Multi-Technology Network Management (MTNM)

Business Agreement

Release 3.5

TMF 513

Member Evaluation, Version 3.1 *March 2007*

Notice

Direct inquiries to the TM Forum office:

240 Headquarters Plaza,

East Tower - 10th Floor,

Morristown, NJ 07960 USA

Tel No:	+1 973 944 5100
Fax No:	+1 973 944 5110
TM Forum Web Page	www.tmforum.org

Table of Contents

List of Figures

List of Tables

List of Requirements

List of Use Cases

List of UML Classes

Executive Summary

The TM Forum's Multi-Technology Network Management (MTNM) effort has the goal of defining a unified open interface to be used between an OS that exhibits Element Management Layer (EML) functionality and an OS that exhibits Network Management Layer (NML) functionality as defined in the ITU-T TMN model.

In terms of the TM Forum's enhanced Telecom Operations Map (eTOM), the intended coverage is Fulfillment and Assurance with respect to the Resource Management and Operations (RM&O) and Service Management and Operations (SM&O) process groupings. In terms of technology, the goal is to cover the management of all technologies from layer 1 (e.g., SONET/SDH) to high layer technologies such as VoIP.

This document provides the requirements and USe Case for MTNM.

1 Introduction

1.1 Document Overview

This document defines the business agreement for the exchange of management information within a model of telecommunication management business functions documented in TM FORUM's enhanced Telecom Operations Map (eTOM). Specifically, the Business Agreement captured in this document defines the information exchange, or interface, between Network Management Systems (NMS) and Element Management Systems (EMS) enabling management of different transport network technologies (e.g., SONET/SDH, ATM, DWDM, etc.).

This document covers the NML-EML interface.

As the problem of managing next-generation networks composed of hybrid network elements become more and more prevalent, previously technology-specific interface solutions become insufficient. The goal of the team is to provide a single common solution to enable management of these multi-technology network elements.

1.2 Document Structure

The Business Agreement contains the business problem statement including the scope, objectives and business scenarios. The bulk of the Business Agreement is the categorized requirement statements, the use cases, as well as the associated diagrams that model the static and dynamic relationships defined by the requirement statements and use cases.

This document has two major parts, the Business Problem Statement and the Business Requirement Model. The following is the structure of the BA:

Section 1	**Introduction**: Provides overview of this the document and outlines its structure and defines essential terms used in the document.

Business Problem Statement:

Business Requirement Model:

Issues and Appendices:

2 Business Problem Description, Project Scope and Objectives

2.1 Business Problem

As the communications industry continues to evolve with deregulation and liberalization, service providers are under increased pressure to deliver a broadened set of services at competitive price points. As a result, service providers must deliver these services in an efficient, cost-effective and timely manner. The ability to effectively manage these networks become key in retaining existing, as well as acquiring new market share. However, the complexities surrounding today's networks present challenges in achieving the effective network management goal.

The composition of today's networks has contributed to the complexity of managing these networks. These networks are commonly composed of network elements provided by various vendors. The task of interoperability extends beyond the network element layer up to the management layers, to include interoperability between multi-vendor Element and Network Management Systems.

Service Providers have also taken advantage of technological advancements in transport network equipment. It is not uncommon for service providers to deploy next generation, multi-technology network elements, (or "hybrid NEs"), as these network elements provide new services and optimal network resource utilization. However, network management solutions that have been specified to date applies only to a specific technology.

There is an industry demand for a full-featured, commercially available, scalable and non-proprietary network management solution, where multi-vendor, multi-technology management systems interoperate in an open architecture environment.

2.2 Supported Business Scenarios

The NML-EML interface has been specified to support a wide variety of network management business scenarios from the provisioning of connections to the retrieval of equipment inventory information. The fact that this document describes the requirements for an network management interface means that there are a large number of business scenarios that may be supported by the a network management interface that meets the requirements specified in this document. Examples of the business scenarios that may be supported include, but are not limited to:

- Inventory Discovery
- Connection Provisioning
- Equipment Provisioning
- Performance Management

For specific cases of these and other business scenarios please refer to MTNM Catalyst IIS.

2.3 Project Scope

There is an immediate industry need, (as expressed by Service Providers) to provide a single, common solution in managing multi-technology networks, (including SONET, SDH, DWDM, and ATM). Today, technology-specific forums, (i.e. NSIF, ATMF, OIF, etc.) produce separate solutions or interfaces specific to their technology areas. The objective of the team is to provide a single common solution to address:

- Multiple transport network technologies – including SONET, SDH DWDM and ATM
- Hybrid network elements that support multiple technologies, (as illustrated in the following diagram)

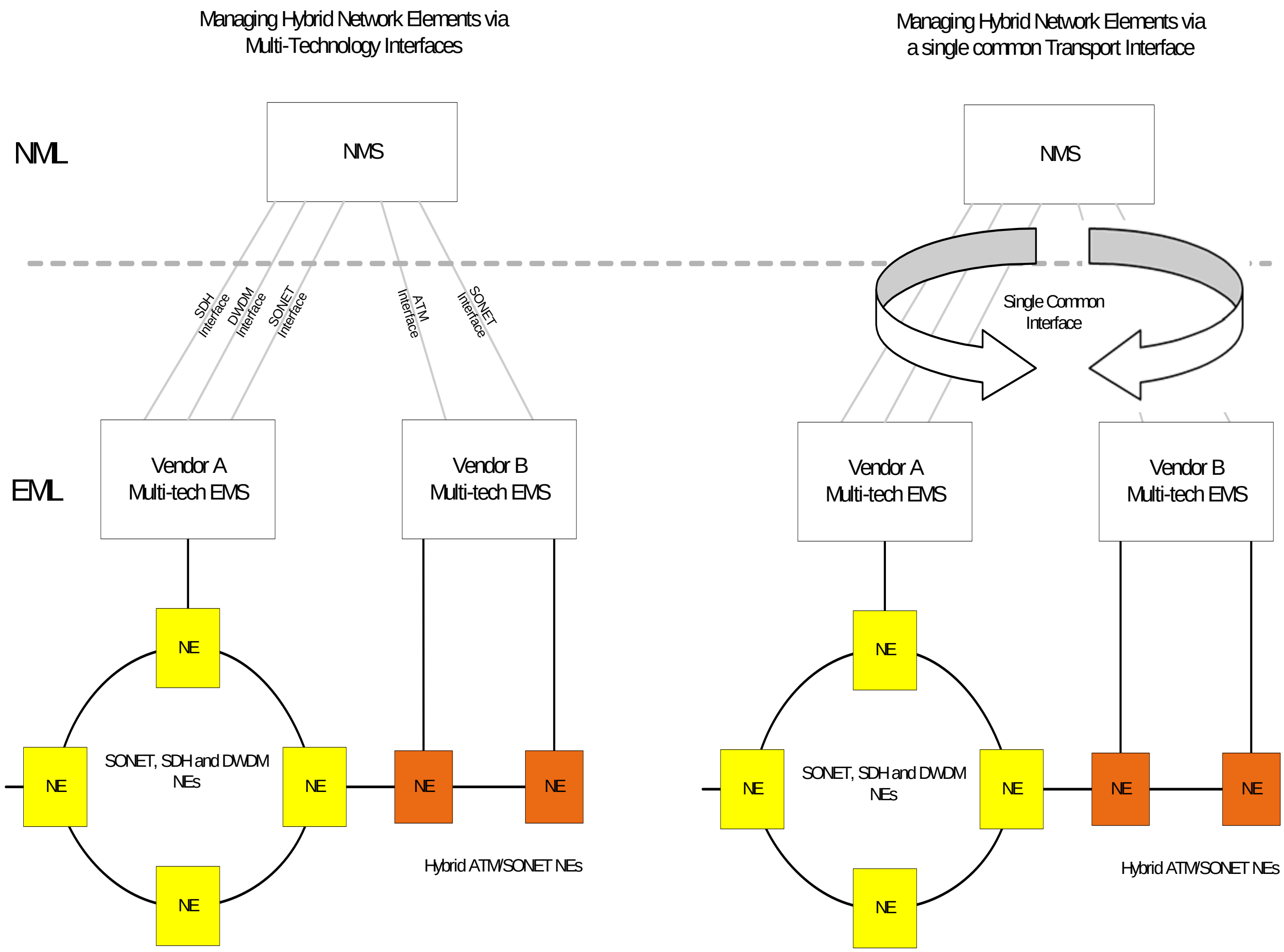


Figure 2.1: Project Scope

2.4 Project Objectives

The specific objective for the Multi-technology Network Management modeling team includes:

- Utilize the work of the Phase I SSIM and ATMIM groups by reviewing, identifying and resolving the difference in the business requirements and UML models to create a common network management model to manage multi-technology networks.
- Build support of DWDM management and enhance management features to the NML-EML Interface.
- The development of Protocol-Independent Information Models, supporting Use Cases for a multi-technology NML-EML Interface.

- Develop CORBA IDL specifications needed to implement the model and demonstrate its viability
- Assurance that the top-down approach is taken and that the Network Management processes from the higher layers, BML, SML and NML, by collaboration with CaSMIM, are consistently fulfilled at the lower layers of the TMN, (NML to EML) to achieve complete flow-through.

2.5 Benefits Gained by Solving this Business Problem - Business Case

2.5.1 Overall benefit of solving business problem

The expected beneficiaries are service providers, customers and vendors. The major benefits are due to a flexible and extensible common solution to the management of multi-technology network elements in a network using multi-vendor, multi-technology management systems.

2.5.2 Benefits to the end user or customer

As a consequence of the Service Providers' benefits, the customers will benefit by:

An improved service: the provisioning of new service will be fast and the unavailability of the service will be limited due to the correlation of network fault to services.

Reduction of prices: The reduction of Service Provider cost will lead to the reduction of service prices.

2.5.3 Benefits to a Service Provider

The multi-technology NML-EML solution will provide Service Providers with an efficient way to manage their next-generation networks composed of hybrid network elements. The solution will ensure multi-vendor interoperability as it will be an open and accessible to the industry.

Service Providers will benefit from a common multi-technology network management interface able to translate the service definitions that are made in Service Management Systems into technology specific implementations. It will result in:

- Rapid Service Delivery by decreasing service activation time
- Fast introduction of new technologies: New Technologies can be introduced in the Network without causing major changes to the interfaces at the Network Management Level.
- Increased Operating Efficiency through automation and integration of network management and element management systems.
- Enable multi-vendor interoperability affording service providers the option of choosing various vendor products.

2.5.4 Benefits to a Vendor and/or Systems Integrator

Agreements at TMF, the ATM Forum, the ADSL Forum, the Network and Services Integration Forum and other Industry fora address how to interconnect technology specific equipment from different vendors and manage the resultant network. This leaves a major area of uncertainty: how to manage mixed technology environments. The providers of management systems can respond to service provider requests with custom designs based on the technologies employed, the interfaces supported by the peer and north/southbound systems, and the specific Service Provider requirements - but this is prohibitively expensive and consumes a large amount of scarce development resources. The result is proposals for network management solutions that take too long and cost too much. Neither Service Provider nor vendor benefits. The goal: solutions that meet the most important needs of the Service Provider and are available quickly, at reasonable cost, and provide flexibility for changes in the futures as new technologies are added. The work here provides requirements for common interfaces to support this goal. Hence, management system vendors and software providers will be able to develop network management systems and applications that can be used by multiple Service Providers, and which support multiple technologies in a multi-vendor environment with increased efficiency and reduced cost.

2.5.5 Benefits to the TM Forum

The work of the team aligns with the TM Forum TOM in the network and element layers in the areas of Network Planning, Provisioning, Inventory Management, and Monitoring. Additionally, the team is solving an immediate industry need for a solution that is applicable across multiple technologies, including SONET/SDH, DWDM and ATM.

