SIF-023-1998

(SIF Document #SIF-IM-9810-146R2)

SIF APPROVED DOCUMENT

WORK GROUP: INFORMATION MODEL

TITLE: Network View Model for Connection Management and Fault Management

DATE: October, 1998

EDITOR: **Name:** Andy Walsh Wendy Teller
Voice: +1-732-758-5648 +1-630-527-6206
email: awalsh@notes.cc.bellcore.com teller@wwa.com

ABSTRACT: This contribution is the draft of the update to the current model (SIF-014-1997). It includes changes for link end objects and for support of protection of connections.

This document has received the approval of the SONET Interoperability Forum (SIF).

December 10, 1998

Table of Contents

1. INTRODUCTION 1

1.1 History of the Development Process and Methodology 1

1.2 Scope 1

1.3 Network Management Architecture 2

1.4 NE+Network-Level Management Architecture Example 3

1.5 Changes from SIF-014-1997 4

1.6 References 4

1.7 Definitions 7
1.7.1 1+1 protection 7
1.7.2 access group 7
1.7.3 add traffic 7
1.7.4 administrative domain 7
1.7.5 bicast 7
1.7.6 bridge 7
1.7.7 Client/Server (C/S) Pointers 8
1.7.8 characteristic information 8
1.7.9 connection management 8
1.7.10 Connection Termination Point (CTP) 8
1.7.11 connection type 8
1.7.12 Downstream Connectivity Pointer (DCP) 8
1.7.13 drop 9
1.7.14 drop and continue 9
1.7.15 drop traffic 9
1.7.16 dual hubbed 9
1.7.17 dual node interconnect 9
1.7.18 dynamic subnetworks 9
1.7.19 Element Management Layer (EML) 9
1.7.20 Element Management System (EMS) 9
1.7.21 exclusive composite 9
1.7.22 exclusive merge 10
1.7.23 extra traffic 10
1.7.24 hardwired multiplex 10
1.7.25 hold off time 10
1.7.26 highly protected 10
1.7.27 layer network 10
1.7.28 layer network domain 10
1.7.29 link 10
1.7.30 link connection 11
1.7.31 link end 11
1.7.32 logical ring 11
1.7.33 M:N protection switching 11
1.7.34 Management Applications Function (MAF) 11
1.7.35 multi-domain ring 11

1.7.36 multiple partitioning views 11
1.7.37 network connection 11
1.7.38 network connection termination point 12
1.7.39 network connection termination point bidirectional 12
1.7.40 network connection termination point sink 12
1.7.41 network connection termination point source 12
1.7.42 Network Element (NE) 12
1.7.43 Network Element Function (NEF) 12
1.7.44 Network Element Layer (NEL) 12
1.7.45 network element view (NE view) 12
1.7.46 Network Management Layer (NML) 13
1.7.47 Network Management System (NMS) 13
1.7.48 NMS Environment - *under study* 13
1.7.49 network trail termination point 13
1.7.50 network trail termination point bidirectional 13
1.7.51 network trail termination point sink 13
1.7.52 network trail termination point source 13
1.7.53 network view 13
1.7.54 NE-view connection termination point 13
1.7.55 NE-view trail termination point 14
1.7.56 nodal view 14
1.7.57 node 14
1.7.58 non-preemptible unprotected channel 14
1.7.59 non-preemptible unprotected traffic 14
1.7.60 partitioning 14
1.7.61 point-to-point route 14
1.7.62 preemptible 14
1.7.63 protected 14
1.7.64 protection channels 15
1.7.65 protection level 15
1.7.66 protection 15
1.7.67 requested effort level 15
1.7.68 restoration 16
1.7.69 ring interconnection 16
1.7.70 ring interworking 16
1.7.71 ring switch 16
1.7.72 route 16
1.7.73 route capacity 16
1.7.74 route separation 16
1.7.75 span switch 16
1.7.76 selector 17
1.7.77 service 17
1.7.78 state 17
1.7.79 subnetwork 17
1.7.80 subnetwork connection (SNC) 17
1.7.81 subnetwork connection protection 17
1.7.82 Subnetwork Management System (subNMS) - *under study* 17
1.7.83 subnetwork capacity 17
1.7.84 subtending rings 17
1.7.85 Termination Point (TP) 18
1.7.86 Time-Slot Assignment (TSA) 18
1.7.87 Time-Slot Interchange (TSI) 18
1.7.88 topological component 18

1.7.89 trail 18
1.7.90 trail termination function 18
1.7.91 Trail Termination Point (TTP) 18
1.7.92 trail protection 18
1.7.93 transport entity 19
1.7.94 unprotected 19
1.7.95 Upstream Connectivity Pointer (UCP) 19

1.8 Acronyms 19

2. STATEMENTS OF APPLICATION 22

2.1 Connection Management SOA 22
2.1.1 Service Provider Marketing Request 22
2.1.2 Subnetwork & NE Considerations 23
2.1.3 Additional Functional and Model Requirements 25
2.1.4 Connection Management Functions Supported 25

2.2 Fault Management SOA 26
2.2.1 Motivations for Fault Management Operations 26
2.2.2 Fault Management Operations 27
2.2.3 Fault Management Operations Supported 29

2.3 Protection SOA 31
2.3.1 Configuration Management 32
2.3.2 Fault Management 35
2.3.3 Protection Operations Supported 35

3. MODEL VERIFICATION SCENARIOS 37

3.1 Connection Management Scenarios 37
3.1.1 Network Creation (and autodiscovery) 38
3.1.2 Network Connection Configuration 44

3.2 Connection Configuration with Status Reporting Scenarios 48
3.2.1 Connection Configuration with Status Reporting 49
3.2.2 Functional Definitions of Object Classes 50

3.3 Fault Management Scenarios 52
3.3.1 Assumptions 53
3.3.2 EMS Application Functions 54
3.3.3 Requirements for Fault Management for NEL Detected Faults 55
3.3.4 Information Included in an RCAA Notification 56
3.3.5 Information Included in the RCAA Record 59

3.4 Management Scenarios for Protection 60
3.4.1 Technology-specific Architecture Scenarios 61
3.4.2 TMN System Level Considerations 64
3.4.3 Highly-protected Connection Scenarios 65
3.4.4 Protected Connection Scenarios 67

4. INFORMATION MODEL 69

4.1 Inheritance Hierarcy 69

4.2 Naming Hierarchy 70

4.3 Entity-Relationship Diagrams 71

4.4 Characteristic Information 73

4.5 States 74

4.6 Information Model 75

List of Figures

FIGURE 1-1 PHYSICAL REALIZATION EXAMPLES OF MULTI-LAYER NETWORK MANAGEMENT ARCHITECTURE 3
FIGURE 1-2 EXAMPLE OF NE+NETWORK-LEVEL MANAGEMENT PHYSICAL CONFIGURATION 4
FIGURE 3-1 EXAMPLE SONET NETWORK ARCHITECTURE 37
FIGURE 3-2 PROCESSES AND MESSAGE FLOWS - BUILD INVENTORY AND MAP TO NODAL/GEOGRAPHIC OBJECTS (NETWORK VIEW ONLY) 39
FIGURE 3-3 OBJECT INSTANCES FOR LAYER NETWORK RESOURCE CREATION 40
FIGURE 3-4 NETWORK AND NE VIEW TERMINATION POINTS RELATIONSHIPS 41
FIGURE 3-5 PROCESSES AND MESSAGE FLOWS - BUILD SUBNETWORKS, LINKS, AND PARTITIONED VIEWS 42
FIGURE 3-6 OBJECT INSTANCES FOR DEFAULT SUBNETWORK TOPOLOGY CREATION 43
FIGURE 3-7 OBJECT INSTANCES FOR PARTITIONED SUBNETWORK TOPOLOGY CREATION44
FIGURE 3-8 PROCESSES AND MESSAGE FLOWS: CREATE LINK CONNECTIONS, SUPPORTING TRAILS IN EACH LAYER 45
FIGURE 3-9 OBJECT INSTANCES FOR LAYER NETWORK CONNECTION CONFIGURATION 46
FIGURE 3-10 PROCESSES AND MESSAGE FLOWS: SET-UP SUBNETWORK CONNECTIONS 47
FIGURE 3-11 OBJECT INSTANCES FOR STS-1 SUBNETWORK CONNECTION CONFIGURATION48
FIGURE 3-12 CONNECTION CONFIGURATION STATUS REPORTING SCENARIO 49
FIGURE 3-13 EMS APPLICATION LOGIC EXAMPLE 50
FIGURE 3-14 NETWORK DETECTED TROUBLE SCENARIO FROM GR-2869 53
FIGURE 3-15 ALARM AND RCAA MESSAGE FLOW 55
FIGURE 3-16 GENERIC CASES FOR PROTECTED CONNECTIONS 62
FIGURE 3-17 UPSR-UPSR INTERCONNECTION (LOGICAL RING) 63
FIGURE 3-18 UPSR-UPSR INTERCONNECTION (DROP AND CONTINUE) 63
FIGURE 3-19 INFORMATION AT TMN INTERFACES 65
FIGURE 3-20 HIGHLY-PROTECTED SCENARIO WITH DUAL-HOMING TO CUSTOMER PREMISES 66
FIGURE 3-21 HIGHLY-PROTECTED SCENARIO WITH SINGLE ACCESS 67
FIGURE 3-22 NMS/EMS CONNECTION ATTRIBUTES EXAMPLES 67
FIGURE 3-23 PROTECTED CONNECTION SCENARIO 68
FIGURE 4-1 INHERITANCE HIERARCHY 69
FIGURE 4-2 INHERITANCE HIERARCHY (CONTINUED) 69
FIGURE 4-3 NAMING HIERARCHY 70
FIGURE 4-4 NAMING HIERARCHY (CONTINUED) 70
FIGURE 4-5 TERMINATION POINTS AND AGGREGATES 71

FIGURE 4-6 SUBNETWORK TOPOLOGY – UNPARTITIONED 71
FIGURE 4-7 SUBNETWORK TOPOLOGY – INTER-LAYER ROLE RELATIONSHIPS 72
FIGURE 4-8 SUBNETWORK TOPOLOGY – PARTITIONING ROLE RELATIONSHIPS 72
FIGURE 4-9 CONNECTIONS – UNPARTITIONED VIEW 73
FIGURE 4-10 CONNECTION PARTITIONING ILLUSTRATION 73

1. Introduction

It is the function of the SIF Network Management Information Model Working Group to derive a CMISE GDMO information model for use between the SONET NMS and SONET EMS. This document provides the information model and the material which was defined in support of the information model.

The document is divided into four sections. This section covers introductory information. Section 2 provides Statements of Applications which define the high level requirements for the Information Model. Section 3 provides scenarios which serve both to refine the requirements defined in Section 2 and also shows how the information models support the requirements defined in Section 2. Section 4 contains the information model.

1.1 History of the Development Process and Methodology

This information model has been aligned with the work of others standards bodies where possible. This has encompassed, and included, methods and models as presented by ITU, ETSI, NMF and ATMF. It is a method that, while it has at times created a need for this working group to reevaluate portions of its efforts, has at least kept this work current with the overall needs of the service providers in managing networks with scopes exceeding those of SONET alone. However; it is the charter of this group to focus on SONET requirements specifically.

1.2 Scope

This document specifies a CMISE GDMO information model for use between the SONET NMS and SONET EMS. It does not specify the OSI stack or CMISE functions which are used to transport the information. Specifications for OSI stack profiles and CMISE functions can be found in GR-253 and in the SIF document “Requirements for SIF OS Platforms”.

This specification addresses the network-view aspects of the NMS/EMS (Network Management System/Element Management System) interface needed to support a SONET network management, i.e. the management of aggregates of Network Elements such as subnetworks. It also relates the SONET Network view and NE view to build coherent management functions. Although it is architecturally permissible to offer NE view only or Network view only management services, it is to be kept in mind that the network view is intended as an aggregate view to provide additional value. In this sense, the network-view should not constitute an opaque obstacle for reaching the NE; but it should be conceived as an organized way of seeking NE details, when needed.

This specification focuses on what is considered to be the initial functionality of SONET network view management. It is understood that this initial set of functions, managed entities, and scenarios will be enhanced in subsequent versions.

This document addresses the following functional areas of SONET network management:

- Transport network connection management (including set-up/ modification for subnetwork connection, link connection, and trails.)
- Transport network configuration provisioning (including subnetwork provisioning, and link provisioning)
- Network fault management (including correlation, localization, notification, for both equipment and connections)
- protection-switching

The following functional areas may be addressed in the future:

- Network connection reservation
- Loopback testing
- Network performance management (including congestion, and connection monitoring)
- Inventory including bandwidth management

Note that functional requirements, but no managed entities may be provided for the following functionality in the current specification:

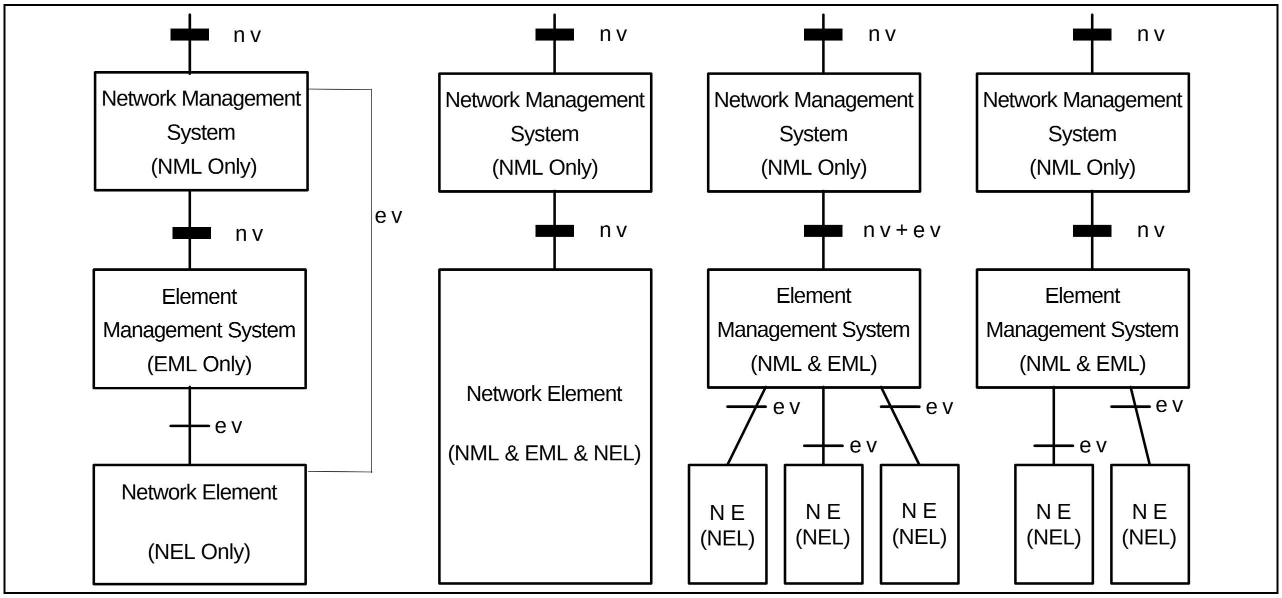
- grouping of subnetwork, or link connections, or of trails,
- routing constraints,
- scheduling or reservation,
- performance monitoring

This specification focuses on the interface functionality to manage the subnetwork, and does not provide requirements on the management systems themselves. Only Public Data Networks are addressed in this specification, the management of a Private Data Network is not included.

1.3 Network Management Architecture

In order to understand which functions will be used in this information model it is necessary to define the philosophy for the network management architecture. One can view the network with its associated network management layers as shown in Figure 1-1. We show here the Network Elements (NEs), EMSs and NMSs, but none of the higher layer management systems. (We are not precluding the use of the higher layers of management by omitting them, but they are not our current focus.) The EMS may address Network Management Layer (NML) and Network Element Layer (NEL) functions along with Element Management Layer (EML) functions. The Network-view (nv) and the NE-view (ev) between the individual management systems are also shown.

Implementation of both the Network View and NE View together represents a specific design choice. Also, implementations that provide a “stand alone” network view (no references to NE view objects) may be defined using the objects described in this document.


nv
Network Management System (NML Only)
nv
ev
Element Management System (EML Only)
ev
Network Element (NEL Only)
nv
Network Management System (NML Only)
nv
Network Element (NML & EML & NEL)
nv
Network Management System (NML Only)
nv+ev
Element Management System (NML & EML)
ev
ev
ev
NE (NEL)
NE (NEL)
NE (NEL)
nv
Network Management System (NML Only)
nv
Element Management System (NML & EML)
ev
ev
NE (NEL)
NE (NEL)

ev - network element view
nv - network view

Figure 1-1 Physical Realization Examples of Multi-layer Network Management Architecture

To support the multiple architectures described in Figure 1-1, the SONET NE-view and the SONET Network-view MIBs can be combined in multiple fashions. The SONET network management interface requirements in this document address the EMS/NMS interface, regardless of the functional layers addressed by the individual EMS or NMS. The primary focus of these requirements is the NMS to EMS interaction needed to support SONET subnetwork management. With respect to Figure 1-1, these requirements are relevant to SONET NEs, EMSs, and NMSs supporting EML functions.

1.4 NE+Network-Level Management Architecture Example

The NE+Network-Level Management Architecture (see Figure 1-2) exposes the “SONET NE-View” between the NMS Environment and the EMS (managing the subnetwork). In this architecture, the NMS has the option to view and manage the SONET network by performing operations on the subnetwork as a whole or by performing operations on select SONET NEs. One could imagine that, for certain NMS applications, a single-entity subnetwork view would be sufficient, while for other applications a detailed view of each SONET NE comprising the subnetwork as well as their interconnections would be desirable. The models defined in this document may also be used without exposing the NE-view.

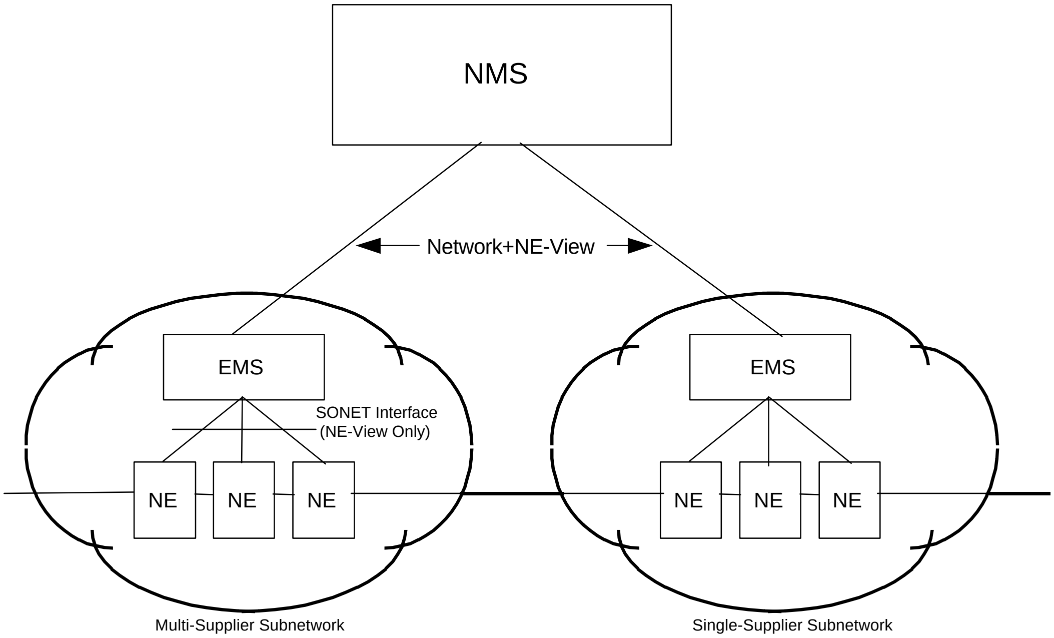


Figure 1-2 Example of NE+Network-Level Management Physical Configuration

1.5 Changes from SIF-014-1997

1. Addition of requirements for protection in Section 2.3 and a mapping of requirements for protection to functions supported by the model.
2. Addition of scenarios for protection in Section 3.4.
3. Introduction of linkEnd and accessGroup as a method of associating network CTPs and TTPs to subnetworks. This eliminated the need for sNTPs which simplified the model and reduced the number of objects required. This change is fully documented in SIF-IM-9807-104R1.
4. Support in the model for protected connections. These changes are fully documented in SIF-IM-9804-072R2.
5. Updated Section 3.1 to reflect changes to model for linkEnd. This change is fully documented in SIF-IM-9807-114R1.
6. Introduced the virtual objects, sonetVirtualCTP, sonetVirtualLink, sonetVirtualLinkEnd, and sonetVirtualLinkConnection.
7. Updated the glossary to include terms for protection, virtual objects and linkEnd.

1.6 References

The following standards contain information which was used while defining the information model in this document.

ITU-T Recommendation G.774 (09/92), *Synchronous Digital Hierarchy (SDH) Management Information Model for the Network Element View*.

ITU-T Recommendation G.774.01 (11/94), *Synchronous Digital Hierarchy (SDH) Performance Monitoring for the Network Element View*.

ITU-T Recommendation G.774.02 (11/94), *Synchronous Digital Hierarchy (SDH) Configuration of the Payload Structure for the Network Element View*.

ITU-T Recommendation G.774.03 (11/94), *Synchronous Digital Hierarchy (SDH) Management of Multiplex-Section Protection for the Network Element View*.

ITU-T Recommendation G.774.04 (07/95), *Synchronous Digital Hierarchy (SDH) Management of the Subnetwork Connection Protection for the Network Element View*.

ITU-T Recommendation G.774.05 (07/95), *Synchronous Digital Hierarchy (SDH) Management of Connection Supervision Functionality (HCS/LCS) for the Network Element View*.

ITU-T Recommendation G.774.07 (11/96), *Synchronous Digital Hierarchy (SDH) Management of Lower Order Path Trace and Interface Labelling for the Network Element View*.

ITU-T Recommendation G.805 (11/95), *Generic Functional Architecture of Transport Networks*.

ITU-T Recommendation G.853.1, *Common Elements of the Information Viewpoint for the Management of a Transport Network*.

ITU-T Recommendation G.853.2, *Subnetwork Connection Management Information Viewpoint*.

ITU-T Recommendation G.855.1, *Management of the Transport Network - Class Library for GDMO Transport Network Model.*

ITU-T Recommendation M.3010 (05/96), *Principles for a Telecommunications management network.*

ITU-T Recommendation M.3100 (07/95), *Generic Network Information Model*.

ITU-T Recommendation Q.821 (03/93), *Stage 2 and Stage 3 Description for the Q3 Interface - Alarm Surveillance*.

ITU-T Recommendation X.710 (03/91), *Common management information service definition for CCITT applications*.

ITU-T Recommendation X.721 (02/92), *Information Technology - Open Systems Interconnection - Structure of Management Information: Definition of Management Information*.

ITU-T Recommendation X.733 (02/92), *Information technology – Open Systems Interconnection – Systems Management: Alarm reporting function*.

ITU-T Recommendation X.734 (09/92), *Information technology – Open Systems Interconnection – Systems Management: Event report management function*.

ITU-T Recommendation X.735 (09/92), *Information technology – Open Systems Interconnection – Systems Management: Log control function*.

AF-NM-0058.000, *NMS/EMS Network View Interface Requirements, and Logical MIB*, (ATM Forum Technical Committee)

AF-NM-0073-000 Letter Ballot, *M4 Network View CMIP MIB Specification* Version 1.0, November 1996.

GR-253-CORE (December 1995), *Synchronous Optical Network (SONET) Transport Systems: Common Generic Criteria (A Module of TSGR, FR-NWT-000440)*, Issue 2, (Bellcore)

TR-NWT-000496 (May 1992), *SONET Add-Drop Multiplex Equipment (SONET ADM) Generic Criteria*, Issue 3, (Bellcore).

GR-836-IMD (September 1996), *Generic Operations Interfaces Using OSI Tools - Information Model Details:Transport Configuration and Surveillance for Network Elements*, Issue 2, (Bellcore)

GR-1042-IMD (September 1996), *Generic Requirements for Operations Interfaces Using OSI Tools - Information Model Overview: Synchronous Optical Network (SONET) Transport Information Model*, Issue 2, (Bellcore)

SR-TSV-002671 (June 1993), *EML Applications for Fault Management: Subnetwork Root Cause Alarm Analysis*, Issue 1 (Bellcore).

SR-TSV-002672 (March 1994), *EML Applications for Fault Management: Intelligent Alarm Filtering for SONET*, Issue 1 (Bellcore).

GR-2869-CORE (October 1996), *Generic Requirements for Operations Based on the TMN Architecture*, Issue 2 (Bellcore).

SIF-016-97, *Requirements for SIF OS Platforms*.

1.7 Definitions

This section lists the terms and definitions used in this document. Many of these terms have different definitions in other standards documents. Attempts were made to adopt the most common definition of each term. The definition includes the context in which the term is used. Where possible, the relationship between a term and the real world is provided.

KEY:

- "double quotes" - term is defined elsewhere in this glossary
- *italics* - element of the SIF Information Model (e.g., object, attribute, etc.)

1.7.1 1+1 protection

Protection switching is considered to be 1+1 if, per direction of transmission, the protection switch occurs without coordination between the endpoints. In 1+1 architectures, per direction of transmission, the receive end can perform a protection switch without coordinating with the transmit end since the signal is permanently bridged at the transmit end, and an APS channel is not necessarily required. (T1.105.01)

1.7.2 access group

A group of co-located "network trail termination points" and "network conntection termination points" within a "layer network domain." The *sonetAccessGroup* managed object class is used to represent an access group in the model.

1.7.3 add traffic

Traffic inserted into working channels on the ring at a ring node. (T1.105.01)

1.7.4 administrative domain

A set of network and administrative resources grouped for management purposes. For the NMS/EMS interface, the grouping of managed objects corresponds to the resources managed by an EMS agent. An administrative domain will typically encompass a number of "layer network domains" associated with distinct transport layers. The *networkR1* managed object class is used to represent an administrative domain in the model.

1.7.5 bicast

A configuration of a uni-directional transport entity originating at a single source termination point and terminating at two sink termination points. A bicast connection can be used as a component of an end-to-end "highly protected" connection.

1.7.6 bridge

The act of transmitting identical traffic on both the working and protection channels. (T1.105.01)

1.7.7 Client/Server (C/S) Pointers

The client pointer attribute (clientLayerLinkEnd) identifies the object instances of the related sonetLinkEnd (and subclasses) in the client layer. The server pointer attribute (serverLayerNTTP) identifies the object instance of the related networkTTP (and subclasses) in the server layer.

1.7.8 characteristic information

A signal with a specific rate and format, which is transferred on “network connections” [G.853-01]. Characteristic information is also associated with termination points independent of whether or not they are supporting connections. The potential signal formats include signals of the SONET hierarchy (VT/STS/OC) and digital tributary signal formats (DS1, DS3).

1.7.9 connection management

A management application designed to manage the set-up, release, and modification of subnetwork connections and trails in a layer network. Connection management will generally require application functions in multiple TMN management layers (i.e., SML, NML, EML, and NEL). Although the focus is on configuration management, functions in other functional areas (e.g., fault, performance management) may be included in the application. SIF functional requirements for a SONET connection management application are described in G.805.

1.7.10 Connection Termination Point (CTP)

A Connection Termination Point is a managed object that terminates a link connection. See M.3100.

1.7.11 connection type

The connection type indicates the basic configuration of a connection including cardinality relationships among endpoints of a subnetwork connection or trail. Possible types include:

- point-to-point
- point-to-multipoint
- bicast
- exclusive merge
- exclusive composite

1.7.12 Downstream Connectivity Pointer (DCP)

Defined in M.3100, the downstream connectivity pointer attribute points to the termination point managed object, within the same managed element, that receives information (traffic) to this termination point at the same layer, or is null.

1.7.13 drop

The port on a SONET network element where the service to an end customer may be connected, e.g., a tributary card on a SONET ADM. For example, a drop for a DS1 customer service may be provided by a VT1.5 card terminating a VT1.5 trail.

1.7.14 drop and continue

The ability of a SONET add-drop multiplex to pass the same signal (STS/VT) that is being dropped onto the outgoing OC-N signal [SR-2672]. A function within a ring node where traffic is both extracted from the working channels on the ring (drop), and transmitted onwards on the ring (continue). (G.842)

1.7.15 drop traffic

Normal or extra traffic extracted from working, protection, or non-preemptible unprotected channels on the ring at a ring node. (G.841)

1.7.16 dual hubbed

A configuration in which customer traffic can be routed to either or both of two hubs to enable diverse route protection. (see G.842)

1.7.17 dual node interconnect

An architecture between two rings where two nodes in each ring are interconnected to enable diverse route protected ring interconnection. (see G.842)

1.7.18 dynamic subnetworks

A management capability that allows modification of a subnetwork's properties in terms of associated resources.

1.7.19 Element Management Layer (EML)

An abstraction of the functions provided by systems which manage each network element on an individual basis.

1.7.20 Element Management System (EMS)

A management system, which provides functions at the Element Management Layer, and could also include functions at the Network Management Layer. The "administrative domain" associated with the EMS agent could be delimited by geographical area, topology, or supplier product (as examples) within the provider's network.

1.7.21 exclusive composite

A configuration of a bi-directional transport entity connecting three bi-directional termination points with an "exclusive merge" configuration in one direction and a "bicast" configuration in the opposite direction. An exclusive composite connection can be used as a component of an end-to-end protected connection configuration; it is not intended for supporting multicast services to customers.

1.7.22 exclusive merge

A configuration of a uni-directional transport entity originating at two source termination points and terminating at a single sink termination point in which only one of the two source signals is passed to the sink at any given time. An exclusive merge connection provides a component of an end-to-end protected connection configuration.

1.7.23 extra traffic

Traffic that is carried over the protection channels when that capacity is not used for the protection of working traffic. Extra Traffic is not protected. Whenever the protection channels are required to protect the working traffic, the Extra Traffic is preempted. Ring interworking on protection is considered Extra Traffic. (T1.105.01)

1.7.24 hardwired multiplex

An add-drop multiplex configuration in which specific VT/STS-1 time-slots are dedicated to specific low-speed ports [SR-2672].

1.7.25 hold off time

The time that a protection switch controller waits after detecting a failure before initiating the switch. (G.842)

1.7.26 highly protected

The “protection level” of a transport entity that is protected against single failures of either a facility or a node (except for bicast, exclusive merge, or composite configurations in which the node that contains the bridge/selector function is not protected). Service traffic on a highly protected transport entity is not preemptible.

1.7.27 layer network

A "topological component" that includes other topological components, transport entities, and transport processing functions that describes the generation, transport and termination of a particular characteristic information [G.853-01].
As an example, a layer network may be associated with SONET STS-1 transport.

1.7.28 layer network domain

The part of a layer network which is managed by a management system [G.853-01]. For the NMS/EMS interface, the relevant management system is the EMS. The *sonetLayerNetworkDomain* managed object class is used to represent a layer network domain in the model.

1.7.29 link

A “topological component” that provides transport capacity between two endpoints in different subnetworks via a fixed (i.e., inflexible routing) relationship[1]. The endpoints

[1] An exception to the “fixed” relationship is when protection mechanisms are applied.

are “link ends.” Multiple links may exist between a pair of subnetworks. A link also represents a set of “link connections.” The *sonetLink* managed object class is used to represent a link in the model.

1.7.30 link connection

A “transport entity” that represents the fixed capacity of transfer of “characteristic information” transparently across a link. A link connection is delineated by “network connection termination points” or “NE-view connection termination points.” The “network connection termination points” can be associated with “subnetwork termination points” by relationship. The *sonetLinkConnection* managed object class is used to represent a link connection in the model.

1.7.31 link end

A link end represents the extremity of a link. It may have an associated set (possibly empty) of networkCTPs. The *linkEnd* managed object class is used to represent a link end in the model.

1.7.32 logical ring

(See subnetwork connection protection.)

1.7.33 M:N protection switching

Protection switching is considered to be M:N if, per direction of transmission, the protection switch requires coordination between endpoints. In 1:1 architectures, for example, per direction of transmission, the receive end must request a bridge from the transmit end in order to perform a protection switch, and an APS channel is required. (T1.105.01)

1.7.34 Management Applications Function (MAF)

An application process participating in system management. The management application function includes an agent (being managed) and/or manager [G.784].

1.7.35 multi-domain ring

A ring architecture (e.g., UPSR, BLSR, interconnected rings, subtending rings) comprised of two or more administrative groupings of NEs each associated with a distinct administrative domain.

1.7.36 multiple partitioning views

A management capability that supports more than one scheme of partitioning subnetworks. This allows the use of different partitioning schemes for different functional areas or different management systems.

1.7.37 network connection

A “transport entity” formed by a series of contiguous "link connections" and/or "subnetwork connections" between subnetwork termination points. A network connection may extend across “layer network domains” associated with more than one “administrative domain.” A network connection is not represented by an object class in the model.

1.7.38 network connection termination point

An extremity of a “link connection” [G.855-01]. It is also a network level abstraction of a nodal (NE) view connection termination point. The *networkCTP* managed object class is used to represent a network connection termination point in the model.

1.7.39 network connection termination point bidirectional

A network connection termination point that represents the functionalities of both a network connection termination point source and network connection termination point sink in the model.

1.7.40 network connection termination point sink

A unidirectional network connection termination point that is intended to be bound to the output of a unidirectional “link connection.” The *sonetNetworkCTPSink* managed object class is used to represent a network connection termination point sink in the model.

1.7.41 network connection termination point source

A unidirectional network connection termination point that is intended to be bound to the input of a unidirectional “link connection.” The *sonetNetworkCTPSource* managed object class is used to represent a network connection termination point source in the model.

1.7.42 Network Element (NE)

A system that supports at least “NEFs” and may also support “Element Management Layer” Functions/Mediation Functions. It cannot be further decomposed into managed elements in the context of a given management function.

1.7.43 Network Element Function (NEF)

A function within a SONET entity that supports the SONET based network transport services, e.g. cross-connections.

1.7.44 Network Element Layer (NEL)

An abstraction of functions related specifically to the technology, vendor, and the network resources or network elements that provide basic communications services.

1.7.45 network element view (NE view)

The network element view is the network view representating information received directly from network elements which is defined in other information models such as GR-1042.

1.7.46 Network Management Layer (NML)

An abstraction of the functions provided by systems which manage network elements on a collective basis as subnetworks, and/or as individual entities.

1.7.47 Network Management System (NMS)

An entity which implements functions at the Network Management Layer. It may also include Element Management Layer functions.

1.7.48 NMS Environment - *under study*

A set of Network Management Systems (NMS) which cooperate to manage one or more subnetworks.

1.7.49 network trail termination point

An extremity of a “trail” [G.855-01]. It is also a network level abstraction of a nodal (NE) view trail termination point. A network trail termination point includes trail termination functions that ensure integrity of information transport on an end-to-end basis (see Figure I.2 in G.855-01 for a mapping between termination point managed objects [G.855-01] and termination functions and access points in a functional network architecture [G.853-01]). The *sonetNetworkTTP* managed object class is used to represent a network trail termination point in the model.

1.7.50 network trail termination point bidirectional

A network trail termination point that represents the functionalities of both a network trail termination point source and network trail termination point sink in the model.

1.7.51 network trail termination point sink

A network trail termination point that is intended to be bound to the output of a unidirectional trail. The *sonetNetworkTTPSink* managed object class is used to represent a network trail termination point sink in the model.

1.7.52 network trail termination point source

A network trail termination point that is intended to be bound to the input of a unidirectional trail. The *sonetNetworkTTPSource* managed object class is used to represent a network trail termination point source in the model.

1.7.53 network view

The network view is an abstracted view of the network which would include some level of NE connectivity.

1.7.54 NE-view connection termination point

A managed object class defined in an information model for network elements which represents the termination of a link connection.

1.7.55 NE-view trail termination point

A managed object class defined in an information model for network elements which represents the termination of a trail.

1.7.56 nodal view

The partitioned view of a network in which some or all of the resources of the NE are represented by a subnetwork. (In some cases, several NEs may be configured in a way that permits them to be managed as a single NE.)

1.7.57 node

A group of objects defined on the basis of mutual proximity to support the routing of highly protected connection services. Alternate paths for a highly protected connection service are not generally permitted to pass through the same node (an exception is the case in which access to a customer location is provided at a single node, i.e. not dual-homed). Objects within a node may be mapped to a single NE or multiple NEs.

1.7.58 non-preemptible unprotected channel

A channel in a BLSR (MS shared protection ring) provisioned bidirectionally to provide transport without MS shared protection ring automatic protection switching. Non-preemptible unprotected channels are provisioned from (corresponding) working and protection channel pairs.

1.7.59 non-preemptible unprotected traffic

Unprotected traffic carried on protection locked-out channel which may not be preempted (e.g., by protection switches).

1.7.60 partitioning

The decomposition of a “subnetwork” into its component “subnetworks” and “links” in a way that reflects the internal structure (topology) of that “subnetwork” or the way that it will be managed. Partitioning may be based on a variety of factors including architecture (e.g., BLSR), supplier product line, or administrative considerations.

1.7.61 point-to-point route

A point-to-point route consists of two end points and, optionally, intermediate points in a layer network. Each end point or intermediate point will consist of a set of termination points in the same layer network.

1.7.62 preemptible

The “protection level” associated with a transport entity provisioned to carry “extra traffic.” The “extra traffic” may be preempted to provide transport capacity for “protected” or “highly protected” transport entities in the event of failure.

1.7.63 protected

The “protection level” associated with a transport entity that is protected against single failures of a facility that supports the entity. Service traffic on a protected transport entity is not preemptible.

1.7.64 protection channels

The channels allocated to transport the working traffic during a switch event. When there is a switch event, traffic on the affected working channels is bridged onto on the protection channels. (T1.105.01)

1.7.65 protection level

The protection level indicates the class of network level protection provided in response to basic types of failure. Possible values of protection level are:

- “highly protected”
- “protected”
- “unprotected”
- “preemptible”

Two types of protection level are included in the protection model: The provisioned protection level indicates the level of protection established via provisioning. The operational protection level indicates the level of protection actually being supported at a given time.

1.7.66 protection

In the network view, protection refers to the ability to switch service from a primary “transport entity” to a preconfigured backup “transport entity” in response to a detected failure on the primary “transport entity.” Protection mechanisms may be entirely within a “layer network” or may involve a server layer mechanism supporting client transport services, and may be activated on the basis of a variety of monitoring. Trail protection is a protection method applied in a “layer network” when a defect condition is detected in the same “layer network.” Subnetwork connection protection is applied in the client layer network when a defect condition is detected in a server layer network, sub-layer or other transport layer network.

1.7.67 requested effort level

The degree of commitment to obtain a requested “protection level” agreed to as part of connection set-up. Three effort levels are identified:

- required minimum - connection cannot be set up if requested protection level or higher is not available
- required exact - connection cannot be set up if requested protection level is not available
- best effort - connection may be set up even if requested protection level is not available

1.7.68 restoration

In the network view, restoration refers to an application in which the NMS responds to a confirmed failure by requesting a new connection. This action may be viewed by the EMS as a release of the failed connection and the set-up of a new connection.

