Connections have failed and they are currently being restored.
- ESTABLISHED\_IN\_SERVICE\_DEGRADED  
The number of failed connections has reached or exceeded the degraded threshold AND the severely degraded threshold has not been reached or exceeded.
- ESTABLISHED\_IN\_SERVICE\_SEVERELY\_DEGRADED  
The number of failed supporting Connections has reached or exceeded the severely degraded threshold (in case of LCAS support).
- ESTABLISHED\_IN\_SERVICE\_DEGRADED\_SEARCHING  
At least one supporting Connection has failed (in the case of LCAS support) AND the severely degraded threshold has not been reached or exceeded.
- ESTABLISHED\_IN\_SERVICE\_SEVERELY\_DEGRADED\_SEARCHING  
The number of failed supporting Connections has reached or exceeded the severely degraded threshold (in case of LCAS support).
- NOT\_SUPPORTED

#### **11.1.6 ConnectionDirectionType**

Enumeration elements could be mapped to appropriate integer values (e.g., BIDIRECTIONAL = 0, UNIDIRECTIONAL = 1) in an implementation, but this is out of scope of this Recommendation.

##### **11.1.6.1 Defined values**

- BIDIRECTIONAL
- UNIDIRECTIONAL
- NOT\_SUPPORTED

#### **11.1.7 ConnectionKindType**

Enumeration elements could be mapped to appropriate integer values (e.g., SPC = 0, PC = 1, SC = 2) in an implementation, but this is out of scope of this Recommendation.

##### **11.1.7.1 Defined values**

- SPC  
Indicates a soft permanent connection type; an SNC that has been created either:
  - as side effect of establishCall(); or
  - by the network, at E-NNI, as side effect of establishCall() performed at another NMS-EMS interface.
- PC  
Indicates a permanent connection type; an SNC that has been created without using the control plane functionality.
- SC  
Indicates a switched connection type; an SNC that has been created by the network, at UNI.
- NOT\_SUPPORTED

### 11.1.8 ConnectionStateType

Enumeration elements could be mapped to appropriate integer values (e.g., SEARCHING = 0, COMPLETE = 1) in an implementation, but this is out of scope of this Recommendation.

#### 11.1.8.1 Defined values

- SEARCHING  
This state indicates that NOT all the resources necessary to support the Connection have been successfully allocated yet.  
This state may not be applicable in a non-control plane environment.
- COMPLETE  
This state indicates that all resources necessary to support the Connection have been successfully allocated.
- NOT\_SUPPORTED

### 11.1.9 InterfaceType

This Enumeration defines the type of the interface.

Enumeration elements could be mapped to appropriate integer values (e.g., UNI = 0, I-NNI = 1, INTERNAL\_E-NNI = 2,...) in an implementation, but this is out of scope of this Recommendation.

#### 11.1.9.1 Defined values

- UNI
- I-NNI
- INTERNAL\_E-NNI
- EXTERNAL\_E-NNI
- UNSPECIFIED
- NOT\_SUPPORTED

### 11.1.10 LadStatusType

Indicates the current status of the Discovery process.

Enumeration elements could be mapped to appropriate integer values (e.g., IDLE = 0, A-Z\_KNOWN = 1, Z-A\_KNOWN = 2,...) in an implementation, but this is out of scope of this Recommendation.

#### 11.1.10.1 Defined values

- IDLE
- A-Z\_KNOWN
- Z-A\_KNOWN
- A-Z\_Z-A\_KNOWN

### 11.1.11 LevelOfEffortType

Enumeration elements could be mapped to appropriate integer values (e.g., NONE = 0, BEST\_EFFORT = 1, MANDATORY = 2,...) in an implementation, but this is out of scope of this Recommendation.

#### 11.1.11.1 Defined values

- NONE
- BEST\_EFFORT

- MANDATORY
- NOT\_SUPPORTED

#### **11.1.12 LinkConnectionStateType**

Enumeration elements could be mapped to appropriate integer values (e.g., FREE = 0, BUSY = 1, NOT\_SUPPORTED = 2,...) in an implementation, but this is out of scope of this Recommendation.

##### **11.1.12.1 Defined values**

- FREE
- BUSY
- NOT\_SUPPORTED

#### **11.1.13 OSPFNeighbourStateType**

Enumeration elements could be mapped to appropriate integer values (e.g., DOWN = 0, ATTEMPT = 1, INIT = 2,...) in an implementation, but this is out of scope of this Recommendation.