3 Business Processes

3.1 Process Definition and Issues

This section defines the business process issues to be addressed in this document. It provides a business process view of the requirements to be satisfied through a standardized NML-EML interface. This document defines the business need, formal requirements, and scenarios in the form of UML Use Cases that are needed to support business process that interact with the Network Elements.

The nature of this project does not directly address any specific business process but is dedicated to provide support for most known business process that rely on an NML-EML interface. The end product shall be an interface that is designed to support a large variety of business processes that depend on and support multiple technologies, including SONET/SDH, DWDM and ATM.

3.2 Mapping of Processes to eTOM Business Process Framework

3.2.1 eTOM Business Process Framework

The eTOM is a business process framework or model that provides the enterprise processes required for a service provider. It is not a service provider business model. In other words, it does not address the strategic issues or questions of who a service provider's target customers should be, what market segments should the service provider serve, what are a service provider's vision, mission, etc. A business process framework is one part of the strategic business model and plan for a service provider.

The eTOM is based on the Telecom Operations Map (TOM). The eTOM broadens the TOM to a total enterprise framework and addresses the impact of ebusiness. Although the eTOM is more complex than the TOM, in some ways it is more intuitive than the TOM in that it closes gaps in enterprise management (i.e., corporate-type) processes, marketing processes, customer retention processes, supplier and partner management processes, etc.

Figure 3.1: eTOM Business Process Framework Conceptual Structure

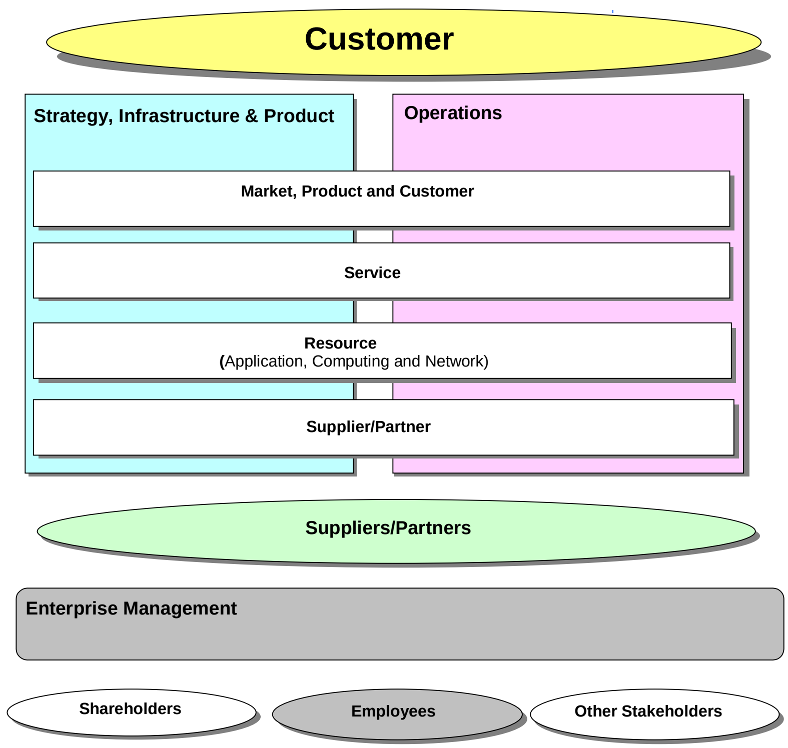


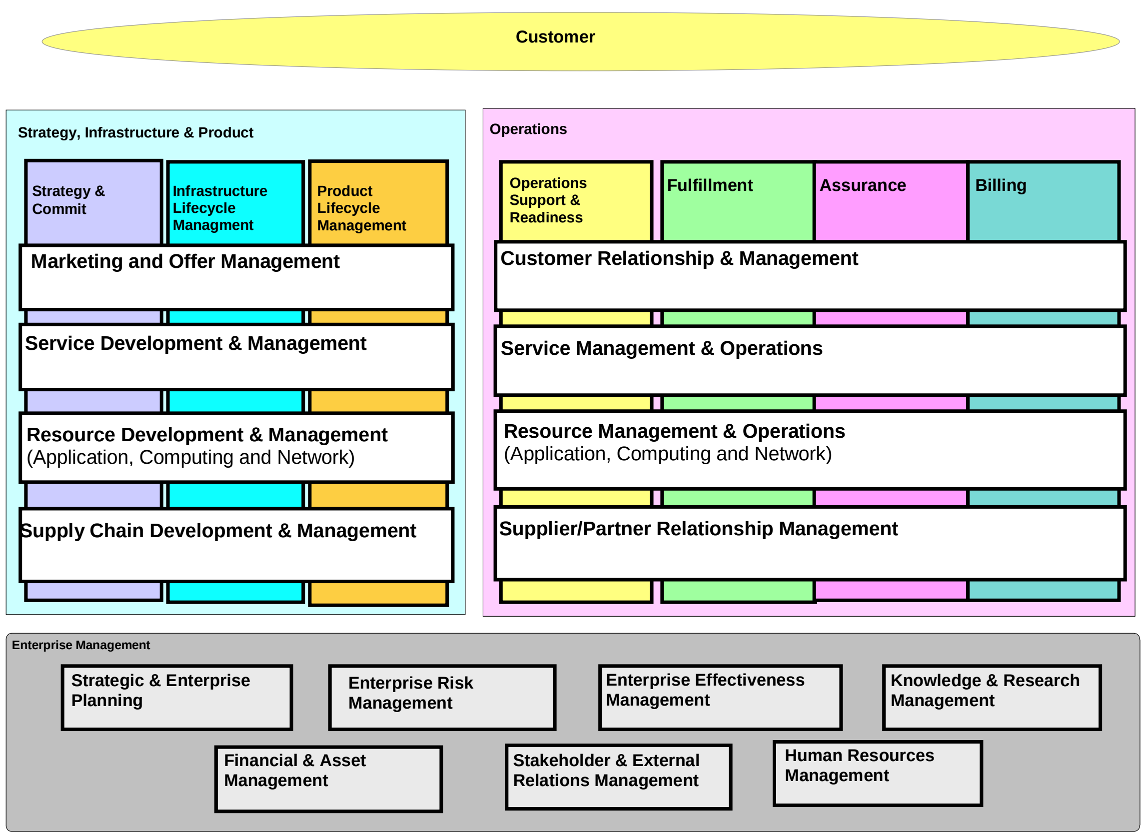
Figure 3.1 shows the highest conceptual view of the eTOM Business Process Framework. This view provides an overall context that differentiates strategy and lifecycle processes from the operations processes in two large process areas, seen as the two major boxes in the upper part of the diagram. It also differentiates the key functional areas as horizontal layers acrtoss these process areas. The third major process area, concerned with the management of the enterprise itself, is shown as a separate box in the lower part of the diagram. In addition, Figure 3.1 shows the internal and external entities that interact with the enterprise (as ovals).

Figure 3.2 shows how the three major process areas - designated as Level 0 processes of the eTOM business process framework - are decomposed into their constituent Level 1 process groupings. This view thus provides the Level 1 decomposition of the Level 0 processes and gives an overall view of the eTOM framework. However, in practice it is the next level - the Level 2 decomposition of the Level 1 processes - at which users tend to work, as this degree of detail is needed in analyzing their businesses.

Figure 3.2 shows seven end-end vertical process groupings, that are the end-to-end processes that are required to support customers and to manage the business. Amongst these end-end Vertical Process Groupings, the focal point of the eTOM farmework is on the core customer operations processes of Fulfillment, Assurance and Billing (FAB). Operations Support & Readiness (OSR) is differentiated from FAB real-time processes to highlight the focus on enabling support and automation in FAB, i.e.. on line and immediate support of customers, with OSR ensuring that the operational environment is in place to let the FAB processes do their job. Outside of the Operations process area - in the Strategy, Infrastructure & Product (SIP) process area - the Strategy & Commit vertical, as well as the two Lifecycle Management verticals, are differentiated. These are distinct because, unlike Operations, they do not directly support the customer, are intrinsically different from the Operations processes and work on different business time cycles.

The horizontal process groupings in Figure 3.2 distinguish functional operations processes and other types of business functional processes, e.g., Marketing versus Selling, Service Development versus Service Configuration, etc. Amongst these Horizontal Functional Process Grouppings, those on the left (that cross the Strategy & Commit, Infrastructure Lifecycle Management and Product Lifecycle Management vertical process groupings) enable, support and direct the work in the Operations process area.

Figure 3.2: eTOM Level 0 View of Level 1 Process Groupings



The MTNM NML-EML Interface addresses the problems arising in the Resource Management & Operations.

Resource Management & Operations (RM&O): this process grouping maintains knowledge of resources (application, computing and network infrastructures) and is responsible for managing all these resources.(e.g. networks, IT systems, servers, routers, etc.) utilized to deliver and support services required by or proposed to customers. It also includes all functionalities responsible for the direct management of all such resources (network elements, computers, servers, etc.) utilized within the enterprise. These processes are responsible for ensuring that the network and information technologies infrastructure supports the end-to-end delivery of the required services. The job of these processes is to ensure that infrastructure runs smoothly, is accessible to services and employees, is maintained and is responsive to the needs, whether directly or indirectly, of services, customers and employees. RM&O also has the basic function to assemble information about the resources (e.g.. from network elements and/or element management systems), and then integrate, correlate, and in many cases, summarize that data to pass on the relevant information to Service Management systems, or to take action in the appropriate resource.

In the original TOM Business Process Framework, the "Network and Systems Management" processes were included at the highest, most general level. This is no longer adequate in an ebusiness world. Application and computing management are as important as network management. Moreover, network, computing and applications resources must increasingly be managed in a joint and integrated fashion. To cope with these needs, eTOM has introduced the Resource Management & Operations process grouping (together with the corresponding Resource Development & Management grouping within SIP), to provide integrated management across these three sets of resources: applications, computing and network. These areas also combine the Network Element Management processes of the TOM, since these processes are actually critical components of any resource management process, as opposed to a separate process layer.

The RM&O processes thus manage the complete service provider network and sub-network and information technology architectures.

eTOM differentiates day-to-day operations and support from planning and development, and other strategy and lifecycle processes. In the TOM, these resource layer processes were not differentiated or were not addressed. The eTOM structure better depicts the structure of an enterprise, especially in an ebusiness era.

4 Requirements

This section lists the single concept, requirement statements to be fulfilled by the Business Requirement Model.

4.1 Category I: Static and Structural Requirements

The requirements for the NML-EML Interface have been specified in terms of the entities (objects) that are visible across the interface and the operations that may be performed on these objects. A number of the objects visible across the interface are required to have a common set of attributes. Table 4.1 identifies the objects that shall have a common set of attributes.

Table 4.1: Objects containing a common set of attributes

	Object Name
1	Alarm Severity Assignment Profile (ASAP)
2	Call
3	Connection
4	Element Management System (EMS)
5	Equipment
6	Equipment Holder
7	Equipment Protection Group (EPG)
8	Flow Domain (FD)
9	Flow Domain Fragment (FDFr)
10	Group Termination Point (GTP)
11	Subnetwork Point Pool Link (SNPP Link)
12	Managed Element (ME)
13	Matrix Flow Domain (MFD)
14	MultiLayer Routing Area (MLRA)
15	MultiLayer Subnetwork (MLSN)
16	MultiLayer Subnetwork Point Pool (MLSNPP)
17	MultiLayer Subnetwork Point Pool Link (MLSNPPLink)
18	Performance Monitoring Point (PMP)
19	Protection Group (PG)
20	Subnetwork Connection (SNC)
21	Termination Point (TP)[a]
22	Termination Point Pool (TP Pool)

Table 4.1: Objects containing a common set of attributes

	Object Name
23	Threshold Crossing Alert (TCA) Parameter Profile
24	Topological Link (TL)
25	Traffic Conditioning (TC) Profile
26	Traffic Descriptor (TD)
27	Transmission Descriptor (TMD)

a. This includes all types of TP (i.e. PTPs, CTPs and FTPs)

4.1.1 Common Attributes

{Requirement I. 060} All objects identified in Table 4.1 shall have the following attributes:

1) Name

   This attribute represents a unique, invariant (for the life of the object) identifier for the object within the Element Management System (EMS) management domain.