1.7.69 ring interconnection

An architecture between two rings where one or more nodes in each ring are interconnected. (G.842)

1.7.70 ring interworking

A network topology whereby two rings are interconnected at two nodes on each ring, and the topology operates such that a failure at either of these two nodes will not cause loss of any working traffic. (G.842)

1.7.71 ring switch

Protection mechanism that applies to both two-fibre and four-fibre rings. During a ring switch, the traffic from the affected span is carried over the protection channels on the long path. (G.841)

1.7.72 route

A sequence of geographical or topological points or components through which “transport entities” may be established. “Transport entities” within a route have common endpoints and common intermediate points (if intermediate points are specified). The role of route endpoint may be played by a “subnetwork” or “access group;” an intermediate point may be associated with a “subnetwork,” or “link end.” In an unpartitioned subnetwork view, the route includes all potential connections in the “subnetwork” between “access groups;” in a fully-partitioned subnetwork view, the route may include all available connections supported by a specific set of paths (links) and NEs (fabrics).

1.7.73 route capacity

A measure of the transport capacity available on an end-to-end basis on a specific “route” within a “layer network domain.”

1.7.74 route separation

A qualitative measure of the relationship between “routes” assigned to primary and protection connections.

1.7.75 span switch

Protection mechanism similar to 1:1 linear APS that applies only to four-fibre rings where working and protection channels are contained in separate fibres and the failure only affects the working channels. During a span switch, the normal traffic is carried over the protection channels on the same span as the failure. (G.841)

1.7.76 selector

The NE function that selects and allows to pass one signal out of two inputs based on predefined criteria.

1.7.77 service

Service or transport service is the communications product purchased by a customer, represented by a "subnetwork connection." Services come in many forms, including switched services, data services, and private line services.

1.7.78 state

State is an attribute type representing the condition or an object instance. See X.721.

1.7.79 subnetwork

A "topological component" used to effect routing of a specific "characteristic information" [G.853-01]. A subnetwork is associated with a specific "layer network." Within a given layer, "partitioning" may be applied to decompose a subnetwork into its component subnetworks and links. The *sonetSubnetwork* managed object class is used to represent a subnetwork in the model.

1.7.80 subnetwork connection (SNC)

A "transport entity" that transfers information across a subnetwork [G.853-01]. A point-to-point subnetwork connection connects two "subnetwork termination points." An SNC may be either a stand-alone SNC, or a concatenation of SNCs and link connections. The *sonetSubnetworkConnection* managed object class is used to represent a subnetwork connection in the model.

1.7.81 subnetwork connection protection

A type of protection architecture (using "1+1 protection switching") applied to a subnetwork connection in which a transmitted signal is bridged onto two paths at the source and selected between the two paths near the receiver based on pre-defined criteria. Also referred to as "logical ring."

1.7.82 Subnetwork Management System (subNMS) - *under study*

A Network Management System, which is managing one or more subnetworks, and which is managed by one or more Network Management Systems.

1.7.83 subnetwork capacity

A measure of the total/available transport capacity within a subnetwork. This measure may be useful in planning for network topology changes. (See "route capacity.")

1.7.84 subtending rings

A SONET architectural configuration which allows a secondary ring to share bandwidth with a primary ring via drop side interfaces at one or two nodes of the primary ring. The secondary ring typically runs at a lower speed compared to the primary ring.

1.7.85 Termination Point (TP)

A Termination Point is a managed object that terminates “transport entites” such as “trails” and connections. See M.3100.

1.7.86 Time-Slot Assignment (TSA)

The capability to flexibly assign add-dropped signals, but not through signals. Through signals maintain the same time-slots on the incoming and outgoing signals [SR-2671].

1.7.87 Time-Slot Interchange (TSI)

The capability to flexibly assign both add-dropped signals and through signals [SR-2671].

1.7.88 topological component

An architectural component, used to describe the transport network in terms of the topological relationships between sets of points within the same “layer network” [G.853-01]. Examples of topological components include “layer network,” “subnetwork,” and “link.”

1.7.89 trail

A "transport entity" which consists of an associated pair of "unidirectional trails" capable of simultaneously transferring information in opposite directions between their respective inputs and outputs [G.853-01]. A trail also represents the transfer of “characteristic information” between “network trail termination points” or “NE-view trail termination points.” A trail is supported by a “subnetwork connection” or “network connection” and in addition includes trail termination functions that ensure integrity of information transport (i.e., via monitoring) on an end-to-end basis. A trail may be established to directly support an end-to-end network service or to provide “link connections” within the client layer. A trail is represented in the model by the *sonetTrail* managed object class.

1.7.90 trail termination function

A transport processing function that allows the monitoring of information transport on trails on an end-to-end basis. Two types of trail termination functions are defined: a trail termination source adds monitoring information to the “characteristic information” input at one end; a trail termination sink removes the monitoring information and presents the “characteristic information” at the other end.

1.7.91 Trail Termination Point (TTP)

A Trail Termination Point is a managed object that terminates a trail. It represents the access point to a subnetwork. See M.3100.

1.7.92 trail protection

Normal traffic is carried over/selected from a protection trail instead of a working trail if the working trail fails, or if its performance falls below a required level. (G.841)

1.7.93 transport entity

An architectural component which transfers information between its inputs and outputs within a “layer network” [G.853-01]. Examples of transport entities include “subnetwork connection,” “link connection,” and “trail.”

1.7.94 unprotected

The protection level associated with a transport entity that has no guarantee of protection against facility failures. Service traffic on an unprotected transport entity is not preemptible.

1.7.95 Upstream Connectivity Pointer (UCP)

Defined in M.3100, the upstream connectivity pointer attribute points to the termination point managed object, within the same managed element, that sends information (traffic) to this termination point at the same layer, or is null.

1.8 Acronyms

ADM	Add Drop Multiplexer
AIS	Alarm Indication Signal
APS	Automatic Protection System
ATMF	Asynchronous Transfer Mode Forum
BLSR	Bi-directional Line Switched Ring
BML	Business Management Layer
C/S	Client/Server
CMISE	Common Management Information Service Element
CTP	Connection Termination Point
DCN	Digital Communication Network
DCP	Downstream Connectivity Pointer
DCS	Digital Cross-connect System
DSn	Digital Signal hierarchy, layer n
EFD	Event Forwarding Discriminator
EML	Element Management Layer
EMS	Element Management System
E-R	Entity-Relationship
ETSI	European Telecommunications Standards Institute
GDMO	Guidelines for the Definition of Managed Objects

ITU-T	International Telecommunications Union
LOF	Loss of Frame
LOP	Loss Of Pointer
LOS	Loss of Signal
MAF	Management Application Function
MIB	Management Information Base
NE	Network Element
NEF	Network Element Function
NEL	Network Element Layer
NMF	Network Management Forum
NML	Network Management Layer
NMS	Network Management System
NSA	Non-Service-Affecting
OC-N	Optical Carrier - level N
OOF	Out Of Frame
OS	Operating System
OSI	Open Systems Interconnection
OTDR	Optical Time Delayed Reflectometry
QOS	Quality Of Service
PM	Performance Monitoring
RCAA	Root Cause Alarm Analysis
RDI	Remote Defect Indication
RFI	Remote Failure Indication
SA	Service-Affecting
SLM	Signal Label Mismatch
SML	Service Management Layer
SNC	SubNetwork Connection
SOA	Statement Of Application
STS-N	Synchronous Transport Signal level N
STS-Nc	Synchronous Transport Signal level N, concatenated
TCA	Threshold Crossing Alarm

TMN	Telecommunications Management Network
TP	Termination Point
TSA	Time-Slot Assignment
TSI	Time-Slot Interchange
TTP	Trail Termination Point
UCP	Upstream Connectivity Pointer
UPSR	Unidirectional Path Switched Ring
VT	Virtual Tributary

2. Statements of Application

A Statement of Application (SOA) defines the service provider's functional requirements for a management application. In this section, SOAs are presented for Connection Management and Fault Management applications. A section on requirements for protection is also included. Each SOA represents a broad set of requirements that may have impact on management applications at various TMN layers as well as interfaces between management systems supporting the applications. Following each SOA section is a section that indicates which particular SOA requirements are supported in this version of the information model for the NMS/EMS interface.

2.1 Connection Management SOA

The objectives of connection management are:

- the ability to setup and tear down connections in a subnetwork
- the ability to report and control the configuration and status of components of a subnetwork

2.1.1 Service Provider Marketing Request

The typical components of a service request are type of service and need for an alternative path. Type of service information may include service rate (e.g.OC-3, OC-12, DS1, DS3), payload mapping (STS-1, STS-3c), Add Drop Multiplexing and QOS.

Alternate paths may be controlled at different TMN levels. Example protection mechanisms include:

1. Protection within a subnetwork where the EMS controls the details of the protection mechanism.

2. Path protection in which the NEs containing the path terminations control the path switching. This mechanism requires that the NMS configure primary and secondary paths and create an association between them.

3. NML restoration, in which the NMS responds to a confirmed failure by requesting a new connection.

Availability of bandwidth is a tool needed for a marketing request. Marketing must do a quick check to see if the service the customer is ordering is available. A service request may require the ability to add scheduling for service turn on or turn off, and the addition of customer name and location. A service request may also include or require time of day, "Start Date", "Termination Date", or some other scheduling mechanism as well as STATE information to indicate that the facilities are reserved.

Marketing will need the ability to locate endpoints which includes the geographic as well as the logical address. Customers may require a specific drop port which implies that specific subnetwork or network element details may be required.

2.1.2 Subnetwork & NE Considerations

In support of an Inventory Application, information about facilities, ports, drop plugs, cross-connects, timeslot availability, intermediate and end test points, and state information (in service, assigned, unassigned, alarmed, outstanding conditions) must be provided at an external interface. Because the network itself is assumed to contain the most important source of network inventory information, stewardship (primary responsibility for creation/deletion/modification) of inventory data is assumed to lie more within the EMS than within the NMS.

In support of a Connection Management Application, the following topology information must be supported in the Connection Management Model:

- Partitioning

- Logical and physical geographic and termination point information are needed at the NMS to allow tariff based path diversity. (This could be a means of "route selection" by the user or presented to the user.) This also relates to finding end points A & Z, and links between subnetworks.

- The model should be rich enough to allow a network wide view at the NML (all subnetworks) and a subnetwork wide view at the EMLs.

- Network Topology Autodiscovery is needed, but can be a later modeling effort.

In support of a Connection Management Application, the following information must be supported in the Connection Management Model:

- rates - selection of a specific facility such as DS1.

- protect scheme which allows the following strategies:

  1. Protection within subnetwork - a protected subnetwork connection or trail within a subnetwork may be requested by the NMS and associated with the subnetwork connection (e.g.: ring protection); the NMS does not have visibility into the details of the protection mechanism.

  2. Path protection in which the NEs containing the path terminations control the path switching requires that the NMS configure primary and secondary paths and create an association between them. The paths and their association are passed to the EMSs whose domains contain the terminating NEs. An EMS requested to establish intermediate subnetwork connections receives two (or possibly one)

requests for subnetwork connections for which an association is not necessary. (Reversion to the original is an application decision, but the model must be rich enough to provide the proper notification upon restoral of service to revert to original configuration).

3. NML restoration, in which the NMS responds to a confirmed failure by requesting a new connection would be viewed by the EMS as a release of the failed connection and the set-up of a new connection. The Manage Pending Network Changes MAF in the NMS coordinates these actions. At the NMS/EMS interface this involves subnetwork connection set-up and release actions. (Reversion to the original is an application decision, but the model must be rich enough to provide the proper notification upon restoral of service to revert to original configuration).

- drops available - We must distinguish between fiber and wire drop interfaces and be able to retrieve rate/format supported at drop.

- bandwidth availability including scheduling which requires the ability to add scheduling for service turn on/off and the addition of customer name and location. A service request may also include or require time of day, "Start Date", "Termination Date", or some other scheduling mechanism as well as STATE information to indicate that the facilities are reserved.

  Reservation of bandwidth will also be required. The ability to query to determine the capacity remaining and trigger a notification to add more, noting time differences from reservations is required. If bandwidth falls below the threshold additional capacity will be ordered. This will require the ability to add THRESHOLDING, or THRESHOLD Crossing Information in the model.

- Assign priorities to circuits/components/sessions. This could be a vehicle for supplying "bandwidth on demand". While assignment of priorities may be an application issue, it may have implications on the model.

- Gather unused facilities (DS1, DS3. STS-1s. etc.) into pool(s) of unused facilities (DS1, DS3. STS-1s. etc.). Information required includes total available bandwidth between two ports on a subnetwork, and partitioned views (contained subnetworks and links) of routes representing that total available bandwidth.

- Ability to accept any manually entered preferred routes or explicitly preferred route separation assignments.

- Query for capacity within a subnetwork or subnetworks.

2.1.3 Additional Functional and Model Requirements

The following additional functions are required:

- Reserve all paths. This information will be forwarded to the network level management function for combining with other subnetwork paths.
- The NML will build the end-to-end connections for Primary and Secondary Paths. All paths are linked to unique circuit order identifier. The circuit order identifier could be a "service number" attribute returned via the information model to the service provider application.
- Monitor all alarms while provisioning.
- Mark all termination points and all links used as in-service. Refine the STATEs within the model.
- Test end-to-end connections to ensure working order. Modeling for testing consists of two parts: state management of resources to be tested and managing testing resources. We need to add states to objects in our model to allow resources to be taken out of service for testing.
- Notify requester of SUCCESS or FAILURE of the connection establishment.
- Notify all downstream OSs (including the Inventory Application) that the circuit is assigned and can't be reused, must be monitored etc.
- Support EFDs. This should be a standard function of CMISE supporting notifications to all management applications that requested this type of notification.

2.1.4 Connection Management Functions Supported

Only a subset of the requested Connection Management Functions are supported in this version of the model. The table below shows which Connection Management Functions are supported by this information model:

CONNECTION MANAGEMENT FUNCTION	**FUNCTION SUPPORTED**
Network Topology	
Partitioning	Yes
A & Z End Points	Yes
Geographic Location	Yes
Route selection	Yes
Rates x	Yes

CONNECTION MANAGEMENT FUNCTION	**FUNCTION SUPPORTED**
Protection Schemes	
Subnetwork Protection	Yes
Path Protection	Yes
NMS initiated Connection Restoration	Not applicable at this interface.
Drops Available	Yes
Scheduling and Reservation	No
Bandwidth Availability	Sufficient information is available to calculate route capacity
Subnetwork Pools of unused facilities	No
Establish Primary and Secondary Connections	
Establish Connection	Yes
Establish Monitored Connection	No
Release Connection	Yes
Mark TPs and Links as IN-SERVICE	No
Notify Connection SUCCESS or FAILURE	Yes

2.2 Fault Management SOA

2.2.1 Motivations for Fault Management Operations

The primary objective of Fault Management is the timely identification, isolation, diagnosis and resolution of network deficiencies. Fault Management provides the following network management benefits:

- decreased time to isolate and resolve network deficiencies
- decreased need for experienced network technicians
- better information from which to manage vendor activities
- increased network availability
- better information from which to manage operations activities
- early detection of potential problems will allow correction before full failure occurs.

The fundamental strategy for fault isolation and problem diagnosis is the sensible collection and classification of alarms from various systems. User and application connectivity will be maintained by submitting a reconfiguration request to the Reconfiguration Management processes.

Fault Management includes functions that enable fault detection and fault isolation and allow for the correction of abnormal behavior. They include alarm and trouble monitoring capabilities such as alarm surveillance, collection, classification, correlation, notification, display, review, suspension and clearing, and fault localization, isolation and testing. SONET Fault Management involves network facilities and network equipment. (For clarification, network facilities include DS1s, OC-3s, etc. Network equipment includes network elements (NEs) such as Digital Cross-Connect Systems (DCSs) and SONET Add-Drop Multiplexers (ADMs).) Alarms can originate from entities defined as networks, subnetworks, and the ring interconnections. (A set of NEs can be defined as a subnetwork. Subnetworks are viewed as logical entities consisting of multiple physical nodes.) Fault Management responds in a reactive mode to network alarms and to Performance Management requests (based on analysis of network events, PM may generate alarms).

Alarms can be considered either service affecting or transient and are categorized as one of four severity codes: critical, major, minor or informational (i.e., an event). Multi-level alarms will identify self-correcting errors, errors not affecting system operation, errors requiring operator intervention, alarms for critical processes and alarms which notify of any service-threatening situation, e.g., overload conditions.

At a minimum, alarms will contain the following information:

- Originating resource (network element, facilities, element management system, Performance Monitoring, etc.)
- Trouble explanation (e.g., defined alarm code)
- Severity code
- Date and time alarm condition occurred
- Duration of alarm condition
- Status of alarm (e.g., active, cleared, acknowledged, etc.)

Fault Management's span of control begins with the receipt of either an external or internal alarm and ends with the request for the generation of a trouble ticket.

2.2.2 Fault Management Operations

The following text describes the operations required by a service provider in performing the SONET Fault Management process.

1. Set Parameters

- Establish criteria for reliability, availability and survivability. As defined in GR-2869-CORE Generic Requirements for Operations Based on the TMN Architecture:
- Reliability refers to measures of the mean time between failure and the mean time to repair equipment.
- Availability refers to the percentage of time services and resources are ready for use.
- Survivability refers to the robustness of the network in the presence of faults.
- Register facilities for monitoring
- Define alarm types to be screened
- Set up alarm/event thresholds and other event criteria
- Assign priorities to circuits/components/sessions
- Set up rule base

2. Monitor Network Status
   - Monitoring events and alarms
   - Maintain log of events and alarms
   - Status request of any component on the network (automated and manual)
   - Collect/poll for prescreened system events or alarms
   - Collect performance information
   - Filter duplicate and informational messages
   - Suppress recurring messages
   - Suppress minor or informational alarms
   - Archive events
   - Continual data analysis and results review
   - Generate event and alarm reports

3. Detect Network Deficiencies
   - Update the user interface (network topology)
   - Retrieve alarms
   - Reformat alarms
   - Identify alarms
   - Identify alarm implications
   - Archive alarms

4. Receive notification of a fault condition from alarms, from the Performance Management System (e.g., exceeding thresholds for PM triggers), or from a customer (via a trouble report or a Customer Network Management system).

5. Isolate Trouble
   - Heuristic (methodical and intuitive) analysis of alarms
   - Correlate alarms
   - Review correlation results
   - Determine source alarms

- Suppress related alarms
- Test to further isolate trouble and recommend further action for problem resolution.

6. Diagnose Fault
   - Analyze event log
   - Perform root cause analysis
   - Make an informed hypothesis concerning the source of the trouble
   - Determine condition severity
   - Suggest tests
   - Utilize remote test access and local diagnostic tools to test resources and provide multiple test points along the circuit path
   - Analyze test results
   - Suggest appropriate actions based upon diagnostic results

7. Resolve Trouble
   - Initiate the fault resolution process
   - Maintain connectivity to production applications
   - Automatically generate trouble ticket request
   - Track status of the trouble
   - Escalate as necessary
   - Send requests to:
   - Reconfiguration Management to reconfigure network or reroute traffic
   - Connection Management to order new services where required

8. Restore Service
   - Initiate full backup, recovery and restore procedures
   - Test the end-to-end connections to assure they work.
   - Clear alarms
   - Modify alarm status indicators
   - Update user interface
   - Mark inventories with failed resources and new routes, all nodes with pertinent state information, etc.
   - Notify all downstream OSs that reconfiguration has occurred.
   - Notify users of reconfiguration, reroute or restoral.
   - Verify restoral of service.
   - Close trouble log and archive

9. Restore Original Configuration

2.2.3 Fault Management Operations Supported

Only a subset of the requested Fault Management Functions are supported in this initial version of the model. The table below shows which Fault Management Operations are supported by this information model:

Fault Management Operations	Function Supported
1. Set Parameters	
* Establish criteria for reliability, availability and survivability. As defined in GR-2869-CORE Generic Requirements for Operations Based on the TMN Architecture:	Not Applicable
* Register facilities for monitoring	Yes
* Define alarm types to be screened	Yes
* Set up alarm/event thresholds and other event criteria	No
* Assign priorities to circuits/components/sessions	No
* Set up rule base	No
2. Monitor Network Status	
* Monitoring events and alarms	Yes
* Maintain log of events and alarms	Yes
* Status request of any component on the network (automated and manual)	No
* Collect/poll for prescreened system events or alarms	Yes
* Collect performance information	Yes
* Filter duplicate and informational messages	Yes
* Suppress recurring messages	Yes
* Suppress minor or informational alarms	Yes
* Archive events	Yes
* Continual data analysis and results review	Yes
* Generate event and alarm reports	Yes
3. Detect Network Deficiencies	
* Update the user interface (network topology)	Not Applicable
* Retrieve alarms	Yes
* Reformat alarms	Yes
* Identify alarms	Yes
* Identify alarm implications	Yes
* Archive alarms	Yes
4. Receive notification of a fault condition from alarms, from the Performance Management System (e.g., exceeding thresholds for PM triggers), or from a customer (via a trouble report or a Customer Network Management system).	Yes for fault conditions from alarms, no for other conditions
5. Isolate Trouble	
* Heuristic (methodical and intuitive) analysis of alarms	Yes
* Correlate alarms	Yes
* Review correlation results	Yes
* Determine source alarms	Yes
* Suppress related alarms	Yes
* Test to further isolate trouble and recommend further action for problem resolution.	Not Applicable
6. Diagnose Fault	
* Analyze event log	Yes

Fault Management Operations	Function Supported
* Perform root cause analysis	Yes
* Make an informed hypothesis concerning the source of the trouble	Yes
* Determine condition severity	Yes
* Suggest tests	No
* Utilize remote test access and local diagnostic tools to test resources and provide multiple test points along the circuit path	No
* Analyze test results	No
* Suggest appropriate actions based upon diagnostic results	No
7. Resolve Trouble	
* Initiate the fault resolution process	No
* Maintain connectivity to production applications	No
* Automatically generate trouble ticket request	No
* Track status of the trouble	No
* Escalate as necessary	No
* Send requests to:	No
* Reconfiguration Management to reconfigure network or reroute traffic	No
* Connection Management to order new services where required	No
8. Restore Service	
* Initiate full backup, recovery and restore procedures	No
* Test the end-to-end connections to assure they work.	No
* Clear alarms	No
* Modify alarm status indicators	No
* Update user interface	No
* Mark inventories with failed resources and new routes, all nodes with pertinent state information, etc.	
* Notify all downstream OSS that reconfiguration has occurred.	No
* Notify users of reconfiguration, reroute or restoral.	No
* Verify restoral of service.	No
* Close trouble log and archive	No
9. Restore Original Configuration	

2.3 Protection SOA

This section identifies high-level functional requirements for SONET protection for point to point connections in the network view. These functional requirements represent an initial refinement of SOA-level requirements and have been grouped in functional categories appropriate for the NMS/EMS interface.

2.3.1 Configuration Management

2.3.1.1 Provisioning

2.3.1.1.1 NE/Subnetwork Configuration

2.3.1.1.1.1 Configure line-switched subnetwork protection

The following configuration functions apply to APS and BLSR.

1. Provision protection parameters such as *working, protection roles to lines*. For the case of BLSR, it may be necessary to inhibit protection on an STS path basis (e.g., STS-3c carrying ATM traffic with ATM-layer path protection).

2. Select priority of lines for 1:n protection.

2.3.1.1.2 Inventory Notification and Query

2.3.1.1.2.1 Report/query current topology configurations

1. *Report/query linear APS current protection configuration.* The current configuration of protecting and protected lines is reported or made available.

2. *Report/query BLSR current protection configuration*. The current subnetwork configuration of protecting and protected lines on a BLSR ring is reported or made available.

3. *Report/query UPSR current protection configuration.*

2.3.1.1.2.2 Report/query NE/subnetwork resources

1. Report*/query NE protection resources*. The NE inventory of redundant equipment modules and line protection capabilities may be needed as part of Inventory Management.

2. *Report/query subnetwork protection resources*. Request subnetwork view of protection capabilities.

3. Report/query network capacity.

2.3.1.1.3 Subnetwork Connection Management

2.3.1.1.3.1 Design route

1. *Request diversified route within subnetwork*. Different scenarios arise depending on the split of functionality between EMS and NMS.

Select diverse-facility link connections on a link (i.e., supported by separate server layer trails).

Select a subnetwork connection routing that is diverse-routed as compared with an existing subnetwork connection within a subnetwork. Two levels of route diversity are defined in support of protected and highly protected end-to-end connections:

- No shared facilities
- No shared facilities or nodes

2.3.1.1.3.2 Configure SNC

1. *Set up protected SNC*

Set up point-to-point subnetwork connection with a requested protection level and effort level. The following protection levels are supported: highly protected, protected, unprotected, and preemptible. The following effort levels are supported: required exact, required minimum, and best effort.

Set up a bicast subnetwork connection with a requested protection level and effort level. The following protection levels are supported: highly protected, protected, and unprotected. The following effort levels are supported: required exact and required minimum.

Set up an exclusive merge subnetwork connection with a requested protection level and effort level. The following protection levels are supported: highly protected, protected, and unprotected. The following effort levels are supported: required exact and required minimum.

Set up exclusive composite subnetwork connection with a requested protection level and effort level. The following protection levels are supported: highly protected, protected, and unprotected. The following effort levels are supported: required exact and required minimum. (An exclusive composite connection provides a bi-directional configuration that combines bicast and exclusive merge connections

Set up a point-to-point subnetwork connection with a routing constraint against a specified subnetwork connection.

2. *Modify Protection of Subnetwork Connection*

Change the provisioned protection level (highly protected, protected, unprotected, preemptible) of a subnetwork connection in accordance with a requested effort level and without interrupting service. If the change cannot be effected within the constraints of the effort level, no change shall be made.

If a protection switching event occurs during the interval between receipt of the protection change request and completion of the action, the protection level shall revert to the initial level.

Modify a uni-directional subnetwork connection to become a bicast subnetwork connection. Input parameters include the identity of the subnetwork connection and the identity of the second terminating endpoint of the multipoint connection.

Modify a uni-directional subnetwork connection to become an exclusive merge subnetwork connection. Input parameters include the identity of the subnetwork connection and the identity of the second originating endpoint of the multipoint connection.

Modify a bidirectional subnetwork connection to become an exclusive composite subnetwork connection. Input parameters include the identity of the subnetwork connection and the identity of the additional endpoint of the multipoint connection.

Modify a bicast subnetwork connection to become a uni-directional subnetwork connection. Input parameters include the identity of the multipoint connection and the identity of the terminating endpoint to be deleted.

Modify an exclusive merge subnetwork connection to become a uni-directional subnetwork connection. Input parameters include the identity of the multipoint connection and the identity of the originating endpoint to be deleted.

Modify an exclusive composite subnetwork connection to become a bidirectional point-to-point subnetwork connection. Input parameters include the identity of the multipoint connection and the identity of the endpoint to be deleted.

3. *Delete SNCs*

Delete a subnetwork connection.

2.3.1.1.3.3 Query

Given a subnetwork connection one can query for its working or protection subnetwork connection counterpart.

Query the provisioned protection level of a subnetwork connection.

Query the operational protection level of a subnetwork connection. The operational protection level indicates the effective protection level at a given time. The operational protection level may differ from the provisioned protection level due to network resource impairments or other effects.

Query the mode of a protected subnetwork connection (revertive, non-revertive)

Query the active endpoint on an exclusive merge or exclusive composite multipoint connection.

2.3.1.2 Status and Control

Changes in the NE view which affect the Network View must be reflected in the Network View Model.

2.3.1.2.1 NE/Subnetwork Status and Control

2.3.1.2.1.1 Report subnetwork state changes

Changes in protection status of a protected transport entity must be reported.

For a protected SNC it shall be possible to report/retrieve the following information:

- Trails or SNCs which support the protected SNC.
- Operational state of the trails or SNC which support the protected SNC.
- Routes of the trails or SNC which support the protected SNC.
- Whether extra traffic is being supported on supporting trail or SNC, and if so, the identity of the preemptible SNC.

Use standard protection switching mechanisms to report protection state changes.

Retrieve the protection status (not-switched, switched) of an APS or BLSR system. For a BLSR in a switched state, the status should also indicate whether all services are supported or not all services are supported.

2.3.1.2.1.2 Control protection mechanisms

For a protected SNC provided via a UPSR or logical ring, it shall be possible to:

- Allow/Prohibit switching to the alternate supporting SNC or trail.
- Force protected SNC traffic onto a supporting SNC or trail.
- Release a force onto a supporting SNC or trail
- Report/retrieve the current status of the above actions

Use standard protection switching mechanisms to control protection switching systems. This includes activiate and deactivate protection.

2.3.1.2.1.3 Query protection status

The protection status of a protected transport entity must be retrievable.
Use standard protection switching mechanisms to query protection status.

2.3.2 Fault Management

2.3.2.1 Testing

2.3.2.1.1 Configure test access

1. *Set up test access*

2.3.2.1.2 Control test

1. *Insert errored signal on protected line*
2. *Insert errored signal on protected path*

2.3.3 Protection Operations Supported

This section shows which protection functions are supported by this model. Functions supported by the Network View model have a Yes under the column headed by NV. Functions supported by the Network Element View model have a yes under the column headed by NEV. Functions which are not supported have both columns blank.

Protection Functions	NV	NEV

Protection Functions	NV	NEV
PPS/BLSR working/protection roles		Yes
Inhibit protection on an STS basis in BLSR		Yes
Report/Query APS/BLSR protection configuration		Yes
Report/Query UPSR protection configuration	Yes	
Report/Query NE protection resources		Yes
Report/Query subnetwork protection resources		Yes
Report network capacity	Yes*	
Request diversified route within subnetwork	Yes	
Setup protected subnetwork connection	Yes	
Modify protection of subnetwork connection	Yes	
Delete subnetwork connection	Yes	
Report subnetwork state changes	Yes	
Control protection mechanisms		Yes
Query protection status	Yes	
Setup test access		
Control test		

*Can be calculated from information provided in the model.

3. Model Verification Scenarios

Scenarios have been developed to validate and refine the information model. Scenarios demonstrate how features of the model are used in the context of an applications process. These scenarios do not prescribe how the model must be used, but show how the model can be used.

3.1 Connection Management Scenarios

The scenarios in this section are Network Creation and Network Connection Configuration. The scenarios include both the network view only and the network view with the NE view. Figure 3-1, shown below, illustrates the assumed network architecture for these scenarios.

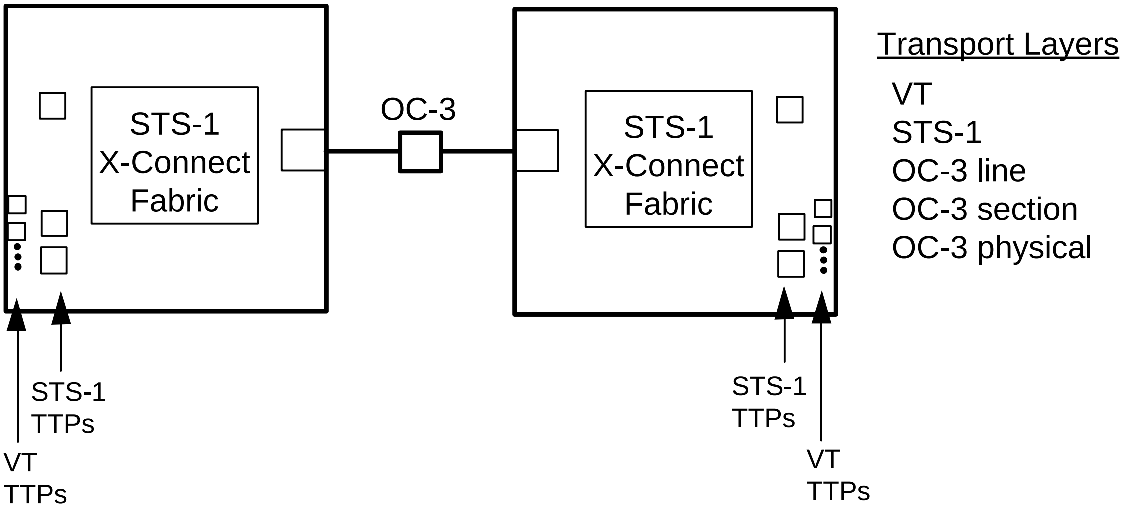


Figure 3-1 Example SONET Network Architecture

Figure 3-1 shows two SONET OC-3 ADMs connected together, through a regenerator, in a point-to-point configuration. These ADMs are configured as terminal multiplexers and they only provide STS-1 cross-connection capability. Not shown are potential DS1 or DS3 drop cards. This network architecture, while extremely simple, allows us to validate the information model without unnecessary complication.

3.1.1 Network Creation (and autodiscovery)

This function involves populating the EMS Management Information Base (MIB) with managed objects representing network resources and reporting the creation of these objects to the NMS. First, layer network resources are created. Then, subnetwork topologies are created. The creation of subnetwork topologies can be a cooperative process between the EMS, providing default views, and the NMS, requesting NMS specific views. This process is discussed below.

3.1.1.1 Layer Network Resource Creation

3.1.1.1.1 Network View Only

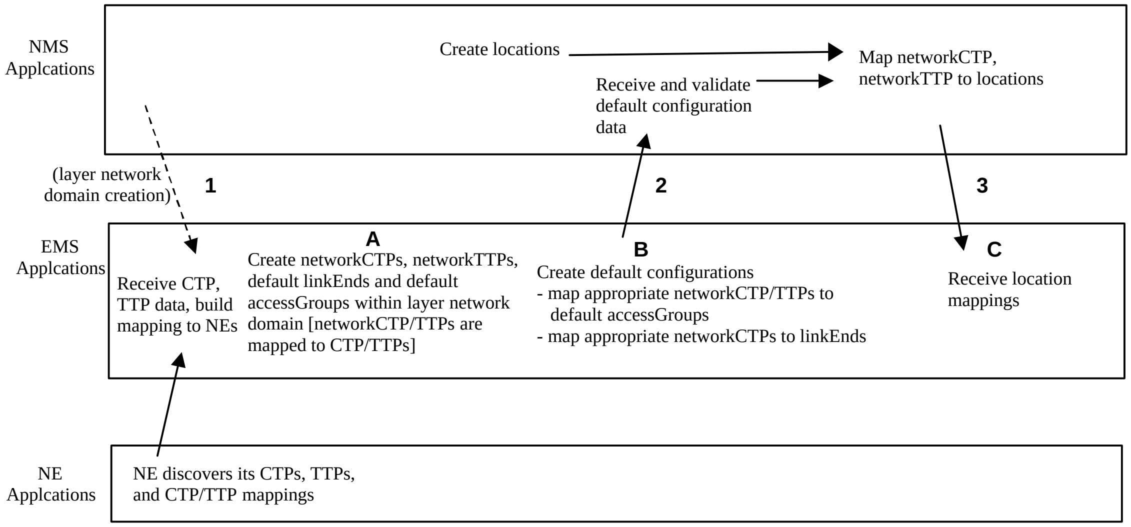

NMS
Applcations
Create locations
Map networkCTP, networkTTP to locations
Receive and validate default configuration data
(layer network domain creation)
1
2
3
EMS
Applcations
A
B
C
Receive CTP, TTP data, build mapping to NEs
Create networkCTPs, networkTTPs, default linkEnds and default accessGroups within layer network domain [networkCTP/TTPs are mapped to CTP/TTPs]
Create default configurations
- map appropriate networkCTP/TTPs to default accessGroups
- map appropriate networkCTPs to linkEnds
Receive location mappings
NE
Applcations
NE discovers its CTPs, TTPs, and CTP/TTP mappings

Figure 3-2 Processes and Message Flows - Build Inventory and Map to Nodal/Geographic Objects (Network View Only)

Figure 3-2 shows the applications processes and message flows used to create layer network resources for the network view. The process starts with an inventory download from the NMS and/or an autodiscovered upload from the NEs and ends with network termination points (trail and connection) which reflect NE terminations points and are mapped to access groups, NEs, and geographic locations. The following message flows illustrate exchanges of information between the NMS and the EMS:

- **Message Flow 1 (if not autodiscovered)** - This flow represents inventory download. Object instance(s) of layerNetworkDomain will be created in the MIB as a result of the NMS sending the EMS M-CREATE message(s). The networkCTP and networkTTP object instances are created by the EMS via autodiscovery.
- **Message Flow 2** - This flow represents notifications from the EMS informing the NMS of object creations and attribute value changes as a result of termination point discovery (layerNetworkDomain, accessGroup, networkCTP, and networkTTP) and termination point mappings to accessGroups and linkEnds.
- **Message Flow 3** - This flow represents the NMS mapping termination points (M-SET on locationName attributes) to geographic locations.

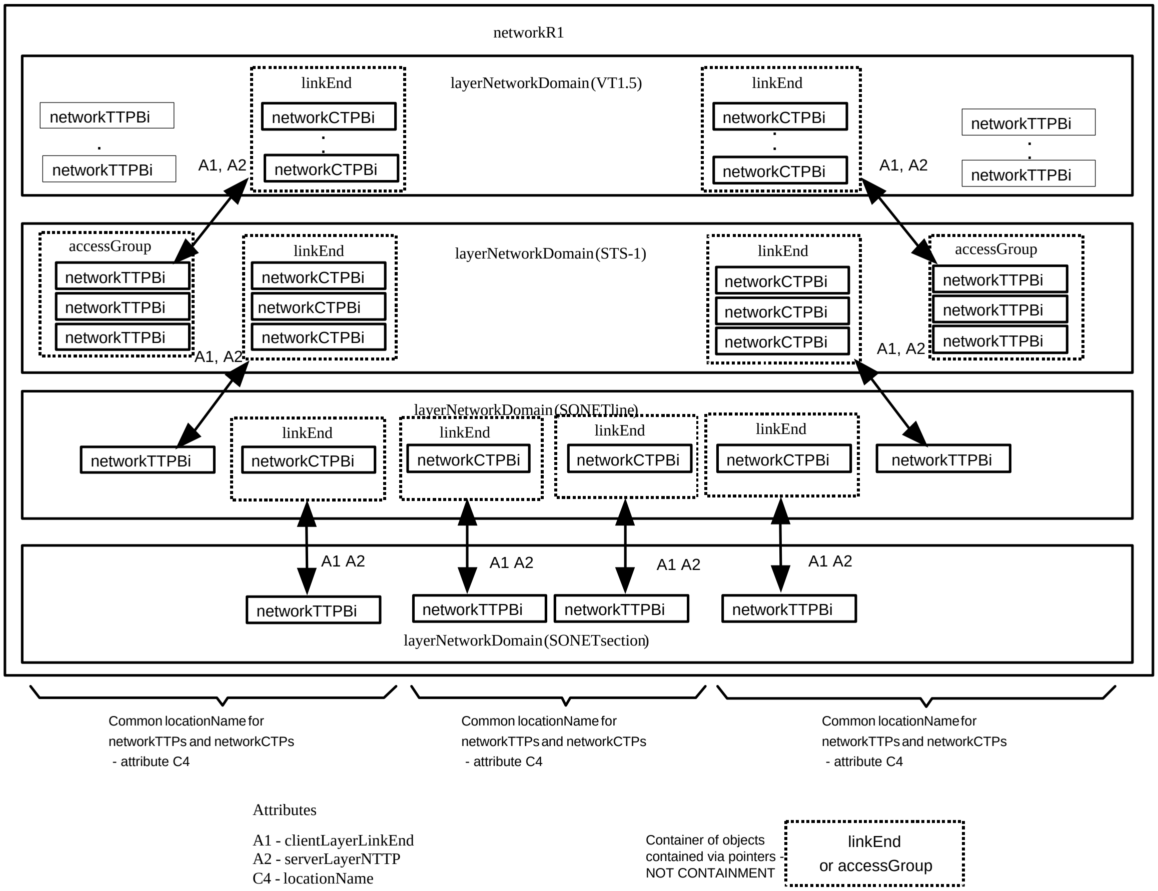


Figure 3-3 Object Instances for Layer Network Resource Creation

Figure 3-3 shows the managed objects created during the layer network resource creation process. Also shown are the key relationships between managed objects, in the form of name bindings and pointer attributes. Name bindings are represented by solid-line boxes contained within solid-line boxes (e.g., layerNetworkDomain is contained within networkR1). Pointer relationships are shown by arrows and by dashed boxes. Finally, the “A”, “B”, and “C” references refer to “A”, “B”, and “C” in the previous figure. These references represent a sequence of object creation from “A” objects to “B” objects to “C” objects (and associated relationships). The optical layer objects have not been shown in Figure 3-3 to simplify the figure.