##### **11.1.13.1 Defined values**

- DOWN
- ATTEMPT
- INIT
- EX\_START
- 2\_WAY
- NOT\_SUPPORTED

#### **11.1.14 PerceivedSeverityType**

Indicates the criticality of the occurred alarm.

Enumeration elements could be mapped to appropriate integer values (e.g., INDETERMINED = 0, CRITICAL = 1, MAJOR = 2,...) in an implementation, but this is out of scope of this Recommendation.

##### **11.1.14.1 Defined values**

- INDETERMINED
- CRITICAL
- MAJOR
- MINOR
- CLEARED
- NOT\_SUPPORTED

#### **11.1.15 PlaneType**

Enumeration elements could be mapped to appropriate integer values (e.g., CONTROL\_PLANE = 0, MANAGEMENT\_PLANE = 1,...) in an implementation, but this is out of scope of this Recommendation.

##### **11.1.15.1 Defined values**

- CONTROL\_PLANE
- MANAGEMENT\_PLANE
- NOT\_SUPPORTED

### **11.1.16 RecoveryType**

Enumeration elements could be mapped to appropriate integer values (e.g., NONE = 0, RESTOREABLE = 1, PROTECTED = 2) in an implementation, but this is out of scope of this Recommendation.

#### **11.1.16.1 Defined values**

- NONE
- RESTOREABLE
- PROTECTED
- NOT\_SUPPORTED

### **11.1.17 RoutingAreaLevelType**

Enumeration elements could be mapped to appropriate integer values (e.g., TOP\_LEVEL\_RA = 0, INTERMEDIATE\_LEVEL\_RA = 1, ROUTING\_NODE\_LEVEL = 3,...) in an implementation, but this is out of scope of this Recommendation.

#### **11.1.17.1 Defined values**

- TOP\_LEVEL\_RA
- INTERMEDIATE\_LEVEL\_RA
- ROUTING\_NODE\_LEVEL
- NOT\_SUPPORTED

### **11.1.18 SNCType**

Enumeration elements could be mapped to appropriate integer values (e.g., SIMPLE = 0, ADD\_DROP\_A = 1, ADD\_DROP\_Z = 3,...) in an implementation, but this is out of scope of this Recommendation.

#### **11.1.18.1 Defined values**

- SIMPLE
- ADD\_DROP\_A
- ADD\_DROP\_Z
- INTERCONNECT
- DOUBLE\_INTERCONNECT
- DOUBLE\_ADD\_DROP
- OPEN\_ADD\_DROP
- EXPLICIT
- NOT\_SUPPORTED

### **11.1.19 SourceIndicatorType**

Indicates whether the notification is a result of a management operation or something that occurred on the managed system.

Enumeration elements could be mapped to appropriate integer values (e.g., RESOURCE\_OPERATION = 0, MANAGEMENT\_OPERATION = 1, UNKNOWN = 2,...) in an implementation, but this is out of scope of this Recommendation.

#### **11.1.19.1 Defined values**

- RESOURCE\_OPERATION
- MANAGEMENT\_OPERATION

- UNKNOWN
- NOT\_SUPPORTED

#### **11.1.20 StandardAdminStateType**

Enumeration elements could be mapped to appropriate integer values (e.g., LOCKED = 0, UNLOCKED = 1, NOT\_SUPPORTED = 2,...) in an implementation, but this is out of scope of this Recommendation.

##### **11.1.20.1 Defined values**

- LOCKED
- UNLOCKED
- NOT\_SUPPORTED

#### **11.1.21 StandardNeighbourStateType**

Enumeration elements could be mapped to appropriate integer values (e.g., UP = 0, DOWN = 1, NOT\_SUPPORTED = 2) in an implementation, but this is out of scope of this Recommendation.

##### **11.1.21.1 Defined values**

- UP
- DOWN
- NOT\_SUPPORTED

#### **11.1.22 StandardOperStateType**

Enumeration elements could be mapped to appropriate integer values (e.g., ENABLED = 0, DISABLED = 1, TESTING = 2,...) in an implementation, but this is out of scope of this Recommendation.