2) User label

   This attribute represents a provisionable, user friendly name for the object. It shall be initially set to the same value as the native EMS name attribute by the EMS. This attribute is owned and shall be provisionable by the NMS.

3) Native EMS name

   This attribute represents the name of the object as presented on the EMS GUI. The native EMS name attribute is owned by the EMS which may or may not support changing this value.

4) Owner

   This attribute represents an identifier for the owner of the object. This attribute shall be provisionable by the NMS.

5) Additional information

   This attribute represents data (i.e., attributes) that has not been explicitly modeled as part of the object and allows for this data to be exchanged across the NML-EML Interface.

Source: Version 3.0.

Mandatory/Optional: Mandatory

4.1.2 Element Management System (EMS)

{Requirement I. 001} The Element Management System (EMS) object shall represent the abstraction of the MultiLayer Subnetwork (MLSN)(s) managed by the EMS (i.e. Subnetworks in the EMS management domain) and the EMS itself.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

{Requirement I. 061} The EMS object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Software version

   This attribute shall represent the version of the EMS vendor software.

2) Type

   This attribute shall represent the vendor and type of the EMS.

3) Alarm severity assignment profile

   This attribute shall represent the name of the Alarm Severity Assignment Profile (ASAP) that has been assigned to the EMS.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.3 Managed Element (ME)

{Requirement I. 002} A Managed Element (ME) object shall represent the EMS view of a Network Element (NE).

Source: Version 2.0.

Mandatory/Optional: Mandatory.

{Requirement I. 003} An ME object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Location

   This attribute shall represent the geographical location of the ME

2) Software version

   This attribute shall represent the software version of the ME.

3) Product name

   This attribute shall represent the ME vendor's name/designation for the product (i.e. the name the vendor uses to identify the Network Element (NE)).

4) Communication state

   This attribute shall represent the state of the current connectivity between the EMS and the Network Element.

5) Supported connection layer rate(s)

   This attribute shall represent a list of the rates for which cross-connects can be established in the ME.

6) Synchronization state

   This attributes shall indicate whether the EMS is able to keep its data synchronized with the Network Element (NE) data and generate all appropriate notifications relating to changes in the data.

7) Network Access Domain

   This attribute shall represent the Network Access Domain to which this ME has been assigned.

8) Manufacturer

   This attribute shall represent the ME vendor name.

9) Alarm severity assignment profile

   This attribute shall represent the name of the Alarm Severity Assignment Profile (ASAP) that has been assigned to the ME.

10) Manufacturer date

   This attribute shall represent the date of manufacture of the ME.

Source: Version 3.5.

Mandatory/Optional: Mandatory.

4.1.4 Termination Point (TP)

{Requirement I. 004} A Termination Point (TP) object shall represent a logical abstraction of an endpoint (actual or potential) of either:

1) a Topological Link (TL) or
2) a Subnetwork Connection (SNC)
3) a Flow Domain Fragment (FDFr)
(if EMS supports connectionless layers).

Source: Version 3.5.

Mandatory/Optional: Mandatory.

{Requirement I. 005} A TP object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Directionality

   This attribute shall represent the directionality of the TP (bidirectional, source, or sink).

   An ATM TP will always have directionality set to bi-directional.

2) Protection association

   This attribute shall represent whether the TP is participating in a Path Switched Ring (PSR) protection scheme with another TP. (For example in a multi-layer subnetwork, if ‘a’, ‘b’, ‘c’ are edge points and a three-ended connection is required from ‘a’ to ‘b’, where ‘b’ is one of the endpoints. If ‘c’ is the constrained choice for ‘b’ as the other end of the three-ended connection, then ‘b’ and ‘c’ are said to be associated by a protection association.)

3) Edge Point

   This attribute represents whether the TP is at the edge of a subnetwork. (Refer to {Requirement I. 008}).

4) Network Access Domain

   This attribute shall represent the Network Access Domain to which this TP has been assigned.

5) Equipment protected

   This attribute shall indicate whether or not the TP is supported by Equipment that is protected.

6) Ingress TMD state

   This attribute shall indicate the state of consistency between a TP and its associated ingress Transmission Descriptor (TMD).

7) Egress TMD state

This attribute shall indicate the state of consistency between a TP and its associated egress Transmission Descriptor (TMD).

8) GTP or TPPool

This attribute shall represent the name of the Group Termination Point (GTP) or Termination Point Pool (TP Pool) of which this TP is a member, if applicable

9) Layered transmission parameters

This attribute shall represent the transmission parameters associated with the different layers that are either encapsulated within the TP or not encapsulated but configured via the TP. Refer to the supporting document SD1-16_LayeredParameters.pdf for details of the currently defined transmission parameters.

10) Ingress TMD

This attribute shall represent the name of the ingress Transmission Descriptor (TMD) associated with this TP.

11) Egress TMD

This attribute shall represent the name of the egress Transmission Descriptor (TMD) associated with this TP.

12) TCA parameter profile

This attribute shall represent the name of the Threshold Crossing Alert (TCA) Parameter Profile associated with this TP.

13) Alarm severity assignment profile

This attribute shall represent the name of the Alarm Severity Assignment Profile (ASAP) that has been assigned to the TP.

14) Performance monitoring point

This attribute shall represent the names of the Performance Monitoring Point (PMP)s associated with this TP.

Source: Version 3.5.

Mandatory/Optional: Mandatory.

4.1.5 Physical Termination Point (PTP)

{Requirement I. 007} A Physical Termination Point (PTP) object shall represent the actual or potential endpoint of a Topological Link (TL).

Essentially, this is a representation of a physical port. Examples of PTPs are T1 ports, T3 ports, OC-N optical ports, etc.

PTPs have a containment relationship with Connection Termination Point (CTP)s. PTPs forming a UPSR pair are related and contain related CTPs. If the PTP is potentially able to support a CTP on a particular layer rate, then that CTP shall be represented at the EML-NML Interface.

CTPs are always clients of the PTP but that the layer relationship may reverse.

Each bidirectional PTP (or ATMNI CTP) may be either endpoint of one bidirectional topological link or of up to two unidirectional topological links being the aEnd (source) of one of these topological links and the zEnd (sink) of the other.

Each unidirectional PTP (or ATMNI CTP) must not be the endpoint of more than one unidirectional topological link.

A PTP object is a type of Termination Point (TP) object that shall have the attributes identified in {Requirement I. 005}.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.5.1 Edge Termination Point (Edge TP)

{Requirement I. 008} An Edge Termination Point (Edge TP) is a Termination Point (TP) that is at an entrance or exit point of a MultiLayer Subnetwork (MLSN), (i.e. add-drop or TPs that terminate topological links between two Subnetworks).

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.6 Connection Termination Point (CTP)

{Requirement I. 006}

A Connection Termination Point (CTP) object shall represent the actual or potential end point of either:

1) A Subnetwork Connection (SNC) or
2) An ATM Network Interface (ATMNI) at the Network Interface layer rate.

A CTP may be contained by and be the client of a Physical Termination Point (PTP), an Floating Termination Point (FTP) or a CTP. A CTP may be contained by and be the server of an FTP or a CTP (via inverse multiplexing).

Source: Version 2.0.

Mandatory/Optional: Mandatory.

{Requirement I. 062}

A CTP object is a type of Termination Point (TP) and therefore shall have, in addition to the attributes identified in {Requirement I. 005}, the following attributes:

1) Connection state

   This attribute shall indicate whether the CTP is involved in an active cross-connection at the CTP's rate. (Refer to {Requirement I. 023}).

2) Mapping mode

   This attribute shall indicate if the CTP is configured such that it is capable of supporting lower rate connections, or if the CTP is configured such that it is capable of supporting cross-connections at the TP's rate. Refer to Section 4.2.2.1.1.

3) Ingress TD

   This attribute shall represent the name of the ingress Traffic Descriptor (TD) associated with this CTP.

4) Egress TD

   This attribute shall represent the name of the egress Traffic Descriptor (TD) associated with this CTP.

Note:

The ingress and egress traffic descriptor attributes may be empty in the following cases:

i) For technologies that don't use the TD concept. (Currently only used on ATM CTPs).
ii) If the CTP is involved in a uni-directional connection.
iii) If there is no assigned bandwidth when an implicit VPCTP is created.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.7 Floating Termination Point (FTP)

{Requirement I. 075} A Floating Termination Point (FTP) object shall represent a Termination Point (TP) that is not directly supported by a physical port.

An FTP is a TP without a physical layer that behaves both like a Physical Termination Point (PTP) and a Connection Termination Point (CTP):

- FTPs (client side) behave like PTPs wherever PTPs are used in NML-EML Interface.
- FTPs (server side) behave like CTPs wherever CTPs are used in NML-EML Interface.

FTPs have a containment relationship with CTPs. The FTP will contain CTPs in client layers and may contain CTPs in Server layer (via inverse multiplexing).

A FTP object is a type of CTP object and also a type of PTP object and shall have the attributes identified in {Requirement I. 062} and {Requirement I. 007}.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.1.8 Group Termination Point (GTP)

{Requirement I. 069} A Group Termination Point (GTP) object shall represent a sequence of Connection Termination Point (CTP)s (with a specific order) in the same Managed Element (ME).

GTPs shall have the following behavior:

1) The CTPs comprising a GTP need not be contiguous.
2) A CTP can not belong to more than one GTP at a time.
3) Once a CTP is included in a GTP, it can not be cross-connected independent of the GTP.
4) All CTPs in a GTP must have the same Connection State and be in the same Network Access Domain.
5) A GTP shall be named with respect to the containing ME.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement I. 070} A GTP object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Contained TPs

   This attribute represents a list of the names of the Termination Point (TP)s that are contained by the GTP.

2) Connection State

   This attribute represents whether the GTP is involved in an active cross-connection. (Refer to {Requirement I. 023}).

3) Network Access Domain

   This attribute represents the Network Access Domain to which this GTP has been assigned.

4) Alarm reporting

   This attribute shall indicate whether alarm reporting for the GTP is enabled or disabled.

5) Alarm severity assignment profile

   This attribute shall represent the name of the Alarm Severity Assignment Profile (ASAP) that has been assigned to the GTP.

6) TP Pool

   This attribute shall represent the name of the Termination Point Pool (TP Pool) of which this GTP is a member, if applicable

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.1.9 Termination Point Pool (TP Pool)

{Requirement I. 009}

A Termination Point Pool (TP Pool) object shall represent a grouping (without a specific order) of Termination Point (TP)s or Group Termination Point (GTP)s from the same Subnetwork, for some administrative management purposes (e.g. bandwidth reservation, common routing etc.).

The members of a TP Pool are administered and used independently of each other but for a common purpose. A Connection Termination Point (CTP) that is contained in a GTP cannot be a member of a TP Pool but the whole GTP can. A CTP that is contained in a TPPool cannot be added to a GTP. A TP or GTP can be a member of at most one TPPool.

For ATM, this object shall be used to support the administrative partitioning of the ATM Network Interface (ATM NI) consisting of a reserved range of Virtual Path Identifier (VPI) values and bandwidth. In this example the set of potential client CTPs of the ATM NI CTP is partitioned into a set of TP Pool objects. While the ATM NI CTP represents a real network interface the TP Pool objects represent virtual network interfaces consisting of a set or range of ATM VP CTPs

If an Element Management System (EMS) does not support the concept of administrative partitioning, then **no** TP Pool objects shall be present at the NML-EML Interface.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

{Requirement I. 094} A TP Pool object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Contained members

   This attribute shall represent a list of the names of the Termination Point (TP)s or Group Termination Point (GTP)s that are contained by the TPPool.