3.1.1.1.2 Network View + NE View

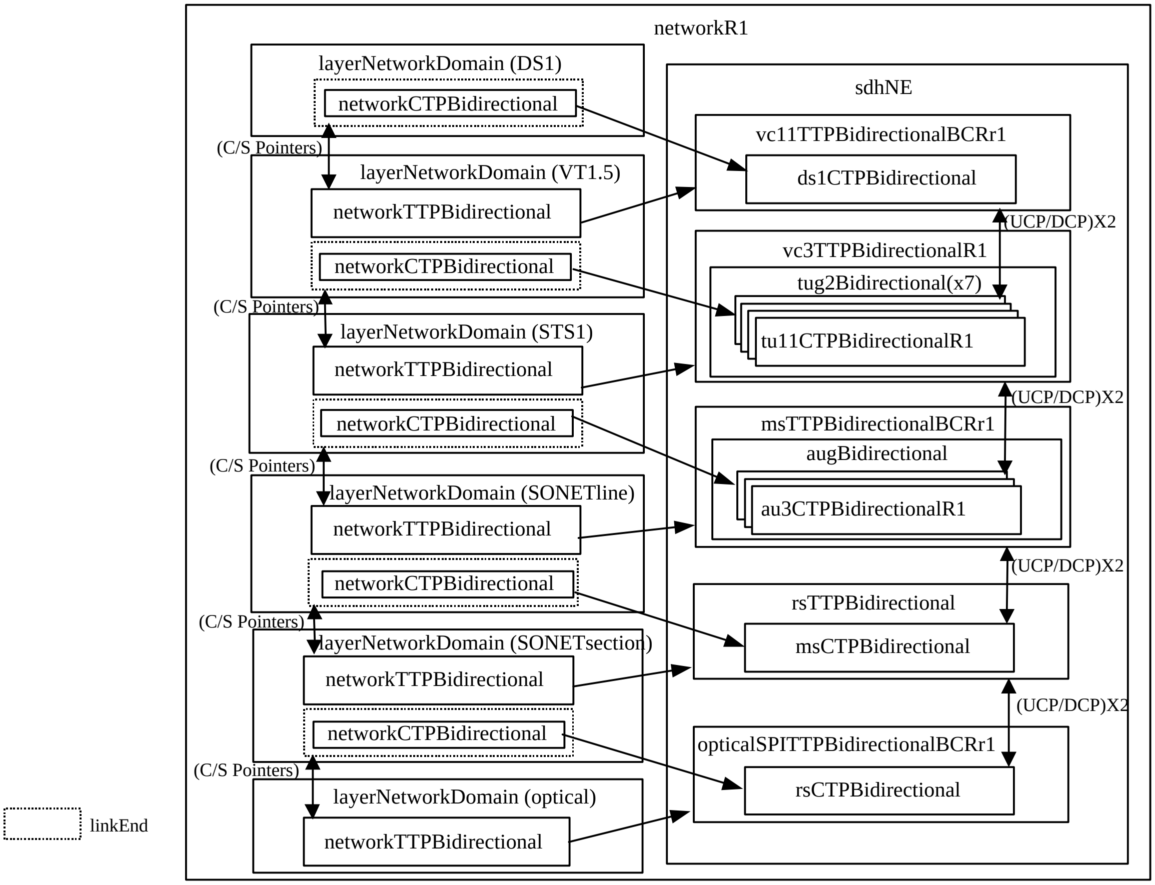


Figure 3-4 Network and NE View Termination Points Relationships

Figure 3-4 shows both network and NE view termination point managed objects for the SONET network when both views are used. This figure illustrates the network termination points that are abstracting the NE termination points. If only the network view is provided by the EMS, the NE view termination points will not be visible to the NMS. If the NE view is provided in addition to the network view, the network termination points will point to the NE View termination points. The linkEnd object is used to show the associations between the CTPs, layerNetworkDomain, and subnetworks. Also, the accessGroup object is used to show the associations between the corresponding CTPs, TTPs, layerNetworkDomain, and subnetworks. The UCP/DCP are upstream and downstream connectivity pointers. The C/S Pointers are the clientLayerLinkEnd and serverLayerNTTP attributes. The one way arrows show the network view TPs that abstract the NE view TPs. All layers from optical to DS1 are shown, unlike the object instance figures which show only the SONET section to VT1.5 layers.

3.1.1.2 Subnetwork Topology Creation

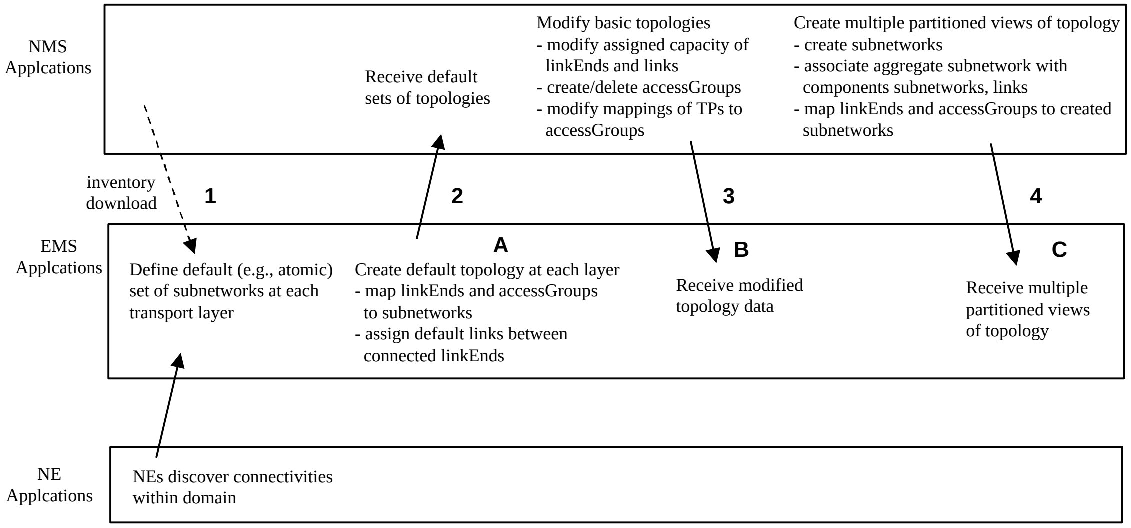

NMS Applcations
EMS Applcations
NE Applcations
Receive default sets of topologies
Modify basic topologies
- modify assigned capacity of linkEnds and links
- create/delete accessGroups
- modify mappings of TPs to accessGroups
Create multiple partitioned views of topology
- create subnetworks
- associate aggregate subnetwork with components subnetworks, links
- map linkEnds and accessGroups to created subnetworks
inventory download
1
2
3
4
A
B
C
Define default (e.g., atomic) set of subnetworks at each transport layer
Create default topology at each layer
- map linkEnds and accessGroups to subnetworks
- assign default links between connected linkEnds
Receive modified topology data
Receive multiple partitioned views of topology
NEs discover connectivities within domain

Figure 3-5 Processes and Message Flows - Build Subnetworks, Links, and Partitioned Views

Figure 3-5 shows the applications processes and message flows used to create subnetwork topologies. The process starts with an inventory download from the NMS and/or an autodiscovered upload from the NEs and ends with partitioned views of subnetworks and links. The following message flows illustrate exchanges of information between the NMS and the EMS:

- **Message Flow 1 (if not autodiscovered)** - This flow represents inventory download. Object instances of subnetwork, linkEnd, and link will be created in the MIB as a result of the NMS sending the EMS M-CREATE messages.
- **Message Flow 2** - This flow represents notifications from the EMS informing the NMS of object creations and attribute value changes as a result of subnetwork topology discovery (subnetwork, linkEnd, and link), mappings of subnetwork to linkEnd and accessGroup, and mappings of linkEnd and accessGroup to network or NE termination points.
- **Message Flow 3** - This flow represents the NMS modifying linkEnds, links, and accessGroups and creating link object instances (M-CREATE) and relating these to existing linkEnds (M-SET on linkPointer).
- **Message Flow 4** - This flow represents the NMS creating partitioned views of subnetworks by creating subnetworks (M-CREATE) and relating these to existing managed objects (M-SET).

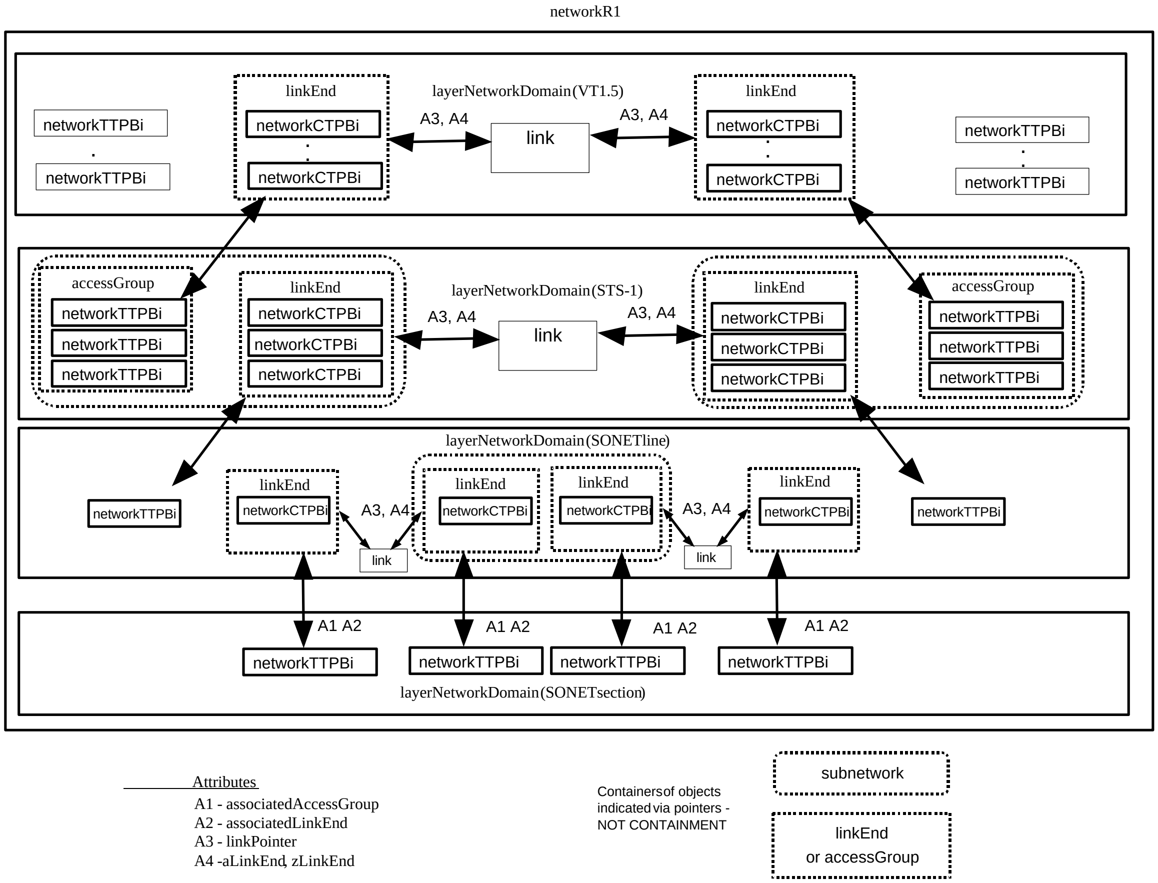


Figure 3-6 Object Instances for Default Subnetwork Topology Creation

Figure 3-6 shows the managed objects created during the default subnetwork topology creation process. Also shown are the key relationships between managed objects, in the form of name bindings and pointer attributes. Name bindings are represented by solid-line boxes contained within solid-line boxes (e.g., layerNetworkDomain is contained within networkR1). Pointer relationships are shown by arrows and by dashed boxes. The “A” label refers to step A in the previous figure involving creation of default topologies Figure 3-7 shows the relationships among managed objects resulting from partitioning actions (message flow 4) in the subnetwork topology creation process. The “A” and “C” labels refer to steps A and C in Figure 3-5. The optical layer objects have not been shown in Figure 3-6 to simplify the figure.

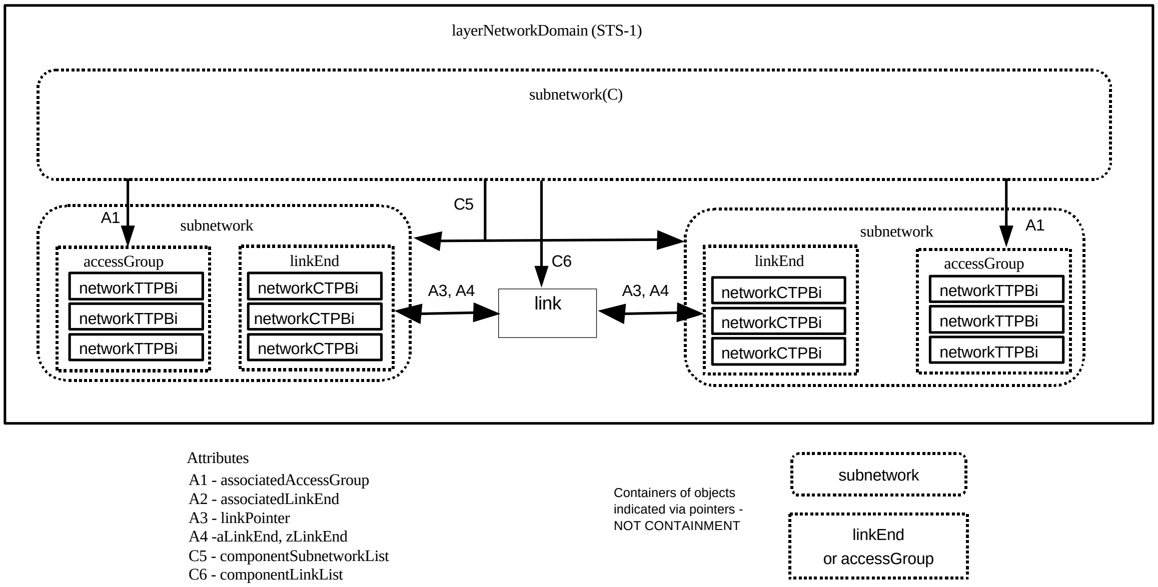


Figure 3-7 Object Instances for Partitioned Subnetwork Topology Creation

3.1.2 Network Connection Configuration

This function involves the set-up and tear-down of connections to support customer services. This scenario illustrates the processes required to set-up connections. First, server layer trails and client layer link connections are created. Then, subnetwork connections are created.

3.1.2.1 Layer Network Connection Configuration

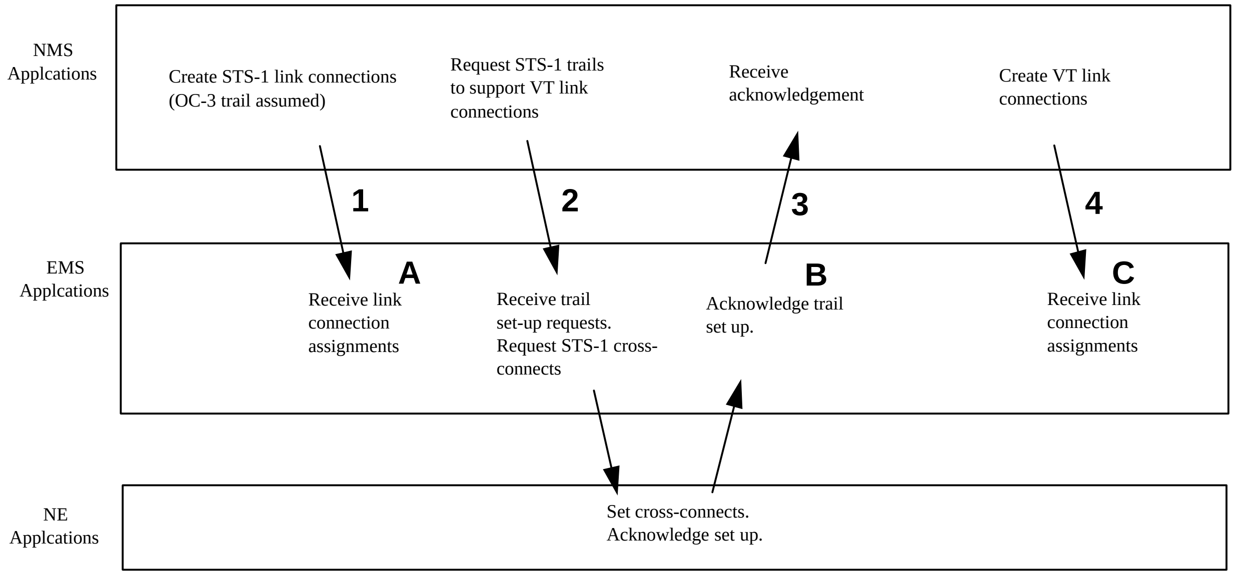
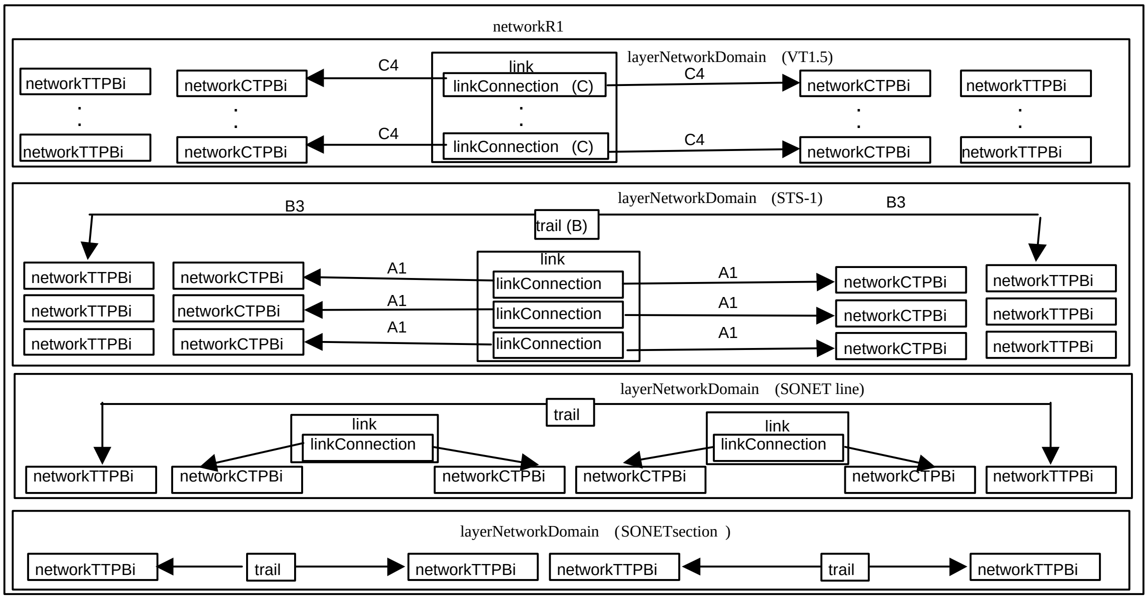

NMS
Applcations
Create STS-1 link connections (OC-3 trail assumed)
Request STS-1 trails to support VT link connections
Receive acknowledgement
Create VT link connections
1
2
3
4
EMS
Applcations
A
Receive link connection assignments
Receive trail set-up requests. Request STS-1 cross-connects
B
Acknowledge trail set up.
C
Receive link connection assignments
NE
Applcations
Set cross-connects.
Acknowledge set up.

Figure 3-8 Processes and Message Flows: Create Link Connections, Supporting Trails in Each Layer

Figure 3-8 shows the applications processes and message flows used to configure layer network connections. The process starts with a request for STS-1 link connections and ends with the creation of VT link connections. The following message flows illustrate exchanges of information between the NMS and the EMS:

- **Message Flow 1** - This flow represents requests for STS-1 link connections. The NMS sends the EMS an M-ACTION on a link object instance requesting a setupLinkConnection. The EMS will respond to this request (not shown) with the identification of the created linkConnections or a failure notification.
- **Message Flow 2** - This flow represents requests for STS-1 trails to support VT link connections. The NMS sends the EMS an M-ACTION on a layer network domain object instance requesting a setupTrail.
- **Message Flow 3** - The EMS will respond to the trail request with the identification of the created trails or a failure notification.
- **Message Flow 4** - This flow represents requests for VT link connections. The NMS sends the EMS an M-ACTION on a link object instance requesting a setupLinkConnection. The EMS will respond to this request (not shown) with the identification of the created linkConnections or a failure notification.


networkR1
layerNetworkDomain (VT1.5)
networkTTPBi
networkCTPBi
link
linkConnection (C)
C4
layerNetworkDomain (STS-1)
B3
trail (B)
linkConnection
A1
layerNetworkDomain (SONET line)
trail
layerNetworkDomain (SONETsection)

<u>Pointer Attributes</u>
A1 - aEndCTP or zEndCTP
B3 - aEndTTPs or zEndTTPs
C4 - aEndCTP or zEndCTP

Figure 3-9 Object Instances for Layer Network Connection Configuration

Figure 3-9 shows the managed objects created during the layer network connection configuration process. Also shown are the key relationships between managed objects, in the form of name bindings and pointer attributes. Name bindings are represented by solid-line boxes contained within solid-line boxes (e.g., layerNetworkDomain is contained within networkR1). Pointer relationships are shown by arrows and by dashed boxes. Finally, the “A”, “B”, and “C” references refer to “A”, “B”, and “C” in the previous figure. These references represent a sequence of object creations from “A” objects to “B” objects to “C” objects (and associated relationships). The optical layer objects have not been shown in Figure 3-9 to simplify the figure.

3.1.2.2 Subnetwork Connection Configuration

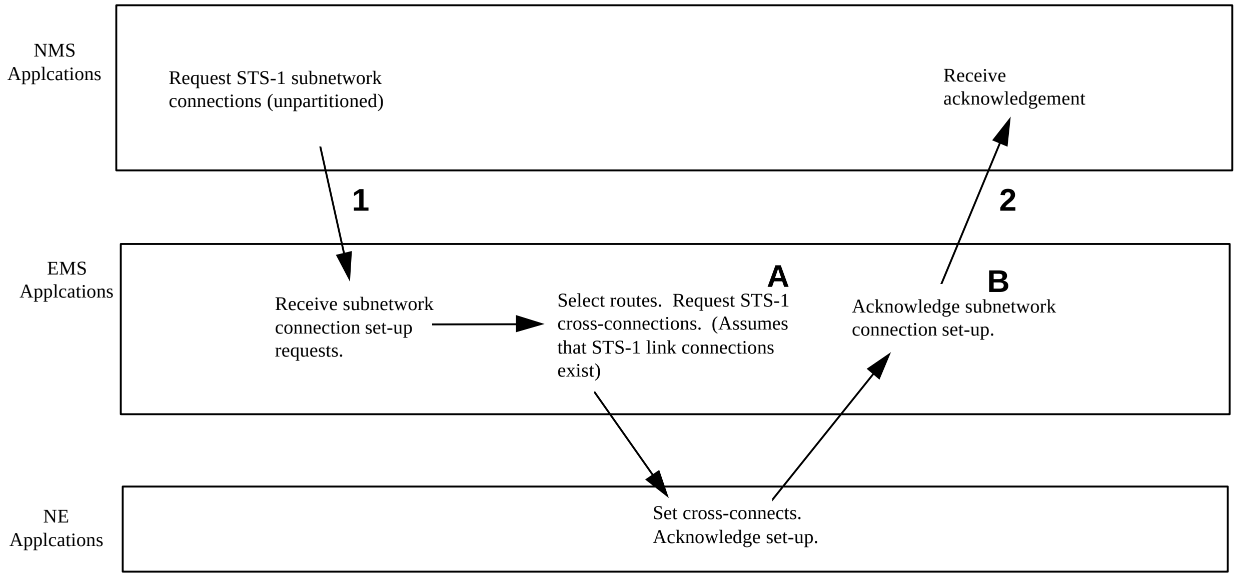

NMS
Applcations
Request STS-1 subnetwork connections (unpartitioned)
Receive acknowledgement
1
2
EMS
Applcations
A
B
Receive subnetwork connection set-up requests.
Select routes. Request STS-1 cross-connections. (Assumes that STS-1 link connections exist)
Acknowledge subnetwork connection set-up.
NE
Applcations
Set cross-connects. Acknowledge set-up.

Figure 3-10 Processes and Message Flows: Set-up Subnetwork Connections

Figure 3-10 shows the applications processes and message flows used to configure subnetwork connections. The process starts with a request for STS-1 subnetwork connections (unpartitioned) and ends with acknowledgement of the creation of subnetwork connections. The following message flows illustrate exchanges of information between the NMS and the EMS:

- **Message Flow 1** - This flow represents requests for STS-1 subnetwork connections. The NMS sends the EMS an M-ACTION on a subnetwork object instance requesting a setupSNC.
- **Message Flow 2** - The EMS will respond to the set-up subnetwork connection request with the identification of the created subnetwork connections or a failure notification.

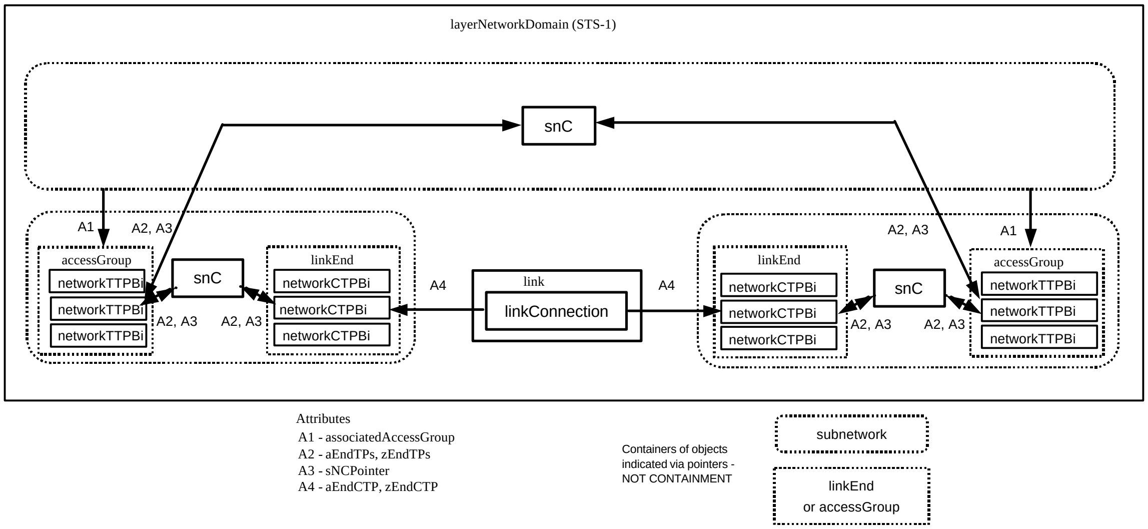


Figure 3-11 Object Instances for STS-1 Subnetwork Connection Configuration

Figure 3-11 shows the managed objects created during the STS-1 subnetwork connection configuration process for both the partitioned and aggregated subnetwork cases. Similar diagrams can be drawn for other network layers. Also shown are the key relationships between managed objects, in the form of name bindings and pointer attributes. Name bindings are represented by solid-line boxes contained within solid-line boxes (e.g., layerNetworkDomain is contained within networkR1). Pointer relationships are shown by arrows and by dashed boxes. Finally, the “A” and “B” references refer to “A” in the previous figure. These references represent a sequence of object creations between “A” objects (and associated relationships). Note: Link connections, though necessary, are not shown in this figure.

3.2 Connection Configuration with Status Reporting Scenarios

This section gives an information model solution for providing Connection Configuration with Status Reporting in support of the following requirements:

R1 - The NMS requires progress information during the setup process.

R2 - On Failure:

- The EMS will report completed portions (e.g., component SNCs/link connections).
- The NMS will have the option to:
  - a) completely undo. EMS will report portions it was not able to undo. As an option, the NML may specify the "undo" at setup time.
  - b) keep partial setup

R3 - The NMS will have the capability to cancel while in progress. In response to a cancel request, the EMS will completely undo. It will also report portions "unable to undo".

This section gives scenarios and functional descriptions of managed objects classes for realizing the application. The approach uses a new object class modelled after the “Current Alarm Summary Control” object class [Q.821] would emit periodic reports via stimulation by a “Scanner” object class. The scanner object class was found to contain many conditional packages not needed for this application so the “Management Operations Schedule” object has been selected in lieu of the scanner object.

3.2.1 Connection Configuration with Status Reporting

Connection Configuration with Status Reporting allows an agent (EMS) to report on the status of in-progress subnetwork connections that have been requested by a manager (NMS). The functional components of the Connection Configuration service are illustrated in Figure 3-12. The agent’s Connection Configuration function receives connection requests, confirms connection completions, and provides data for status reporting. Status reporting for a given subnetwork connection object instance begins when the setup/release request is received and ceases when the agent completes all steps in the operation (e.g., sets up NE cross connections). A Connection Configuration Status Control function contains the criteria for reporting and generates the status reports.

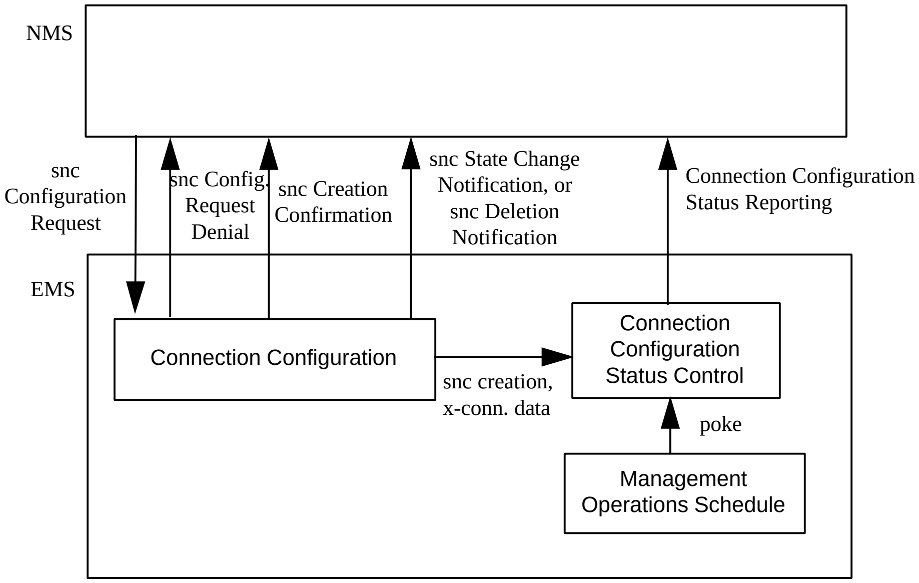

NMS
snc Configuration Request
snc Config. Request Denial
snc Creation Confirmation
snc State Change Notification, or snc Deletion Notification
Connection Configuration Status Reporting
EMS
Connection Configuration
snc creation, x-conn. data
Connection Configuration Status Control
poke
Management Operations Schedule

Figure 3-12 Connection Configuration Status Reporting Scenario

A report contains the identity of each subnetwork connection under setup, an indicator of whether the operation is setup or release, and a percentage measure of the progress achieved in completing the operation. The percentage indicates the number of NE cross-connects (completed or released) out of the total needed the complete the operation. Status reporting is done periodically at close intervals (on the order of seconds) as directed (via a "poke") by a Management Operations Schedule function.

An example of the logic flows in a scenario for Connection Configuration with status reporting is shown in Figure 3-13. This illustrates the NMS/EMS interactions in Figure 3-12 in somewhat more detail and indicates the logical context for each request/response and notification.

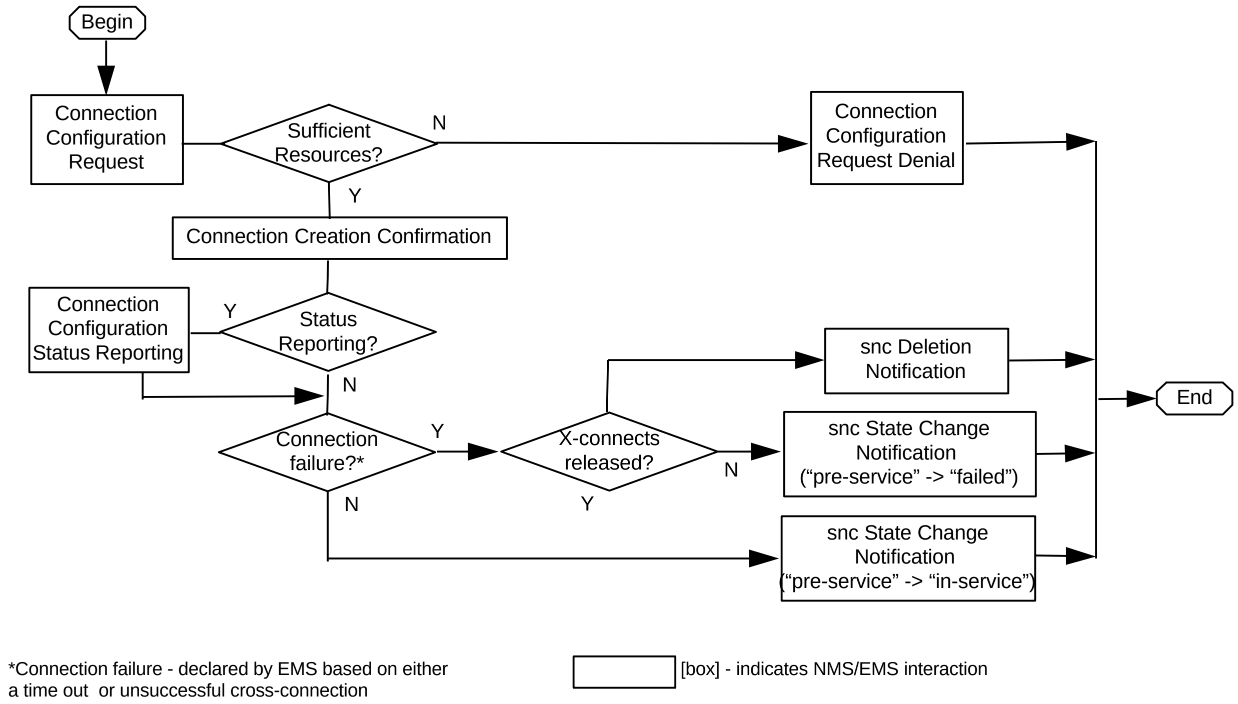

Begin
Connection Configuration Request
Sufficient Resources?
N
Y
Connection Configuration Request Denial
Connection Creation Confirmation
Connection Configuration Status Reporting
Y
Status Reporting?
N
snc Deletion Notification
End
Connection failure?*
Y
X-connects released?
N
Y
snc State Change Notification ("pre-service" -> "failed")
N
snc State Change Notification ("pre-service" -> "in-service")
*Connection failure - declared by EMS based on either a time out or unsuccessful cross-connection
[box] - indicates NMS/EMS interaction

Figure 3-13 EMS Application Logic Example

The names of states for an snc are in quotes to indicate that work is needed to define these states in detail.

3.2.2 Functional Definitions of Object Classes

3.2.2.1 Connection Configuration Status Control Object Class

This object class is a somewhat modified form of the currentAlarmSummaryControl object class [Q.821]. The Connection Configuration Status Control object class is a class of support objects that enables the generation of reports on the status of subnetwork connections that have been requested by a manager and are in the process of being established or released through NE cross connections. An object is included in this summary report if:

- the object is a subnetwork connection object instance, and
- the manager requests reporting

An object will be removed from the list when it is setup, released, or failed.

The semantics of associated attributes are as follows:

- Connection Configuration Status Control Id - distinguished value can be used as a Relative Distinguished Name (RDN) for instance of object class.
- Object List - identifies a list of objects to be included in the connection configuration summary report.

The semantics of the associated notification is as follows:

- Connection Configuration Summary Report - identifies subnetwork connection object instances currently being setup or released and gives the current value of the percentage of associated cross-connects that have been successfully established or released.

The semantics of an associated conditional package is as follows:

- Empty List Suppression - suppresses notifications when the object list is empty.

3.2.2.2 Management Operations Schedule [Q.821]

The Management Operations Schedule object class is a class of support objects that provide the ability to schedule a management service to occur periodically. The period is specified by an Interval, with the first occurrence of the service (coinciding with the start of the first interval) specified as the Begin Time. The end of the time span during which the service can occur is defined by the End Time.

The objects that will supply the service are defined by the Affected Object Class and Affected Object Instances (e.g. the Current Alarm Summary Control object when providing the Current Alarm Summary Reporting Service). The Destination Address specifies the destination of the service. The Administrative State is used to allow/inhibit the operation of the schedule. The Operational State describes whether the object is capable of performing its function(s).

This object class is a subclass of the Top object class.

The semantics of associated attributes are as follows:

a) Administrative State

   The semantics of the Administrative State attribute type is described in X.731.

b) Affected Object Class

   The Affected Object Class attribute type identifies the object class affected by a scheduled management operation.

c) Affected Object Instances

   The Affected Object Instances attribute type identifies the object instances on which a scheduled management operation will be performed.

d) Begin Time

   The Begin Time attribute type indicates the starting time for a management function.

e) Destination Address

The Destination Address attribute type identifies the destination to which selected event reports will be sent. The Destination Address may be an application entity title or address group. If no Destination Address is specified in the request, the address of the invoker is assumed.

f) End Time

The End Time attribute type indicates the termination time of a management function.

g) Interval

The Interval attribute type indicates the time between occurrences of a given activity described by an instance of the Management Operations Schedule object class. The interval can be specified in seconds, minutes, hours, or days.

h) Operational State

The semantics of the Operational State attribute type is described in X.731.

I) Schedule Id

The Schedule Id is an attribute type whose distinguished value can be used as an RDN when naming an instance of the Management Operations Schedule object class.

3.3 Fault Management Scenarios

This section describes general characteristics of SONET Fault Management scenarios that originate with the detection of a failure(s) by network elements. Examples of root cause for the failures in these scenarios include fiber cuts and equipment failures. Other scenarios for fault management such as response to a customer reported fault may be handled in future versions of the model.

The GR-2869 fault management scenario for “Network Detected Trouble” [GR-2869, Section 6.7.2] describes a set of basic functions and flows between functions. Figure 3-14 shows a portion of the functional flow for this scenario focussing on the subset of EMS functions applicable to a network-detected fault scenario such as a fiber cut. Functions in the Business Management Layer (BML) are omitted for simplicity.