##### **11.1.22.1 Defined values**

- ENABLED
- DISABLED
- DEGRADED
- TESTING
- NOT\_SUPPORTED

#### **11.1.23 TCESStatusType**

Enumeration elements could be mapped to appropriate integer values (e.g., STOPPED = 0, CLOSING = 1, STOPPING = 2,...) in an implementation, but this is out of scope of this Recommendation.

##### **11.1.23.1 Defined values**

- STOPPED
- CLOSING
- STOPPING
- CAP\_REQ\_SENT
- A-Z\_Z-A\_UNKNOWN
- A-Z\_OK\_Z-A\_UNKNOWN
- Z-A\_OK\_A-Z\_UNKNOWN
- A-Z\_OK\_Z-A\_OK



- Z-A\_OK\_A-Z\_RENEG

## **11.2 Structures**

### **11.2.1 AVPType**

Contains an attribute name together with its value.

#### **11.2.1.1 Attributes**

- attributeName: String = NA  
Contains the name of the attribute.
- attributeValue: AnyType = NA  
Contains the value of the attribute.

### **11.2.2 AttributeValueChangeType**

An Attribute Value Change structure is used in a notification to report an attribute that has been changed.

#### **11.2.2.1 Attributes**

- attributeName: String = NA
- oldValue: AnyType = NA
- newValue: AnyType = NA

### **11.2.3 CallAndConnectionsType**

This structure defines a Call and its associated Connections.

#### **11.2.3.1 Attributes**

- call: Call = NA  
This attribute contains the Call object.
- connections: SequenceOf Connection = empty list  
This attribute contains a list of Connection objects.

### **11.2.4 CallEndType**

This structure provides the names of one Call end.

#### **11.2.4.1 Attributes**

- snppId: ObjectIdType = Null
- snpId: ObjectIdType = Null
- triNameOrGroupTRIName: ObjectIdType = Null

### **11.2.5 CurrentProblemType**

#### **11.2.5.1 Attributes**

- managedObject: ObjectIdType = NA
- alarmStatus: AlarmStatusType = NA
- probableCause: String

## 11.2.6 DiversityDataType

This structure defines the diversity related parameters:

- coroutingLevelOfEffort
- nodeDiversityLevelOfEffort
- linkDiversityLevelOfEffort
- nodeSRGType
- linkSRGType.

NOTE – The diversity and co-routing parameter set includes in addition a Route Group Label attached to each Connection/SNC supporting a Call.

### 11.2.6.1 Attributes

- coroutingLevelOfEffort: LevelOfEffortType = NA
- nodeDiversityLevelOfEffort: LevelOfEffortType = NA
- linkDiversityLevelOfEffort: LevelOfEffortType = NA
- nodeSRGType: CIS CharacterString = empty string
- linkSRGType: CIS CharacterString = empty string

## 11.2.7 ObjectIdType

### 11.2.7.1 Attributes

- OBJECT\_CLASS: String
- INSTANCE\_ID: String

## 11.2.8 SNPPLinkOrSNPPType

This structure defines an SNPPLink object or an SNPP object.

### 11.2.8.1 Attributes

- snppLink: SNPPLink = Null
- snpp: SNPP = Null

## 11.2.9 SharedRiskGroupType

### 11.2.9.1 Attributes

- name: String = NA
- values: StringListType = NA

## 11.2.10 TimeOfDayType

### 11.2.10.1 Attributes

- time: String
- date: String

## 11.3 TypeDefinitions

### 11.3.1 AVPListType

Sequence of AVPType.

#### 11.3.1.1 Attributes

- avpList: SequenceOf AVPType = empty list

### **11.3.2 AttributeValueChangeListType**

Sequence of AttributeValueType.

#### **11.3.2.1 Attributes**

- attributeValueChangeList: SequenceOf AttributeValueType = empty list

### **11.3.3 CallAndConnectionsListType**

Sequence of CallAndConnectionType.

#### **11.3.3.1 Attributes**

- callAndConnectionsList: SequenceOf CallAndConnectionsType = empty list

### **11.3.4 CurrentProblemListType**

Sequence of CurrentProblemType.

#### **11.3.4.1 Attributes**

- currentProblemList: SequenceOf CurrentProblemType = empty list

### **11.3.5 GeneralizedTimeType**

typedef GeneralizedTimeType

### **11.3.6 IntegerListType**

Sequence of Integer.