2) Number of members

   This attribute shall represent the total number of currently contained TPs or GTPs.

3) Number of idle members

   This attribute shall represent the number of currently contained idle TPs or GTPs that are free to be used for the intended management purpose (e.g., potential ATM VP CTPs that are currently not "in use").

4) Layered transmission parameters

   This attribute shall represent the common layers and transmission parameters (if any) associated with the contained TPs (or the TPs contained in contained GTPs). (Refer to {Requirement I. 005}.)

5) Description of use

   This attribute shall optionally describe the specific use of the TP pool, in particular how its members are collected and administered. For example, the description of use of an ATM VP TP pool could be "Virtual UNI".

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.1.10 Topological Link (TL)

{Requirement I. 010}

A Topological Link (TL) is a physical link between two Physical Termination Point (PTP)s or a trail between two Termination Point (TP)s (e.g., an ATM link between two ATM NI CTPs), which are called aEnd TP and zEnd TP of the TL.

The layer rate of a TL is determined by the lowest common layer rate of the two end point TPs, about which the EMS has knowledge (with the physical layer being the very lowest).

A TL reported by an EMS to the NMS will usually be between two Network Elements (NEs) that are managed by the same EMS but need not be adjacent. Depending on the capabilities of the EMS and the NEs, a TL may or may not be auto-discovered by the EMS.

The end points of a TL may belong to different MultiLayer Subnetwork (MLSN)s, in which case it is called a "*top-level TL*", or to the same Subnetwork, and then the TL is called an "inner TL".

For a TP that is connected outside of the EMS' span of control, if the EMS knows about the remote end, the EMS may provide this information via a single-ended TL, called an "*off-network TL*". Such a TL is referred to as single-ended since only one end point, namely the aEnd TP, belongs to the EMS managing the TL the other (i.e., the zEnd TP) being off-network and being reported as a remote address (and possibly being managed by another EMS). An off-network TL is considered as a top-level TL.

The end points of a TL may also belong to the same ME, in which case it is called an "*internal TL*".

The TL may be created/deleted by the NMS and is implemented by the EMS.

The TL is contained in the EMS. A TL may traverse through DWDM equipment and optical transport networks managed by another EMS. The TL reported by the first EMS in that case could correspond to a (regenerator) section under its management domain. It is up to the NMS user to put together the exact topology of the network in such cases, based on the reported MEs and TLs.

For unidirectional TLs the aEnd TP marks the traffic source whereas the zEnd TP marks the traffic sink.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

{Requirement I. 011} A TL object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Directionality

This attribute shall represent the directionality of the TL. A TL may be unidirectional or bidirectional. A unidirectional TL may connect to bidirectional TPs.

2) aEnd Termination Point (TP)

This attribute shall represent the name of the A end TP.

3) zEnd Termination Point (TP)

This attribute shall represent the name of the Z end TP.

4) Layer rate

This attribute shall represent the layer rate of the TL. Refer to {Requirement I. 022}.

5) Network Access Domain

This attribute represents the Network Access Domain to which this TL has been assigned.

6) Alarm reporting

This attribute shall indicate whether alarm reporting for the TL is enabled or disabled.

7) Alarm severity assignment profile

This attribute shall represent the name of the Alarm Severity Assignment Profile (ASAP) that has been assigned to the TL.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.11 MultiLayer Routing Area (MLRA)

{Requirement I. 116}

{R I. 001}

The MultiLayerRoutingArea (MLRA) object shall represent a MultiLayer Routing Area that is managed by the EMS system.

In the Control Plane context an MLRA consists of multiple congruent Routing Areas (RAs) at different layers.

The Control Plane RA consists of a set of subordinate routing areas, the Subnetwork Point Pool Link (SNPP Link)s that interconnect them, and the Subnetwork Point Pool (SNPP)s representing the ends of the SNPP Links exiting that RA. The way that RAs and SNPP Links are grouped in RAs is entirely determined by the EMS.

MLRAs are organized in a hierarchical structure based upon “containment” (in which the lower level MLRAs are completely contained within a single higher level MLRA). The MLRA can contain other MLRAs and this recursion can continue as many times as necessary.

A top-level RA coincides with the whole Control Plane Network. The information on MLRA (subordinated MLRAs) can be accessed through operations on the containing MLRA.

In a non-Control Plane context an MLRA consists of multiple MultiLayer Subnetwork (MLSN)s and provides a mechanism to allow for the provisioning of Connections across multiple MLSNs.

Source: Version 3.5.

Mandatory/Optional: Optional

{Requirement I. 117}

{R I. 002}

The MultiLayerRoutingArea object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Layered Routing Area List

   This attribute shall specify a list of the Control Plane names for the congruent RAs, one for each layer rate.

2) Routing Area Level

   This attribute shall specify the level of the MLRA in the routing hierarchy. Valid values for this attribute are: “TopLevelRA”, “IntermediateLevelRA”, and “RoutingNodeLevel”.

3) Supporting ME Name

   This attribute shall specify the name of the supporting Managed Element (ME). When the MLRA is a routing node,

4) Shared Risk Group

   This attribute is applicable when the MLRA is a routing node. This attribute shall be a sequence of {risk type, and sequence of strings}.

Note:

   The behavior when routing a connection that specifies a risk type that is not present in a RA is dictated by the Control Plane implementation.

5) Superior MultiLayer Routing Area

   This attribute shall identify the superior MLRA of the prescribed MLRA.

6) Supporting MultiLayer Subnetwork List

   This attribute shall specify a list of MultiLayer Subnetwork (MLSN) names that support the MLRA.

Source: Version 3.5.

Mandatory/Optional: Mandatory

4.1.12 MultiLayer Subnetwork (MLSN)

{Requirement I. 012}

A MultiLayer Subnetwork (MLSN) object shall represent the topology provided by the EMS system.

The main services provided within a Subnetwork are the set-up and tear-down of Subnetwork Connection (SNC).

A Managed Element (ME) may belong to more than one MLSN, at different layer rates (e.g. SDH & ATM). However, MLSNs cannot overlap at the same layer rate.

MLSNs are created/deleted/modified by the EMS only. (Note that the NMS may internally define its own representation of an MLSN).

The NMS initially discovers the existing MLSNs and is later notified by the EMS of MLSN creations, deletions or changes.

Refer to supporting document SD1-18_layers.pdf for more information on the concept of MLSNs.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

{Requirement I. 013}

An MLSN object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Type

   Refer to {Requirement I. 038}

2) Supported SNC layer rate(s)

   Refer to {Requirement I. 022}

3) Network Access Domain

   This attribute represents the Network Access Domain to which this MLSN has been assigned.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.12.1 MultiLayer Subnetwork (MLSN) Type

{Requirement I. 038} The NML-EML Interface shall support the management of the following network topologies or MultiLayer Subnetwork (MLSN) types:

1) Singleton

Used for a single NE that is managed independently of its Topological Link connectivity to other NEs. It may for example be a member of a ring that is managed by a number of EMS'. It is acceptable for an EMS to represent all NEs as being in Singleton Subnetworks regardless of the actual network configuration. A singleton Subnetwork does not contain internal topological links.

2) Chain

Used to cover the case where two or more NEs are managed by the same EMS and are connected by Topological Links in a chain.

3) Path Switched Ring (PSR)

Used to cover the case where two or more NEs are managed by the same EMS and are connected by Topological Links in a ring that is capable of supporting Subnetwork Connection protection. (e.g. UPSR, SNCP Ring)

4) Shared Protection Ring (SPRing)

Used to cover the case where two or more NEs are managed by the same EMS and are connected by Topological Links in a complete ring that supports Shared Line Protection (e.g. BLSR)

5) Open PSR

Used to cover the case where two or more NEs of a PSR (but not the entire ring) are managed by the same EMS. (e.g. Open UPSR)

6) Open SPRing

Used to cover cases where two or more NEs of an SPRing (but not the entire ring) are managed by one EMS. (e.g. Open BLSR)

7) Mesh

Used to cover an arbitrary set of two or more NEs not covered by any other type.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.13 Connection

{Requirement I. 129}

{R I. Connection 1}

The term (top-level) Connection is used in these requirements as a general identification of connectivity through a (top-level) MultiLayer Routing Area (MLRA)[a].

In a network **with** Control Plane functionality the (top-level) Connection is defined as described in {Requirement I. 130} and {Requirement I. 131}.

In a network **without** Control Plane functionality the top-level Connection is defined as described in {Requirement I. 131}.

Note:

In **ALL** cases the (top-level) Connections are "implemented" in the Transport Plane as Subnetwork Connection (SNC)s as described in {Requirement I. 014} and {Requirement I. 015}.

Source: Version 3.5.

Mandatory/Optional: Optional

a. The term MultiLayer Routing Area (MLRA) is also used as a general term to describe an area through which Connections can be routed over more than one MultiLayer Subnetwork (MLSN)s. It is not relevant whether the routing is done within or outside the EMS domain.

{Requirement I. 130}

{R I. Connection 4}

The Connection object shall represent the relationship between Subnetwork Point (SNP)s or Termination Point (TP)s.

A Connection may encompass a sequence of Connections (which are referred to as "Connection Segments", but not represented in this model) and thus create a recursive relationship between Connections (i.e., Connections supporting Connections).

The Connection(s) at the highest level of recursion are referred to as "top-level" Connection(s).

A Call may have one or more parallel top-level Connection(s).

Source: Version 3.5.

Mandatory/Optional: Mandatory

{Requirement I. 131}

{R I. Connection 2}

The Connection object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes.

A Connection name is unique with the context of a MultiLayer Routing Area (MLRA) and not the EMS as stated in {Requirement I. 060}.

1) Connection ID

   This attribute shall represent the unique identifier of the Connection. The identifier is assigned by the Control Plane and is used for Control Plane signalling. This attribute is equivalent to the Connection Name attribute in ITU-T Recommendation G.7713.

2) Layer Rate

   This attribute shall represent the layer rate of the Connection. Refer to {Requirement I. 022}.

3) A-end ID

   This attribute shall represent the A-end of the Connection in the Control Plane name space, i.e., the value of this attribute shall identify an Subnetwork Point (SNP) or an Subnetwork Point Pool (SNPP).

4) Z-end ID

   This attribute shall represent the Z-end of the Connection in the Control Plane name space, i.e., the value of this attribute shall identify an Subnetwork Point (SNP) or an Subnetwork Point Pool (SNPP).

5) A-end TP

   This attribute shall represent the A-end of the Connection in the Transport Plane name space, i.e., the value of this attribute shall identify a Termination Point (TP).

6) Z-end TP

   This attribute shall represent the Z-end of the Connection in the Transport Plane name space, i.e., the value of this attribute shall identify a Termination Point (TP).

7) A-end TNA Name

   This attribute shall represent the A-end TNA or Group-TNA of the Connection in the Control Plane name space.

   If there is a value in the A-end TNA Name then the A-end TP attribute must be empty.

8) Z-end TNA Name

This attribute shall represent the Z-end TNA or Group-TNA of the Connection in the Control Plane name space.

If there is a value in the Z-end TNA Name then the Z-end TP attribute must be empty.

9) Directionality

This attribute shall represent the directionality of the Connection (bidirectional or unidirectional).

10) Reroute Allowed

This attribute shall indicate whether reroute is allowed on the Connection.

This attribute specifies if the EMS/Control Plane is allowed and/or required to reroute the Connection if there is a failure on the Connection, periodically to optimize the routes, or for any other reason. It is an EMS/ Control Plane implementation decision as to whether this is achieved using network routing protocols or if the EMS/Control Plane detects the failure and takes appropriate action to attempt to fix the Connection.