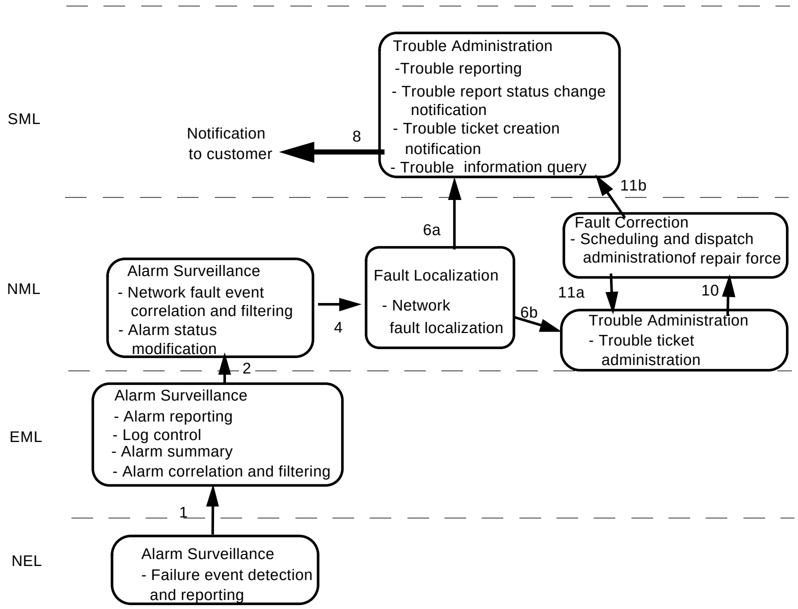

SML
NML
EML
NEL
Trouble Administration
-Trouble reporting
- Trouble report status change notification
- Trouble ticket creation notification
- Trouble information query
Notification to customer
8
11b
6a
Fault Correction
- Scheduling and dispatch administration of repair force
Alarm Surveillance
- Network fault event correlation and filtering
- Alarm status modification
4
Fault Localization
- Network fault localization
6b
11a
10
Trouble Administration
- Trouble ticket administration
2
Alarm Surveillance
- Alarm reporting
- Log control
- Alarm summary
- Alarm correlation and filtering
1
Alarm Surveillance
- Failure event detection and reporting

Figure 3-14 Network Detected Trouble scenario from GR-2869

Failure events detected in the Network Element Layer (NEL) are passed to Element Management Layer (EML) applications functions belonging to the Alarm Surveillance group. EML surveillance functions for SONET have been described in SR-2671 (root cause alarm analysis) and SR-2672 (alarm filtering). Many of the concepts used for our fault management model are derived from SR-2671.

3.3.1 Assumptions

1. The EMS will likely have knowledge of detailed operating characteristics and behaviors of the subnetwork/NEs that will not be visible to an NMS. This will enable the EMS to provide unique management capabilities related to:

- algorithms for root cause failure determination
- protection and reconfiguration mechanisms
- affected services
- states/status of transmission resources

Therefore, even though the NMS has visibility on the subnetwork connections/trails within an EMS's domain, the EMS may contain additional, useful knowledge.

2. The capability of the EMS to perform root cause analysis will help to reduce traffic on the DCN between the EMS and NMS.

3. The EMS will not have visibility on higher layer connections within a subnetwork connection unless the higher layer connections terminate within the EMS's domain.

4. The NMS can query the EMS for the list of current affected services and current alarms in the subnetwork.

5. There may be more than one fault event detected due to a single failure mechanism.

The EML network detected fault management application involves analysis functions, supporting data and supporting data acquisition, and results reporting functions. The following sections address EML application functions and interface data.

3.3.2 EMS Application Functions

A generic scenario for the network detected faults EMS application involves six functional steps:

1. Receive, terminate, log, and understand autonomous messages received from NEs within the subnetwork being managed.

2. Analyze the information received using supporting data (equipment hierarchy, topology, and transmission resource assignment) and knowledge of the current status for the subnetwork to filter alarms and report the filtered alarm information.

3. Analyze the information received using supporting data (equipment hierarchy, topology, and transmission resource assignment) and knowledge of the current status for the subnetwork to recognize and identify the root cause of the received message flow.

4. Analyze the information received using the supporting data and knowledge of the current status of the subnetwork to characterize the severity of the root cause problem (e.g., Service-affecting (SA) vs. Non-service-affecting (NSA), Critical/Major/Minor).

5. Analyze the information received using the supporting data and knowledge of the current status of the subnetwork to determine the impact of the root cause problem (i.e., identify affected transmission resources).

6. Report the root cause information along with the associated severity and services impact.

In summary alarms will be received from the NEs, logged at the EMS, and fed into the alarm analysis routine. The analysis routine will create records in the RCAA log and issue RCAA notifications to be sent to an Event Forwarding Discriminator (EFD). The EFD will filter the notifications and send the resulting Event Reports to the NMS. Figure 3-15, which is a variant of a picture from SR 2671, shows the process.

It is possible that the alarm analysis routine cannot find a root cause. In this case the alarm analysis routine will place the information in the uncorrelated NEL alarms in RCAA alarm records.

The EMS may be configured to pass high priority alarms to the NMS (via a specific EFD). This will allow the NMS to be immediately informed of high priority alarms even though the Alarm Analysis Routine takes a long time.

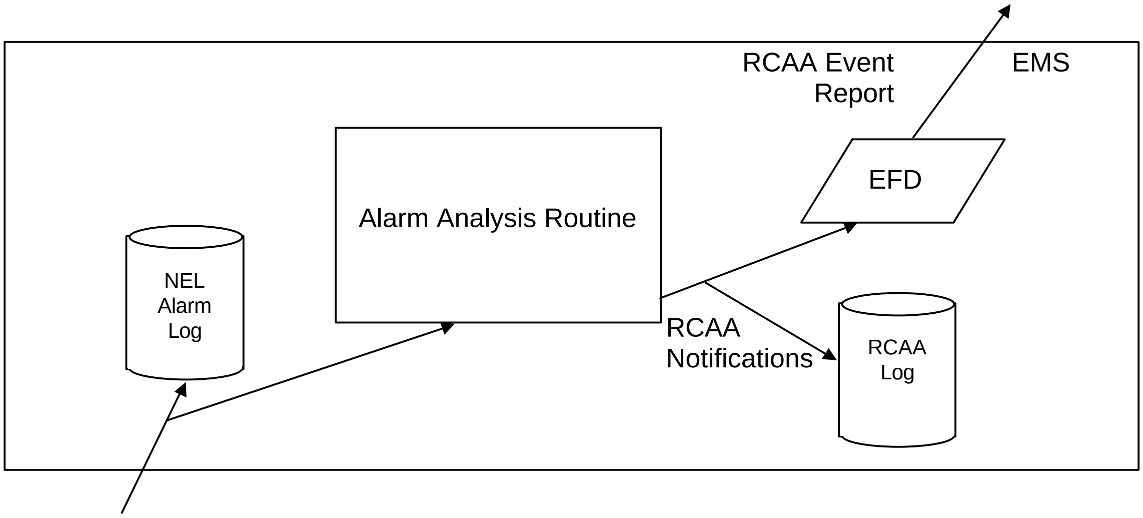


Figure 3-15 Alarm and RCAA Message Flow

The following explain the objects and notifications used to model Fault Management:

Log object from X.721 and X.735 is used to store incoming event reports and local system notifications. The NEL Alarm Log and the RCAA Log will be instances of the Log object class.

Records in the NEL Alarm Log will be instances of the alarmRecord class defined in X.721 and will contain information about NEL Alarm Event Reports from the NEs.

A RCAA Record is a subclass of the alarmRecord object class. It will contain information about the root cause of alarms.

An Alarm Analysis Routine object is used as the source of RCAA notifications.

The notification is sent to an EFD. EFDs are defined in X.721 and X.734 and are used to define the conditions that shall be satisfied by a potential event report before that event report is forwarded to a particular destination.

If the notification satisfies the conditions of the EFD an RCAA Event Report is sent from the EMS to the NMS to report the results of the Alarm Analysis Routine.

3.3.3 Requirements for Fault Management for NEL Detected Faults

Requirements were derived from the scenario defined in the previous section. The following requirements are supported by the model presented in this document.

1. The EMS shall log alarms and events from NEs including TCAs.

2. The EMS shall filter via Intelligent Alarm Filtering and/or Root Cause Alarm Analysis alarms/events/TCAs from the NE and pass the results to the NMS within a discrete period of time.

3. Results passed to NMS (as defined in 2) shall include a reference to the set of NEL alarm events that have been resolved within that message.

4. The EMS shall maintain a log of results passed to the NMS (as defined in 2), retrievable by the NMS.

5. Subsequent analysis messages can follow those messages defined in 2. These messages shall also be passed to the NMS and tagged to the original.

6. The NMS shall be able to filter RCAA messages from the EMS related to severity, time, type and location.

7. The NMS shall be able to query the EMS for a list of the EMS's outstanding alarms.

8. The EMS shall identify the affected transmission resources.

9. "Clear" messages shall be passed to the NMS as soon as alarms are "cleared" at the lower layers.

10. The NMS needs access to the CMIP NE view of the equipment.

The following requirements are not supported with the current model, but will be supported with future versions of the model.

1. The NMS shall be able to optionally block "subsequent analysis messages" (as described in above) from the EMS. Blocked messages shall be logged.
2. The NMS shall be capable to ask the EMS to "resync" fully or partially its NE alarm info.
3. Acknowledgements made by the NMS may be passed to the EMS.
4. The discrete time period defined above shall be configurable.

3.3.4 Information Included in an RCAA Notification

This section discusses the information included in a RCAA notification that is sent from the EMS to the NMS.

The following requirements were derived from the requirements and from the Fault Management scenarios:

1. The RCAA notification should contain at least all the information contained in the associated NEL notification if the root cause was a NEL alarm.
2. The RCAA notification should contain information equivalent to the information contained in the associated NEL notification if the root cause was not a NEL alarm. For example if the root cause was a fiber cut the alarmed object will be a link which is not in the NE view.
3. The RCAA notification should contain sufficient information so that RCAA alarms can be filtered on severity, time, type and location.
4. The RCAA notification should contain information so that the high priority NEL alarms that have been resolved by this alarm message can be retrieved.
5. The RCAA notification should contain a reference to any earlier RCAA message(s)

that are related to this message.

6. The RCAA notification should contain an indication whether this message represents a setting, update or clearing of an alarm.
7. The RCAA notification should indicate whether the alarm is Service Affecting (SA) or Not Service Affecting (NSA).
8. The RCAA notification should indicate whether this alarm represents one or more than one NEL alarms.
9. The RCAA notification should indicate the transmission resources affected by this RCAA alarm. “Transmission resources” means link, link connection, subnetwork connection or trail.

Given these requirements Table 3-1 lists the parameters that are included in the RCAA notification. The first column identifies the parameter. The second column indicates the standard that defines the parameter. If the parameter has not been defined in a standard this is indicated with a “n/a”. The third column indicates whether the parameter is optional or required in the standard from which it is drawn. The fourth column indicates whether this parameter should be optional for RCAA notifications. The last column indicates which of the requirements above motivated the inclusion of this parameter.

Parameter	**Standard**	**Optional in standard**	**Optional in RCAA Notification**	**Requirements**
Managed Object Class (Alarm Analysis Routine Object Class)	X.710	Mandatory	Mandatory	1,2
Managed Object Instance (Alarm Analysis Routine Object instance)	X.710	Mandatory	Mandatory	1,2
Event Type	X.710	Mandatory	Mandatory	1,2,3
Event Time	X.710	Optional	Mandatory	1,2,3
probableCause	X.733	Mandatory	Mandatory	1,2
SpecificProblems	X.733	Optional	Optional	1,2
perceivedSeverity	X.733	Mandatory	Mandatory	1,2,3,6
backedUpStatus	X.733	Optional	Optional	1,2
backUpObject	X.733	Optional	Optional	1,2
trendIndication	X.733	Optional	Optional	1,2
thresholdInfo	X.733	Optional	Optional	1,2
notificationIdentifier	X.733	Optional	Mandatory	1,2,4
correlatedNotification	X.733	Optional	Optional	1,2,5
stateChangeDefinition	X.733	Optional	Optional	1,2
monitoredAttributes	X.733	Optional	Optional	1,2
proposedRepairAction	X.733	Optional	Optional	1,2
additionalText	X.733	Optional	Optional	1,2
additionalInformation	X.733	Optional	Optional	1,2
alarmedObjectClass	n/a	n/a	Mandatory	1,2
alarmedObjectInstance	n/a	n/a	Mandatory	1,2
location	n/a	n/a	Mandatory	3
serviceAffecting	n/a	n/a	Mandatory	7
numberOfNELAlarms	n/a	n/a	Mandatory	8
affectedTransmissionR	n/a	n/a	Mandatory	9

esources				
highestPriorityNEAlarmRecords	n/a	n/a	Mandatory	4

Table 3-1 Parameters Included in RCAA Notification

Any parameter which was mandatory in X.710 or X.733 is mandatory in an RCAA notification. Any parameter that was optional in X.710 or X.733 is optional here unless the parameter allowed us to meet one of the requirements noted above.

The sonetAlarmAnalysisRoutine object emits the RCAA notifications. We must also send the object class and the object instance of the object which is actually in alarm. The alarmedObjectClass and alarmedObjectInstance provide this information.

The attribute: affectedTransmissionResources should provide sufficient object pointers so that the NMS can determine the set of the transmission resources that are affected by the single failure. Here transmission resources refers to links, link connections, trails and subnetwork connections. A single failure can affect many transmission resources. Consider for example that a OC-48 fiber which has been cut will affect over 1344 DS1s. To limit the information which must be sent we will send only the “bottom of the transmission hierarchy”. This will be enough information so that the NMS can determine the other transmission resources either because it is aware of the topology at the EMS or through querying the EMS. We must be sure that we have sufficient information in the EMS so that the NMS can determine all affected resources.

To show that we can derive all dependent transmission resources given the transmission resource at the bottom of the transmission hierarchy we must consider both a hierarchy based on partitioning and layering. Considering partitioning first, given a failure in a subnetwork connection or trail we can scope and filter to find all subnetwork connections which have the failed subnetwork connection as a component. A similar strategy will work if the key transmission source is a link connection. If the key transmission resource is a link one can follow a similar strategy for all link connections contained in the link.

Deriving dependent transmission resources across layering must depend on the connection between TTPs and CTPs again. Suppose that trail has failed. The trail points to TTPs. The TTPs will point to CTPs which support link connections. The link connections will have failed because the supporting TTPs have failed. The CTPs may point to subnetwork connections which have failed. By following the trail of CTPs, subnetwork connections, and TTPs through the layering and using those entities to find the associated transmission entities one can derive the transmission entities associated by key transmission entity.

The use of notificationIdentifier and correlatedNotification allow the correlation between RCAA notifications.

3.3.5 Information Included in the RCAA Record

This section discusses the information that is included in a RCAA Alarm Record.
The following requirements were derived from the Fault Management SOA and from the Fault Management scenarios:

1. The RCAA Alarm Record should contain at least all the information contained in the associated RCAA notification.
2. The RCAA Alarm Record should indicate whether the root cause represented by this alarm is still active.
3. The RCAA Alarm Record should contain information so that all NEL alarms that have been resolved by this alarm message can be retrieved.

One can accomplish the requirements 1 and 2 above by including all the information defined for the RCAA notification. In order to determine which alarms are currently active the NMS will have to correlate the rcaaRecord which clear past alarms with the rcaaRecord which annunciated the clears. This can be done by looking at the perceived severity and the entity instance. One can also place a reference to the rcaaRecord which represents the setting of the alarm in the correlatedNotifications attribute of the rcaaRecord which represents the clearing of the alarm.

To meet the last requirement we must add an additional attribute which lists all correlated NEL alarms which have not already been included in the highestPriorityNELalarmRecords. This new attribute is called lowerPriorityNELalarm records.

The table below shows the information which will be kept in the rcaaRecord.

Parameter	Standard	Optional in standard	Optional in RCAA Notification
Managed Object Class	X.710	Mandatory	Mandatory
Managed Object Instance	X.710	Mandatory	Mandatory
Event Type	X.710	Mandatory	Mandatory
Event Time	X.710	Optional	Mandatory
probableCause	X.733	Mandatory	Mandatory
SpecificProblems	X.733	Optional	Optional
perceivedSeverity	X.733	Mandatory	Mandatory
backedUpStatus	X.733	Optional	Optional
backUpObject	X.733	Optional	Optional
trendIndication	X.733	Optional	Optional
thresholdInfo	X.733	Optional	Optional
notificationIdentifier	X.733	Optional	Mandatory
correlatedNotification	X.733	Optional	Optional
stateChangeDefinition	X.733	Optional	Optional
monitoredAttributes	X.733	Optional	Optional
proposedRepairAction	X.733	Optional	Optional
additionalText	X.733	Optional	Optional
additionalInformation	X.733	Optional	Optional
alarmedObjectClass	n/a	n/a	Mandatory
alarmedObjectInstance	n/a	n/a	Mandatory
location	n/a	n/a	Mandatory
serviceAffecting	n/a	n/a	Mandatory
numberOfNELAlarms	n/a	n/a	Mandatory
affectedTransmissionResources	n/a	n/a	Mandatory
highestPriorityNELalarmRecords	n/a	n/a	Mandatory
lowerPriorityNELalarmRecords	n/a	n/a	Mandatory

Table 3-2 Information Kept in rcaaRecord

3.4 Management Scenarios for Protection

Three types of scenarios are considered in support of protection:

- TMN system scenarios – to identify the high-level scenarios needed to be supported from the perspective of the user of the overall TMN system.
- NMS/EMS connection scenarios – to identify specific scenarios from the perspective of the NMS user of the EMS connection configuration application.
- Technology-specific architecture scenarios – to identify the technology-specific subnetwork architectures to be supported by the SIF information model.

The TMN system scenarios provide the functional basis for identifying the NMS/EMS connection scenarios. The NMS/EMS scenarios are used in combination with the different technology-specific scenarios to ensure that the required scope of application in terms of functions and architecture is addressed by the model.

3.4.1 Technology-specific Architecture Scenarios

The technology-specific architectures of interest include linear Automatic Protection Switching (APS), Unidirectional Path Switched Ring (UPSR), and Bidirectional Line Switched Ring (BLSR). The architectures listed in Table 3-3 must be supported by the model.

PROTECTION SYSTEM	SOURCE
Linear 1+1 Protection	GR-253
1:N Protection	GR-253
2-Fiber BLSR Protection	GR-1230
4-Fiber BLSR Protection	GR-1230
UPSR Protection	GR-1400
VT-Acess Ring with VT Squelching	GR-1230
Interconnection of Two BLSRS Using Same-Side Routing - Drop and Continue	GR-1230
4- Fiber BLSR Interconnection Using Same-Side Routing - Drop and Continue	GR-1230
Interconnection of Two BLSRs Using Opposite-Side Routing and Drop and Continue	GR-1230
Interconnection of a BLSR and a UPSR Using Dual Transmit	GR-1230
Interconnection of Two UPSRs Using Dual Transmit	GR-1400
Interconnection of Two BLSRs using Dual Transmit	GR-1230
Subtending Ring	GR-1230
Virtual Ring	Existing Application

Table 3-3 Typical Protection Systems

The case of setting up a protected subnetwork connection across a generic (i.e., non-specified) subnetwork architecture is examined prior to addressing the technology-specific architectures. Two cases arise: if the network provides the protection, the protected connection extends between two endpoints (for point-to-point) as shown in the left diagram of Figure 3-16; if the customer provides the protection mechanisms (or if protection is provided within another administrative domain), a pair of diversely routed connections between two pairs of endpoints are required as shown in the right diagram of Figure 3-16. Each connection of the connection pair may be protected or not, depending upon specific service agreements.

Protected connection (network provides protection)

Diverse-routed connection pair (e.g., customer provides protection mechanisms)

Figure 3-16 Generic cases for protected connections

A number of interconnected architecture scenarios can be addressed. In UPSR-UPSR interconnection, at least two configurations are possible: logical ring and drop-and-continue. The logical ring case is illustrated in Figure 3-17 for the case of a unidirectional (for simplicity) connection across subnetworks A and B. In subnetwork A, a point-to-multipoint connection connects point 1 with points 2 and 3. This configuration, established for strictly protection purposes (using a bridge function in an NE), is termed a *bicast* connection. Because the transport service among the three points in this configuration is not itself protected, it is classified as an unprotected bicast configuration. In subnetwork B, the configuration is termed an *exclusive merge* connection since only one of the inputs is selected for transfer to the output at point 1 (using a selector function in an NE). It is also unprotected when considered by itself. From the perspective of the end-to-end configuration across subnetworks A and B, we have a highly protected point-to-point connection. It is noted that in the case of bidirectional transport, the connection within each subnetwork containing both bicast and exclusive merge functions is termed an *exclusive composite* connection.

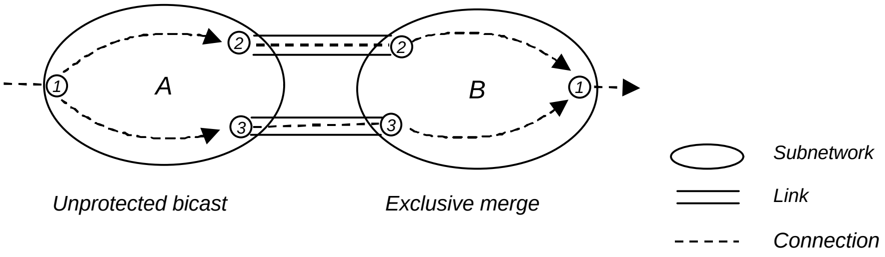


Figure 3-17 UPSR-UPSR Interconnection (logical ring)

The case of drop-and-continue is illustrated in Figure 3-18 for a uni-directional connection. In subnetwork A, a bicast configuration provides transport from point 1 to both 2 and 3. Because the transport service among the three points in this configuration is protected against a single failure except at entry and exit nodes, it is classified as a highly protected bicast connection. In subnetwork B, an unprotected *exclusive merge* connection is used from points 2 and 3 to point 1. From the perspective of the end-to-end configuration across subnetworks A and B, we have a highly protected point-to-point connection as in Figure 3-17. However, in this case, each subnetwork provides independent protection actions.

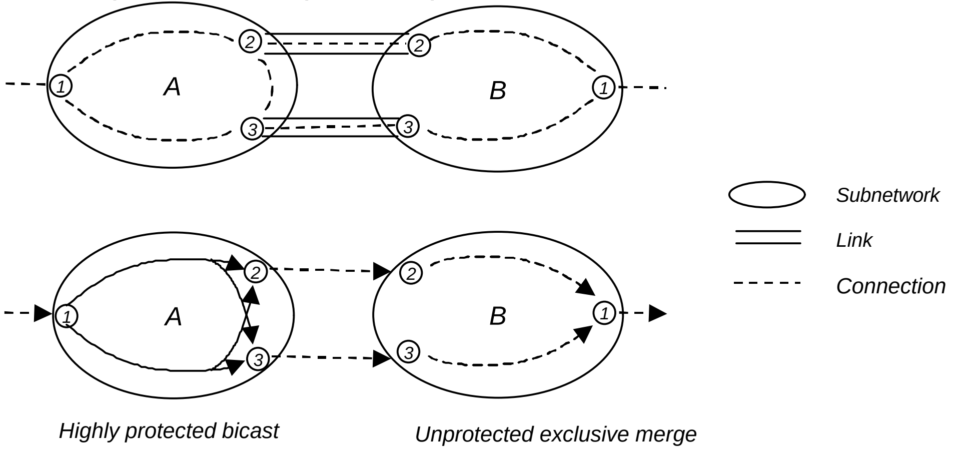


Figure 3-18 UPSR-UPSR Interconnection (drop and continue)

For the case of BLSR-BLSR interconnection, an end-to-end uni-directional, highly protected, point-to-point connection may be set up across a pair of dual-node interconnected rings via the bicast and exclusive merge combination similar to the UPSR-UPSR case. The line level protection switching gives a highly protected classification to each connection. In addition, protection actions in each ring operate independently of each other.

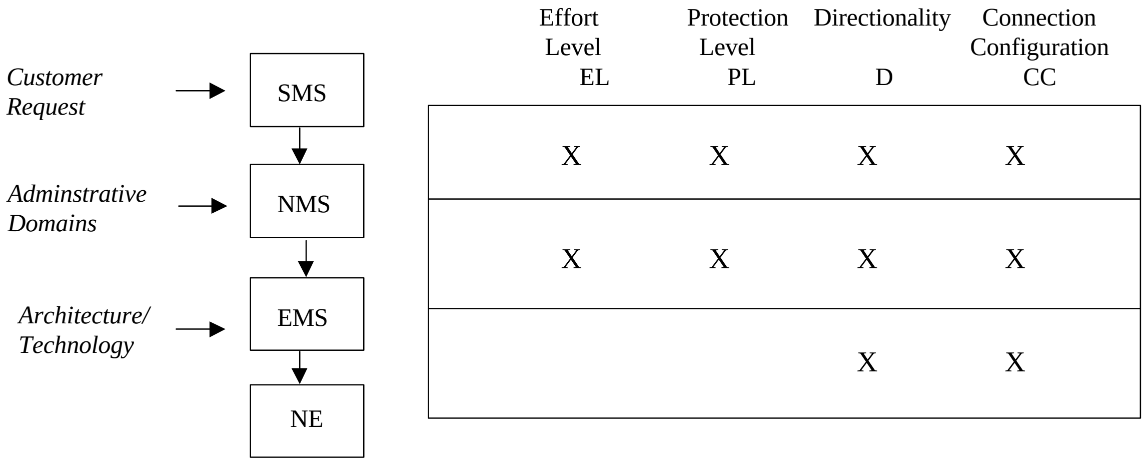
The connection types bicast, exclusive merge, exclusive composite and the protected variations of these connection types will handle the protection methods used in SONET, for example, APS, UPSR, BLSR, and combinations of these.

3.4.2 TMN System Level Considerations

The context of the user of the overall TMN system is primarily at the Service Management Layer (SML) of the TMN hierarchy. SML scenarios address different combinations of service that may be requested by end customers of the service provider. This helps to enumerate different possible scenarios at the NMS/EMS interface.

The types of information that are pertinent to management of protected connections at various TMN interfaces are indicated in Figure 3-19. The left side of the figure shows the general flow of information across the different level systems. The inputs from the left side indicate key information types that distinguish the systems' processing from one another. The box on the right side indicates (by an "X") which types of information may be relevant at each interface. The Connection Configuration attribute represents the point-to-point or point-to-multipoint nature of the connection. Directionality is either bidirectional or unidirectional. The Protection Level can take on several values (highly protected, protected, unprotected, and preemptible). The Effort Level may be either Required Minimum, Required Exact, or Best Effort as defined in the glossary.

The Service Management System (SMS) receives a customer request for connection service of a particular quality between specified end points or locations. The SMS passes relevant aspects of the request to the NMS. The NMS designs the end-to-end path(s) and determines which EMS adminstrative domains need to be involved in setting up the necessary component connections. The EMS receives the request for a connection and determines on the basis of the particular technology/architecture deployed the optimum path(s) through the subnetworks within its domain. The EMS sends requests for cross-connects to the appropriate NEs.



Effort Level EL	Protection Level PL	Directionality D	Connection Configuration CC
X	X	X	X
X	X	X	X
		X	X

Figure 3-19 Information at TMN Interfaces

For a given service request, there may be considerable differences between the information at the SMS/NMS and NMS/EMS interfaces. Different configuration types may be involved at each interface. Different values of Protection Level and Effort Level may also result.

The following scenarios address network connection setup for protection levels of highly protected and protected. For each protection level, the cases of single and multiple administrative domains (EMS domain) are discussed and the different possible connection configurations within multiple domains are identified. The TMN-level scenarios are described in terms of connections across generic subnetworks; technology-specific subnetwork architectures do not enter the discussion at this level.

3.4.3 Highly-protected Connection Scenarios

A highly protected connection may be set up in two ways: dual-homed access to customer locations and single access. Example configurations that support dual-homed access are shown in Figure 3-20 for the cases of a connection within a single EMS domain and within multiple domains. In the single domain case, it is necessary to set up a pair of connections that share neither nodes nor facilities. This configuration is referred to as a *disjoint-route connection pair*. For such a connection configuration, a node containing the endpoints or any intermediate points may not be shared by both connections nor may the facilities be shared by the connections. The multiple domain case represents a view of an end-to-end network connection that is partitioned by EMS domain; the unpartitioned view would appear the same as the single domain case. Within an EMS domain, the possible configurations include a disjoint-route connection pair and a point-to-point subnetwork connection.

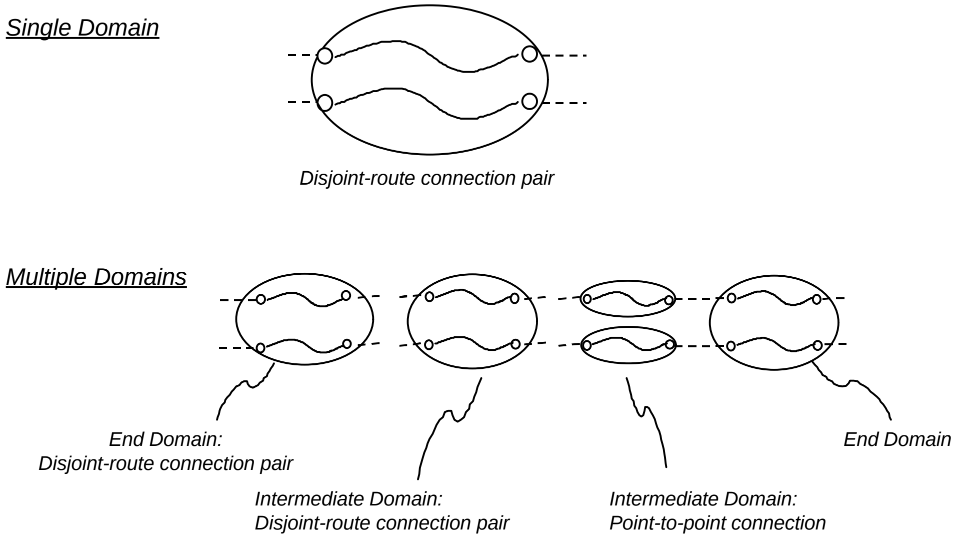

Single Domain
Disjoint-route connection pair
Multiple Domains
End Domain:
Disjoint-route connection pair
End Domain
Intermediate Domain:
Disjoint-route connection pair
Intermediate Domain:
Point-to-point connection

Figure 3-20 Highly-protected scenario with dual-homing to customer premises

Example configurations that support single access are shown in Figure 3-21 for the cases of a connection within a single EMS domain and within multiple domains. In the single domain case, it is necessary to set up a highly protected connection between two points. The subnetwork must implement the highly protected connection based on its particular technology restrictions. Thus, the NMS would pass essentially the same connection attribute information to the EMS that it received from the SMS.

In the multiple domain case, each end domain must implement bicast/exclusive merge connections to explicitly provide for two paths through intervening subnetworks. For example, in the uni-directional case the domain on the left side of the figure sets up a bicast connection, the domain on the right sets up an exclusive merge connection. For bi-directional transport, each end domain sets up exclusive composite connections. Intermediate domains must support either a disjoint-route connection pair or a point-to-point subnetwork connection depending on the routing.

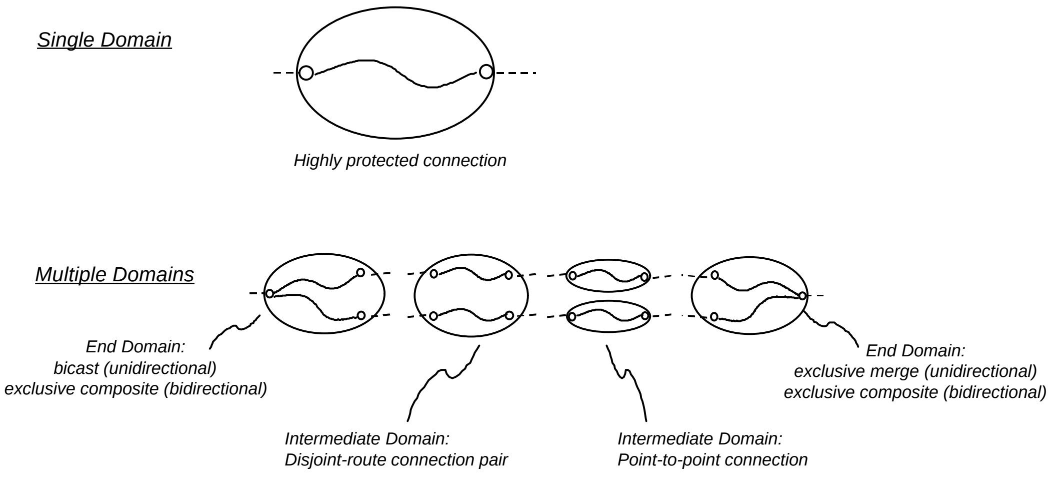
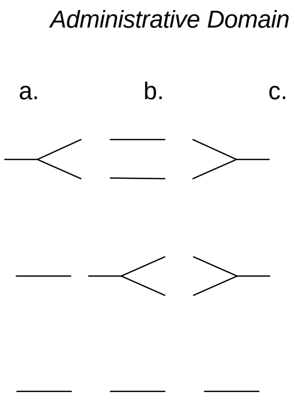

Single Domain
Highly protected connection
Multiple Domains
End Domain:
bicast (unidirectional)
exclusive composite (bidirectional)
End Domain:
exclusive merge (unidirectional)
exclusive composite (bidirectional)
Intermediate Domain:
Disjoint-route connection pair
Intermediate Domain:
Point-to-point connection

Figure 3-21 Highly-protected scenario with single access

Particular types of point-to-multipoint connections are used in the end domains in Figure 3-21. Assume that the protected connection is a unidirectional connection. In this case we will need a bicast connection at the source of the protected connection, a disjoint-route connection pair and/or point to point connections in intermediate domains and an exclusive merge at the sink of the protected connection.

It is useful to consider how an NMS may implement an SMS request for a highly protected connection in the case of multiple administrative domains. Consider the scenario in which an SMS issues the NMS a request for a unidirectional connection with a protection level of "highly protected" at an effort level of "best effort" as indicated in Figure 3-22. In this case, the NMS would first generate a request as indicated in Attempt #1. Bicast and exclusive merge configurations are requested in each end domain; disjoint-routed connection pairs are requested in intermediate domains. If parallel domains are used for routing, the intermediate connections may be point-to-point connections.



Attempt	Admin. Domain AD	Effort Level EL	Protection Level PL	Direction-ality D	Connection Configuration CC	Disjoint Routing DR
SMS/NMS Interface		*BE*	*HP*	*U*	*PP*	
1.	a.	RM	HP	U	BC	N
	b.	BE	P	U	PP	Y
		BE	P	U	PP	Y
	c.	RM	HP	U	XM	N
2.	a.	RM	HP	U	PP	N
	b.	BE	P	U	BC	N
	c.	BE	P	U	XM	N
3.	a.	BE	HP	U	PP	N
	b.	BE	HP	U	PP	N
	c.	BE	HP	U	PP	N

RM - Required Minimum
BE - Best Effort

HP - Highly-Protected
P - Protected
UP - Unprotected
PE - Pre-emptible

U - Uni-directional
BC - Bicast
PP - Point-to-Point
XM - Exclusive Merge

Y/N - Yes/No

Figure 3-22 NMS/EMS Connection Attributes Examples

If any of the EMSs are unable to fulfill the request it receives, e.g., the end domain is unable to support a point-to-multipoint configuration, the NMS may then generate a new request for a somewhat lesser level of protection as indicated in Attempt # 2. In this case the EMS for the end domain is requested to set up a protected connection, and the intermediate domain provides the bicast configuration. The third attempt may be three highly protected connections.

3.4.4 Protected Connection Scenarios

The configurations for a protected network connection are shown in Figure 3-23. Within a single domain, a protected point-to-point subnetwork connection is set up. In the multiple domain case, protected point-to-point subnetwork connections must be established within each domain. In addition, the link connections that make up the segments of the network connection between the EMS domains must be protected. This may be achieved by protection of the server layer trail, or by existing entirely within an NE such as a DCS. In any case, the NMS must have knowledge of the protection level of the links or link connections between subnetworks in different EMS domains.

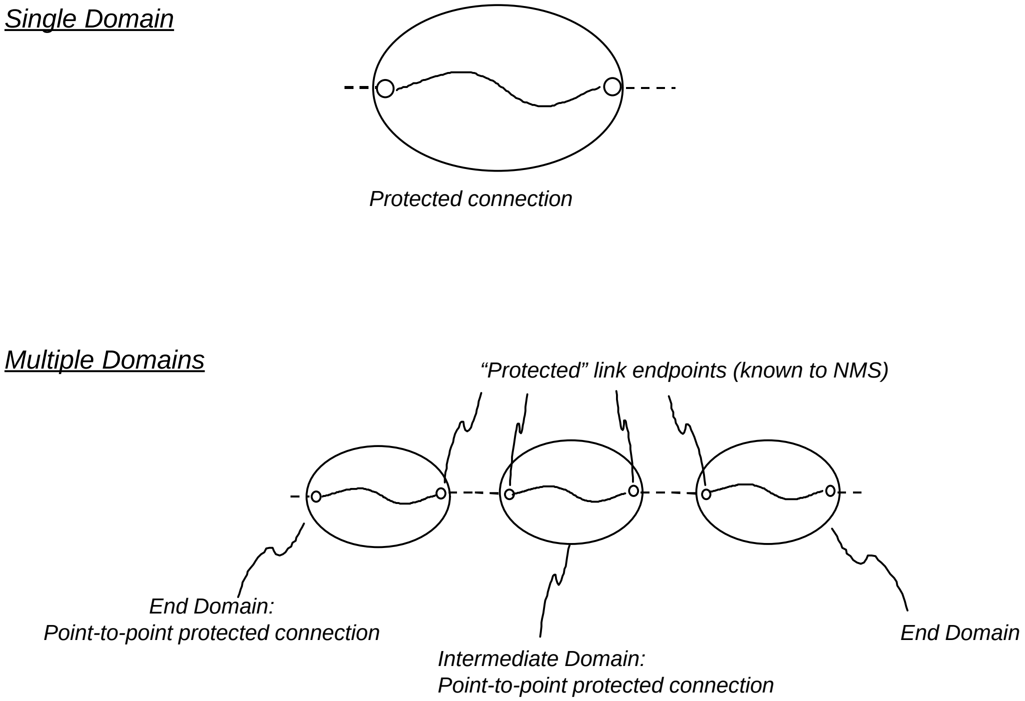

Single Domain
Protected connection
Multiple Domains
"Protected" link endpoints (known to NMS)
End Domain:
Point-to-point protected connection
Intermediate Domain:
Point-to-point protected connection
End Domain

Figure 3-23 Protected connection scenario

Consider how an NMS may implement an SMS request for a protected connection. In the single domain scenario in which an SMS issues the NMS a request for a protection level of "protected" at an effort level of "required," the NMS would generate a request with essentially the same values of attributes.

In the multiple domain case, the sequence of requests that would be generated by the NMS for a "best effort" SMS request can be identified as done above for the highly protected case.