#### **11.3.6.1 Attributes**

- integerList: SequenceOf Integer = empty list

### **11.3.7 LayerRateType**

There is a standardized list of LayerRates in the [TMF MTNM] SD1-17\_LayerRates supporting document. LayerRates from this standard list should be used wherever possible. The standardized list will be augmented in subsequent releases through the normal approval process. The LayerRate definition includes a range of layer identifiers for proprietary usage.

The conditions for use of this capability, to augment the list of supported layers between releases, are described in the supporting document, "Layer rates". This text is available in the electronic attachment in pdf format as SD1-17\_V3.0\_LayerRates.pdf.

#### **11.3.7.1 Attributes**

- layerRateValue: Integer = NA

### **11.3.8 ObjectIdListType**

Sequence of ObjectIdType.

#### **11.3.8.1 Attributes**

- objectIdList: SequenceOf ObjectIdType = empty list

### **11.3.9 SNPPLinkOrSNPPListType**

Sequence of SNPPLinkOrSNPPType.

#### **11.3.9.1 Attributes**

- snppLinkOrSNPPList: SequenceOf SNPPLinkOrSNPPType = empty list

### **11.3.10 SharedRiskGroupListType**

Sequence of SharedRiskGroupType.

#### **11.3.10.1 Attributes**

- sharedRiskGroupList: SequenceOf SharedRiskGroupType = empty list

### **11.3.11 StringListType**

Sequence of String.

#### **11.3.11.1 Attributes**

- stringList: SequenceOf String = empty list

## Appendix I

### G.7718.1 and TMF 608v3.0

(This appendix does not form an integral part of this Recommendation.)

This appendix describes the relationship between work in the TMF on the NMS-EMS management interface (i.e., the MTNM product) and G.7718.1 network element interface. This discussion describes why these models are strongly related, and describes the measures taken to keep them no tighter coupled than strictly necessary.

The TMF MTNM Team are working on a network level interface between NMS and EMS, and the approach that has been taken was to add managed objects for the control plane into the existing network model. The assumption is that the network elements have already been configured, and the NMS can view the results of the configuration and operate on those results.

G.7718.1 complements this work by providing an EMS "South" bound interface to an NE. This interface provides additional operations needed to perform the network element configuration in the first place, and all the data necessary for the EMS to construct the managed objects specified at the "North" bound interface.

Because the scope of the EMS is normally larger than the scope of a network element, there are some differences between the managed objects at the North bound interface and at the South bound interface. The most notable difference is that the North bound interface uses special multi-layer and multi-level constructs to reduce the amount of data required on the interface. Recognizing the smaller scope of a network element, and protocol neutral specification of the South bound interface, G.7718.1 provides single layer and single level objects, and leaves any optimization to the protocol specific design stage.

A second important difference is that the North interface reuses the subnetwork connection as the construct to group multiple connections, while G.7718.1 uses the G.8080 Call construct for this purpose. This results in several operations against the Connection in the North interface being against the Call in the south interface.

These differences also make the two interface models less tightly coupled, and this reduction in coupling will make subsequent development easier.

## **Appendix II**

### **Alternate representations**

(This appendix does not form an integral part of this Recommendation.)

The UML description (clauses 8, 9, 10 and 11) of this Recommendation is available in navigable HTML format in the electronic attachment. Note that the browser must have Java enabled in order to be able to use this representation.

The second attachment contains the UML description (clauses 8, 9, 10 and 11) of this Recommendation in the original Rational Rose format.

ITU-T Y-SERIES RECOMMENDATIONS

**GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS**

GLOBAL INFORMATION INFRASTRUCTURE

General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899

INTERNET PROTOCOL ASPECTS

General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
<b>Operation, administration and maintenance</b>	<b>Y.1700–Y.1799</b>
Charging	Y.1800–Y.1899

NEXT GENERATION NETWORKS

Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Numbering, naming and addressing	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899

*For further details, please refer to the list of ITU-T Recommendations.*







## SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
<b>Series G</b>	<b>Transmission systems and media, digital systems and networks</b>
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
<b>Series Y</b>	<b>Global information infrastructure, Internet protocol aspects and next-generation networks</b>
Series Z	Languages and general software aspects for telecommunication systems