A value of “NO” means that the EMS/Control Plane is not allowed to reroute the Connection.

A value of “YES” means that the EMS/Control Plane is allowed to reroute the Connection and is required to attempt to reroute it upon failure.

A value of “NA” is used when the NMS does not want to specify the exact EMS/Control Plane behaviour. In this case it is left up to the EMS/Control Plane to decide whether rerouting will be provided. It is also used if the EMS/Control Plane is allowed to reroute the Connection but not required to attempt to reroute it upon failure.

11) Static Protection Level

This attribute shall indicate the level of static protection of the Connection. The level of static protection shall be one of “unprotected”, “preemptible”, “partially protected”, “highly protected”, or “fully protected”.

12) Maximum Cost

A number that indicates the maximum cost allowed for a Connection. The cost of a Connection is calculated based on the cost of the supporting MultiLayer Subnetwork Point Pool Link (MLSNPPLink)s. This link cost represents a vector of one or more metrics, each of which indicates the relative desirability of a particular link over another during path selection.

13) Protection Effort

This attribute shall indicate the protection effort of the Connection, and shall be one of “whatever”, “same-or-better”, “same-or-worse”, or “same”.

14) Routing Constraint Effort

This attribute shall indicate the requested level of effort for meeting the routing constraints requested for the creation of the Connection, and may be either “exact-match” or “best-effort”.

15) Connection Type

This attribute shall perform the same function as the SNC Type attribute and shall have the same set of values (refer to {Requirement I. 040}) with the following qualifications:

- Simple - for the top-level Connections this is the only type supported.
- Add Drop A - for non top-level (i.e. subordinate) Connections this is the only type supported.
- Add Drop Z - for non top-level (i.e. subordinate) Connections this is the only type supported.

16) Connection Setup Type

This attribute shall identify the setup type of the connection. The type shall be one of the following:

- Permanent Connection (PC)
- Switched Connection (SC) or
- Soft Permanent Connection (SPC).

17) Connection State

Refer to {Requirement I. 132}.

18) Revertive

This attribute shall indicate whether the Connection shall always attempt to return to it's home (intended) route.

19) Network Access Domain

This attribute shall represent the Network Access Domain to which this Connection has been assigned.

20) Priority

This attribute shall represent the priority of the Connection (i.e., highest (0) to lowest).

21) Alarm Reporting Indication

This attribute shall indicate whether alarm reporting for the Connection is administratively activated or de-activated.

22) Alarm Severity Assignment Profile Pointer

This attribute shall represent the name of the Alarm Severity Assignment Profile (ASAP) that has been assigned to the Connection.

23) Route Group Label

This attribute shall represent the Route Group of the Connection.

24) Using Home Route

This attribute describes whether the Connection is currently using the home route.

25) Call ID

This attribute shall contain the name of the associated Call.

26) Supporting SNCs

This attribute shall contain the names of the supporting Subnetwork connnections if any.

Note:

This attribute is only applicable for top-level connections in a non-Control PLane environment if SNCs support indirectly the Call

Source: Version 3.5.

Mandatory/Optional: Mandatory

4.1.13.1 Connection State

{Requirement I. 132}

{R I. Connection 3}

The Connection State attribute shall represent the state of the Connection object. This attribute shall have one of the following values:

1) Searching

   This state indicates that NOT all the resources necessary to support the Connection have been successfully allocated yet.

2) Complete

   This state indicates that all resources necessary to support the Connection have been successfully allocated.

3) Not Applicable

   This state indicates that the Connection is used in a non Control Plane environment.

Note:

State 1) may not be applicable in a non Control Plane environment.

Source: Version 3.5.

Mandatory/Optional: Optional

4.1.14 Call

{Requirement I. 133}

{R I. Call 1}

A Call exists in the context of a MultiLayer Subnetwork (MLSN) if the routing domain is equivalent to the MLSN domain.

A Call exists in the context of a MultiLayer Routing Area (MLRA) if the routing domain is spread over more than one MLSN domain.

In both cases the Call is named by the EMS. It represents the association between endpoints in support of an instance of a service or an infrastructure trail.

A Call is a grouping object that is used to facilitate and manage (establish, modify, delete) a set of Subnetwork Connection (SNC)s that provide end-to-end transport of user information. Because the Call manages the set of Connections and/or Subnetwork Connection (SNC)s as a group, it is possible to take diversity constraints between the individual Connections and/or Subnetwork Connection (SNC)s into account when they are created/modified.

Calls which exist in the context of an MLSN manage the Subnetwork Connection (SNC)s directly.

Calls which exist in the context of an MLRA:

- Manage the Subnetwork Connection (SNC)s indirectly via top-level Connections (refer to Figure 4.1)

  OR

- Manage Connections indirectly via top-level Connections

There is a relationship between Calls and top-level Connections. Zero or more top-level Connections may be associated to a Call. A top-level Connection may not be part of more than one Call.

The bandwidth of a Call is provided by one or more Connections.

Note: Creating a Call between Termination Point (TP)s in the same Managed Element (ME) is possible.

Source: Version 3.5.

Mandatory/Optional: Optional

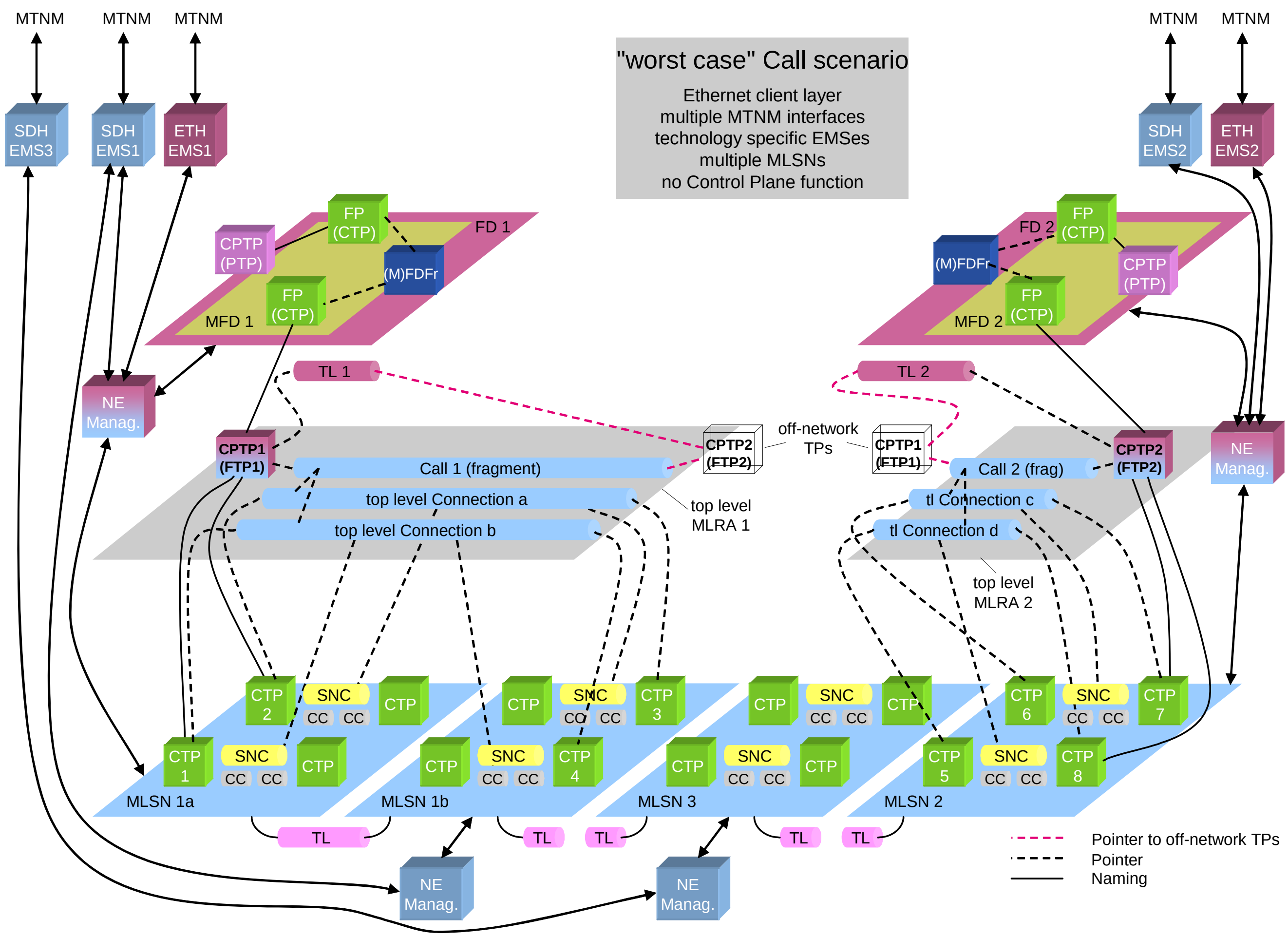


Figure 4.1: Call/Connection modeling

{Requirement I. 134}

{R I. Call 2}

The Call object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Call ID

This attribute shall represent the unique identifier of the Call. The identifier is assigned by the Control Plane and is only used for Control Plane Signalling.

Note:

This attribute corresponds to the Call Name attribute in ITU-T Recommendation G.7713.

2) Network Access Domain

This attribute represents the Network Access Domain to which this Call has been assigned to.

3) A-end

This attribute shall represent the source end point of the Call. The value of this attribute has the following fields:

- Subnetwork Point Pool (SNPP)
- Subnetwork Point (SNP)
- Termination Point (TP), which identifies either a Connectionless Port TP (CPTP), a Physical Termination Point (PTP) or a Connection Termination Point (CTP)
- TNA Name, which identifies either a TNA or a Group-TNA Name.

All the fields may be empty.

Note:

All TPs that are managed by different EMSs are “off-network” TPs which are addressed by remote addresses (refer to Figure 4.1).

In a Control Plane environment the A-end TP and/or Z-end TP attributes may also contain a “null-name”.

4) Z-end

This attribute shall represent the destination end point of the Call. The value of this attribute has the following fields:

- Subnetwork Point Pool (SNPP)
- Subnetwork Point (SNP)
- Termination Point (TP), which identifies either a Connectionless Port TP (CPTP), a Physical Termination Point (PTP) or a Connection Termination Point (CTP)

- TNA Name, which identifies either a TNA or a Group-TNA Name.

All the fields may be empty.

5) Call parameter

This attribute represents the following parameters:

- Severely Degraded Threshold
- Degraded Threshold
- Class Of Service
- Class Of Service specific Parameters.

6) Call state

Refer to {Requirement I. 135}

7) Call diversity

This attribute represents the diversity status of the Call which is identified by the following values:

- Co-routing Level of Effort

  This value shall represent whether the top-level Connections / Subnetwork Connection (SNC)s in a Route Group are co-routed or not. The value of this attribute shall be one of the following:

  - None (i.e., no co-routing requirement in any of the Route Groups)
  - Best Effort
  - Mandatory.

- Link Diversity Level of Effort

  This value shall represent whether the Route Groups in a Call are link diversely routed or not. The value of this attribute shall be one of the following:

  - None (i.e., no link diversity requirement for the Route Groups)
  - Best Effort
  - Mandatory.

- Node Diversity Level of Effort

  This value shall represent whether the Route Groups in a Call are node diversely routed or not. The value of this attribute shall be one of the following:

- None (i.e., no node diversity requirement for the Route Groups)
- Best Effort
- Mandatory

- Link SRG Type

  This value shall represent the Shared Risk Group Type to be used for link diversity.