4. Information Model

This section provides a model for the Network Connection Management and Fault Management applications. This model addresses a subset of the functions covered in the statements of applications for Connection Management and Fault Management at the NMS/EMs interface.

4.1 Inheritance Hierarcy

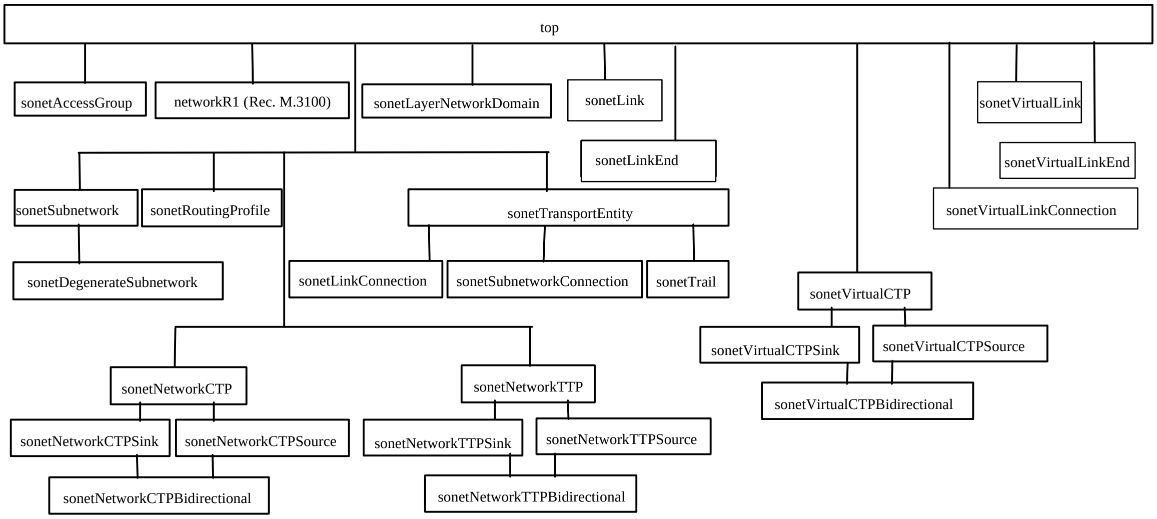


Figure 4-1 Inheritance Hierarchy

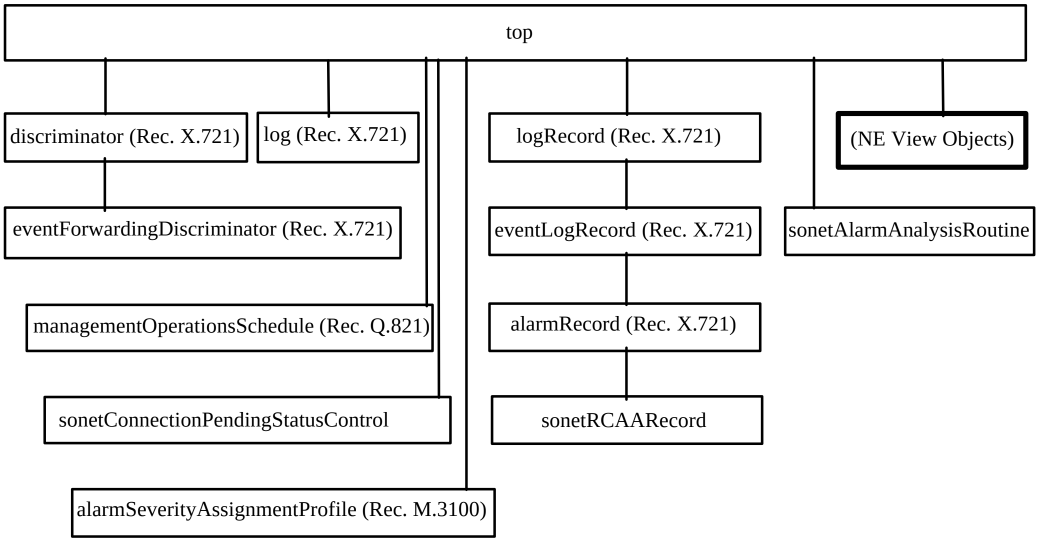

top
discriminator (Rec. X.721)
log (Rec. X.721)
logRecord (Rec. X.721)
(NE View Objects)
eventForwardingDiscriminator (Rec. X.721)
eventLogRecord (Rec. X.721)
sonetAlarmAnalysisRoutine
managementOperationsSchedule (Rec. Q.821)
alarmRecord (Rec. X.721)
sonetConnectionPendingStatusControl
sonetRCAARecord
alarmSeverityAssignmentProfile (Rec. M.3100)

Figure 4-2 Inheritance Hierarchy (Continued)

4.2 Naming Hierarchy

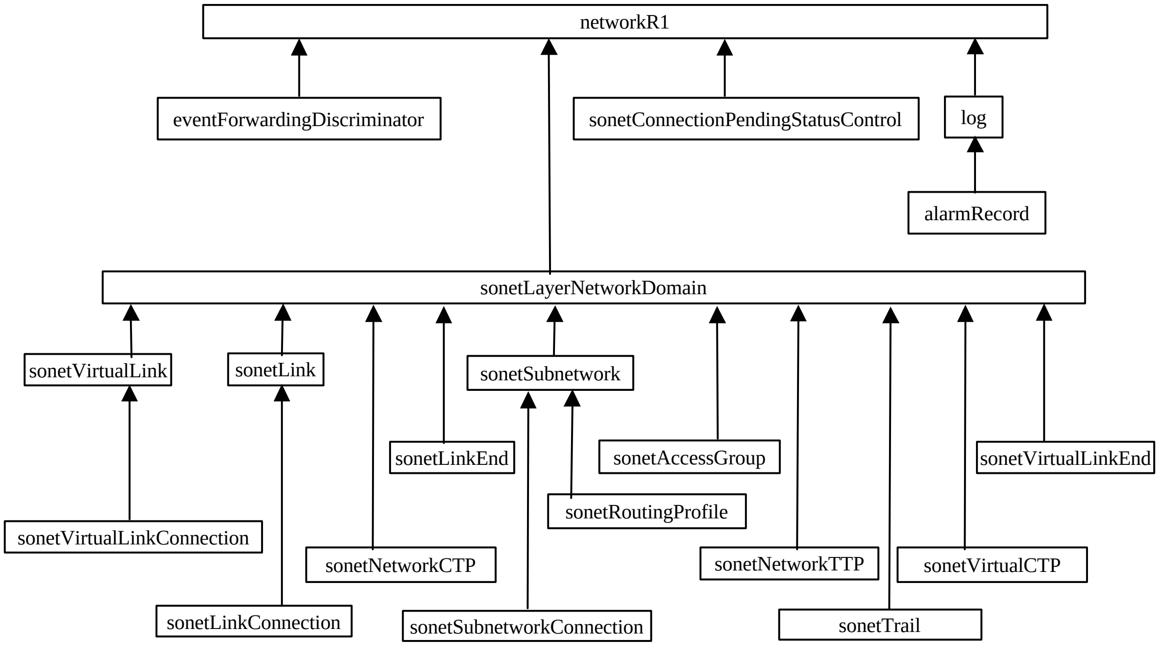


Figure 4-3 Naming Hierarchy

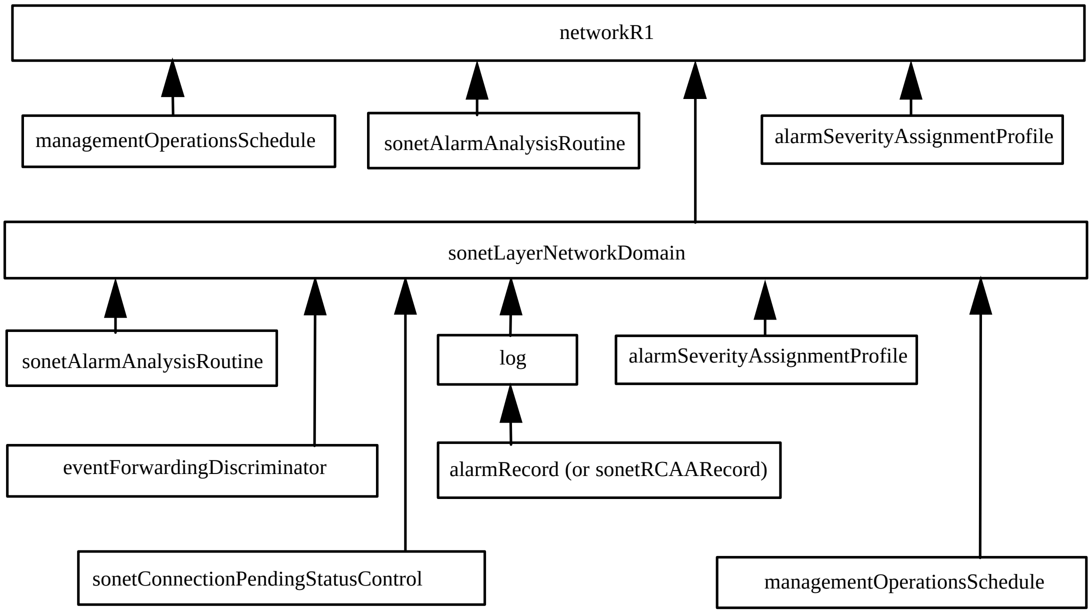

networkR1
managementOperationsSchedule
sonetAlarmAnalysisRoutine
alarmSeverityAssignmentProfile
sonetLayerNetworkDomain
sonetAlarmAnalysisRoutine
log
alarmSeverityAssignmentProfile
eventForwardingDiscriminator
alarmRecord (or sonetRCAARecord)
sonetConnectionPendingStatusControl
managementOperationsSchedule

Figure 4-4 Naming Hierarchy (Continued)

The bold box in the Naming Hierarchy above indicates objects that are defined in the NE view. Not shown is the allowable, for existing implementations only, use of either managedElement or managedElementComplex as the top of the NE View naming hierarchy (rather than networkR1). The Network View would still use networkR1 as top.

The NE View objects may be included in an implementation where they are referenced from the defined Network View objects described in this document, and where both the NE View and the Network View are supported. Implementation of both the NE View and Network View together represents a specific design choice. Also, implementations that provide a “stand alone” network view (no references to NE view objects) may be defined using the objects described in this document. In this case the objects represented by the bold box would not be referenced by the Network View objects.

4.3 Entity-Relationship Diagrams

An E/R diagram of the basic subnetwork topology relationships is shown in Figure 4-6 for the unpartitioned case.

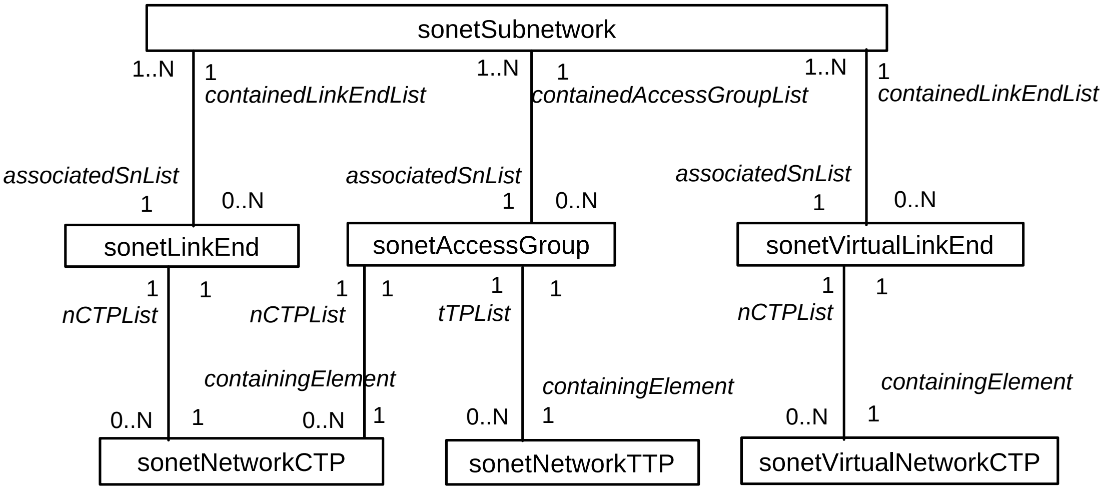


Figure 4-5 Termination Points and Aggregates

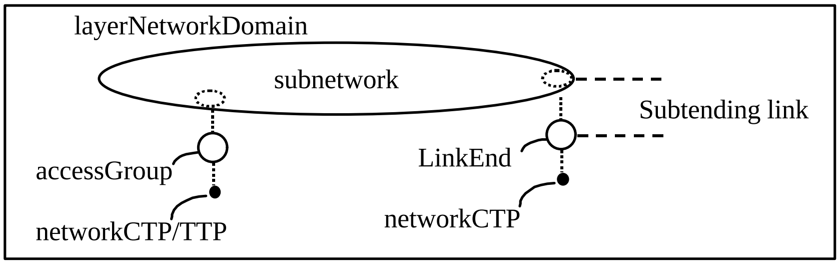
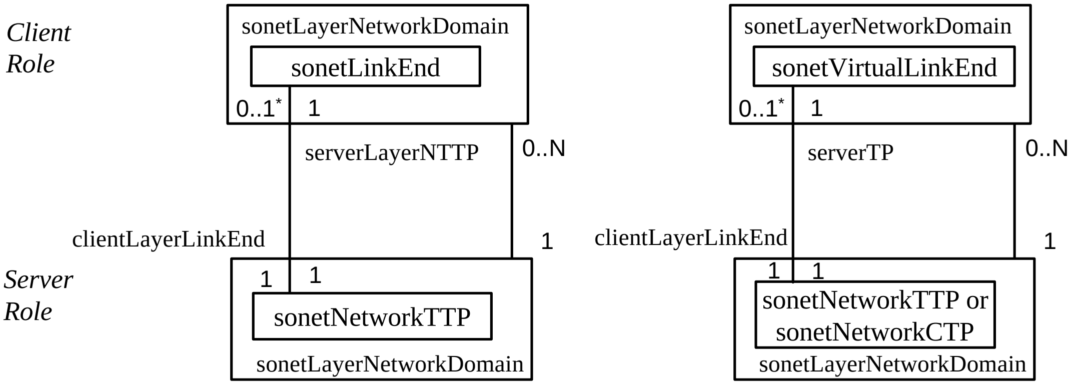


Figure 4-6 Subnetwork topology – unpartitioned

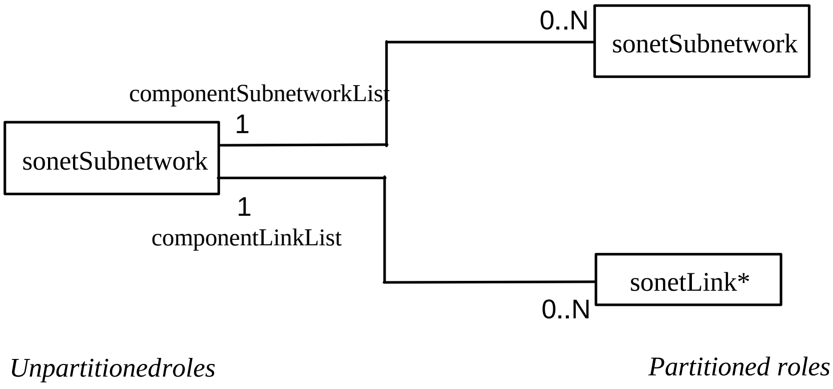
Inter-layer relationships are represented by the role relationships shown in Figure 4-7. In a role relationship diagram, a distinction is drawn between different instances of an object class that play different roles. Here, the distinction is between client and server layerNetworkDomain entities. Changes to the existing model are summarized in the lower part of the figure.



* A sonetNetworkTTP can support only one sonetLinkEnd instance per client sonetLayerNetworkDomain. However, more than one client sonetLayerNetworkDomain may be supported, as indicated by the cardinalities between server and client sonetLayerNetworkDomains.

Figure 4-7 Subnetwork topology – inter-layer role relationships

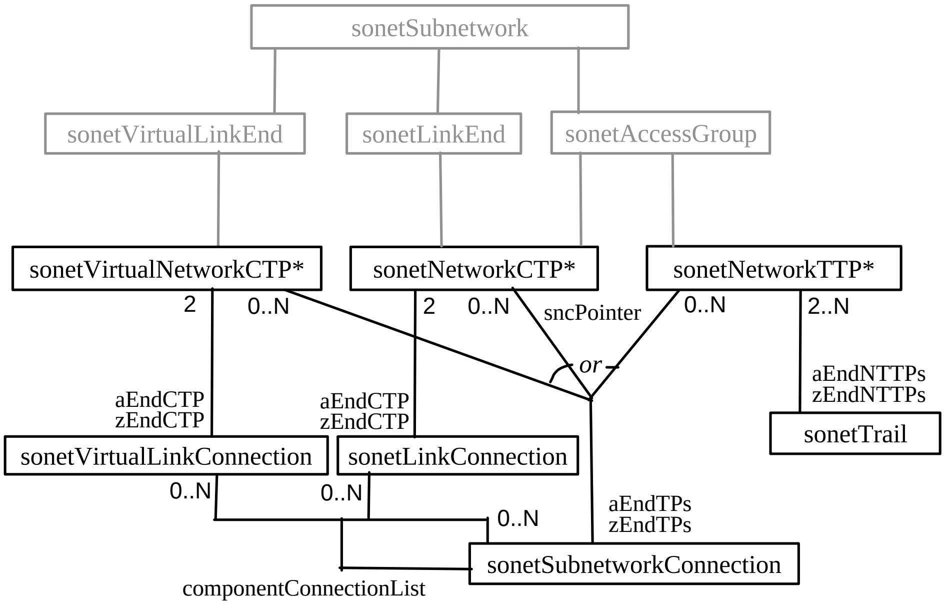
Role relationships representing subnetwork partitioning are shown in Figure 4-8.



* Items in the componentLinkList may include either sonetLink or virtualLink

Figure 4-8 Subnetwork topology – partitioning role relationships

The E/R diagram describing relationships between connection/trail objects and the relevant termination points is shown in Figure 4-9.



* sink, source, bidirectional with restrictions on combinations

Figure 4-9 Connections – unpartitioned view

An illustration of the relationships between connections and termination points under partitioning is shown in Figure 4-10.

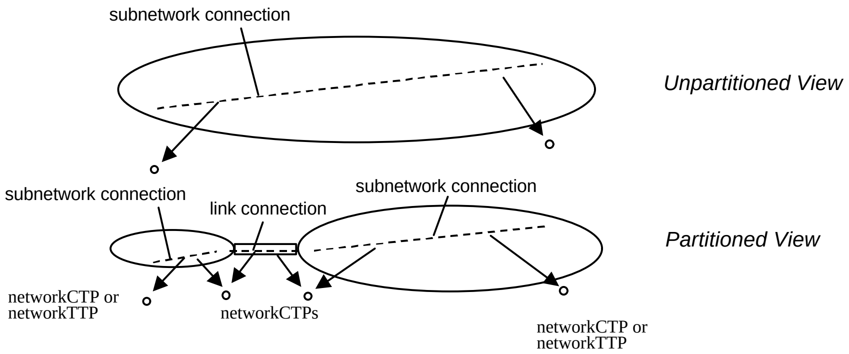


Figure 4-10 Connection partitioning illustration

4.4 Characteristic Information

SONET Layer	Characteristic Information	Source
Optical OC-192	opticalSTM64SPICI	SIF NLM*
Optical OC-48	opticalSTM16SPICI	ITU-T M.3100
Optical OC-12	opticalSTM4SPICI	ITU-T M.3100
Optical OC-3	opticalSTM1SPICI	ITU-T M.3100
Electrical STS-3	electricalSTM1SPICI	ITU-T M.3100
Electrical STS-1	electricalSTS1SPICI	SIF NLM*

Section STS-192	rsSTM64SPICI	SIF NLM*
Section STS-48	rsSTM16SPICI	ITU-T M.3100
Section STS-12	rsSTM4SPICI	ITU-T M.3100
Section STS-3	rsSTM1SPICI	ITU-T M.3100
Line STS-192	msSTM64SPICI	SIF NLM*
Line STS-48	msSTM16SPICI	ITU-T M.3100
Line STS-12	msSTM4SPICI	ITU-T M.3100
Line STS-3	msSTM1SPICI	ITU-T M.3100
Path STS-12c	au44cCI	SIF NLM*
Path STS-3c	au4VC4CI	ITU-T M.3100
Path STS-1	au3TU3VC3CI	ITU-T M.3100
Path VT1.5	tu11VC11CI	ITU-T M.3100
Line DS3	ds3LineCI	SIF NLM*
Line DS1	ds1LineCI	SIF NLM*
Path DS3	ds3PathCI	SIF NLM*
Path DS1	ds1PathCI	SIF NLM*

Table 4-2 Characteristic Information Values

* These values of characteristic information are not currently defined in any other documents. These values are scheduled for addition to GR-836-IMD and/or GR-1042-IMD. When these values are added to the GR(s), they will be removed from this document and referenced as appropriate.

4.5 States

States are supported for some object classes. The model only supports states where the need was recognized. The table below shows the states supported by each object class. See the object class definitions for the acceptable values of each state. (Note: In the object class definitions, “sonet” is added to each object class shown below.)

Classes	administrative State	operational State	tPConnection State	configuration State
AccessGroup	N	N	N	N
AlarmAnalysis Routine	N	N	N	N
Connection Pending StatusControl	N	N	N	N
LayerNetwork Domain	N	N	N	N
Link	Y	N	N	N
LinkConnection	N	Y	N	N
LinkEnd	Y	N	N	N
NetworkCTP	N	C	Y	N
NetworkTTP	Y	Y	Y	N
RCAARecord	N	N	N	N
RoutingProfile	N	N	N	N
Subnetwork	N	N	N	N
Subnetwork Connection	Y	Y	N	Y
Trail	C	Y	N	Y
Trail	C	Y	N	Y
VirtualCTP	N	N	Y	N
VirtualLink	N	N	N	N
VirtualLinkConnection	N	N	N	N
VirtualLinkEnd	N	N	N	N

Table 4-3 States Supported(Y), Conditionally Supported (C), or Not Supported(N) by Object Class

4.6 Information Model

```
sonetAccessGroup  MANAGED OBJECT CLASS
    DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Rec. M.3100 : 1995":attributeValueChangeNotificationPackage,
        "Rec. M.3100 : 1995":createDeleteNotificationsPackage,
        "Rec. M.3100 : 1995":userLabelPackage,
        sonetAccessGroupPackage PACKAGE
                BEHAVIOUR sonetAccessGroupBehaviour;
                ATTRIBUTES
                        signalId
                                GET,
                        sonetAccessGroupId
                                GET,
                        "Rec. M.3100 : 1995":totalTpCount
                                GET,
                        "Rec. M.3100 : 1995":connectedTpCount
```

```
                                        GET,
                                "Rec. M.3100 : 1995":idleTpCount
                                        GET,
                                associatedSnList
                                        GET,
                                tPList
                                        GET-REPLACE     ADD-REMOVE;;;
REGISTERED AS {sIFNLMObjectClass 1};

sonetAccessGroupBehaviour BEHAVIOUR
    DEFINED AS
```

"The sonetAccessGroup object class represents a group of co-located networkTTPs and networkCTPs (and subclasses) within a layer network domain. The tPList attribute points to the set of networkTTPs or networkCTPs that are grouped together.

The totalTpCount is the number of termination points which belong to the accessGroup. The connectedTpCount is the number of termination points which belong to the accessGroup and which are used in a subnetwork connection. The idleTpCount is the number of termination points in the accessGroup which are not used in a subnetwork connection and are available for connection.

The associateSnList attribute points to the subnetworks to which this accessGroup is associated.

A change in the value of the tPList shall cause an attributeValueChange notification:

";

```
sonetAlarmAnalysisRoutine       MANAGED OBJECT CLASS
    DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        sonetAlarmAnalysisRoutinePackage PACKAGE
                BEHAVIOUR sonetAlarmAnalysisRoutineBehaviour;
                ATTRIBUTES
                        "Rec. M.3100 : 1995":alarmSeverityAssignmentProfilePointer
                                GET-REPLACE,
                        sonetAlarmAnalysisRoutineId
                                GET;
                NOTIFICATIONS
                        rcaaNotification;;;
REGISTERED AS {sIFNLMObjectClass 2};

sonetAlarmAnalysisRoutineBehaviour BEHAVIOUR
```

```
    DEFINED AS
        "The sonetAlarmAnalysisRoutine is an object which represents an application
which analyzes alarms. Once analysis is complete it issues the rcaaNotification. The
attribute sonetAlarmAnalysisRoutineId is used as an RDN.

    The alarmSeverityAssignmentProfilePointer attribute is used to flexibly assign the
severity of the alarms issued by the sonetAlarmAnalysisRoutine.";

sonetConnectionPendingStatusControl  MANAGED OBJECT CLASS
    DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Rec. M.3100 : 1995":operationalStatePackage,
        "Rec. M.3100 : 1995":stateChangeNotificationPackage,
        sonetConnectionPendingStatusControlPackage PACKAGE
                BEHAVIOUR

    sonetConnectionPendingStatusControlBehaviour;
                ATTRIBUTES
                        sonetConnectionPendingStatusControlId
                                GET,
                        "Rec. Q.821":objectList
                                GET-REPLACE      ADD-REMOVE;
                NOTIFICATIONS
                        connectionConfigurationSummaryReport;;;
    CONDITIONAL PACKAGES
        emptyListSuppressionPackage PRESENT IF "an instance supports it";
REGISTERED AS { sIFNLMObjectClass 3};

sonetConnectionPendingStatusControlBehaviour BEHAVIOUR
    DEFINED AS
        "This managed object class is a class of support objects that enables the
generation of reports on the status of subnetwork connections and trails that have been
requested by the managing system and are in the process of being established or
released through NE cross connections.

    The objectList attribute identifies the sonetSubnetworkConnection and sonetTrail
object instances that are to be included in the connectionConfigurationSummaryReport.
Objects will be added to the list by the setupSNC, releaseSNC, setupTrail, and
releaseTrail actions.  An object will be removed from the list when the object instance
has been deleted or when the value of the configurationState in the object instance has
changed.

    The connectionConfigurationSummaryReport notification provides a summary report
of the sonetSubnetworkConnection and sonetTrail object instances that are being
established or released.  The report includes the object instance of the
```

sonetSubnetworkConnection or sonetTrail object, whether it is being established or released, and the percentage of completed or released cross-connections.

The emptyListSuppressionPackage suppresses the connectionConfigurationSummaryReport if the objectList is empty.

A change in the value of the operationalState attribute shall cause a stateChange notification.

A change in the value of any of the following attributes shall cause an attributeValueChange notification:

objectList

";

```
sonetDegenerateSubnetwork              MANAGED OBJECT CLASS
    DERIVED FROM sonetSubnetwork;
    CHARACTERIZED BY
        sonetDegenerateSubnetworkPackage    PACKAGE
                BEHAVIOUR sonetDegenerateSubnetworkBehaviour;;;
REGISTERED AS { sIFNLMObjectClass 4};

sonetDegenerateSubnetworkBehaviour BEHAVIOUR
    DEFINED AS
```

"This managed object class represents the degenerate case of a subnetwork. That is, this subnetwork allows no flexibility in the assignment of subnetwork connections between termination points (networkTTPs, networkCTPs, or virtualCTPs). Only fixed relationships may exist between termination points and these relationships cannot be altered by the managing system.

The componentLinkList and componentSubnetworkList attributes are set to null.

The setupSNC and releaseSNC actions are used to setup and release subnetwork connections between termination points with no flexibility.";

```
sonetLayerNetworkDomain          MANAGED OBJECT CLASS
    DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Rec. M.3100 : 1995":createDeleteNotificationsPackage,
        "Rec. M.3100 : 1995":userLabelPackage,
        sonetLayerNetworkDomainPackage PACKAGE
                BEHAVIOUR sonetLayerNetworkDomainBehaviour;
                ATTRIBUTES
                        signalId
```

```
                                GET,
                        sonetLayerNetworkDomainId
                                GET;
                ACTIONS
                        addLink,
                        releaseTrail,
                        removeLink,
                        setupTrail;;;
REGISTERED AS { sIFNLMObjectClass 5};
```

```
sonetLayerNetworkDomainBehaviour BEHAVIOUR
    DEFINED AS
```

"The sonetLayerNetworkDomain object class represents the part of a layer network managed by a single administrative domain. It acts as a container for instantiations of network resource object classes including subnetwork, link, linkEnd, accessGroup, networkCTP, networkTTP, virtualCTP, virtualLink, virtualLinkEnd, and trail object classes. A trail that extends across multiple administrative domains, however, is not explicitly included in the model.

The setupTrail action is used to explicitly create a sonetTrail object instance and associate it with the terminating sonetNetworkTTPs.

The releaseTrail action is used to explicitly delete a sonetTrail object instance and remove it's associations with sonetNetworkTTPs.

The addLink action is used to explicitly create an sonetLink object instance and associate it with the terminating sonetLinkEnds.

The removeLink action is used to explicitly delete an sonetLink object instance and remove it's associations with sonetLinkEnds." ;

```
sonetLink       MANAGED OBJECT CLASS
    DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Rec. X.721 | ISO/IEC 10165-2 : 1992":administrativeStatePackage,
        "Rec. M.3100 : 1995":attributeValueChangeNotificationPackage,
        "Rec. M.3100 : 1995":createDeleteNotificationsPackage,
        "Rec. M.3100 : 1995":stateChangeNotificationPackage,
        "Rec. M.3100 : 1995":userLabelPackage,
        sonetLinkPackage PACKAGE
                BEHAVIOUR sonetLinkBehaviour;
                ATTRIBUTES
                        aLinkEnd
                                GET,
                        signalId
                GET,
            sonetLinkId
                GET,
            zLinkEnd
                GET;
        ACTIONS
            addLinkCapacity,
            releaseLinkConnection,
            removeLinkCapacity,
            setupLinkConnection;;;
REGISTERED AS { sIFNLMObjectClass 6};
```

```
sonetLinkBehaviour BEHAVIOUR
    DEFINED AS
```

"A sonetLink is a topological component that provides transport capacity between two endpoints via a fixed (i.e., inflexible routing) relationship. The two endpoints are sonetLinkEnd object instances associated with peer sonetSubnetworks (i.e., the subnetworks are at the same level of partitioning and are not related via componentSubnetworkList pointer attributes. A single sonetLink object instance has pointer relationships, through the aLinkEnd and zLinkEnd attributes, with a pair of sonetLinkEnd object instances. A sonetLink also represents a set of link connections.

The addLinkCapacity action is used to add additional capacity to the link. This action may require the server layer to provide additional bandwidth. If this action is successful, additional sonetNetworkCTPs will be created and associated with the terminating sonetLinkEnds.

The removeLinkCapacity action is used to remove excess capacity from the link. This action causes the deletion of related sonetNetworkCTPs. This action requires that terminated link connections be explicitly released before the related sonetNetworkCTPs are deleted.

The setupLinkConnection action is used to explicitly create a sonetLinkConnection object instance and associate it with the terminating sonetNetworkCTPs.

The releaseLinkConnection action is used to explicitly delete a sonetLinkConnection object instance and remove it's associations with sonetNetworkCTPs.

The userLabel associated with an object instance of this object class is provided for the convenience of the manager.
The administrativeState is used for administratively locking and unlocking the sonetLink. When unlocked, the sonetLink functions normally. When in the locked state, the sonetLink is prohibited from adding or removing linkConnections. Also, the

sonetLink is prohibited from adding or removing link capacity. Locking a sonetLink does not automatically lock the contained linkConnections. Shutting down is not supported.

A change in the value of the administrativeState attribute shall cause a stateChange notification.
";

```
sonetLinkConnection        MANAGED OBJECT CLASS
    DERIVED FROM sonetTransportEntity;
    CHARACTERIZED BY
        sonetLinkConnectionPackage PACKAGE
                BEHAVIOUR sonetLinkConnectionBehaviour;
                ATTRIBUTES
                        aEndCTP
                                GET,
                        zEndCTP
                                GET;;;
REGISTERED AS { sIFNLMObjectClass 7};
```

sonetLinkConnectionBehaviour BEHAVIOUR
    DEFINED AS

"A sonetLinkConnection is a transport entity that represents the fixed capacity of transfer of information of a given signalId between sonetNetworkCTPs. Only point-to-point link connections are supported by this object class. The aEndCTP and zEndCTP attributes point to the terminating sonetNetworkCTPs.

The operationalState is used to describe the operability of the linkConnection. If the state is enabled, the linkConnection is fully or partially operational. If the state is disabled, the linkConnection is totally inoperable.

The userLabel associated with an object instance is provided for the convenience of the manager.";

```
sonetLinkEnd        MANAGED OBJECT CLASS
    DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Rec. X.721 | ISO/IEC 10165-2 : 1992":administrativeStatePackage,
        "Rec. M.3100 : 1995":attributeValueChangeNotificationPackage,
        "Rec. M.3100 : 1995":createDeleteNotificationsPackage,
        "Rec. M.3100 : 1995":stateChangeNotificationPackage,
        sonetLinkEndPackage PACKAGE
                BEHAVIOUR sonetLinkEndBehaviour;
                ATTRIBUTES
                        sonetLinkEndId
                                GET,
                        serverLayerNTTP
                                GET SET-BY-CREATE,
                        signalId
                                GET,
                        "Rec. M.3100 : 1995":totalTpCount
                                GET,
                        "Rec. M.3100 : 1995":connectedTpCount
                                GET,
                        "Rec. M.3100 : 1995":idleTpCount
                                GET,
                        associatedSnList
                                GET,
                        linkPointer
                                GET,
                        nCTPList
                                GET-REPLACE      ADD-REMOVE;;;
    CONDITIONAL PACKAGES
        linkEndCapacityManagementPackage    PRESENT    IF    "the    sonetLinkEnd
represents the end of a link which crosses the EMS domain boundary.";
REGISTERED AS { sIFNLMObjectClass 8};
```

sonetLinkEndBehaviour BEHAVIOUR
    DEFINED AS

A linkEnd object represents the extremity of a link. It may have an associated set (possibly empty) of networkCTPs.

The totalTpCount is the number of termination points which belong to the accessGroup. The connectedTpCount is the number of termination points which belong to the accessGroup and which are used in a subnetwork connection. The idleTpCount is the number of termination points in the accessGroup which are not used in a subnetwork connection and are available for connection.

The associatedSnList attribute points to the subnetworks which are associated with this linkEnd.

The serverLayerNTTP attribute points to the networkTTP in the server layer which supports this linkEnd.

The linkPointer attribute points to the link which this linkEnd terminates.

The nCTPList attribute points to the CTPs which are included inthis linkEnd.

A change in the value of any of the following attributes shall cause an attributeValueChange notification:
nCTPList

```
";

sonetNetworkCTP   MANAGED OBJECT CLASS
    DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Rec. M.3100 : 1995":attributeValueChangeNotificationPackage,
        "Rec. M.3100 : 1995":createDeleteNotificationsPackage,
        "Rec. M.3100 : 1995":locationNamePackage,
        "Rec. M.3100 : 1995":stateChangeNotificationPackage,
        sonetNetworkCTPPackage          PACKAGE
                BEHAVIOUR sonetNetworkCTPBehaviour;
                ATTRIBUTES
                        channelIdentifier
                                GET,
                        signalId
                                GET,
                        sonetNetworkCTPId
                                GET,
                        containingElement
                                GET,
                        tPConnectionState
                                GET,
                        tPPointer
                                GET;;;
    CONDITIONAL PACKAGES
        "Rec. M.3100 : 1995":operationalStatePackage
                PRESENT IF "the NE View CTP the sonetNetworkCTP is
        abstracting supports the operational state attribute.",
        sNCPointerPackage
                PRESENT IF "the termination point may be flexibly
assigned to another termination point or the
assignment of the termination point to another
termination point can be set up and torn down",
        connectivityPointerPackage
                PRESENT IF "the termination point is always
permanently assigned to another termination point";
REGISTERED AS { sIFNLMObjectClass 9};

sonetNetworkCTPBehaviour BEHAVIOUR
    DEFINED AS
```

"The sonetNetworkCTP (or subclasses) object class represents the extremity of a potential or actual link connection. It is also a network level abstraction of an NE view connection termination point.

The channelIdentifier attribute provides a group number and a channel number to identify the timeslot associated with this NetworkCTP.

The locationName attribute in the locationNamePackage represents a specific geographic location where the NetworkCTP is located.

The sNCPointer identifies the sonetSubnetworkConnection, if any, that the networkCTP is terminating.

The containingElement attribute identifies the sonetLinkEnd and/or sonetAccessGroup which contains the networkCTP.

The tPConnectionState attribute provides the connection status for the NetworkCTP.

The tPPointer attribute points to the NE View termination point which this termination point abstracts.

If supported, the operational state of the sonetNetworkCTP will reflect the operational state of the abstracted NE View CTP.

If supported, a change in the value of the operationalState attribute shall cause a stateChange notification. A change in the value of the tPConnectionState attribute shall cause a stateChange notification.

A change in the value of any of the following attributes shall cause an attributeValueChange notification:

containingElement [potential for deletion]
";

```
sonetNetworkCTPBidirectional             MANAGED OBJECT CLASS
    DERIVED FROM sonetNetworkCTPSource,sonetNetworkCTPSink;
    CHARACTERIZED BY
        sonetNetworkCTPBidirectionalPackage  PACKAGE
                BEHAVIOUR sonetNetworkCTPBidirectionalBehaviour;;;
REGISTERED AS { sIFNLMObjectClass 10};

sonetNetworkCTPBidirectionalBehaviour BEHAVIOUR
    DEFINED AS
        "This managed object class represents both a sink and a source
    network connection termination point.";

sonetNetworkCTPSink              MANAGED OBJECT CLASS
    DERIVED FROM sonetNetworkCTP;
    CHARACTERIZED BY
```

```
        sonetNetworkCTPSinkPackage    PACKAGE
                BEHAVIOUR sonetNetworkCTPSinkBehaviour;;;
REGISTERED AS { sIFNLMObjectClass 11};

sonetNetworkCTPSinkBehaviour BEHAVIOUR
    DEFINED AS
        "This managed object class is a unidirectional networkCTP which terminates a
unidirectional linkConnection.";


sonetNetworkCTPSource   MANAGED OBJECT CLASS
    DERIVED FROM sonetNetworkCTP;
    CHARACTERIZED BY
        sonetNetworkCTPSourcePackage          PACKAGE
                BEHAVIOUR sonetNetworkCTPSourceBehaviour;;;
REGISTERED AS { sIFNLMObjectClass 12};

sonetNetworkCTPSourceBehaviour BEHAVIOUR
    DEFINED AS
        "This managed object class is a unidirectional networkCTP which originates a
unidirectional linkConnection.";

sonetNetworkTTP    MANAGED OBJECT CLASS
    DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Rec. X.721 | ISO/IEC 10165-2 : 1992":administrativeStatePackage,
        "Rec. M.3100 : 1995":attributeValueChangeNotificationPackage,
        "Rec. M.3100 : 1995":createDeleteNotificationsPackage,
        "Rec. M.3100 : 1995":locationNamePackage,
        "Rec. M.3100 : 1995":operationalStatePackage,
        "Rec. M.3100 : 1995":stateChangeNotificationPackage,
        sonetNetworkTTPPackage PACKAGE
                BEHAVIOUR sonetNetworkTTPBehaviour;
                ATTRIBUTES
                        clientLayerLinkEnd
                                GET,
                        containingElement
                                GET,
                        signalId
                                GET,
                        sonetNetworkTTPId
                                GET,
                        tPConnectionState
                                GET,
                        tPPointer
```

```
                              GET;;;
    CONDITIONAL PACKAGES
        sNCPointerPackage
                PRESENT IF "the termination point may be flexibly assigned to
                        another termination point or the assignment of the
                        termination point to another termination point can be set up
                        and torn down",
        connectivityPointerPackage
                PRESENT IF "the termination point is always permanently assigned
        to another termination point";
REGISTERED AS { sIFNLMObjectClass 13};

sonetNetworkTTPBehaviour BEHAVIOUR
    DEFINED AS
```

"The sonetNetworkTTP object class represents the potential extremity of a sonetTrail. It is also a network level abstraction of an NE View trail termination point.