Note:

If the “Link Diversity Level of Effort” attribute specifies link diversity (i.e., Best Effort or Mandatory), but no value is specified for the Link SRG Type attribute, then basic link diversity is followed, i.e., the Call is routed by using separate Links, and no other link-related risk types, (e.g., duct, bridge, etc.) will be used.

- Node SRG Type

  This attribute shall represent the Shared Risk Group Type to be used for node diversity.

Note:

If the “Node Diversity Level of Effort” attribute specifies node diversity (i.e., Best effort or Mandatory), but no value is specified for the Node SRG Type attribute, then basic node diversity is followed, i.e., the call is routed by using separate nodes, and no other node-related risk types (e.g., building, floor, etc.) will be used.

8) Diversity violations

This attribute represents information on the current state of diversity violations of the Call. It contains the following information:

- Diversity synthesis

  This value shall represent a synthesis of the current state of diversity violations of the Call. The possible values are

  - NoViolations
  - Violations
  - LinkViolations
  - NodeViolations
  - LinkAndNodeViolations
  - Unknown

- Link diversity violations

  This value shall represent the link diversity violations with the following information:

  - SRG type where the violation occured.
  - name of the link involved in the violation
  - list of connections sharing the link.

- Node diversity violations

  This value shall represent the node diversity violations with the following information:

  - SRG type where the violation occured
  - name of the node involved in the violation
  - list connections sharing the node.

- Link partial diversity list

  This value shall represent the actual link diversity achieved where the diversity requirements were not satisfied with the following indormation:

  - SRG types where the violation ioccured
  - list of links involved, in the violation
  - list of connections.

  It indicates the SRG types where the provisioned diversity has been partially satisfied. E.g. if "bridge" diversity was asked, and over a given bridge only the "duct" diversity is achieved, then the duct names are listed. The operator is then enabled to verify e.g. whether such partial achievement is anyway satisfactory or not.

- Node partial diversity list

  This value shall represent the actual node diversity achieved where the diversity requirements were not satisfied with the following information:

  - SRG types where the violation occured
  - list of nodes involved in the violation,
  - list of connections.

  It indicates the SRG names where the provisioned diversity has been partially satisfied.

Source: Version 3.5.

Mandatory/Optional: Optional

4.1.14.1 Call State

{Requirement I. 135}

{R I. Call 3}

The Call state attribute of the Call object shall represent one of the following values:

1) In progress

   The supporting Connections are currently being created.

2) Established - In Service

   All the supporting Connections have been created successfully.

3) Established - In Service - Searching

   A Call has been modified through the addition of Connections and not all new Connections have been successfully created.

4) Established - Out Of Service

   All Connections have failed and they are not being restored.

5) Established - Out Of Service - Searching

   All Connections have failed and they are currently being restored.

6) Established - In Service - Degraded

   The number of failed connections has reached or exceeded the degraded threshold AND the severely degraded threshold has not been reached or exceeded.

7) Established - In Service - Severely degraded

   The number of failed supporting Connections has reached or exceeded the severely degraded threshold (in case of LCAS support).

8) Established - In Service - Degraded - Searching

   At least one supporting Connection has failed (in the case of LCAS support) AND the severely degraded threshold has not been reached or exceeded. The failed Connections are being restored.

9) Established - In Service - Severely degraded - Searching

   The number of failed supporting Connections has reached or exceeded the severely degraded threshold (in case of LCAS support). The failed Connections are being restored.

Note:

States 1), 3), 5), 8), and 9) may not be applicable in a non Control Plane environment.

Source: Version 3.5.

Mandatory/Optional: Optional

4.1.15 Subnetwork Connection (SNC)

{Requirement I. 014}

A Subnetwork Connection (SNC) object shall represent the relationship between two of the following types on end points:

- Physical Termination Point (PTP)
- Connection Termination Point (CTP) or
- Group Termination Point (GTP) (refer to {Requirement I. 076}) or
- Floating Termination Point (FTP)

An SNC represents a transparent end-to-end connection or a trail (closed or half-open) through or within a MultiLayer Subnetwork (MLSN) according to the roles associated to its end points.

If it represents a connection, its end points are CTPs or FTPs with the SNC's layer rate as connectable layer rate. In the case of GTPs (i.e. a Bundled connection) the SNC does not have an explicit layer rate

If it represents a trail, its end points are CTPs or FTPs or PTPs.

An SNC shall be contained in a Subnetwork.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

{Requirement I. 015} An SNC object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) State

Refer to {Requirement I. 017}.

2) Directionality

This attribute shall represent the directionality of the SNC (bidirectional or unidirectional).

3) Layer rate

Refer to {Requirement I. 022}.

4) Static protection level

This attribute shall represent the degree of internal resilience/protection of the SNC e.g., to indicate whether the Subnetwork Connection should be Protected, Preemtible, or Unprotected. The EMS will be required to create a SNC with the specified Protection Level.

5) Type

This attribute shall indicate the specific traffic flow through the SNC. (Refer to {Requirement I. 040})

6) aEnd TPs

This attribute shall represent a list of the following aEnd Termination Point (TP)s:

- Physical Termination Point (PTP)
- Connection Termination Point (CTP)
- Group Termination Point (GTP)
- Floating Termination Point (FTP)

7) zEnd TPs

This attribute shall represent a list of the following zEnd Termination Point (TP)s:

- Physical Termination Point (PTP)
- Connection Termination Point (CTP)
- Group Termination Point (GTP)
- Floating Termination Point (FTP)

8) Network routed

This attribute shall indicate if the route of the SNC was computed by either the network or the EMS during activation.

9) Reroute allowed

This attribute shall indicate if an SNC may be rerouted.

10) Network reroute

This attribute shall indicate if the reroute (if allowed) shall be computed by the network, by the EMS, or by either.

11) Revertive

This attribute shall indicate whether the SNC shall always attempt to return to it's intended Route.

12) Network Access Domain

This attribute represents the Network Access Domain to which this SNC has been assigned.

13) Alarm reporting

This attribute shall indicate whether alarm reporting for the SNC is enabled or disabled.

14) Correlation identifier

This attribute shall contain information about relationships that this subnetwork connection may have to other objects.

15) Bundled SNC

This parameter shall indicate if the SNC to be created is a bundled SNC

16) GTP deletion

The attribute shall only be used when creating bundled SNCs. It shall indicate that the EMS has to delete all the interior GTPs supporting the bundled SNC when the SNC is deleted.

17) Fixed

This attribute shall indicate whether the SNC is fixed (i.e. cannot be deleted by the NMS) or flexible. A fixed SNC is defined a SNC whose all cross-connects are fixed.

18) Alarm severity assignment profile

This attribute shall represent the name of the Alarm Severity Assignment Profile (ASAP) that has been assigned to the SNC.

19) Retain SNC

This attribute shall indicate if when modifying an SNC whether the original SNC shall be deleted or put into the pending state.

20) Priority

This attribute shall represent the priority of the SNC (i.e highest (0) to lowest).

21) aEnd point role

This attribute shall represent the role of the aEnd Termination Point (TP)s of the SNC. Refer to {Requirement I. 092}.

22) zEnd point role

This attribute shall represent the role of the zEnd Termination Point (TP)s of the SNC. Refer to {Requirement I. 092}.

23) Route Group Label

This attribute shall represent the Route Group to which the SNC belongs.

24) CallID

This attribute shall represent the name of the Call of which the SNC is a component, if it's a component of an Call.

Source: Version 3.5.

Mandatory/Optional: Mandatory.

4.1.15.1 Subnetwork Connection (SNC) End Point Role

{Requirement I. 092} The EMS shall ensure that the Termination Point (TP) role assigned to the end point TPs of a Subnetwork Connection (SNC) shall indicate whether the TP is either:

- an SNC-like end point - referred to as a *connection matrix* end point
- a trail-like end point - referred to as a *link connection* end point

At a connection matrix end point the span of the SNC starts at a G.805 Connection Point (CP) with a fixed or flexible connection through the Network Element (NE) at the SNC layer, i.e. the connectable layer of the end point.

At a link connection (LC) end point connectivity is adapted from a server layer and the span of the SNC starts with a G.805 Termination Connection Point (TCP).

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.1.15.2 Subnetwork Connection (SNC) Naming

{Requirement I. 016} The EMS shall ensure that each name that is assigned to an Subnetwork Connection (SNC) shall be unique and must not be re-used.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.15.3 Subnetwork Connection (SNC) States

{Requirement I. 017} The SNC State attribute of the Subnetwork Connection indicates one of the following values:

1) Active

   The SNC is not in pending state, a route has been assigned to the SNC and all XCs for the SNC are active in the network.

2) Pending

   The SNC has been created by an NMS and has not been activated by any NMS; or the SNC has been successfully deactivated by an NMS. That state has no relationship with the network state of the XCs of the SNC.

3) Partial

   The SNC is not in pending state, and either a route has not been assigned to the SNC, or not all of the cross-connects of the SNC are active in the network. This may or may not include activated SNCs for which there are currently no active cross-connects in the network, depending on the SNC management mode of operation.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.15.4 Subnetwork Connection (SNC) Types

{Requirement I. 040} The NML-EML Interface shall support the following complex types of Subnetwork Connection (SNC):

1) Simple
2) Add Drop A
3) Add Drop Z
4) Double Add Drop
5) Interconnect
6) Double Interconnect
7) Open Add Drop
8) Explicit

Refer to supporting document SD1-36_SNCTypes.pdf for further information regarding Subnetwork Connection types.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.15.5 Subnetwork Connection (SNC) Routing Constraints

{Requirement I. 018} The NMS may provide routing constraint information to the EMS as part of the creation and activation of an SNC. The routing constraint information in a specific SNC create shall include:

EITHER:

1) Resources that must not be part of the route chosen as a result of the request (i.e. excluded resources). The resources may be:

- Managed Element (ME)s
- Termination Point (TP)s
- Subnetwork Connection (SNC)s
- Topological Link (TL)s
- Group Termination Point (GTP)s

OR:

2) Resources that must form part of the route chosen as a result of the request (i.e. included resources). The resources may be:

- Cross-Connect (XC)s
- Termination Point (TP)s
- Managed Element (ME)s
- Topological Link (TL)s
- Group Termination Point (GTP)s

Additional information related to the capabilities of the specific EMS may be provided.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.1.15.6 Subnetwork Connection (SNC) Configurations

{Requirement I. 039} The NML-EML Interface shall support the following Subnetwork Connection (SNC) configurations:

1) Unidirectional, Point-to-Point
2) Unidirectional, Point-to-Multipoint
3) Bidirectional, Point-to-Point

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.16 Bundled Subnetwork Connection (B-SNC)

{Requirement I. 076}

A bundled Subnetwork Connection (B-SNC) is a type of Subnetwork Connection (SNC) where the end points are Group Termination Point (GTP)s. The Route of a B-SNC is also comprised of a collection of cross-connected GTPs.

Figure 4.2 depicts a B-SNC (see the dotted line).

The following conditions apply to a B-SNC:

1) The Connection Termination Point (CTP) ordering is preserved between the aEnd and zEnd of the B-SNC, i.e., the ith CTP in the aEnd GTP in Managed Element #1 is mapped to the ith CTP in the zEnd GTP in Managed Element #3, refer to Figure 4.2.
2) The CTPs within the GTPs along the route of the B-SNC must match. So, each GTP along the route should contain a sequence CTPs of a given set of layer rates and in a particular order. For example, if one endpoint of a B-SNC is a GTP whose first 3 CTPs are of layer rate STS-1, the next three CTPs are of layer rate VT1.5 and the last two CTPs are of layer rate STS3c, then all other GTPs supporting the B-SNC must have the same number of CTPs of each layer rate and in the same order as the given GTP.
3) The CTPs comprising a GTP that supports a bundled SNC service need not be contiguous,
4) For a B-SNC each supporting GTP shall be contained in a Physical Termination Point (PTP) or a Floating Termination Point (FTP).
5) For a B-SNC service spanning a non-singleton subnetwork, the aEnd and zEnd attributes are GTPs such that the CTPs comprising each GTP are contained in the same PTP or FTP

Source: Version 3.0.

Mandatory/Optional: Optional.

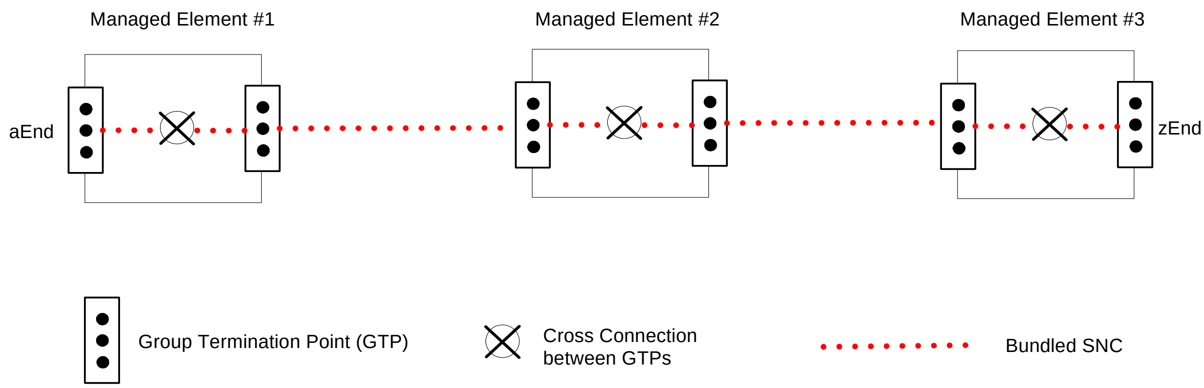


Figure 4.2: Bundle Subnetwork Connection

4.1.17 Cross-Connect (XC)

{Requirement I. 019}

A Cross-Connect (XC) object shall represent a physical connection within a Network Element (NE).

An XC is atomic and is identified, similarly to an Subnetwork Connection (SNC) in a singleton MultiLayer Subnetwork (MLSN), based on its external shape.

An XC is primarily used in the specification of an SNC Route.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

{Requirement I. 020} A XC object shall have the following attributes:

1) Type

This attribute shall indicate the specific traffic flow through the XC. (Refer to Section 4.1.15.4).

2) Directionality

This attribute shall represent the directionality of the XC (bidirectional or unidirectional).

3) Active

This attribute shall indicate if the XC is active in the Network Element (NE).

4) Fixed

This attribute shall indicate if the XC is fixed (i.e. cannot be deleted by the NMS) or is flexible.

5) Additional information

Refer to {Requirement I. 060}.

6) aEnd TP(s)

This attribute shall represent a list of the following aEnd Termination Point (TP)s:

- Connection Termination Point (CTP)
- Group Termination Point (GTP)
- Floating Termination Point (FTP)

7) zEnd TP(s)

This attribute shall represent a list of the following zEnd Termination Point (TP)s:

- Connection Termination Point (CTP)
- Group Termination Point (GTP)
- Floating Termination Point (FTP)

8) Connection ID

This attribute shall represent the identifier of the associated Connection.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.18 Route

{Requirement I. 021}

The Route object shall represent the route of a Subnetwork Connection (SNC). An SNC route shall be represented as a partially ordered series of Cross-Connect (XC)s through which the SNC traverses. Only XCs on the SNC's layer rate are part of the route.

The XCs that are part of the Route shall be listed from the NE on which the SNC starts (first entry) to the NE on which the SNC ends (last entry).

A SNC may be associated with more than one Route

- always 1 intended Route, i.e. the preferred, or default Route
- 0..n backup/alternative Route(s)

A route belongs to only one SNC. However XCs/Connection Termination Point (CTP)s can be shared by Routes of different SNCs.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement I. 089} A Route object shall have the following attributes:

1) Identifier

This attribute shall represent a unique identifier for the route within the context of the Subnetwork Connection (SNC) name.

2) Contained XCs

This attribute shall represent the partially ordered list of Cross-Connect (XC)s that constitute the SNC route.

3) Intended

This attribute shall indicate whether the Route is the intended Route (also referred to as the preferred, or default route) or the backup Route. The intended route could be simply the first time provisioned route, or the preferred route for a number of factors, from network engineering to intrinsic media reliability. The backup route shall be partly or totally different from intended route (but with same end points), and its main use is for restoration and maintenance purposes.

4) Actual state

This attribute shall represent a summary state of the actual states of the XCs in the network, regardless of the SNC that the XCs are currently serving. It can assume only the following values:

- Inactive - none of the contained XCs are active in the network
- Active - all of the contained XCs are active in the network. So it is the route where SNC traffic is currently carried. There can be at most one active route per SNC. The in use by attribute shall indicate that the Route is not being used by another SNC.
- Partial - one or more, but not all the XCs are active in the network.

5) Administrative state

This attribute shall represent whether the Route is allowed to be active or not.

6) In use by

This attribute shall indicate whether the Route is being used by another SNC. At Route is considered to be in use by another SNC if at least one of its XCs or CTPs is carrying traffic of another SNC.

7) Exclusive

This attribute shall indicate that the Route can only be associated with a single SNC. This means that the XCs and CTPs that are contained by this Route can only be used by the SNC to which the Route is associated.

8) Additional information

Refer to {Requirement I. 060}.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.1.19 Layer Rate

{Requirement I. 022} The Layer rate is used to identify:

1) The supported rate(s) of a Managed Element (ME),
2) The layer rate of a Connection Termination Point (CTP),
3) The layer rate of a Physical Termination Point (PTP),
4) The layer rate of a Subnetwork Connection (SNC),
5) The supported rate(s) of a MultiLayer Subnetwork (MLSN),
6) The layer rate of a Topological Link (TL).
7) The layer rate of a Protection Group (PG).
8) The layer rate of a Threshold Crossing Alert (TCA) Parameter Profile.
9) The layer rate of a Performance Monitoring Point (PMP).
10) The Layer rate of a Connection.

Refer to supporting document SD1-17_LayerRates.pdf for the currently defined layer rates.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.1.20 Connection State

{Requirement I. 023} The Connection state attribute of a Connection Termination Point (CTP) or Group Termination Point (GTP) object shall be have the following values:

1) Sink connected
2) Source connected
3) Bi-directionally connected
4) Not connected

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.21 Transmission Parameter

{Requirement I. 024} A Transmission Parameter is a characteristic of a Termination Point (TP) that shall be identified by a name value pair.

Refer to supporting document SD1-16_LayeredParameters.pdf for the currently identified TP parameters.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.1.22 Traffic Descriptor (TD)

{Requirement I. 025} A Traffic Descriptor (TD) object represents a collection of attributes, which are used to define bandwidth and Quality of Service (QoS) characteristics on a Connection Termination Point (CTP).

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement I. 063} A TD object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Service category

   This attribute shall represent the ATM Service Category which relates quality requirements and traffic characteristics to network behavior (procedures and parameters). It is intended to specify a combination of Quality of Service (QoS) commitment and traffic parameters that is suitable for a given set of applications (user interpretation) and that allows for specific multiplexing schemes at the ATM layer (network interpretation. Refer to {Requirement I. 027} for a list of supported service categories.

2) Conformance profile

   This attribute shall represent the conformance profile which characterizes an ATM connection. The conformance definitions are taken from the ATM Forum UNI 4.1, UNI 4.0, and UNI 3.1 standards. Refer to {Requirement I. 028} for a list of the supported profiles.

3) Traffic parameters

   This attribute shall represent a set of traffic and QoS parameters. Refer to {Requirement I. 029} for a list of the supported parameters.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

{Requirement I. 026} The NML-EML Interface shall support the Traffic Descriptor (TD) combinations as specified in supporting document SD1-5_ATMConformanceDefinitions.pdf.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.1.22.1 Traffic Descriptor (TD) Service Category

{Requirement I. 027} The NML-EML Interface shall support the following Traffic Descriptor (TD) Service Categories:

1) CBR (Constant Bit Rate)
2) rt-VBR (Real-Time Variable Bit Rate)
3) nrt-VBR (Non-Real-Time Variable Bit Rate)
4) UBR (Unspecified Bit Rate)
5) ABR (Available Bit Rate)
6) GFR (Guaranteed Frame Rate)

Source: Version 2.0.

Mandatory/Optional: Optional.

4.1.22.2 Traffic Descriptor (TD) Conformance Profile

{Requirement I. 028} The NML-EML Interface shall support the following Traffic Descriptor (TD) Conformance Profiles listed below.

The TD Conformance Profile combined with the Traffic Descriptor (TD) Service Category uniquely identifies the columns in the tables in the supporting document SD1-5_ATMConformanceDefinitions.pdf.

1) CBR Legacy 1 (CBR.L1)
2) CBR Legacy 2 (CBR.L2)
3) VBR Legacy 1 (VBR.L1)
4) VBR Legacy 2 (VBR.L2)
5) VBR Legacy 3 (VBR.L3)
6) VBR Legacy 4 (VBR.L4)
7) CBR.1
8) VBR.1
9) VBR.2
10) VBR.3
11) UBR.1
12) UBR.2
13) GFR.1
14) GFR.2
15) USER

Other combination of standard and vendor-specific TD parameters may also be supported.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.1.22.3 Traffic Descriptor (TD) Parameters

{Requirement I. 029} The NML-EML Interface shall support the Traffic Descriptor (TD) parameters as identified in supporting document SD1-16_LayeredParameters.pdf.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.1.23 Transmission Descriptor (TMD)

{Requirement I. 078} A Transmission Descriptor (TMD) object represents a collection of attributes, which are used to define multi-layered transmission parameters, and additional information parameters on a Termination Point (TP) or on a Matrix Flow Domain (MFD). Refer to the supporting document SD1-16_LayeredParameters.pdf for details of the currently defined transmission parameters.

Note that the transmission parameters contained in chapter "Traffic conditioning parameters" must not be used in a TMD.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement I. 079} A TMD object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Layered transmission parameters

   This attribute shall represent a list of transmission parameters which can be set at a specified layer on a Termination Point (TP) or on a Matrix Flow Domain (MFD) having this TMD assigned as an egress or ingress TMD.

   Specific parameters include, for example, frame format, line code, alarm reporting control (enable/disable), TP service state (In Service, Out Of Service, connectionless parameters, traffic conditioning parameters, etc.). For each layer, a layer-specific Alarm Severity Assignment Profile (ASAP) can be embedded into the respective single-layer list of transmission parameters.

2) Additional Object information

   This attribute shall represent additional information parameters which can be set on a Termination Point (TP) or on a Matrix Flow Domain (MFD) by association with this TMD.

3) External representation

   This attribute shall represent a reference to the external representation of the TMD (e.g., an XML file name). The content of this information is opaque at the NML-EML Interface and not utilized.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.1.24 NML-EML Interface Version

{Requirement I. 030} A unique identifier, referred to as the Interface Version Number, shall be assigned to a specific version of the NML-EML Interface.

The EMS shall maintain the Interface Version Number associated with the specific version of the NML-EML Interface that the EMS is using.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.24.1 Interface Versioning

{Requirement I. 031} The NML-EML Interface shall fulfill the following requirements.

1) Support of multiple versions of an interface.
2) Support of Naming Context conventions.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.25 Equipment

{Requirement I. 064} An Equipment object shall represent the manageable physical components of a NE such as the circuit packs, the fans and any other type of replaceable unit within the NE.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

{Requirement I. 032} An Equipment object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Service state

This attribute shall indicate the current administrative state of the equipment. The administrative states that shall be supported are In Service, Out of Service and Out of Service for Maintenance.