The clientLayerLinkEnd attribute identifies the object instances of the related sonetLinkEnd objects (and subclasses) in the client layer. The client sonetLinkEnds must belong to different sonetLayerNetworkDomains.

The containingElement attribute points to the accessGroup which contains this TTP.
The sNCPointer identifies the sonetSubnetworkConnection, if any, that the networkTTP is terminating.

The administrativeState is used for administratively locking and unlocking the sonetNetworkTTP. When unlocked, the sonetNetworkTTP functions normally. When in the locked state, the sonetNetworkTTP is removed from service.

The operationalState describes the operability of the NetworkTTP. When enabled, the NetworkTTP is either partially or fully operable. When disabled, the NetworkTTP is totally inoperable.

The locationName attribute in the locationNamePackage represents a specific geographic location where the NetworkTTP is located.

The tPConnectionState attribute provides the connection status for the NetworkTTP.

The tPPointer attribute points to the NE View termination point which this termination point abstracts.

A change in the value of the administrativeState attribute, operationalState attribute, or tPConnectionState attribute shall cause a stateChange notification.

A change in the value of any of the following attributes shall cause an

```
attributeValueChange notification:

        clientLayerLinkEnd
";

sonetNetworkTTPBidirectional        MANAGED OBJECT CLASS
  DERIVED FROM sonetNetworkTTPSource,sonetNetworkTTPSink;
  CHARACTERIZED BY
    sonetNetworkTTPBidirectionalPackage  PACKAGE
        BEHAVIOUR sonetNetworkTTPBidirectionalBehaviour;;;
REGISTERED AS { sIFNLMObjectClass 14};

sonetNetworkTTPBidirectionalBehaviour BEHAVIOUR
  DEFINED AS
    "This managed object class represents both a sink and a source network trail
termination point.";

sonetNetworkTTPSink         MANAGED OBJECT CLASS
  DERIVED FROM sonetNetworkTTP;
  CHARACTERIZED BY
    sonetNetworkTTPSinkPackage    PACKAGE
        BEHAVIOUR sonetNetworkTTPSinkBehaviour;;;
REGISTERED AS { sIFNLMObjectClass 15};

sonetNetworkTTPSinkBehaviour BEHAVIOUR
  DEFINED AS
    "This managed object class is a unidirectional sonetNetworkTTP which
terminates a unidirectional sonetTrail.";

sonetNetworkTTPSource   MANAGED OBJECT CLASS
  DERIVED FROM sonetNetworkTTP;
  CHARACTERIZED BY
    sonetNetworkTTPSourcePackage        PACKAGE
        BEHAVIOUR sonetNetworkTTPSourceBehaviour;;;
REGISTERED AS { sIFNLMObjectClass 16};

sonetNetworkTTPSourceBehaviour BEHAVIOUR
  DEFINED AS
    "This managed object class is a unidirectional sonetNetworkTTP which
originates a unidirectional sonetTrail.";

sonetRCAARecord  MANAGED OBJECT CLASS
  DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":alarmRecord;
  CHARACTERIZED BY
    sonetRCAARecordPackage PACKAGE
```

```
                BEHAVIOUR sonetRCAARecordBehaviour;
                ATTRIBUTES
                        alarmedObjectClass
                                GET,
                        alarmedObjectInstance
                                GET,
                        faultLocation
                                GET,
                        serviceAffecting
                                GET,
                        numberOfNELAlarms
                                GET,
                        affectedTransmissionResources
                                GET,
                        highestPriorityNELalarmRecords
                                GET,
                        lowerPriorityNELalarmRecords
                                GET;;;
REGISTERED AS { sIFNLMObjectClass 17};

sonetRCAARecordBehaviour BEHAVIOUR
    DEFINED AS
        "The sonetRCAARecord is generated by the root cause alarm analysis routine
based on one or more NEL alarm event reports. It contains all the information
contained in the RCAA Notification as well as the lowerPriorityNELalarmRecords
attribute which lists all correlated NEL alarm records which were not included in the
highestPriorityNELalarmRecords. The correlatedNotifications will contain the
notificationIdentifier of previous RCAA notifications to which this notification is either an
update or a clear.
Instances of the sonetRCAARecord will be created by the alarm analysis routine.
Instances will be deleted per the configuration of the log.";

sonetRoutingProfile MANAGED OBJECT CLASS
    DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Rec. M.3100 : 1995":createDeleteNotificationsPackage,
        sonetRoutingProfilePackage PACKAGE
                BEHAVIOUR sonetRoutingProfileBehaviour;
                ATTRIBUTES
                        maxHops
                                GET SET-BY-CREATE,
                        routeDescriptionList
                                GET SET-BY-CREATE,
                        sonetRoutingProfileId
                                GET;;;
```

REGISTERED AS { sIFNLMObjectClass 18};

sonetRoutingProfileBehaviour BEHAVIOUR
DEFINED AS

"The sonetRoutingProfile object class represents a set of topological routing constraints that can be applied to a new sonetSubnetworkConnection or sonetTrail during setup. Subnetwork connections or trails shall be rejected if the routing criteria cannot be met.

The maxHops attribute is the maximum number of network elements that the new connection or trail may traverse. This attribute may be set to null to indicate that the maxHops criteria does not apply.

The routeDescriptionList attribute is a list of instantiated objects (that may include sonetLink, sonetVirtualLink, sonetSubnetwork, sonetLinkEnd object instances, etc.) and their use in routing (prohibited, mandatory, preferred).

Objects may be referenced by the routeDescriptionList as being excluded, mandatory, or preferred. A mandatory object must be used. An excluded object must not be used. A preferred object should be used if possible.

A sonetSubnetworkConnection or sonetTrail may be referenced by the routeDescriptionList as same route, diverse route, or diverse route and node. A new sonetSubnetworkConnection or sonetTrail being created must follow the same route as a sameRoute referenced object. A new sonetSubnetworkConnection or sonetTrail must follow a different route than a referenced object referred to as diverseRoute. A new sonetSubnetworkConnection or sonetTrail must follow a physically diverse route and have no nodes in common with a referenced object referred to as diverseRouteAndNode.";

```
sonetSubnetwork           MANAGED OBJECT CLASS
    DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Rec. M.3100 : 1995":attributeValueChangeNotificationPackage,
        "Rec. M.3100 : 1995":createDeleteNotificationsPackage,
        "Rec. M.3100 : 1995":userLabelPackage,
        sonetSubnetworkPackage PACKAGE
                BEHAVIOUR sonetSubnetworkBehaviour;
                ATTRIBUTES
                        accessGroupList
                                GET-REPLACE     ADD-REMOVE,
                        componentLinkList
                                GET-REPLACE     ADD-REMOVE,
                        componentSubnetworkList
                                GET-REPLACE     ADD-REMOVE,
                        linkEndList
                                GET-REPLACE      ADD-REMOVE,
                        signalId
                                GET,
                        sonetSubnetworkId
                                GET,
                        subnetworkType     GET;
                ACTIONS
                        setupSNC,
                        releaseSNC;;;
    CONDITIONAL PACKAGES
        lineSwitchPackage PRESENT IF "the subnetworkType is APS, BLSR, or
         other line-based protection mechanism.";
REGISTERED AS { sIFNLMObjectClass 19};
```

sonetSubnetworkBehaviour BEHAVIOUR
DEFINED AS

"This managed object class represents a topological component used to effect routing of information of a specific signalId. A sonetSubnetwork is associated with a specific layer network. Within a given layer, partitioning may be applied to decompose a sonetSubnetwork into its component sonetSubnetworks and links.

The accessGroupList points to accessGroup managed object instances associated with this subnetwork. The linkEndList points to linkEnd managed object instances associated with this subnetwork.

The componentLinkList and componentSubnetworkList attributes identify the components of the sonetSubnetwork which exist at the next lower level of partitioning.

The subnetworkType attribute provides information on the element level resource(s) this object instance is abstracting.

The setupSNC and releaseSNC actions are used to setup and release sonetSubnetworkConnections within the sonetSubnetwork.

A change in the value of any of the following attributes shall cause an attributeValueChange notification:

```
                componentLinkList
                componentSubnetworkList
                subnetworkType
";
sonetSubnetworkConnection       MANAGED OBJECT CLASS
    DERIVED FROM sonetTransportEntity;
    CHARACTERIZED BY
        "Rec. X.721 | ISO/IEC 10165-2 : 1992":administrativeStatePackage,
    sonetSubnetworkConnectionPackage    PACKAGE
        BEHAVIOUR sonetSubnetworkConnectionBehaviour;
        ATTRIBUTES
            aEndTPs
                GET,
            componentConnectionAZList
                GET,
            componentConnectionZAList
                GET,
            configurationState
                GET,
            relatedRoutingProfile
                GET,
            zEndTPs
                GET,
            connectionType
                GET-REPLACE,
            operationalProtectionLevel
                GET,
            provisionedProtectionLevel
                GET-REPLACE;;;
  CONDITIONAL PACKAGES
    primaryTpPackage PRESENT IF "the subnetworkConnectionType is exclusive
merge or exclusive composite and is protected via a revertive switching mechanism.";
REGISTERED AS { sIFNLMObjectClass 20};
```

sonetSubnetworkConnectionBehaviour BEHAVIOUR
DEFINED AS

"A sonetSubnetworkConnection object instance represents a transport entity that transfers information across a sonetSubnetwork. A sonetSubnetwork connection is terminated by network termination points (aEndTPs and zEndTPs) which may be either networkTTP, networkCTP, or virtualCTP objects.

A sonetSubnetworkConnection can be composed of sonetSubnetworkConnections and sonetLinkConnections which exist at the next lower level of partitioning. The componentConnectionAZList attribute identifies the currently active components of the sonetSubnetworkConnection going from A to Z. The componentConnectionZAList attribute identifies the currently active components of the sonetSubnetworkConnection going from Z to A. .

The configurationState attribute describes the procedural status relative to the process of creation or deletion of a sonetSubnetworkConnection object instance. The configurationState indicates that the sonetSubnetworkConnection is in either a preservice, inservice, postservice, or configuration failure state.

The relatedRoutingProfile attribute points to the sonetRoutingProfile object instance which contains the routing profile for the sonetSubnetworkConnection.

A change in the value of the administrativeState attribute or configurationState attribute shall cause a stateChange notification.

The administrativeState is used for administratively locking and unlocking the sonetSubnetworkConnection. When unlocked, the sonetSubnetworkConnection functions normally. When in the locked state, the sonetSubnetworkConnection is prohibited from the transport of characteristic information.

The operationalState describes the operability of the sonetSubnetworkConnection. When enabled, the sonetSubnetworkConnection is either partially or fully operable. When disabled, the sonetSubnetworkConnection is totally inoperable.

The userLabel associated with an object instance of this object class is provided solely for the convenience of the manager.

The connectionType attribute indicates connectivity and cardinality relationships among endpoints of the sonetSubnetworkConnection. Possible types include: point-to-point, bicast, exclusive merge, and exclusive composite.

The provisionedProtectionLevel attribute indicates the level of protection established via provisioning. The operationalProtectionLevel attribute indicates the level of protection actually being supported at a given time. Although it is desirable that the operationProtectionLevel be at least as good as the provisionedProtectionLevel, fault conditions or management actions may not permit it. Possible values of either of these attributes are:

highly protected
protected
unprotected
preemptible

There are no restrictions on the relative combinations of values that may be attained by the provisionedProtectionLevel and operationalProtectionLevel attributes.";

```
sonetTrail     MANAGED OBJECT CLASS
    DERIVED FROM sonetTransportEntity;
    CHARACTERIZED BY
        sonetTrailPackage  PACKAGE
                BEHAVIOUR sonetTrailBehaviour;
                ATTRIBUTES
                        aEndTTPs
                                GET,
                        componentConnectionAZList
                                GET,
                        componentConnectionZAList
```

```
                                GET,
                        configurationState
                                GET,
                        relatedRoutingProfile
                                GET,
                        zEndTTPs
                                GET,
                        connectionType
                                GET-REPLACE,
                        operationalProtectionLevel
                                GET,
                        provisionedProtectionLevel
                                GET-REPLACE;;;
    CONDITIONAL PACKAGES
        "Rec. X.721 | ISO/IEC 10165-2 : 1992":administrativeStatePackage
        PRESENT IF "the trail can be administratively locked and unlocked."
;
REGISTERED AS {sIFNLMObjectClass 21};
```

sonetTrailBehaviour BEHAVIOUR
DEFINED AS

"A sonetTrail object class represents the transfer of information between sonetNetworkTTPs. The aEndTTPs and zEndTTPs attributes point to the terminating sonetNetworkTTPs.

A sonetTrail can be composed of sonetSubnetworkConnections and sonetLinkConnections which exist at the next lower level of partitioning. The componentConnectionAZList attribute identifies the currently active components of the sonetTrail going from A to Z. The componentConnectionZAList attribute identifies the currently active components of the sonetTrail going from Z to A.

If supported, the administrativeState is used for administratively locking and unlocking the sonetTrail. When unlocked, the sonetTrail functions normally. When in the locked state, the sonetTrail is prohibited from the transport of characteristic information.

The relatedRoutingProfile attribute points to the sonetRoutingProfile object instance which contains the routing profile for the sonetTrail.

If supported, a change in the value of the administrativeState attribute shall cause a stateChange notification. A change in the value of the configurationState attribute shall cause a stateChange notification.

The configurationState attribute describes the procedural status relative to the process of creation or deletion of a sonetTrail object instance. The configurationState indicates that the sonetTrail is in either a preservice, inservice, postservice, or configuration failure state.

The operationalState describes the operability of the sonetTrail. When enabled, the sonetTrail is either partially or fully operable. When disabled, the sonetTrail is totally inoperable.

The userLabel associated with an object instance of this object class is provided solely for the convenience of the manager.

The connectionType attribute indicates connectivity and cardinality relationships among endpoints of the sonetSubnetworkConnection. Only point-to-point trails are supported in the model.

The provisionedProtectionLevel attribute indicates the level of protection established via provisioning. The operationalProtectionLevel attribute indicates the level of protection actually being supported at a given time. Although it is desirable that the operationProtectionLevel be at least as good as the provisionedProtectionLevel, fault conditions or management actions may not permit it. Possible values of either of these attributes are:

highly protected
protected
unprotected
preemptible

There are no restrictions on the relative combinations of values that may be attained by the provisionedProtectionLevel and operationalProtectionLevel attributes.";

```
sonetTransportEntity                MANAGED OBJECT CLASS
    DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Rec. M.3100 : 1995":createDeleteNotificationsPackage,
        "Rec. M.3100 : 1995":operationalStatePackage,
        "Rec. M.3100 : 1995":stateChangeNotificationPackage,
        "Rec. M.3100 : 1995":userLabelPackage,
        sonetTransportEntityPackage PACKAGE
                BEHAVIOUR sonetTransportEntityBehaviour;
                ATTRIBUTES
                        "Rec. M.3100 : 1995":directionality       GET,
                        signalId        GET,
                        sonetTransportEntityId      GET;;;
REGISTERED AS { sIFNLMObjectClass 22};
```

```
sonetTransportEntityBehaviour BEHAVIOUR
    DEFINED AS
```

"This managed object represents a transport entity which transfers information transparently from input to output, either uni-directionally or bidirectionally.

A change in the value of the operationalState attribute shall cause a stateChange notification.";

```
sonetVirtualLink      MANAGED OBJECT CLASS
    DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Rec. M.3100 : 1995":createDeleteNotificationsPackage,
        "Rec. M.3100 : 1995":userLabelPackage,
        sonetVirtualLinkPackage PACKAGE
                BEHAVIOUR sonetVirtualLinkBehaviour;
                ATTRIBUTES
                        aLinkEnd
                                GET,
                        signalId
                                GET,
                        sonetLinkId
                                GET,
                        zLinkEnd
                                GET;;;
REGISTERED AS { sIFNLMObjectClass 23};

sonetVirtualLinkBehaviour BEHAVIOUR
    DEFINED AS
```

"A sonetVirtualLink is a topological component that provides transport capacity between two endpoints via a fixed (i.e., inflexible routing) relationship. The two endpoints are sonetVirtualLinkEnd object instances associated with peer sonetSubnetworks (i.e., the subnetworks are at the same level of partitioning and are not related via componentSubnetworkList pointer attributes). A single sonetVirtualLink object instance has pointer relationships, through the aLinkEnd and zLinkEnd attributes, with its endpoints. A sonetVirtualLink also represents a set of virtual link connections.

A sonetVirtualLink is automatically created when its endpoints are created. A sonetVirtualLink is automatically deleted when its endpoints are deleted.

The userLabel associated with an object instance of this object class is provided for the convenience of the manager.

";

```
sonetVirtualLinkConnection         MANAGED OBJECT CLASS
    DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Rec. M.3100 : 1995":createDeleteNotificationsPackage,
        "Rec. M.3100 : 1995":userLabelPackage,
        sonetVirtualLinkConnectionPackage PACKAGE
                BEHAVIOUR sonetVirtualLinkConnectionBehaviour;
                ATTRIBUTES
                        aEndCTP
                                GET,
                        "Rec. M.3100 : 1995":directionality
                                GET,
                        signalId
                                GET,
                        virtualLinkConnectionId
                                GET,
                        zEndCTP
                                GET;;;
REGISTERED AS { sIFNLMObjectClass 24};

sonetVirtualLinkConnectionBehaviour BEHAVIOUR
    DEFINED AS
```

"A sonetVirtualLinkConnection is an objet class that represents the fixed capacity of transfer of information of a specific signalId between sonetVirtualCTPs. Only point-to-point virtual link connections are supported by this object class. The aEndCTP and zEndCTP attributes point to the endpoints.

The userLabel associated with an object instance is provided for the convenience of the manager.";

```
sonetVirtualLinkEnd         MANAGED OBJECT CLASS
    DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Rec. M.3100 : 1995":createDeleteNotificationsPackage,
        sonetVirtualLinkEndPackage PACKAGE
                BEHAVIOUR sonetVirtualLinkEndBehaviour;
                ATTRIBUTES
                        sonetLinkEndId
                                GET,
                        serverLayerTP
                                GET,
                        "Rec. M.3100 : 1995":totalTpCount
                                GET,
                        "Rec. M.3100 : 1995":connectedTpCount
                                GET,
                        "Rec. M.3100 : 1995":idleTpCount
                                GET,
                        associatedSnList
                                GET,
            signalId
                GET,
            virtualLinkPointer
                GET,
            virtualCTPList
                GET;;;
REGISTERED AS { sIFNLMObjectClass 25};

sonetVirtualLinkEndBehaviour BEHAVIOUR
  DEFINED AS
```

A virtualLinkEnd object represents the extremity of a virtualLink. It may have an associated set of virtualCTPs.

The totalTpCount is the number of termination points which belong to the virtualLinkEnd. The connectedTpCount is the number of termination points which belong to the virtualLinkEnd and which are used in a subnetwork connection. The idleTpCount is the number of termination points in the virtualLinkEnd which are not used in a subnetwork connection and are available for connection.

The associatedSnList attribute points to the subnetworks which are associated with this linkEnd.

```
";
sonetVirtualCTP     MANAGED OBJECT CLASS
  DERIVED FROM "Rec. X.721 | ISO/IEC 10165-2 : 1992":top;
  CHARACTERIZED BY
    "Rec. M.3100 : 1995":createDeleteNotificationsPackage,
    "Rec. M.3100 : 1995":locationNamePackage,
    sonetVirtualCTPPackage   PACKAGE
        BEHAVIOUR sonetVirtualCTPBehaviour;
        ATTRIBUTES
            virtualCTPChannelIdentifier
                GET,
            sonetNetworkCTPId
                GET,
            containingElement
                GET,
            tPConnectionState
                GET,
            signalId
                GET,
            containingTP
                GET;;;
CONDITIONAL PACKAGES
    sNCPointerPackage
```

PRESENT IF "the termination point may be flexibly
assigned to another termination point or the
assignment of the termination point to another
termination point can be set up and torn down",
connectivityPointerPackage
PRESENT IF "the termination point is always
permanently assigned to another termination point";
REGISTERED AS { sIFNLMObjectClass 26};

sonetVirtualCTPBehaviour BEHAVIOUR
DEFINED AS
"The sonetVirtualCTP (or subclasses) object class represents the extremity of a potential or actual link connection.

The virtualCTPChannelIdentifier attribute defines the slot number used by this virtual point in the containingTP.

The locationName attribute in the locationNamePackage represents a specific geographic location where the virtualCTP is located.

The containingTP is the networkCTP or networkTTP of which this virtual point is a channel termination.

The sNCPointer identifies the sonetSubnetworkConnection, if any, that the virtualCTP is terminating.

The containingElement attribute identifies the sonetVirtualLinkEnd which contains the virtualCTP.

The tPConnectionState attribute provides the connection status for the virtualCTP.

A change in the value of the tPConnectionState attribute shall cause a stateChange notification.

";

sonetVirtualCTPBidirectional MANAGED OBJECT CLASS
DERIVED FROM sonetVirtualCTPSource,sonetVirtualCTPSink;
CHARACTERIZED BY
sonetVirtualCTPBidirectionalPackage PACKAGE
BEHAVIOUR sonetVirtualCTPBidirectionalBehaviour;;;
REGISTERED AS { sIFNLMObjectClass 27};

sonetVirtualCTPBidirectionalBehaviour BEHAVIOUR

```
    DEFINED AS
        "This managed object class represents both a sink and a source virtual
connection termination point.";

sonetVirtualCTPSink               MANAGED OBJECT CLASS
    DERIVED FROM sonetVirtualCTP;
    CHARACTERIZED BY
        sonetVirtualCTPSinkPackage      PACKAGE
                BEHAVIOUR sonetVirtualCTPSinkBehaviour;;;
REGISTERED AS { sIFNLMObjectClass 28};

sonetVirtualCTPSinkBehaviour BEHAVIOUR
    DEFINED AS
        "This managed object class is a unidirectional virtualCTP
which terminates a unidirectional virtualLinkConnection.";

sonetVirtualCTPSource     MANAGED OBJECT CLASS
    DERIVED FROM sonetVirtualCTP;
    CHARACTERIZED BY
        sonetVirtualCTPSourcePackage  PACKAGE
                BEHAVIOUR sonetVirtualCTPSourceBehaviour;;;
REGISTERED AS { sIFNLMObjectClass 29};

sonetVirtualCTPSourceBehaviour BEHAVIOUR
    DEFINED AS
        "This managed object class is a unidirectional virtualCTP
which originates a unidirectional virtualLinkConnection.";

connectivityPointerPackage PACKAGE
ATTRIBUTES
sonetConnectivityPointer
GET;
REGISTERED AS {sIFNLMPackage 1};

emptyListSuppressionPackage PACKAGE
    BEHAVIOUR emptyListSuppressionBehaviour;
REGISTERED AS {sIFNLMPackage 2};

emptyListSuppressionBehaviour BEHAVIOUR
    DEFINED AS
        "The emptyListSuppressionPackage suppresses the
connectionConfigurationSummaryReport if the objectList in the
connectionConfigurationStatusControl object is empty .";

lineSwitchPackage PACKAGE
```

```
    BEHAVIOUR lineSwitchPackageBehaviour;
ATTRIBUTES
subnetworkProtectionStatus
GET;
REGISTERED AS {sIFNLMPackage 3};

lineSwitchPackageBehaviour BEHAVIOUR
    DEFINED AS
        "The lineSwitchPackage indicates the current protection status in a line-based
protection switched subnetwork.";

linkEndCapacityManagementPackage    PACKAGE
    BEHAVIOUR linkEndCapacityManagementPackageBehaviour;
    ACTIONS
        addLinkEndCapacity,
        removeLinkEndCapacity;
REGISTERED AS {sIFNLMPackage 4};

linkEndCapacityManagementPackageBehaviour BEHAVIOUR
    DEFINED AS
        "Indicates that the sonetLinkEnd (or sonetVirtualLinkEnd) represents the end of
a link (or sonetVirtualLink) which crosses the EMS domain boundary.";

primaryTpPackage PACKAGE
    BEHAVIOUR primaryTpPackageBehaviour;
    ATTRIBUTES
        primaryTP
GET;
REGISTERED AS {sIFNLMPackage 5};

primaryTpPackageBehaviour BEHAVIOUR
    DEFINED AS
        "The primaryTpPackage indicates the termination point that is assigned to the
working path under normal conditions in a revertive protection switching arrangement.";

sNCPointerPackage PACKAGE
ATTRIBUTES
sNCPointer
GET;
REGISTERED AS {sIFNLMPackage 6};


sonetAccessGroupId          ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.NameType;
    MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
    BEHAVIOUR              sonetAccessGroupIdBehaviour;
REGISTERED AS {sIFNLMAttribute 1};

sonetAccessGroupIdBehaviour BEHAVIOUR
    DEFINED AS
        "The sonetAccessGroupId is an attribute type whose distinguished value can be
used as an RDN when naming an instance of the sonetAccessGroup object class.";

accessGroupList     ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.AccessGroupList;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
    BEHAVIOUR              accessGroupListBehaviour;
REGISTERED AS {sIFNLMAttribute 2};

accessGroupListBehaviour BEHAVIOUR
    DEFINED AS
        "The accessGroupList attribute lists the accessGroup managed object instances
associated with a subnetwork";

aLinkEnd      ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.EndList;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
    BEHAVIOUR              aLinkEndBehaviour;
REGISTERED AS {sIFNLMAttribute 3};

aLinkEndBehaviour BEHAVIOUR
    DEFINED AS
        "This attribute identifies the aEnd linkEnd object that terminates the link.";

aEndCTP      ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.CTP;
    MATCHES FOR EQUALITY;
    BEHAVIOUR              aEndCTPBehaviour;
REGISTERED AS {sIFNLMAttribute 4};

aEndCTPBehaviour BEHAVIOUR
    DEFINED AS
        "The aEndCTP attribute is a pointer to the subclass of sonetNetworkCTP object
instance terminating the aEnd of the link connection or to the subclass of
sonetVirtualCTP object instance terminating the aEnd of the virtual link connection.";

aEndTPs      ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.EndList;
    MATCHES FOR EQUALITY;
    BEHAVIOUR              aEndTPsBehaviour;
```

```
REGISTERED AS {sIFNLMAttribute 5};

aEndTPsBehaviour BEHAVIOUR
    DEFINED AS
        "This attribute identifies the aEnd networkTTP(s), networkCTP(s), or
virtualCTP(s) that terminate the sonetSubnetwork connection.  In the case of a
unidirectional sonetSubnetwork connection, the object instances identified will be of the
networkTTPSource, networkCTPSource, or virtualCTPSource object class.  In the case
of a bidirectional sonetSubnetwork connection, the object instances identified will be of
the networkTTPBidirectional object class, networkCTPBidirectional object class, or
virtualCTPBidirectional object class.";

aEndTTPs    ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.TTPs;
    MATCHES FOR EQUALITY;
    BEHAVIOUR               aEndTTPsBehaviour;
REGISTERED AS {sIFNLMAttribute 6};

aEndTTPsBehaviour BEHAVIOUR
    DEFINED AS
        "This attribute identifies the aEnd sonetNetworkTTP(s) that terminate the
sonetTrail.  In the case of a unidirectional sonetTrail, the object instances identified
may be of the sonetNetworkTTPSource object class.  In the case of a bidirectional
sonetTrail, the object instances identified will be of the sonetNetworkTTPBidirectional
object class.";

affectedTransmissionResources  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.AffectedTransmissionResources;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
    BEHAVIOUR               affectedTransmissionResourcesBehaviour;
REGISTERED AS {sIFNLMAttribute 7};

affectedTransmissionResourcesBehaviour BEHAVIOUR
    DEFINED AS
        "The affectedTransmissionResources attribute indicates the key transmission
resources which are affected by this alarm.  Any other transmission resource which
depends on this resource either through layering or partitioning will also be affected.";

sonetAlarmAnalysisRoutineId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.NameType;
    MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
    BEHAVIOUR               sonetAlarmAnalysisRoutineIdBehaviour;
REGISTERED AS {sIFNLMAttribute 8};

sonetAlarmAnalysisRoutineIdBehaviour BEHAVIOUR
```

```
    DEFINED AS
        "The sonetAlarmAnalysisRoutineId is an attribute type whose distinguished
value can be used as an RDN when naming an instance of the
sonetAlarmAnalysisRoutine object class.";

alarmedObjectClass        ATTRIBUTE
    WITH ATTRIBUTE SYNTAX  SIFNLMV2Mod.ObjectClass;
    MATCHES FOR EQUALITY;
    BEHAVIOUR alarmedObjectClassBehaviour;
REGISTERED AS {sIFNLMAttribute 9};

alarmedObjectClassBehaviour BEHAVIOUR
    DEFINED AS
        "The object class of the object instance which the alarm analysis routine has
determined to be the root cause of the event.";

alarmedObjectInstance     ATTRIBUTE
    WITH ATTRIBUTE SYNTAX  SIFNLMV2Mod.ObjectInstance;
    MATCHES FOR EQUALITY;
    BEHAVIOUR alarmedObjectInstanceBehaviour;
REGISTERED AS {sIFNLMAttribute 10};

alarmedObjectInstanceBehaviour BEHAVIOUR
    DEFINED AS
        "The object instance of the object which the alarm analysis routine has
determined to be the root cause of the event.";

associatedSnList    ATTRIBUTE
    WITH ATTRIBUTE SYNTAX  SIFNLMV2Mod.AssociatedSnList;
    MATCHES FOR EQUALITY;
    BEHAVIOUR associatedSnListBehaviour;
REGISTERED AS {sIFNLMAttribute 11};

associatedSnListBehaviour BEHAVIOUR
    DEFINED AS
        "A subnetwork object instance associated with the accessGroup, linkEnd, or
virtualLinkEnd object.";

channelIdentifier ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.ChannelIdentifier;
    MATCHES FOR EQUALITY, ORDERING;
    BEHAVIOUR channelIdentifierBehaviour;
REGISTERED AS {sIFNLMAttribute 12};

channelIdentifierBehaviour BEHAVIOUR
```

```
    DEFINED AS
        "This attributes provides a group number (if applicable to the network domain) and a channel number.  The group number (if applicable) specifies the timeslot of the AUG or TUG (VT Group) within its server multiplex. The value shall be the integer which represents the position of the group timeslot in temporal order.  The first timeslot shall be numbered one.  The channel number specifies the timeslot of the networkCTP within its group or server multiplex. The value shall be the integer which represents the position of the timeslot in temporal order.  The first timeslot shall be numbered one.";

clientLayerLinkEnd  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.ClientLinkEnd;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
    BEHAVIOUR               clientLayerLinkEndBehaviour;
REGISTERED AS {sIFNLMAttribute 13};

clientLayerLinkEndBehaviour BEHAVIOUR
    DEFINED AS
        "This attribute identifies the object instances of the related sonetLinkEnd object or sonetVirtualLinkEnd object(and subclasses) in the client layer.  If there are multiple sonetLinkEnd objects they must belong to different sonetLayerNetworkDomains.";

componentLinkList  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.ComponentList;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
    BEHAVIOUR               componentLinkListBehaviour;
REGISTERED AS {sIFNLMAttribute 14};

componentLinkListBehaviour BEHAVIOUR
    DEFINED AS
        "The componentLinkList attribute represents an association between a sonetSubnetwork and its component links or virtualLinks at the next lower level of partitioning.  A sonetSubnetwork may be associated with zero or more component links or virtualLinks.";

componentConnectionAZList       ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.ComponentList;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
    BEHAVIOUR               componentConnectionAZListBehaviour;
REGISTERED AS {sIFNLMAttribute 15};

componentConnectionAZListBehaviour BEHAVIOUR
    DEFINED AS
        "The componentLinkConnectionList attribute represents an association between a sonetSubnetworkConnection or a sonetTrail and its currently active component subnetwork connections, link connections, and/or virtual link connections going in the A
```

to Z direction. A sonetSubnetwork connection or a sonetTrail may be associated with zero or more component connections.";

```
componentSubnetworkList ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.ComponentList;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
    BEHAVIOUR             componentSubnetworkListBehaviour;
REGISTERED AS {sIFNLMAttribute 16};
```

componentSubnetworkListBehaviour BEHAVIOUR
DEFINED AS

"The componentSubnetworkList attribute represents an association between a sonetSubnetwork and its component sonetSubnetworks at the next lower level of partitioning. A sonetSubnetwork may be associated with zero or more component sonetSubnetworks.";

```
configurationState ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.ConfigurationState;
    MATCHES FOR EQUALITY;
    BEHAVIOUR configurationStateBehaviour;
REGISTERED AS {sIFNLMAttribute 18};
```

configurationStateBehaviour BEHAVIOUR
DEFINED AS

"The configurationState attribute describes the procedural status relative to the process of creation or deletion of sonetSubnetworkConnection and sonetTrail object instances. The possible states of the configurationState attribute are: preservice, inservice, postservice, and configuration failure state.

Preservice: The state that exists after the creation of the transport entity object instance (resulting from a set-up request) and prior to either the successful completion of all supporting NE-level cross-connections or the failure of a set-up process. The transport entity does not support service during the preservice state.

Inservice: The state that results after the successful completion of all supporting NE-level cross-connections. The transport entity may provide service while in the inservice state.

Postservice: The state that exists after a request for deletion of the transport entity object instance has been received and prior to either the successful tear-down of all supporting NE-level cross-connections or the failure of the tear-down process. The transport entity does not support service while in the postservice state.

Configuration failure: The state of a transport entity that exists after either:
a) the failure to undo the cross-connections in an aborted set-up process, or
b) an aborted tear-down process.
The transport entity does not support service during the configuration failure state.";

sonetConnectionPendingStatusControlId ATTRIBUTE
WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.NameType;
MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
BEHAVIOUR sonetConnectionPendingStatusControlIdBehaviour;
REGISTERED AS {sIFNLMAttribute 19};

sonetConnectionPendingStatusControlIdBehaviour BEHAVIOUR
DEFINED AS
"The sonetConnectionPendingStatusControlId is an attribute type whose distinguished value can be used as an RDN when naming an instance of the sonetConnectionSetupStatusControl object class.";

connectionType ATTRIBUTE
WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.ConnectionType;
MATCHES FOR EQUALITY;
BEHAVIOUR connectionTypeBehaviour;
REGISTERED AS {sIFNLMAttribute 20};

connectionTypeBehaviour BEHAVIOUR
DEFINED AS
"The connectionType attribute describes connectivity and cardinality relationships among endpoints of the sonetSubnetworkConnection and sonetTrail. Currently, the sonetTrail object supports only the pointToPoint type. Possible types for sonetSubnetworkConnections include:

pointToPoint - unidirectional or bidirectional transport between two points

bicast - unidirectional signal split at a single source into two identical signals

pointToMultipoint - unidirectional transport from a single source to two or more sinks

exclusiveMerge - unidirectional signals from two aEndTPs sources are accepted; at the zEndTPs only one of the two signals is selected and passed on via the sink.

exclusiveComposite - bidirectional transport involving bicast and exclusive merge behavior between a single point and two other points.

dropAndContinueComplex - connection type used at the point of dual ring interworking at an interworking node on a UPSR. The connection type requires in the unidirectional case that the signal coming from one high speed side be continued to the other high speed side and that the low speed output is selected from the the inputs of both high speed sides. In the bidirectional case the configuration adds a signal going from the low speed side to the high speed side which is the source of the drop and

```
continue.";


sonetConnectivityPointer ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.ObjectInstance;
    MATCHES FOR EQUALITY;
    BEHAVIOUR     sonetConnectivityPointerBehaviour;
REGISTERED AS {sIFNLMAttribute 21};

sonetConnectivityPointerBehaviour BEHAVIOUR
    DEFINED AS
        "The sonetConnectivityPointer attribute points to the permanently assigned
termination point.";


containingElement  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.ContainingElement;
    MATCHES FOR EQUALITY;
    BEHAVIOUR containingElementBehaviour;
REGISTERED AS {sIFNLMAttribute 22};

containingElementBehaviour BEHAVIOUR
    DEFINED AS
        "This attribute identifies the sonetLinkEnd or sonetAccessGroup the networkCTP
is related to, the sonetAccessGroup the networkTTP is related to, or the
sonetVirtualLinkEnd the virtualCTP is related to.";


highestPriorityNELalarmRecords ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.AlarmRecords;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
    BEHAVIOUR            highestPriorityNELalarmRecordsBehaviour;
REGISTERED AS {sIFNLMAttribute 23};

highestPriorityNELalarmRecordsBehaviour BEHAVIOUR
    DEFINED AS
        "The highestPriorityNELalarmRecords attribute is a set of NEL alarm record
instances which are correlated to this RCAA record.  These NEL alarms are considered
to be of primary importance in determining the RCAA result.";


sonetLayerNetworkDomainId      ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.NameType;
    MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
    BEHAVIOUR            sonetLayerNetworkDomainIdBehaviour;
```

```
REGISTERED AS {sIFNLMAttribute 24};

sonetLayerNetworkDomainIdBehaviour BEHAVIOUR
    DEFINED AS
        "The sonetLayerNetworkDomainId is an attribute type whose distinguished value
can be used as an RDN when naming an instance of the sonetLayerNetworkDomain
object class.";


linkEndList    ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.LinkEndList;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
    BEHAVIOUR               linkEndListBehaviour;
REGISTERED AS {sIFNLMAttribute 25};

linkEndListBehaviour BEHAVIOUR
    DEFINED AS
        "The linkEndList attribute lists the linkEnd and/or virtualLinkEnd managed object
instances associated with a subnetwork";



sonetLinkId   ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.NameType;
    MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
    BEHAVIOUR               sonetLinkIdBehaviour;
REGISTERED AS {sIFNLMAttribute 26};

sonetLinkIdBehaviour BEHAVIOUR
    DEFINED AS
        "The sonetLinkId is an attribute type whose distinguished value can be used as
an
RDN when naming an instance of the sonetLink or sonetVirtualLink object class.";


linkPointer    ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.LinkPointer;
    MATCHES FOR EQUALITY;
    BEHAVIOUR               linkPointerBehaviour;
REGISTERED AS {sIFNLMAttribute 27};

linkPointerBehaviour BEHAVIOUR
    DEFINED AS
        "The linkPointer attribute identifies the sonetLink object instance which is
associated to the sonetLinkEnd object instance or the sonetVirtualLink object instance
```

which is associated to the sonetVirtualLinkEnd object instance.";

faultLocation ATTRIBUTE
WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.FaultLocation;
MATCHES FOR EQUALITY, SUBSTRINGS, SET-COMPARISON, SET-INTERSECTION;
BEHAVIOUR faultLocationBehaviour;
REGISTERED AS {sIFNLMAttribute 28};

faultLocationBehaviour BEHAVIOUR
DEFINED AS
"The faultLocation will contain a possibly empty list of geographic locations at which the root cause alarm is present. For example if the root cause is determined to be a fiber cut the faultLocation would list the geographic locations of the end points of the fiber.";

lowerPriorityNELalarmRecords ATTRIBUTE
WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.AlarmRecords;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
BEHAVIOUR lowerPriorityNELalarmRecordsBehaviour;
REGISTERED AS {sIFNLMAttribute 29};

lowerPriorityNELalarmRecordsBehaviour BEHAVIOUR
DEFINED AS
"The lowerPriorityNELalarmRecords attribute is a set of NEL alarm record instances which are correlated to this RCAA record and which have not been referenced in highestPriorityNELalarmRecords.";

maxHops ATTRIBUTE
WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.MaxHops;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR maxHopsBehaviour;
REGISTERED AS {sIFNLMAttribute 30};

maxHopsBehaviour BEHAVIOUR
DEFINED AS
"The maxHops attribute is the maximum number of network elements that a connection or trail may traverse.";

sonetNetworkCTPIdATTRIBUTE
WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.NameType;

MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
BEHAVIOUR sonetNetworkCTPIdBehaviour;
REGISTERED AS {sIFNLMAttribute 31};

sonetNetworkCTPIdBehaviour BEHAVIOUR
DEFINED AS
"The sonetNetworkCTPId is an attribute type whose distinguished value can be used as an RDN when naming an instance of the sonetNetworkCTP (or subclasses) or sonetVirtualCTP object class.";

numberOfNELAlarms ATTRIBUTE
WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.Integer;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR numberOfNELAlarmsBehaviour;
REGISTERED AS {sIFNLMAttribute 32};

numberOfNELAlarmsBehaviour BEHAVIOUR
DEFINED AS
"The numberOfNELAlarms attribute will indicate the number of NEL alarms which are correlated to this record. If only one alarm is correlated the alarm analysis routine was unable to correlate the NEL alarm to other alarms and this is a raw alarm.";

serviceAffecting ATTRIBUTE
WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.Boolean;
MATCHES FOR EQUALITY;
BEHAVIOUR serviceAffectingBehaviour;
REGISTERED AS {sIFNLMAttribute 33};

serviceAffectingBehaviour BEHAVIOUR
DEFINED AS
"The serviceAffecting attribute will indicate whether the alarm analysis routine has determined this alarm to be service affecting.";

sonetNetworkTTPId ATTRIBUTE
WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.NameType;
MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
BEHAVIOUR sonetNetworkTTPIdBehaviour;
REGISTERED AS {sIFNLMAttribute 34};

sonetNetworkTTPIdBehaviour BEHAVIOUR
DEFINED AS
"The sonetNetworkTTPId is an attribute type whose distinguished value can be

```
used as an RDN when naming an instance of the sonetNetworkTTP (or subclasses)
object class.";

operationalProtectionLevel ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod. ProtectionLevel;
    MATCHES FOR EQUALITY;
    BEHAVIOUR operationalProtectionLevelBehaviour;
REGISTERED AS {sIFNLMAttribute 35};

operationalProtectionLevelBehaviour BEHAVIOUR
    DEFINED AS
        " The operationalProtectionLevel attribute indicates the level of protection
actually being supported at a given time. Although it is desirable that the
operationProtectionLevel be at least as good as the provisionedProtectionLevel, fault
conditions or management actions may not permit it. Possible values of this attribute
are:
        highly protected
        protected
        unprotected
        preemptible
There are no restrictions on the relative combinations of values that may be attained by
the provisionedProtectionLevel and operationalProtectionLevel attributes.";

primaryTP ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.ObjectInstance;
    MATCHES FOR EQUALITY;
    BEHAVIOUR primaryTPBehaviour;
REGISTERED AS {sIFNLMAttribute 36};

primaryTPBehaviour BEHAVIOUR
    DEFINED AS
        "The primaryTP attribute describes the termination point associated with the
normally working path in a revertive mode protection scheme.";

provisionedProtectionLevel ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod. ProtectionLevel;
    MATCHES FOR EQUALITY;
    BEHAVIOUR provisionedProtectionLevelBehaviour;
REGISTERED AS {sIFNLMAttribute 37};

provisionedProtectionLevelBehaviour BEHAVIOUR
    DEFINED AS
        "The provisionedProtectionLevel attribute indicates the level of protection
established via provisioning. Possible values of this attribute are:
        highly protected
        protected
        unprotected
        preemptible.";

tPList          ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.TPList;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
    BEHAVIOUR               tPListBehaviour;
REGISTERED AS {sIFNLMAttribute 38};

tPListBehaviour BEHAVIOUR
    DEFINED AS
        "This attribute identifies the (subclasses of) networkTTPs or networkCTPs
grouped in the sonetAccessGroup.";


relatedRoutingProfile       ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.RoutingProfile;
    MATCHES FOR EQUALITY;
    BEHAVIOUR               relatedRoutingProfileBehaviour;
REGISTERED AS {sIFNLMAttribute 39};

relatedRoutingProfileBehaviour BEHAVIOUR
    DEFINED AS
        "The relatedRoutingProfile attribute is a pointer to the sonetRoutingProfile object
instance which is associated with the sonetSubnetworkConnection or sonetTrail object
instance.";

routeDescriptionListATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.RouteDescriptionList;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
    BEHAVIOUR               routeDescriptionListBehaviour;
REGISTERED AS {sIFNLMAttribute 40};

routeDescriptionListBehaviour BEHAVIOUR
    DEFINED AS
        "The routeDescriptionList attribute is a list of objects (such as links,
sonetSubnetworks, sonetLinkEnds, sonetSubnetworkConnections etc.) and their use in
routing (excluded, mandatory, preferred, no common facilities).";

sonetRoutingProfileId       ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.NameType;
    MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
    BEHAVIOUR               sonetRoutingProfileIdBehaviour;
REGISTERED AS {sIFNLMAttribute 41};
```

```
sonetRoutingProfileIdBehaviour BEHAVIOUR
    DEFINED AS
        "The sonetRoutingProfileId is an attribute type whose distinguished value can be
used as an RDN when naming an instance of the sonetRoutingProfile object class.";

serverLayerNTTP    ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.ServerNTTP;
    MATCHES FOR EQUALITY;
    BEHAVIOUR            serverLayerNTTPBehaviour;
REGISTERED AS {sIFNLMAttribute 42};

serverLayerNTTPBehaviour BEHAVIOUR
    DEFINED AS
        "This attribute identifies the object instance of the related networkTTP (and
subclasses) in the server layer.";


signalId ATTRIBUTE
WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.SignalId;
MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
        BEHAVIOUR
        signalIdBehaviour BEHAVIOUR
                DEFINED AS "
        This attribute defines the characteristic information of the
layer (in the G.805 sense) to which the entity under consideration belongs.
 It is used to determine whether sub-network connection/connectivity is
possible. The signal Id may be a simple rate and format or may be a bundle
of entities with the same characteristic information which form an aggregate
signal.";;
REGISTERED AS { sIFNLMAttribute 43};

sNCPointer         ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.SNCPointer;
    MATCHES FOR EQUALITY;
    BEHAVIOUR            sNCPointerBehaviour;
REGISTERED AS {sIFNLMAttribute 44};

sNCPointerBehaviour BEHAVIOUR
    DEFINED AS
        "This attribute identifies the object instance of the sonetSubnetworkConnection
that the networkTTP, networkCTP, or virtualCTP terminates.";


sonetSubnetworkId  ATTRIBUTE
```

WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.NameType;
MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
BEHAVIOUR sonetSubnetworkIdBehaviour;
REGISTERED AS {sIFNLMAttribute 45};

sonetSubnetworkIdBehaviour BEHAVIOUR
DEFINED AS
"The sonetSubnetworkId is an attribute type whose distinguished value can be used as an RDN when naming an instance of the sonetSubnetwork object class.";

subnetworkProtectionStatus ATTRIBUTE
WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.SnProtectionStatus;
MATCHES FOR EQUALITY;
BEHAVIOUR subnetworkProtectionStatusBehaviour;
REGISTERED AS {sIFNLMAttribute 46};

subnetworkProtectionStatusBehaviour BEHAVIOUR
DEFINED AS
"The subnetworkProtectionStatus atttribute indicates if the line level protection switching mechanisms in a subnetwork are in a normal state or if one or more line protection switches are in affect.";

subnetworkType ATTRIBUTE
WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.SubnetworkType;
MATCHES FOR EQUALITY;
BEHAVIOUR subnetworkTypeBehaviour;
REGISTERED AS {sIFNLMAttribute 47};

subnetworkTypeBehaviour BEHAVIOUR
DEFINED AS
"The subnetworkType attribute provides information about the element level resource(s) the subnetwork is abstracting.";

tPConnectionState ATTRIBUTE
WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.TPConnectionState;
MATCHES FOR EQUALITY;
BEHAVIOUR tPConnectionStateBehaviour;
REGISTERED AS {sIFNLMAttribute 48};

tPConnectionStateBehaviour BEHAVIOUR
DEFINED AS
"This attribute provides the connection status for networkCTPs, virtualCTPs and networkTTPs. The value of this attribute will be not connected if the networkCTP, virtualCTP, or networkTTP is not involved in a subnetwork connection. Otherwise, the value will reflect whether the networkCTP, virtualCTP, or networkTTP is sinking,

```
sourcing, or sinking and sourcing a subnetwork connection(s).";

tPPointer ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.TPPointer;
    MATCHES FOR EQUALITY;
    BEHAVIOUR tPPointerBehaviour;
REGISTERED AS {sIFNLMAttribute 49};

tPPointerBehaviour BEHAVIOUR
    DEFINED AS
        "This attribute points to the NE View termination point the object is abstracting.
The attribute is a choice of an object instance if the NE View termination point is
represented by a CMIP object, a string if the termination point is defined by a TL1
interface, or NULL if the NE View object is not present.";

sonetTransportEntityId      ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.TransportID;
    MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
    BEHAVIOUR           sonetTransportEntityIdBehaviour;
REGISTERED AS {sIFNLMAttribute 50};

sonetTransportEntityIdBehaviour BEHAVIOUR
    DEFINED AS
        "The sonetTransportEntityId is an attribute type whose distinguished value can
be used as an RDN when naming an instance of the sonetTransportEntity object class
or subclasses.";

zEndCTP     ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.CTP;
    MATCHES FOR EQUALITY;
    BEHAVIOUR           zEndCTPBehaviour;
REGISTERED AS {sIFNLMAttribute 51};

zEndCTPBehaviour BEHAVIOUR
    DEFINED AS
        "The zEndCTP attribute is a pointer to the sonetNetworkCTP (and subclasses)
object instance terminating the zEnd of the link connection.";

zLinkEnd    ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.EndList;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
    BEHAVIOUR           zLinkEndBehaviour;
REGISTERED AS {sIFNLMAttribute 52};

zLinkEndBehaviour BEHAVIOUR
```

DEFINED AS
"This attribute identifies the zEnd sonetLinkEnd terminating the link or the zEnd sonetVirtualLinkEnd terminating the sonetVirtualLink.";

zEndTPs ATTRIBUTE
WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.EndList;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
BEHAVIOUR zEndTPsBehaviour;
REGISTERED AS {sIFNLMAttribute 53};

zEndTPsBehaviour BEHAVIOUR
DEFINED AS
"This attribute identifies the zEnd networkTTP, networkCTP, or virtualCTP that terminates the sonetSubnetwork connection. In the case of a unidirectional sonetSubnetwork connection, the object instances identified will be of the networkTTPSink, networkCTPSink, or virtualCTPSink object class. In the case of a bidirectional sonetSubnetwork connection, the object instances identified will be of the networkTTPBidirectional, networkCTPBidirectional, or virtualCTPBidirectional object class.";

zEndTTPs ATTRIBUTE
WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.TTPs;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
BEHAVIOUR zEndTTPsBehaviour;
REGISTERED AS {sIFNLMAttribute 54};

zEndTTPsBehaviour BEHAVIOUR
DEFINED AS
"This attribute identifies the zEnd networkTTPs that terminate the sonetTrail. In the case of a unidirectional sonetTrail, the object instances identified will be of the networkTTPSink object class. In the case of a bidirectional sonetTrail, the object instances identified will be of the networkTTPBidirectional object class.";

sonetLinkEndId ATTRIBUTE
WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.NameType;
MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
BEHAVIOUR sonetLinkEndIdBehaviour;
REGISTERED AS {sIFNLMAttribute 55};

sonetLinkEndIdBehaviour BEHAVIOUR
DEFINED AS
"The sonetLinkEndId is an attribute type whose distinguished value can be used as an RDN when naming an instance of the sonetLinkEnd or sonetVirtualLinkEnd object class.";

```
nCTPList                ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.TPList;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
    BEHAVIOUR               nCTPListBehaviour;
REGISTERED AS {sIFNLMAttribute 56};

nCTPListBehaviour BEHAVIOUR
    DEFINED AS
        "This attribute identifies the (subclasses of) sonetNetworkCTPs grouped in the
sonetLinkEnd or sonetVirtualCTPs grouped in the sonetVirtualLinkEnd.";

componentConnectionZAList       ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.ComponentList;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
    BEHAVIOUR               componentConnectionZAListBehaviour;
REGISTERED AS {sIFNLMAttribute 57};

componentConnectionZAListBehaviour BEHAVIOUR
    DEFINED AS
        "The componentConnectionList attribute represents an association between a
sonetSubnetworkConnection or a sonetTrail and its currently active component
subnetwork connections, link connections, and virtual link connections going in the Z to
A direction.  A sonetSubnetwork connection or a sonetTrail may be associated with
zero or more component connections.";

containingTP ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.TP;
    MATCHES FOR EQUALITY;
    BEHAVIOUR               containingTPBehaviour;
REGISTERED AS {sIFNLMAttribute 58};

containingTPBehaviour BEHAVIOUR
    DEFINED AS
        "The containingTP is the networkCTP or networkTTP of which a virtual point is a
channel termination.";

serverLayerTP        ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.TP;
    MATCHES FOR EQUALITY;
    BEHAVIOUR               serverLayerTPBehaviour;
REGISTERED AS {sIFNLMAttribute 59};

serverLayerTPBehaviour BEHAVIOUR
    DEFINED AS
        "This attribute identifies the object instance of the related networkTTP or
```

```
networkCTP (and subclasses) in the server layer.";

virtualCTPChannelIdentifier        ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.VirtualCTPChannelIdentifier;
    MATCHES FOR EQUALITY;
    BEHAVIOUR              virtualCTPChannelIdentifierBehaviour;
REGISTERED AS {sIFNLMAttribute 60};

virtualCTPChannelIdentifierBehaviour BEHAVIOUR
    DEFINED AS
        "This attribute identifies the slot that a virtual point
occupies in a containing point. The information contained in this
attribute depends on the characteristic information of the virtual
point and of the containing point as follows:

For a DS1 or VT1.5 over an Ocn: a-b-c where a is the STS1 number, 1
through n, b is the VTG number, 1 through 7, and c is the VT1.5 number,
1 through 4.

For a DS1 or VT1.5 over an STS1: b-c where b is the VTG number 1 through
7 and c is the VT1.5 number, 1 through 4.

For a DS2 or VT6 over an Ocn: a-b where a is the STS1 number 1 through n, and b is
the VTG number 1 through 7.

For DS2 or VT6 over an STS1: b where b is the STS1 number 1 through n.

For DS3 or STS1 over Ocn: a where a is the STS1 number 1 through n.

For STS3c over OCn: a where a=3*((X-1)/3)+1, X = 1 through n.

For STS12c over OCn: a where a = 12*((X-1)/12) + 1, X = 1 through n.

For STS48c over OCn: a where a = 48*((X-1)/48) + 1, X = 1 through n.

For ATM over OCn: a-b-c where a is the sts number, 1 through n, b is the VPI, 1
through 4095, and c is the VCI number 0 through 65535.
";

virtualCTPList                ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.TPList;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
    BEHAVIOUR              virtualCTPListBehaviour;
REGISTERED AS {sIFNLMAttribute 61};
```

```
virtualCTPListBehaviour BEHAVIOUR
    DEFINED AS
        "This attribute identifies the (subclasses of)
sonetVirtualCTPs grouped in the sonetVirtualLinkEnd.This list will
never be empty.";

virtualLinkConnectionId     ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.NameType;
    MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
    BEHAVIOUR               virtualLinkConnectionIdBehaviour;
REGISTERED AS {sIFNLMAttribute 62};

virtualLinkConnectionIdBehaviour BEHAVIOUR
    DEFINED AS
        "The virtualLinkConnectionId is an attribute type whose distinguished value can
be used as an RDN when naming an instance of the sonetVirtualLinkConnection object
class or subclasses.";

virtualLinkPointer    ATTRIBUTE
    WITH ATTRIBUTE SYNTAX SIFNLMV2Mod.ObjectInstance;
    MATCHES FOR EQUALITY;
    BEHAVIOUR               virtualLinkPointerBehaviour;
REGISTERED AS {sIFNLMAttribute 63};

virtualLinkPointerBehaviour BEHAVIOUR
    DEFINED AS
        "The virtualLinkPointer attribute identifies the sonetVirtualLink object instance
which is associated to the sonetVirtualLinkEnd object instance.";

addLink               ACTION
    BEHAVIOUR     addLinkBehaviour;
    MODE CONFIRMED;
    PARAMETERS
        "Rec. M.3100 : 1995":generalErrorParameter;
    WITH INFORMATION SYNTAX SIFNLMV2Mod.AddLinkInformation;
    WITH REPLY SYNTAX SIFNLMV2Mod.AddLinkReply;
REGISTERED AS {sIFNLMAction 1};

addLinkBehaviour BEHAVIOUR
    DEFINED AS
        "The addLink action request is sent by a managing system (e.g., the NMS) to the
managed system (e.g., the EMS) to direct the sonetLayerNetworkDomain (and
subclasses) to setup a link.  Upon receipt of this request, the managed system creates
a sonetLink object instance and sets the related linkPointer attribute in the
sonetLinkEnd object instances. Also, the userLabel is returned to the managing
```

```
system.";

addLinkCapacity     ACTION
    BEHAVIOUR     addLinkCapacityBehaviour;
    MODE CONFIRMED;
    PARAMETERS
        "Rec. M.3100 : 1995":generalErrorParameter;
    WITH INFORMATION SYNTAX SIFNLMV2Mod.AddLinkCapacityInformation;
    WITH REPLY SYNTAX SIFNLMV2Mod.AddLinkCapacityReply;
REGISTERED AS {sIFNLMAction 2};

addLinkCapacityBehaviour BEHAVIOUR
    DEFINED AS
        "The addLinkCapacity action request is sent by a managing system (e.g., the
NMS) to the managed system (e.g., the EMS) to direct the sonetLink (and subclasses)
to add link capacity.  Upon receipt of this request, if server layer bandwidth is available,
the managed system creates two (subclasses of) sonetNetworkCTP object instances
for each integral increment in link capacity specified in the request.  In addition, the
managed system will also add pointers to the newly created sonetNetworkCTPs in the
nCTPList attributes of the linkEnd object instances.  If server layer bandwidth is not
available, the managed system can either reconfigure the server layer to provide
additional bandwidth or deny the request.";


addLinkEndCapacity          ACTION
    BEHAVIOUR addLinkEndCapacityBehaviour;
    MODE CONFIRMED;
    PARAMETERS
        "Rec. M.3100 : 1995":generalErrorParameter;
       WITH INFORMATION SYNTAX SIFNLMV2Mod.AddLinkEndCapacityInformation;
       WITH REPLY SYNTAX SIFNLMV2Mod.AddLinkEndCapacityReply;
REGISTERED AS {sIFNLMAction 3};

addLinkEndCapacityBehaviour BEHAVIOUR
    DEFINED AS
        "The addLinkEndCapacity action request is sent by a managing system (e.g., the
NMS) to the managed system (e.g., the EMS) to direct the sonetLinkEnd (and
subclasses) to add capacity.  Upon receipt of this request, if server layer bandwidth is
available, the managed system creates one sonetNetworkCTP object instance
(subclasses of) for each integral increment in capacity specified in the request. If server
layer bandwidth is not available, the managed system can either reconfigure the server
layer to provide additional bandwidth or deny the request.";

modifySNCConnectionType                ACTION
    BEHAVIOUR     modifySNCConnectionTypeBehaviour;
    MODE CONFIRMED;
    PARAMETERS
        "Rec. M.3100 : 1995":generalErrorParameter;
    WITH INFORMATION SYNTAX SIFNLMV2Mod.ModifySNCConnectionTypeInformation;
    WITH REPLY SYNTAX SIFNLMV2Mod.ModifySNCConnectionTypeReply;
REGISTERED AS {sIFNLMAction 4};

modifySNCConnectionTypeBehaviour BEHAVIOUR
    DEFINED AS
```

"The modifySNCConnectionType action request is sent by a managing system (e.g., the NMS) to the managed system (e.g., the EMS) to direct the sonetSubnetwork (and subclasses) to modify the connectionType attribute of a sonetSubnetworkConnection. Upon receipt of this request, the managed system determines whether or not the resources exist to fulfill the request. If the resources are available, the managed system fulfills the request and notifies the managing system of the successful action.

The requestedEffortLevel describes the level of commitment or guarantee that is sought by the requestor in achieving the requested modify connection type action. Possible values are:

required minimum - grant the requested action if the connection can be established with a provisioned protection level that is the same as or higher than the requested protection level
required exact - grant the requested action if the connection can be established with a provisioned protection level that is the same as the requested protection level
best effort - grant the requested action if the connection can be established with any value of provisioned protection ";

```
modifySNCProtectionLevel        ACTION
    BEHAVIOUR    modifySNCProtectionLevelBehaviour;
    MODE CONFIRMED;
    PARAMETERS
        "Rec. M.3100 : 1995":generalErrorParameter;
        WITH INFORMATION SYNTAX SIFNLMV2Mod.ModifySNCProtectionLevelInformation;
    WITH REPLY SYNTAX SIFNLMV2Mod.ModifySNCProtectionLevelReply;
REGISTERED AS {sIFNLMAction 5};

modifySNCProtectionLevelBehaviour BEHAVIOUR
    DEFINED AS
```

"The modifySNCProtectionLevel action request is sent by a managing system (e.g., the NMS) to the managed system (e.g., the EMS) to direct the sonetSubnetwork (and subclasses) to modify the provisionedProtectionLevel attribute of a sonetSubnetworkConnection. Upon receipt of this request, the managed system determines whether or not the resources exist to fulfill the request. If the resources are available, the managed system fulfills the request and notifies the managing system of the successful action.

The requestedEffortLevel describes the level of commitment or guarantee that is sought by the requestor in achieving the requested protection level in a modify connection protection action. Possible values are:

required minimum - grant the requested action if the connection can be established with a provisioned protection level that is the same as or higher than the requested protection level
required exact - grant the requested action if the connection can be established with a provisioned protection level that is the same as the requested protection level
best effort - grant the requested action if the connection can be established with any value of provisioned protection ";

```
modifyTrailProtectionLevel          ACTION
    BEHAVIOUR     modifyTrailProtectionLevelBehaviour;
    MODE CONFIRMED;
    PARAMETERS
        "Rec. M.3100 : 1995":generalErrorParameter;
    WITH INFORMATION SYNTAX
SIFNLMV2Mod.ModifyTrailProtectionLevelInformation;
    WITH REPLY SYNTAX SIFNLMV2Mod.ModifyTrailProtectionLevelReply;
REGISTERED AS {sIFNLMAction 6};

modifyTrailProtectionLevelBehaviour BEHAVIOUR
    DEFINED AS
```

"The modifyTrailProtectionLevel action request is sent by a managing system (e.g., the NMS) to the managed system (e.g., the EMS) to direct the sonetSubnetwork (and subclasses) to modify the provisionedProtectionLevel attribute of a sonetTrail. Upon receipt of this request, the managed system determines whether or not the resources exist to fulfill the request. If the resources are available, the managed system fulfills the request and notifies the managing system of the successful action.

The requestedEffortLevel describes the level of commitment or guarantee that is sought by the requestor in achieving the requested protection level in a modify trail protection action. Possible values are:

required minimum - grant the requested action if the trail can be established with a provisioned protection level that is the same as or higher than the requested protection level
required exact - grant the requested action if the trail can be established with a provisioned protection level that is the same as the requested protection level

```
best effort - grant the requested action if the trail can be established with any value of
provisioned protection ";


releaseLinkConnection     ACTION
    BEHAVIOUR     releaseLinkConnectionBehaviour;
    MODE CONFIRMED;
    PARAMETERS
        "Rec. M.3100 : 1995":generalErrorParameter;
    WITH INFORMATION SYNTAX
    SIFNLMV2Mod.ReleaseLinkConnectionInformation;
REGISTERED AS {sIFNLMAction 7};

releaseLinkConnectionBehaviour BEHAVIOUR
    DEFINED AS
        "The releaseLinkConnection action request is sent by a managing system (e.g.,
the NMS) to the managed system (e.g., the EMS) to direct the sonetLink (and
subclasses) to release a link connection.  Upon receipt of this request, the managed
system deletes the link connection object instance. The information provided by the
managing system will include the sonetTransportEntityID.";


releaseSNC  ACTION
    BEHAVIOUR     releaseSNCBehaviour;
    MODE CONFIRMED;
    PARAMETERS
        "Rec. M.3100 : 1995":generalErrorParameter;
    WITH INFORMATION SYNTAX SIFNLMV2Mod.ReleaseSNCInformation;
REGISTERED AS {sIFNLMAction 8};

releaseSNCBehaviour BEHAVIOUR
    DEFINED AS
        "The releaseSNC action request is sent by a managing system (e.g., the NMS)
to the managed system (e.g., the EMS) to direct the sonetSubnetwork (and subclasses)
to release a sonetSubnetwork connection.  Upon receipt of this request, the managed
system deletes the sonetSubnetworkConnection object instance and sets the related
sNCPointers to null. The information provided by the managing system will include the
sonetTransportEntityID.";


releaseTrail  ACTION
    BEHAVIOUR     releaseTrailBehaviour;
    MODE CONFIRMED;
    PARAMETERS
        "Rec. M.3100 : 1995":generalErrorParameter;
    WITH INFORMATION SYNTAX SIFNLMV2Mod.ReleaseTrailInformation;
REGISTERED AS {sIFNLMAction 9};

releaseTrailBehaviour BEHAVIOUR
    DEFINED AS
        "The releaseTrail action request is sent by a managing system (e.g., the NMS) to
the managed system (e.g., the EMS) to direct the sonetLayerNetworkDomain (and
subclasses) to release a sonetTrail.  Upon receipt of this request, the managed system
deletes the sonetTrail object instance. The information provided by the managing
system will include the sonetTransportEntityID.";

removeLink   ACTION
    BEHAVIOUR     removeLinkBehaviour;
    MODE CONFIRMED;
    PARAMETERS
        "Rec. M.3100 : 1995":generalErrorParameter;
    WITH INFORMATION SYNTAX
    SIFNLMV2Mod.RemoveLinkInformation;
REGISTERED AS {sIFNLMAction 10};

removeLinkBehaviour BEHAVIOUR
    DEFINED AS
        "The removeLink action request is sent by a managing system (e.g., the NMS) to
the managed system (e.g., the EMS) to direct the sonetLayerNetworkDomain (and
subclasses) to remove the specified link.  Upon receipt of this request, the managed
system deletes the link object instance and sets the linkPointer attributes in the related
sonetLinkEnd object instances to null. The information provided by the managing
system will include the sonetTransportEntityID. This action will fail if there are any
sonetLinkConnection object instances named by the specified link.";


removeLinkCapacity          ACTION
    BEHAVIOUR     removeLinkCapacityBehaviour;
    MODE CONFIRMED;
    PARAMETERS
        "Rec. M.3100 : 1995":generalErrorParameter;
    WITH INFORMATION SYNTAX
    SIFNLMV2Mod.RemoveLinkCapacityInformation;
REGISTERED AS {sIFNLMAction 11};

removeLinkCapacityBehaviour BEHAVIOUR
    DEFINED AS
        "The removeLinkCapacity action request is sent by a managing system (e.g., the
NMS) to the managed system (e.g., the EMS) to direct the sonetLink (and subclasses)
to remove excess link capacity.  Upon receipt of this request, the managed system
```

deletes the pointers from each sonetLinkEnd to two (subclasses of) sonetNetworkCTP object instances for each integral decrement specified in the request. This action will fail if there are an insufficient number of available sonetNetworkCTPs.";

```
removeLinkEndCapacity    ACTION
    BEHAVIOUR     removeLinkEndCapacityBehaviour;
    MODE CONFIRMED;
    PARAMETERS
        "Rec. M.3100 : 1995":generalErrorParameter;
    WITH INFORMATION SYNTAX
    SIFNLMV2Mod.RemoveLinkEndCapacityInformation;
    REGISTERED AS {sIFNLMAction 12};
```

removeLinkEndCapacityBehaviour BEHAVIOUR
    DEFINED AS

"The removeLinkEndCapacity action request is sent by a managing system (e.g., the NMS) to the managed system (e.g., the EMS) to direct the sonetLinkEnd (and subclasses) to remove excess capacity. Upon receipt of this request, the managed system deletes pointers from the sonetLinkEnd to one sonetNetworkCTP (subclasses of) for each integral decrement specified in the request. This action will fail if there are an insufficient number of available sonetNetworkCTPs.";

```
setupLinkConnection                ACTION
    BEHAVIOUR     setupLinkConnectionBehaviour;
    MODE CONFIRMED;
    PARAMETERS
        "Rec. M.3100 : 1995":generalErrorParameter;
    WITH INFORMATION SYNTAX SIFNLMV2Mod.SetupLinkConnectionInformation;
    WITH REPLY SYNTAX SIFNLMV2Mod.SetupLinkConnectionReply;
REGISTERED AS {sIFNLMAction 13};
```

setupLinkConnectionBehaviour BEHAVIOUR
    DEFINED AS

"The setupLinkConnection action request is sent by a managing system (e.g., the NMS) to the managed system (e.g., the EMS) to direct the sonetLink (and subclasses) to setup a link connection. Upon receipt of this request, the managed system creates a sonetLinkConnection object instance. Also, the userLabel is returned to the managing system.";

```
setupSNC            ACTION
    BEHAVIOUR     setupSNCBehaviour;
    MODE CONFIRMED;
    PARAMETERS
```

```
        "Rec. M.3100 : 1995":generalErrorParameter;
    WITH INFORMATION SYNTAX SIFNLMV2Mod.SetupSNCInformation;
    WITH REPLY SYNTAX SIFNLMV2Mod.SetupSNCReply;
REGISTERED AS {sIFNLMAction 14};

setupSNCBehaviour BEHAVIOUR
    DEFINED AS
        "The setupSNC action request is sent by a managing system
(e.g., the NMS) to the managed system (e.g., the EMS) to direct the
sonetSubnetwork (and subclasses) to setup a sonetSubnetworkConnection
with a requested protection level, a connectionType and an associated
effort level.  Upon receipt of this request, the managed system creates
an sonetSubnetworkConnection object instance and sets the related
sNCPointers.  Also, the userLabel is returned to the managing system.

The administrative state of the resultant SNC will be set to locked
if the administative state parameter is set to locked in the setup
action. The administrative state of the resultant SNC will be set to
unlocked if the administrative state parameter is set to unlocked
in the setup action.

The requestedEffortLevel describes the level of commitment or
guarantee that is sought by the requestor in achieving the requested
protection level in a connection set-up action. Possible values are:

required minimum - grant the requested action if the connection can be
established with a provisioned protection level that is the same as or
higher than the requested protection level

required exact - grant the requested action if the connection can be established with a
provisioned protection level that is the same as the requested protection level
best effort - grant the requested action if the connection can be established with any
value of provisioned protection.

For a point to point connection one aTP and one zTP must be provided.

For a bicast connection one aTP and two zTPs must be provided.

For an exclusive merge two aTPs and one zTP must be provided.

For an exclusive compsite one a TP and two zTPs must be provided.";


setupTrail          ACTION
    BEHAVIOUR       setupTrailBehaviour;
    MODE CONFIRMED;
    PARAMETERS
        "Rec. M.3100 : 1995":generalErrorParameter;
    WITH INFORMATION SYNTAX SIFNLMV2Mod.SetupTrailInformation;
    WITH REPLY SYNTAX SIFNLMV2Mod.SetupTrailReply;
REGISTERED AS {sIFNLMAction 15};

setupTrailBehaviour BEHAVIOUR
    DEFINED AS
```

"The setupTrail action request is sent by a managing system (e.g., the NMS) to the managed system (e.g., the EMS) to direct the sonetLayerNetworkDomain (and subclasses) to setup a sonetTrail with a requested protection level, a connectionType and an associated effort level. Upon receipt of this request, the managed system creates a sonetTrail object instance. Also, the userLabel is returned to the managing system.

The administrative state of the resultant SNC will be set to locked
if the administative state parameter is set to locked in the setup
action. The administrative state of the resultant SNC will be set to
unlocked if the administrative state parameter is set to unlocked
in the setup action.

The requestedEffortLevel describes the level of commitment or guarantee that is sought by the requestor in achieving the requested protection level in a connection set-up action. Possible values are:
required minimum - grant the requested action if the connection can be established with a provisioned protection level that is the same as or higher than the requested protection level
required exact - grant the requested action if the connection can be established with a provisioned protection level that is the same as the requested protection level
best effort - grant the requested action if the connection can be established with any value of provisioned protection.

The connectionType of a trail may be point to point.";

```
connectionConfigurationSummaryReport NOTIFICATION
    BEHAVIOUR connectionConfigurationSummaryReportBehaviour;
    WITH INFORMATION SYNTAX
    SIFNLMV2Mod.ConnectionConfigurationSummaryReportInformation;
REGISTERED AS {sIFNLMNotification 1};

connectionConfigurationSummaryReportBehaviour BEHAVIOUR
    DEFINED AS
```

"This notification is sent by the managed system to the managing system to report the status of setupSNC, releaseSNC, setupTrail, and releaseTrail requests. This notification is sent by the managed system at intervals determined by the managementOperationsSchedule object instance.

The information provided in this report includes: object instances of sonetSubnetworkConnections and sonetTrails in the objectList attribute, an indication of whether the sonetSubnetworkConnection or sonetTrail is being established or released, and each sonetSubnetworkConnection's or sonetTrail's associated percentage of completion or release. The percentage of completed or released cross-connections is used to denote percentage of completion or release.";

```
rcaaNotification  NOTIFICATION
     BEHAVIOUR      rcaaNotificiationBehaviour;
     WITH INFORMATION SYNTAX SIFNLMV2Mod.RcaaInfo
          AND ATTRIBUTE IDS
          probableCause                        "Rec. X.721 | ISO/IEC 10165-2 :
1992":probableCause,
          specificProblems              "Rec. X.721 | ISO/IEC 10165-2 :
1992":specificProblems,
          perceivedSeverity             "Rec. X.721 | ISO/IEC 10165-2 :
1992":perceivedSeverity,
          backedUpStatus                "Rec. X.721 | ISO/IEC 10165-2 :
1992":backedUpStatus,
          backUpObject                  "Rec. X.721 | ISO/IEC 10165-2 :
1992":backUpObject,
          trendIndication               "Rec. X.721 | ISO/IEC 10165-2 :
1992":trendIndication,
          thresholdInfo                 "Rec. X.721 | ISO/IEC 10165-2 :
1992":thresholdInfo,
          notificationIdentifier        "Rec. X.721 | ISO/IEC 10165-2 :
1992":notificationIdentifier,
          correlatedNotifications       "Rec. X.721 | ISO/IEC 10165-2 :
1992":correlatedNotifications,
          stateChangeDefinition                "Rec. X.721 | ISO/IEC 10165-2 :
1992":stateChangeDefinition,
          monitoredAttributes           "Rec. X.721 | ISO/IEC 10165-2 :
1992":monitoredAttributes,
          proposedRepairActions         "Rec. X.721 | ISO/IEC 10165-2 :
1992":proposedRepairActions,
          additionalText                "Rec. X.721 | ISO/IEC 10165-2 :
1992":additionalText,
          additionalInformation                "Rec. X.721 | ISO/IEC 10165-2 :
1992":additionalInformation,
          alarmedObjectClass            alarmedObjectClass,
          alarmedObjectInstance         alarmedObjectInstance,
          faultLocation                 faultLocation,
            serviceAffecting                   serviceAffecting,
            numberOfNELAlarms                  numberOfNELAlarms,
            affectedTransmissionResources      affectedTransmissionResources,
            highestPriorityNELalarmRecords     highestPriorityNELalarmRecords;
REGISTERED AS   { sIFNLMNotification 2};

rcaaNotificiationBehaviour BEHAVIOUR
    DEFINED AS
        "The RCAA Notification is generated by the root cause alarm analysis routine
based on one or more NEL alarm event reports.  It will contain all the fields which are
contained in a alarm record as defined in X.733. The notificationIdentifier parameter
shall be required.  The correlatedNotification parameter shall be included if this RCAA
Notification is a update or a clear of an earlier RCAA Notification.  The Event Time
shall be included and will be set to the time of the arrival of the first NEL alarm which is
correlated to this RCAA Notification.

        In addition the alarmedObjectClass, alarmedObjectInstance, faultLocation,
serviceAffecting, numberOfNELAlarms, affectedTransmissionResources  and the
higherPriorityNELalarmRecords parameters shall be included. These parameters shall
have the same values and behaviors as the associated attributes in the RCAArecord.";


sonetAccessGroup-sonetLayerNetworkDomain NAME BINDING
    SUBORDINATE OBJECT CLASS sonetAccessGroup AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetLayerNetworkDomain AND
    SUBCLASSES;
    WITH ATTRIBUTE sonetAccessGroupId;
    BEHAVIOUR sonetAccessGroup-sonetLayerNetworkDomainBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ;
REGISTERED AS {sIFNLMNameBinding 1};

sonetAccessGroup-sonetLayerNetworkDomainBehaviour BEHAVIOUR
    DEFINED AS
        "Instances of sonetAccessGroup are created and deleted by the managing
system. Instances of this class may also be automatically instantiated at initialization of
the EMS. A deletion request is denied if the sonetAccessGroup has any associated
termination points, that is if tPList is not empty.";

sonetConnectionPendingStatusControl-sonetLayerNetworkDomain          NAME
BINDING
    SUBORDINATE  OBJECT  CLASS   sonetConnectionPendingStatusControl     AND
```

```
SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetLayerNetworkDomain AND
    SUBCLASSES;
    WITH ATTRIBUTE sonetConnectionPendingStatusControlId;
    BEHAVIOUR
        sonetConnectionPendingStatusControl-sonetLayerNetworkDomainBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT ,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS   {sIFNLMNameBinding 2};

sonetConnectionPendingStatusControl-sonetLayerNetworkDomainBehaviour
    BEHAVIOUR
    DEFINED AS
        "Instances of sonetConnectionPendingStatusControl are created and deleted by
the managing system. Instances of this class may also be automatically instantiated at
initialization of the EMS.";

sonetConnectionPendingStatusControl-networkR1      NAME  BINDING
    SUBORDINATE  OBJECT  CLASS
    sonetConnectionPendingStatusControl AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS "Rec. M.3100 : 1995":networkR1 AND
    SUBCLASSES;
    WITH ATTRIBUTE sonetConnectionPendingStatusControlId;
    BEHAVIOUR sonetConnectionPendingStatusControl-networkR1Behaviour;
    CREATE
        WITH-REFERENCE-OBJECT ,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS   {sIFNLMNameBinding 3};

sonetConnectionPendingStatusControl-networkR1Behaviour BEHAVIOUR
    DEFINED AS
        "Instances of sonetConnectionPendingStatusControl are created and deleted by
the managing system. Instances of this class may also be automatically instantiated at
initialization of the EMS.";


sonetLayerNetworkDomain-networkR1 NAME BINDING
    SUBORDINATE OBJECT CLASS sonetLayerNetworkDomain AND
    SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS "Rec. M.3100 : 1995":networkR1
    AND SUBCLASSES;
    WITH ATTRIBUTE sonetLayerNetworkDomainId;
    BEHAVIOUR sonetLayerNetworkDomain-networkR1Behaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {sIFNLMNameBinding 4};

sonetLayerNetworkDomain-networkR1Behaviour BEHAVIOUR
    DEFINED AS
        "Instances of sonetLayerNetworkDomain are created and deleted by the
managing system. Instances of this class may also be automatically instantiated at
initialization of the EMS.";


sonetLink-sonetLayerNetworkDomain NAME BINDING
    SUBORDINATE OBJECT CLASS sonetLink AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetLayerNetworkDomain AND
    SUBCLASSES;
    WITH ATTRIBUTE sonetLinkId;
    BEHAVIOUR sonetLink-sonetLayerNetworkDomainBehaviour;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {sIFNLMNameBinding 5};

sonetLink-sonetLayerNetworkDomainBehaviour BEHAVIOUR
    DEFINED AS
        "Instances of sonetLink are created and deleted by the managing system using
the addLink and removeLink actions on the sonetLayerNetworkDomain superior object.
Some instances of this class may be automatically instantiated at initialization of the
EMS. Some may also be automatically created or deleted by the EMS as a side effect
of the creation or deletion of trails in server layers.";


sonetLinkConnection-sonetLink NAME BINDING
    SUBORDINATE OBJECT CLASS sonetLinkConnection AND     SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetLink AND SUBCLASSES;
    WITH ATTRIBUTE sonetTransportEntityId;
    BEHAVIOUR sonetLinkConnection-sonetLinkBehaviour;
REGISTERED AS {sIFNLMNameBinding 6};

sonetLinkConnection-sonetLinkBehaviour BEHAVIOUR
    DEFINED AS
```

"Instances of sonetLinkConnection are created and deleted by the managing system using the setUpLinkConnection and releaseLinkConnection actions on the superior link object. Some instances of this class may be automatically instantiated at initialization of the EMS. Some may also be automatically created or deleted by the EMS as a side effect of the creation or deletion of trails in server layers.";

```
sonetLinkEnd-sonetLayerNetworkDomain NAME BINDING
    SUBORDINATE OBJECT CLASS sonetLinkEnd AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetLayerNetworkDomain AND
    SUBCLASSES;
    WITH ATTRIBUTE sonetLinkEndId;
    BEHAVIOUR sonetLinkEnd-sonetLayerNetworkDomainBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ;

REGISTERED AS {sIFNLMNameBinding 7};

sonetLinkEnd-sonetLayerNetworkDomainBehaviour BEHAVIOUR
    DEFINED AS
```

"Instances of sonetLinkEnd are created and deleted by the managing system. An instance may also be automatically created or deleted by the EMS as a side effect of the creation or deletion of a networkTTP in a server layer. A deletion request is denied if the linkEnd has any associated termination points, that is if nCTPList is not empty.";

```
eventForwardingDiscriminator-networkR1      NAME  BINDING
    SUBORDINATE  OBJECT  CLASS "Rec. X.721 | ISO/IEC 10165-2 : 1992":
        eventForwardingDiscriminator AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS "Rec. M.3100 : 1995":networkR1
    AND SUBCLASSES;
    WITH ATTRIBUTE  "Rec. X.721 | ISO/IEC 10165-2 : 1992":discriminatorId;
    CREATE
        WITH-REFERENCE-OBJECT ,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS   {sIFNLMNameBinding 8};

log-networkR1      NAME  BINDING
    SUBORDINATE OBJECT CLASS "Rec. X.721 | ISO/IEC 10165-2 : 1992": log
    AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS "Rec. M.3100 : 1995":networkR1
    AND SUBCLASSES;
    WITH ATTRIBUTE  "Rec. X.721 | ISO/IEC 10165-2 : 1992":logId;
    CREATE
        WITH-REFERENCE-OBJECT ,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS   {sIFNLMNameBinding 9};

sonetAlarmAnalysisRoutine-networkR1      NAME  BINDING
    SUBORDINATE  OBJECT  CLASS sonetAlarmAnalysisRoutine AND
    SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS "Rec. M.3100 : 1995":networkR1
    AND SUBCLASSES;
    WITH ATTRIBUTE sonetAlarmAnalysisRoutineId;
    BEHAVIOUR sonetAlarmAnalysisRoutine-networkR1Behaviour;
    CREATE
        WITH-REFERENCE-OBJECT ,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS   {sIFNLMNameBinding 10};

sonetAlarmAnalysisRoutine-networkR1Behaviour BEHAVIOUR
    DEFINED AS
        "Instances of sonetAlarmAnalysisRoutine are created and deleted by the
managing system. Instances of this class may also be automatically instantiated at
initialization of the EMS.";

alarmSeverityAssignmentProfile-networkR1      NAME  BINDING
    SUBORDINATE  OBJECT  CLASS "Rec.    M.3100                                    :
1995":alarmSeverityAssignmentProfile
    AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS "Rec. M.3100 : 1995":networkR1
    AND SUBCLASSES;
    WITH ATTRIBUTE "Rec. M.3100 : 1995":alarmSeverityAssignmentProfileId;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS   {sIFNLMNameBinding 11};
```

```
eventForwardingDiscriminator-sonetLayerNetworkDomain NAME  BINDING
    SUBORDINATE  OBJECT  CLASS "Rec. X.721 | ISO/IEC 10165-2 : 1992":
        eventForwardingDiscriminator AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetLayerNetworkDomain AND
    SUBCLASSES;
    WITH ATTRIBUTE "Rec. X.721 | ISO/IEC 10165-2 : 1992":discriminatorId;
    CREATE
        WITH-REFERENCE-OBJECT ,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS   {sIFNLMNameBinding 12};

log-sonetLayerNetworkDomain   NAME  BINDING
    SUBORDINATE OBJECT CLASS "Rec. X.721 | ISO/IEC 10165-2 : 1992": log AND
SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetLayerNetworkDomain AND
    SUBCLASSES;
    WITH ATTRIBUTE  "Rec. X.721 | ISO/IEC 10165-2 : 1992":logId;
    CREATE
        WITH-REFERENCE-OBJECT ,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS   {sIFNLMNameBinding 13};

sonetAlarmAnalysisRoutine-sonetLayerNetworkDomain    NAME  BINDING
    SUBORDINATE  OBJECT  CLASS sonetAlarmAnalysisRoutine AND
    SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetLayerNetworkDomain AND
    SUBCLASSES;
    WITH ATTRIBUTE sonetAlarmAnalysisRoutineId;
    BEHAVIOUR sonetAlarmAnalysisRoutine-sonetLayerNetworkDomainBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT ,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS   {sIFNLMNameBinding 14};

sonetAlarmAnalysisRoutine-sonetLayerNetworkDomainBehaviour BEHAVIOUR
    DEFINED AS
        "Instances of sonetAlarmAnalysisRoutine are created and deleted by the
managing system. Instances of this class may also be automatically instantiated at
```

```
initialization of the EMS.";

alarmSeverityAssignmentProfile-sonetLayerNetworkDomain     NAME  BINDING
    SUBORDINATE OBJECT CLASS "Rec. M.3100 :
1995":alarmSeverityAssignmentProfile
    AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetLayerNetworkDomain
    AND SUBCLASSES;
    WITH ATTRIBUTE "Rec. M.3100 : 1995":alarmSeverityAssignmentProfileId;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS   {sIFNLMNameBinding 15};


managementOperationsSchedule-sonetLayerNetworkDomain  NAME BINDING
    SUBORDINATE  OBJECT  CLASS
        "Rec. Q.821":managementOperationsSchedule AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetLayerNetworkDomain AND
    SUBCLASSES;
    WITH ATTRIBUTE "Rec. Q.821":scheduleId;
    CREATE;
    DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS   {sIFNLMNameBinding 16};

managementOperationsSchedule-networkR1      NAME  BINDING
    SUBORDINATE  OBJECT  CLASS
        "Rec. Q.821":managementOperationsSchedule AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS "Rec. M.3100 : 1995":networkR1
    AND SUBCLASSES;
    WITH ATTRIBUTE "Rec. Q.821":scheduleId;
    CREATE;
    DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS   {sIFNLMNameBinding 17};


sonetNetworkCTP-sonetLayerNetworkDomain NAME BINDING
    SUBORDINATE OBJECT CLASS sonetNetworkCTP AND
    SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetLayerNetworkDomain AND
    SUBCLASSES;
    WITH ATTRIBUTE sonetNetworkCTPId;
    BEHAVIOUR sonetNetworkCTP-sonetLayerNetworkDomainBehaviour;
REGISTERED AS {sIFNLMNameBinding 18};

sonetNetworkCTP-sonetLayerNetworkDomainBehaviour BEHAVIOUR
    DEFINED AS
        "Instances of sonetNetworkCTP (subclasses of) are created and deleted by the
managing system using the addLinkCapacity and removeLinkCapacity actions on
associated link or sonetLinkEnd objects. Instances of this class (subclasses of) can
also be created and deleted by the EMS (for example, as a consequence of operations
at the element level or in server layers).";


sonetNetworkTTP-sonetLayerNetworkDomain NAME BINDING
    SUBORDINATE OBJECT CLASS sonetNetworkTTP AND
    SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetLayerNetworkDomain AND
    SUBCLASSES;
    WITH ATTRIBUTE sonetNetworkTTPId;
    BEHAVIOUR sonetNetworkTTP-sonetLayerNetworkDomainBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS {sIFNLMNameBinding 19};

sonetNetworkTTP-sonetLayerNetworkDomainBehaviour BEHAVIOUR
    DEFINED AS
        "Instances of sonetNetworkTTP (subclasses of) can be created and deleted by
the managing system. Instances of this class (subclasses of) can also be created and
deleted by the EMS (for example, as a consequence of operations at the element level
or in server layers).";


sonetRoutingProfile-sonetLayerNetworkDomain NAME BINDING
    SUBORDINATE OBJECT CLASS sonetRoutingProfile AND      SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetLayerNetworkDomain AND
    SUBCLASSES;
    WITH ATTRIBUTE sonetRoutingProfileId;
    BEHAVIOUR sonetRoutingProfile-sonetLayerNetworkDomainBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
```

```
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS {sIFNLMNameBinding 20};
```

sonetRoutingProfile-sonetLayerNetworkDomainBehaviour BEHAVIOUR
    DEFINED AS
        "Instances of sonetRoutingProfile are created and deleted automatically by the EMS as a result of the setupSNC, setupTrail and releaseSNC, releaseTrail actions. Some instances of this class may be automatically instantiated at initialization of the EMS. Some may also be created and deleted directly by the managing system.";

```
sonetRoutingProfile-sonetSubnetwork NAME BINDING
    SUBORDINATE OBJECT CLASS sonetRoutingProfile AND      SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetSubnetwork AND
    SUBCLASSES;
    WITH ATTRIBUTE sonetRoutingProfileId;
    BEHAVIOUR sonetRoutingProfile-sonetSubnetworkBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS {sIFNLMNameBinding 21};
```

sonetRoutingProfile-sonetSubnetworkBehaviour BEHAVIOUR
    DEFINED AS
        "Instances of sonetRoutingProfile are created and deleted automatically by the EMS as a result of the setupSNC, setupTrail and releaseSNC, releaseTrail actions. Some instances of this class may be automatically instantiated at initialization of the EMS. Some may also be created and deleted directly by the managing system.";

```
sonetSubnetworkConnection-sonetSubnetwork NAME BINDING
    SUBORDINATE OBJECT CLASS sonetSubnetworkConnection AND
    SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetSubnetwork AND
    SUBCLASSES;
    WITH ATTRIBUTE sonetTransportEntityId;
    BEHAVIOUR sonetSubnetworkConnection-sonetSubnetworkBehaviour;
REGISTERED AS {sIFNLMNameBinding 22};
```

sonetSubnetworkConnection-sonetSubnetworkBehaviour BEHAVIOUR
    DEFINED AS
        "Instances of sonetSubnetworkConnection are created and deleted by the managing system using the setupSNC and releaseSNC actions on superior subnetwork

```
object. Some instances of this class may be automatically instantiated at initialization of
the EMS.";


sonetSubnetwork-sonetLayerNetworkDomain NAME BINDING
    SUBORDINATE OBJECT CLASS sonetSubnetwork AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetLayerNetworkDomain AND
    SUBCLASSES;
    WITH ATTRIBUTE sonetSubnetworkId;
    BEHAVIOUR sonetSubnetwork-sonetLayerNetworkDomainBehaviour;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {sIFNLMNameBinding 23};

sonetSubnetwork-sonetLayerNetworkDomainBehaviour BEHAVIOUR
    DEFINED AS
        "Instances of sonetSubnetwork are created and deleted by the managing
system. Some instances of this class may be automatically instantiated at initialization
of the EMS.";


sonetTrail-sonetLayerNetworkDomain NAME BINDING
    SUBORDINATE OBJECT CLASS sonetTrail AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetLayerNetworkDomain AND
    SUBCLASSES;
    WITH ATTRIBUTE sonetTransportEntityId;
    BEHAVIOUR sonetTrail-sonetLayerNetworkDomainBehaviour;
REGISTERED AS {sIFNLMNameBinding 24};

sonetTrail-sonetLayerNetworkDomainBehaviour BEHAVIOUR
    DEFINED AS
        "Instances of sonetTrail are created and deleted by the managing system using
the addTrail and releaseTrail actions on sonetLayerNetworkDomain superior object.
Some instances of this class may be automatically instantiated at initialization of the
EMS.  Some may also be automatically created or deleted by the EMS as a side effect
of the creation or deletion of links in the same layer.";

sonetVirtualCTP-sonetLayerNetworkDomain NAME BINDING
    SUBORDINATE OBJECT CLASS sonetVirtualCTP AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetLayerNetworkDomain AND
    SUBCLASSES;
    WITH ATTRIBUTE sonetNetworkCTPId;
    BEHAVIOUR sonetVirtualCTP-sonetLayerNetworkDomainBehaviour;
CREATE;
    DELETE;
REGISTERED AS {sIFNLMNameBinding 25};

sonetVirtualCTP-sonetLayerNetworkDomainBehaviour BEHAVIOUR
    DEFINED AS
        "Instances of sonetVirtualCTP (subclasses of) are created and deleted by the
managing system using management create and delete operations. When the
sonetVirtualCTP is created a virtualLinkEnd is automatically created if an appropriate
virtualLinkEnd associated with the containing point does not already exist. ??? When
the sonetVirtualCTP is deleted the associated virtualLinkEnd will be deleted if this is
the last associated sonetVirtualCTP in the virtualLinkEnd. Instances of this class
(subclasses of) can also be created and deleted by the EMS (for example, in support of
the creation of an SNC across a partitioning.).";

sonetVirtualLink-sonetLayerNetworkDomain NAME BINDING
    SUBORDINATE OBJECT CLASS sonetVirtualLink AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetLayerNetworkDomain AND
    SUBCLASSES;
    WITH ATTRIBUTE sonetLinkId;
    BEHAVIOUR sonetVirtualLink-sonetLayerNetworkDomainBehaviour;
REGISTERED AS {sIFNLMNameBinding 26};

sonetVirtualLink-sonetLayerNetworkDomainBehaviour BEHAVIOUR
    DEFINED AS
        "Instances of sonetVirtualLink are created as a side effect of the creation of the
endpoints of the sonetVirtualLink. Instances of the sonetVirtualLink are deleted as a
side effect of the deletion of the endpoints of the sonetVirtualLink.";


sonetVirtualLinkConnection-sonetVirtualLink NAME BINDING
    SUBORDINATE OBJECT CLASS sonetVirtualLinkConnection
        AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetVirtualLink AND SUBCLASSES;
    WITH ATTRIBUTE virtualLinkConnectionId;
    BEHAVIOUR sonetVirtualLinkConnection-sonetVirtualLinkBehaviour;
REGISTERED AS {sIFNLMNameBinding 27};

sonetVirtualLinkConnection-sonetVirtualLinkBehaviour BEHAVIOUR
    DEFINED AS
        "Instances of sonetVirtualLinkConnection are created and deleted
automatically when the endpoints are created and deleted.";

sonetVirtualLinkEnd-sonetLayerNetworkDomain NAME BINDING
```

```
    SUBORDINATE OBJECT CLASS sonetVirtualLinkEnd AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS sonetLayerNetworkDomain AND
    SUBCLASSES;
    WITH ATTRIBUTE sonetLinkEndId;
    BEHAVIOUR sonetVirtualLinkEnd-sonetLayerNetworkDomainBehaviour;

REGISTERED AS {sIFNLMNameBinding 28};

sonetVirtualLinkEnd-sonetLayerNetworkDomainBehaviour BEHAVIOUR
    DEFINED AS
        "Instances of sonetVirtualLinkEnd are automatically created  by the EMS as a
side effect of the creation of the first virtualCTP of the specific signalId associated with
the containing CTP. Instances of the sonetVirtualLinkEnd are automatically deleted by
the EMS as a side effect of the deletion of the last virtualCTP associated with the
virtualLinkEnd.";


-- The current top level registration arcs are temporary until SIF gets an arc from ANSI

SIFNLMV2Mod { joint-iso-ccitt recommendation (0) indStd(3) sifnw(777)
informationModel(1) asn1Modules(2) asn1DefinedTypesModule(0) }
DEFINITIONS IMPLICIT TAGS ::= BEGIN

-- EXPORTS everything

IMPORTS

ProbableCause,
SpecificProblems,
PerceivedSeverity,
BackedUpStatus,
TrendIndication,
ThresholdInfo,
NotificationIdentifier,
CorrelatedNotifications,
MonitoredAttributes,
ProposedRepairActions,
AdditionalText,
AdditionalInformation,
AttributeValueChangeDefinition,
AttributeList
FROM
Attribute-ASN1Module {joint-iso-ccitt(2) ms(9) smi(3) part2(2) asn1Module(2) 1}

Boolean,
Bundle,
ChannelNumber,
CharacteristicInformation,   -- This will be removed in next release of document
Directionality,
LocationName,
NameType,
RelatedObjectInstance,
UserLabel
FROM
ASN1DefinedTypesModule {ccitt recommendation m(13) gnm(3100)
informationModel(0) asn1Modules(2) asn1DefinedTypesModule(0)}

DistinguishedName
FROM
InformationFramework {joint-iso-ccitt ds(5) modules(1) informationFramework(1)}

Attribute,
AttributeId,
ObjectClass,
ObjectInstance
FROM
CMIP-1 {joint-iso-ccitt ms(9) cmip(1) modules(0) protocol(3)};

sIFNLMInformationModel  OBJECT IDENTIFIER ::= {joint-iso-ccitt  recommendation (0)
indStd (3) sifnw(777) informationModel(1) }
sIFNLMSpecificExtension OBJECT IDENTIFIER ::= {sIFNLMInformationModel
standardSpecificExtension(0)}
sIFNLMObjectClass OBJECT IDENTIFIER ::= {sIFNLMInformationModel
managedObjectClass(3)}
sIFNLMPackage OBJECT IDENTIFIER ::= {sIFNLMInformationModel package(4)}
sIFNLMAttribute OBJECT IDENTIFIER ::= {sIFNLMInformationModel attribute(7)}
sIFNLMNameBinding OBJECT IDENTIFIER ::= {sIFNLMInformationModel
nameBinding(6)}
sIFNLMAction OBJECT IDENTIFIER ::= {sIFNLMInformationModel action(9)}
sIFNLMNotification OBJECT IDENTIFIER ::= {sIFNLMInformationModel
notification(10)}
sIFNLMParameter OBJECT IDENTIFIER ::= {sIFNLMInformationModel  parameter(11)}

-- This section will be removed when these values are included in GR-836-IMD
-- and/or GR-1042-IMD

-- Characteristic Information OBJECT IDENTIFIERs


sIFNLMCharacteristicInformation OBJECT IDENTIFIER ::= {sIFNLMSpecificExtension}
```

```
electricalSTS1SPICI CharacteristicInformation ::= {sIFNLMCharacteristicInformation 1}

opticalSTM64SPICI CharacteristicInformation ::= {sIFNLMCharacteristicInformation 2}

rsSTM64SPICI CharacteristicInformation ::= {sIFNLMCharacteristicInformation 3}

msSTM64SPICI CharacteristicInformation ::= {sIFNLMCharacteristicInformation 4}

au44cCI CharacteristicInformation ::= {sIFNLMCharacteristicInformation 5}

ds3LineCI CharacteristicInformation ::= {sIFNLMCharacteristicInformation 6}

ds3PathCI CharacteristicInformation ::= {sIFNLMCharacteristicInformation 7}

ds1LineCI CharacteristicInformation ::= {sIFNLMCharacteristicInformation 8}

ds1PathCI CharacteristicInformation ::= {sIFNLMCharacteristicInformation 9}

-- End CharacteristicInformation Section

-- module definitions

AccessGroupList ::= SET OF ObjectInstance

AdditionalCapacity ::= Integer

AddLinkInformation ::= SEQUENCE {
    aLinkEnd ObjectInstance,
    zLinkEnd          ObjectInstance,
    userLabelUserLabel
    }

AddLinkReply ::= SEQUENCE {
    link              ObjectInstance,
    userLabel         UserLabel
    }

AddLinkCapacityInformation ::= AdditionalCapacity

AddLinkCapacityReply ::= SEQUENCE {
    aEndCTPs          SET OF ObjectInstance,
    zEndCTPs          SET OF ObjectInstance
    }
```

```
AddLinkEndCapacityInformation ::= AdditionalCapacity

AddLinkEndCapacityReply ::= SEQUENCE {
    cTPs        SET OF ObjectInstance
    }

AdministrativeState ::= ENUMERATED {
    unlocked (0),
    locked (1)
}
AffectedTransmissionResources ::= SET OF ObjectInstance

AlarmRecords ::= SET OF ObjectInstance

AssociatedSnList ::= SET OF ObjectInstance

Atmocn ::= SEQUENCE {
    sts1Number      INTEGER,
    vpi         INTEGER (1..4095),
    vci         INTEGER (1..65535)
    }

Atmsts ::= SEQUENCE {
    vpi         INTEGER (1..4095),
    vci         INTEGER (1..65535)
    }

ChannelIdentifier ::= SEQUENCE {
    groupNumber [0] GroupNumber OPTIONAL,
    channelNumber [1] ChannelNumber
    }

ClientLinkEnd ::= SET OF RelatedObjectInstance

ComponentList ::= SET OF ObjectInstance

ConfigRequestType ::= ENUMERATED {
    setup (0),
    release (1)
    }

ConfigurationState ::= ENUMERATED {
    preService (0),
    inService (1),
    postService (2),
    configurationFailure (3)
    }

ConfigurationStatus ::= SEQUENCE {
    sNCorTrail                                      [0]     ObjectInstance,
    configRequestType                       [1]     ConfigRequestType,
    percentCompletedOrReleased      [2]     Integer
    }

ConnectionConfigurationSummaryReportInformation ::= SET OF ConfigurationStatus

ConnectionType ::= ENUMERATED {
    broadcast (0), -- point-to-multipoint unidirectional
    pointToPoint (4),  -- point-to-point
    bicast (5), -- unidirectional signal is split at its source into two identical signals
    exclusiveMerge (6), -- exclusive merge at sink end of two unidirectional signals into
one output signal
    exclusiveComposite (7), -- bicast  plus exclusive merge
    dropAndContinue (8) -- drop and continue on UPSR interworking node.
    -- "..." these ellipses defined in ASN.1 amendment are used here to --
    --indicate that this is an  extensible type and additional enumerations --
    -- may be added in future --
    }

ContainingElement ::= SET OF ObjectInstance

CTP ::= RelatedObjectInstance

EffortLevel ::= ENUMERATED {
    requiredMinimum (0), -- required minimum level
    requiredExact (1), -- required exact level
    bestEffort (2) -- best effort to achieve level
}

EndList ::= SET OF ObjectInstance

EndPoint ::= CHOICE {
    terminationPoint                [0]             ObjectInstance,
    sonetLinkEndorAccessGroup  [1]          ObjectInstance
    }

Format          ::= OBJECT IDENTIFIER

GroupNumber ::= INTEGER
```

```
Integer ::= INTEGER

LinkEndList ::= SET OF ObjectInstance

LinkPointer ::= RelatedObjectInstance

FaultLocation ::= SET OF LocationName

MaxHops ::= CHOICE {
    notApplicable    NULL,
    maxHops INTEGER
    }

ModifySNCConnectionTypeInformation ::= SEQUENCE {
    snc          [0]ObjectInstance,
    preConnType     [1]ConnectionType,
    postConnType   [2]ConnectionType
    }

ModifySNCConnectionTypeReply ::= SEQUENCE {
    snc                  [0]ObjectInstance,
    preConnType                  [1]ConnectionType
    }

ModifySNCProtectionLevelInformation ::= SEQUENCE {
    snc          [0]ObjectInstance,
    preProtLevel      [1]ProtectionLevel,
    postProtLevel    [2]ProtectionLevel
    }

ModifySNCProtectionLevelReply ::= SEQUENCE {
    snc          [0]ObjectInstance,
    postProtLevel    [1]ProtectionLevel
    }

ModifyTrailConnectionTypeInformation ::= ObjectInstance

ModifyTrailConnectionTypeReply ::= ObjectInstance

ModifyTrailProtectionLevelInformation ::= SEQUENCE {
    trail          [0]ObjectInstance,
    preProtLevel      [1]ProtectionLevel,
    postProtLevel    [2]ProtectionLevel
    }
```

```
ModifyTrailProtectionLevelReply ::= SEQUENCE {
    snc          [0]ObjectInstance,
    postConnType   [1]ProtectionLevel
    }

ProtectionLevel ::= ENUMERATED {
    highlyProtected (0), -- highly protected
    protected (1),  -- protected
    unprotected (2),  -- unprotected
    preemptible (3)  -- preemptible
}

RcaaInfo ::=  SEQUENCE {
     probableCause                ProbableCause,
      specificProblems                    [1]SpecificProblems OPTIONAL,
      perceivedSeverity                   PerceivedSeverity,
      backedUpStatus                      BackedUpStatus  OPTIONAL,
      backUpObject                        [2]ObjectInstance  OPTIONAL,
      trendIndication                     [3]TrendIndication  OPTIONAL,
      thresholdInfo               [4]ThresholdInfo   OPTIONAL,
      notificationIdentifier      [5]NotificationIdentifier  OPTIONAL,
      correlatedNotifications             [6]CorrelatedNotifications,
      stateChangeDefinition               [7]AttributeValueChangeDefinition OPTIONAL,
      monitoredAttributes                 [8]MonitoredAttributes  OPTIONAL,
      proposedRepairActions               [9]ProposedRepairActions OPTIONAL,
      additionalText                      AdditionalText  OPTIONAL,
      additionalInformation               [10]AdditionalInformation OPTIONAL,
     alarmedObjectClass           ObjectClass,
     alarmedObjectInstance                ObjectInstance,
     faultLocation                FaultLocation,
     serviceAffecting             Boolean,
     numberOfNELAlarms            Integer,
     affectedTransmissionResources            AffectedTransmissionResources,
     highestPriorityNELalarmRecords           AlarmRecords }

ReflectedTP ::= ObjectInstance

ReleaseLinkConnectionInformation ::=   TransportID

ReleaseSNCInformation ::= TransportID

ReleaseTrailInformation ::= TransportID

RemoveLinkInformation ::= TransportID
```

```
RemoveLinkCapacityInformation ::= Integer

RemoveLinkEndCapacityInformation ::= Integer

RouteDescription ::= SEQUENCE {
    referenceObject          ObjectInstance,
    option                   RoutingOption
    }

RouteDescriptionList ::= SEQUENCE OF RouteDescription

RoutingInfo ::= CHOICE {
    routeDescriptionList [0] RouteDescriptionList,
    routingProfilePointer [1]        ObjectInstance
    }

RoutingOption ::= ENUMERATED {
    mandatory1 (0), -- must use the object in establishing the connection NOT
ACCEPTING mandatory (may be keyword)
    preferred (1), -- attempt to use the object in establishing the connection
    exclude (2), -- do not use the object in establishing the connection
    sameRoute (3), -- use the same route as the reference object
    diverseRoute (4), -- use different route as the reference object
    diverseRouteAndNodes (5) -- use different route and no nodes in common with the
reference object
    }

RoutingProfile ::= RelatedObjectInstance

ServerNTTP ::= RelatedObjectInstance

SetupLinkConnectionInformation ::= SEQUENCE {
    aEndCTP          [0]ObjectInstance    OPTIONAL,
    zEndCTP          [1]ObjectInstance    OPTIONAL,
    directionality   Directionality,
    userLabel        UserLabel
    }

SetupLinkConnectionReply ::= SEQUENCE {
    linkConnection   ObjectInstance,
    userLabel        UserLabel,
    aEndCTP ObjectInstance,
    zEndCTP ObjectInstance
    }
```

```
SetupSNCInformation ::= SEQUENCE {
    tPAs                SET OF EndPoint,
    tPZs                SET OF EndPoint,
    connectionType  ConnectionType,
    directionality      Directionality,
    routingInfo         RoutingInfo   OPTIONAL,
    userLabel           UserLabel,
    requestedProtectionLevel        ProtectionLevel,
    requestedEffortLevel            EffortLevel,
    administrativeState             AdministrativeState
    }

SetupSNCReply ::= SEQUENCE {
    sNC         ObjectInstance,
    tPAs        SET OF ObjectInstance,
    tPZs        SET OF ObjectInstance,
    userLabel           UserLabel,
    provisionedProtectionLevel      ProtectionLevel,
    provisionedEffortLevel  EffortLevel
    }

SetupTrailInformation ::= SEQUENCE {
    tTPAs       SET OF EndPoint,
    tTPZs       SET OF EndPoint,
    connectionType  ConnectionType,
    directionality      Directionality,
    routingInfo         RoutingInfo   OPTIONAL,
    userLabel           UserLabel,
    requestedProtectionLevel        ProtectionLevel,
    requestedEffortLevel            EffortLevel,
    administrativeState             AdministrativeState

    }

SetupTrailReply ::= SEQUENCE {
    trail       ObjectInstance,
    tTPAs       SET OF ObjectInstance,
    tTPZs       SET OF ObjectInstance,
    userLabel           UserLabel,
    provisionedProtectionLevel      ProtectionLevel,
    requestedEffortLevel            EffortLevel
    }

SignalId ::= CHOICE {
    simple              [0] CharacteristicInformation,
    bundle              [1] Bundle,
    complex             [2] SEQUENCE OF Bundle}

SnProtectionStatus  ::= ENUMERATED {
    noSwitch (0), -- there are no switches in the subnetwork
    switches (1) -- one or more switches are in effect in the subnetwork
      }

SNCPointer ::= RelatedObjectInstance

SubnetworkType ::= ENUMERATED {
    tSIADM (0),
    tSAADM (1),
    hardwiredADM (2),
    terminalMux (3),
    dCS (4),
    otherNE (5),
    uPSR (6),
    bLSR (7),
    linearChain (8),
    pointtopoint (9),
    otherSubnetwork (10)
    -- "..." these ellipses defined in ASN.1 amendment are used here to --
    --indicate that this is an  extensible type and additional enumerations --
    -- may be added in future --
}

TPConnectionState ::= ENUMERATED {
    notConnected (0),
    sinkConnected (1),
    sourceConnected (2),
    bidirectionallyConnected (3)
    }
TPPointer ::= CHOICE {
    cmipObjectPointer [0]   ObjectInstance,
    tl1ObjectPointer [1]    PrintableString,
    noObject  [3]           NULL
}

TransportID ::= NameType
TTP ::= ObjectInstance

TP ::= ObjectInstance

TPList ::= SET OF ObjectInstance
```

```
TTPList ::= SET OF ObjectInstance

TTPs ::= SET OF ObjectInstance

VirtualCTPChannelIdentifier ::= CHOICE {
    vt15ocn    [0]     Vt15ocn,
    vt15sts1   [1]     Vt15sts1,
    vt6ocn     [2]     Vt6ocn,
    vt6sts1    [3]     INTEGER (1..7),
    sts1ocn    [4]     INTEGER,
    sts3cocn   [5]     INTEGER,
    sts12cocn          [6]     INTEGER,
    sts48cocn          [7]     INTEGER,
    atmocn     [8]     Atmocn,
    atmsts     [9]     Atmsts
}

Vt15ocn ::= SEQUENCE {
    sts1Number       INTEGER,
    vtgNumber        INTEGER (1..7),
    vt15Number       INTEGER (1..4)
    }

Vt15sts1 ::= SEQUENCE {
    vtgNumber        INTEGER (1..7),
    vt15Number       INTEGER (1..4)
    }

Vt6ocn ::= SEQUENCE {
    sts1Number       INTEGER,
    vtgNumber        INTEGER (1..7)
    }


END
```

Appendix 1: Scenarios Where the Nodal View is Required

This appendix shows a configuration where desired subnetwork connections can be provided by using the nodal view of the subnetwork, but these subnetwork connections are not supported at higher levels of partitioning. It is possible that in the future additional connection types will be defined so that the subnetwork connections can be setup and torn down with a single command at higher levels of partitioning for this and other such configurations.

The basic configuration for dual ring interconnection is shown in Figure A -1. The UPSR ring on the left joins nodes A, B, C, D, and E. Dual interconnection to the right ring (UPSR or BLSR) is established via drop-side interfaces at nodes C and D. A highly protected uni-directional connection from point a to point x employs drop & continue, split, and selector functions at both node C and D to provide two low-speed connection paths to the second ring.

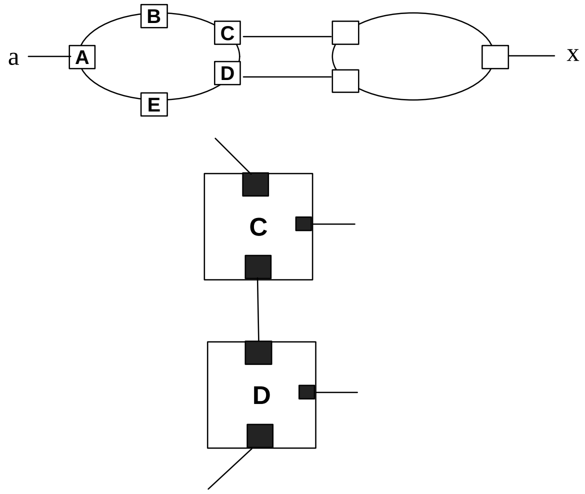


Figure A-1 Dual Ring Interconnection

The nodal subnetwork view at interconnection node C which is supported by VT cross-connection fabric is shown in Figure A -2. The connection scenario of interest is a uni-directional highly protected VT layer connection between a and x using a drop and continue configuration. (It should be noted that this scenario can be modeled in the ring subnetwork view by setting up a highly protected bicast connection on the left ring and an exclusive merge on the right ring. However, a different configuration is visible in the nodal subnetwork view.) The connection between termination points on the nodal subnetwork at C is shown in the lower right of Figure A-2. Termination points associated with the high-speed (ring) interfaces are labeled z1 and z2; the termination point on the low-speed side is labeled c. The signal entering at z1 is split to provide a path to c and the drop and continue path to node D (exits as a sink at z2). The signal which was passed on by the *continue* at node D enters as a source TP at z2 and the selector function is applied to this signal and the signal from z1 forming an exclusive merge.

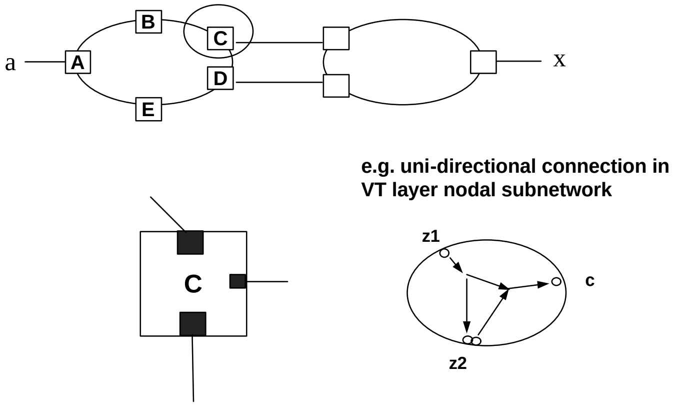


Figure A-2 Nodal Subnetwork at Node C

The combined configuration of split and exclusive merge among the four uni-directional TPs is modeled as a drop and continue connection type.

In Figure A-3 the DCS at node C is assumed to be managed by a different EMS than that managing the ring. The connection across the left-side (open) ring is shown on the lower diagram in the figure. In this case, five uni-directional termination points are involved.

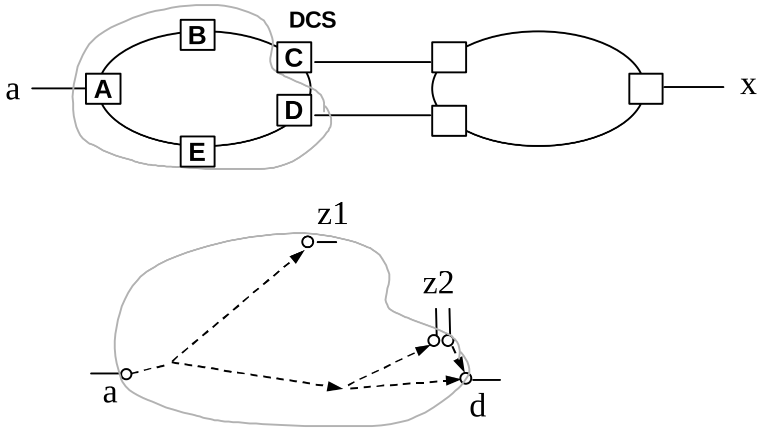


Figure A-3 Ring subnetwork view – open at interconnection node

Several alternative methods have been proposed address the more complex connection types such as the one described in this appendix.

- Add new connection types
- Allow a TP to participate in more than one subnetwork connection
- Make internal TPs explicit and/or subdivide the subnetwork into multiple subnetworks
- Model an open ring as a closed ring with a phantom node
- Use the NE view model to make cross-connections

The fifth method is a real alternative but is not discussed here since there are benefits when the network view is generally applicable.

It is difficult to foresee all possible combinations that may be needed in supporting various interconnection and administrative partitioning scenarios for SONET networks. The scenario in this appendix generates configurations that are artifacts of forced administrative partitionings that do not lend themselves very well to simple modeling constructs in the ring subnetwork partitioned view.

The resulting complexity is a natural consequence of assembling an integrated topology such as a ring from different NEs that are not managed in an integrated manner. The case of multi-vendor rings managed by a single EMS does not present this problem. To support scenarios such as the one in this appendix we currently require the nodal subnetwork view be used.