2) Alarm reporting

This attribute shall indicate whether alarm reporting for this equipment is enabled or disabled.

3) Expected equipment type

This attribute shall represent the type of the expected equipment. This attribute may have no value if there is no expected equipment.

4) Installed equipment type

This attribute shall represent the type of installed equipment. This is attribute may have no value if there is no installed equipment.

5) Installed part number

This attribute shall represent the vendor's part number of the installed equipment.

6) Installed serial number

This attribute shall represent the vendor's serial number of the installed equipment. The combination of the installed part number and the installed serial number for a specific piece of vendor equipment shall uniquely identify that equipment.

7) Installed version

This attribute shall represent the vendor's version of the installed equipment.

8) Manufacturer

This attribute shall represent the name of the equipment vendor.

9) Protection role

This attribute shall represent the protection role (e.g. primary or secondary) that the equipment plays in case it takes part in an equipment protection scheme.

10) Protection scheme state

This attribute shall indicate the current state of the protection scheme (i.e. whether it is active or locked).

11) Alarm severity assignment profile

This attribute shall represent the name of the Alarm Severity Assignment Profile (ASAP) that has been assigned to the Equipment.

12) Manufacturer date

This attribute shall represent the date of manufacture of the ME.

Source: Version 3.5.

Mandatory/Optional: Mandatory.

4.1.26 Equipment Holder

{Requirement I. 033} An Equipment Holder object shall represent resources of the Network Element (NE) that are capable of holding other physical components.

Specific resources that are represented by an Equipment Holder object shall be for instance racks (bays), shelves, and slots or sub-slots.

Source: Version 2.0.

Mandatory/Optional: Mandatory

{Requirement I. 065} An Equipment Holder shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Alarm reporting

This attribute shall indicate whether alarm reporting for this equipment holder is enabled or disabled.

2) Type

This attribute shall indicate the type of the physical container represented by the Equipment Holder e.g. a rack, a shelf, a sub-shelf, a slot or sub-slot.

3) Expected or installed equipment

This attribute shall represent the equipment that is installed or is expected to be installed in the physical container represented by the Equipment Holder.

4) Acceptable equipment types

This attribute shall represent a list of the types of Equipment that can be contained by the Equipment Holder. Note: this shall apply when the Equipment Holder represents a slot.

5) State

This attribute shall represent the current condition of the Equipment Holder with respect to the contained equipment.

6) Alarm severity assignment profile

This attribute shall represent the name of the Alarm Severity Assignment Profile (ASAP) that has been assigned to the Equipment Holder.

7) Location

This attribute shall represent the geographical location of the Equipment Holder

8) Manufacturer

This attribute shall represent the name of the equipment vendor.

9) Manufacturer date

This attribute shall represent the date of manufacture of the ME.

Source: Version 2.0.

Mandatory/Optional: Mandatory

4.1.27 Equipment Protection Group (EPG)

{Requirement I. 072} The Equipment Protection Group (EPG) object shall represent Equipment protection.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement I. 073} An EPG object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Type

   This attribute shall represent the type of the EPG (e.g. M:N).

2) Protection scheme state

   This attribute shall indicate the current state of the protection scheme (i.e. whether it is active or locked).

3) Reversion mode

   This attribute shall indicate whether the protection scheme is revertive or not.

4) Protected Equipment

   These attribute shall represent a list of the protected Equipment instances.

5) Protecting Equipment

   This attribute shall represent a list of the protecting Equipment instances.

6) PG parameter list

   This attribute shall represent the EPG specific parameters. For example SwitchMode, SwitchPosition, wait to restore time.

7) Alarm severity assignment profile

   This attribute shall represent the name of the Alarm Severity Assignment Profile (ASAP) that has been assigned to the EPG.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.1.28 Protection Group (PG)

This section is not applicable to ATM technology.

{Requirement I. 034} The Protection Group (PG) object shall represent trail protection schemes.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement I. 066} A PG object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Type

   This attribute shall represent the type of the PG.

2) Protection scheme state

   This attribute shall indicate the current state of the protection scheme (i.e. whether it is active or locked).

3) Reversion mode

   This attribute shall indicate whether the protection scheme is revertive or not.

4) Layer Rate

   Refer to {Requirement I. 022}.

5) Protection related PTPs

   This attribute shall represent a list of the Physical Termination Point (PTP)(s) related by the PG.

6) PG parameters

   This attribute shall represent the Protection Group specific parameters (e.g. switch mode, switch position, wait to restore time etc.).

7) APS protocol type

   This attribute shall indicate the type of APS protocol supported by the PG.

8) Alarm severity assignment profile

   This attribute shall represent the name of the Alarm Severity Assignment Profile (ASAP) that has been assigned to the PG

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.29 Threshold Crossing Alert (TCA) Parameter

{Requirement I. 088}

A Threshold Crossing Alert (TCA) Parameter object shall represent the the TCA parameters contained with a Threshold Crossing Alert (TCA) Parameter Profile.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement I. 035}

A Threshold Crossing Alert (TCA) Parameter object shall have the following attributes:

1) Name

   This attribute shall represent the name of the TCA parameter. Refer to supporting document SD1-28_PerformanceParameters.pdf for the currently defined TCA parameter names.

2) Granularity

   This attribute shall represent the time granularity of the TCA parameter, either 15 minutes or 24 hours. This attribute is not applicable for instantaneous measurements (i.e. gauge type measurements).

3) Location

   This attribute shall represent the location of the TCA parameter relative to the signal flow.

4) Threshold type

   This attribute shall represent the type of the TCA parameter, (shall indicate the level at which the threshold is triggered or cleared).Refer to supporting document SD1-37_TCAs.pdf for more details of the threshold type

5) Trigger

   This attribute shall indicate whether the threshold type shall trigger a raise or a clear TCA.

6) Value

   This attribute shall represent the value for the TCA parameter.

7) Measurement units

   This attribute shall represent the unit of measurement for the TCA parameter.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.30 Threshold Crossing Alert (TCA) Parameter Profile

{Requirement I. 067} A Threshold Crossing Alert (TCA) Parameter Profile object shall represent for a specific layer rate a set of Threshold Crossing Alert (TCA) Parameters associated with a set of Termination Point (TP)s.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

{Requirement I. 087} A TCA Parameter Profile object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Layer rate

   This attribute shall represent the layer to which the PM threshold values apply. (Refer to {Requirement I. 022}).

2) Associated TPs

   This attribute shall represent a list of the Termination Point (TP) that are associated with the TCA Parameter Profile.

3) TCA Parameters

   This attribute shall represent a list of Threshold Crossing Alert (TCA) Parameters.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.1.31 Alarm Severity Assignment Profile (ASAP)

{Requirement I. 080}

The Alarm Severity Assignment Profile (ASAP) object shall represent a set of severities that can be assigned to specific alarm probable causes.

An ASAP is contained within a EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement I. 081}

An ASAP object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Fixed

   This attribute shall indicate whether the ASAP is modifiable by NMS or not. If not, the ASAP can be neither modified nor deleted by the NMS, but only assigned/de-assigned.

2) Alarm severity assignments

   This attribute shall represent the set of alarm severity assignments. Refer to {Requirement I. 082}.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.1.31.1 Alarm Severity Assignment (ASA)

{Requirement I. 082} The Alarm Severity Assignment (ASA) object shall represent the specific severities for the various service affecting conditions that are to be assigned to a specific alarm probable cause.

1) Probable cause

   This attribute shall represent the name of specific probable cause to which the severities are to be assigned. Refer to supporting document for

2) Probable cause qualifier

   This attribute shall represent the probable cause qualifier and shall be present if the probable cause attribute is not sufficient to uniquely identify an alarm. *OPTIONAL*

3) Native probable cause

   This attribute shall represent the native probable cause. *OPTIONAL*

4) Service affecting severity

   This attribute shall represent the value to be assigned in case the reportable alarm is service affecting.

5) Non-service affecting severity

   This attribute shall represent the severity value to be assigned in case the reportable alarm is non-service affecting.

6) Service independent severity

   This attribute shall represent the severity value to be assigned in case the reportable alarm is service independent. This severity value may also be assigned in the case where the EMS is unable to determine whether the alarm is service affecting or not.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.1.32 Performance Monitoring Point (PMP)

{Requirement I. 084} The Performance Monitoring Point (PMP) object shall represent an access point at which performance monitoring and threshold supervision are provided for a set of PM parameters.

It is contained in a Termination Point (TP).

All PMPs contained in a TP constitute the PM capabilities of the TP.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement I. 085} A PMP object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Layer rate

   This attribute shall represent the layer rate of the PMP. The layer specified must be supported by the containing TP. Refer to {Requirement I. 022}.

2) Location

   This attribute shall represent the location of the performance monitoring measurement.

3) Granularity

   This attribute shall represent the time granularity of the PMP, either 15 minutes or 24 hours. This attribute is not applicable for instantaneous measurements (i.e. gauge type measurements)

4) Supervision state

   This attribute shall represent whether threshold supervision is enabled or disabled.

5) Monitoring state

   This attribute shall represent whether performance monitoring is enabled or disabled.

6) PM parameters

   This attribute shall represent a list of the names of the PM parameter associated with the PMP. Refer to supporting document SD1-28_PerformanceParameters.pdf for the list of currently defined performance parameters.

7) PM thresholds

   This attribute shall represent a list of the names of the thresholds associated with each PM parameter. Refer to {Requirement I. 099}.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.1.32.1 PM Threshold

{Requirement I. 099} The PM Threshold object shall represent the specific severities for the various service affecting conditions that are to be assigned to a specific alarm probable cause

1) Threshold type

   This attribute shall represent the type of the PM threshold, (shall indicate the level at which the threshold is triggered or cleared).

2) Trigger

   This attribute shall indicate whether the PM threshold shall trigger a raise or a clear TCA.

3) Value

   This attribute shall represent the value for the PM threshold parameter.

4) Measurement units

   This attribute shall represent the unit of measurement for the PM threshold parameter.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.33 Log

{Requirement I. 090} A Log object shall represent a repository on the EMS used to store notifications.

The Log object shall be based on the managed object defined in the ITU-T Recommendation X.735, Information Technology - Open systems Interconnection Systems Management: Log Control Function.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement I. 091} A Log object shall have the following attributes defined in ITU-T Recommendation X.735.

1) Operational state
2) Administrative state
3) Size
4) Full action
5) Duration
6) Scheduling
7) Availability status
8) Record compaction
9) Capacity alarm thresholds
10) Discriminator construct

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.1.34 Naming Requirements

4.1.34.1 Termination Point Naming

4.1.34.1.1 Physical Termination Point (PTP) Naming

{Requirement I. 036} A Physical Termination Point (PTP) shall be named relative to the Managed Element (ME) in which it is contained.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.34.1.2 Connection Termination Point (CTP) Naming

{Requirement I. 037} A Connection Termination Point (CTP) shall be named relative to a containing Physical Termination Point (PTP) or a containing Connection Termination Point (CTP).

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.35 Event Notifications

{Requirement I. 041} The NML-EML Interface shall support a reliable mechanism for the EMS to send event notifications to the NMS.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

Table 4.2 identifies the different event types that have been defined for the NML-EML Interface.

Table 4.2: Event Notification Types

	Event Type
1	Object Creation Notification
2	Object Deletion Notification
3	Attribute Value Change Notification
4	State Change Notification

Table 4.2: Event Notification Types

	Event Type
5	Protection Switch Notification
6	Equipment Protection Switch Notification
7	Threshold Crossing Alert (TCA) Notification
8	Alarm Notification
9	File Transfer Status Notification
10	Route Change Notification
11	Heartbeat Notification
12	Performance Monitoring Point (PMP) State Change Notification
13	Software Backup Status Notification
14	Log Attribute Value Change Notification
15	Log Capacity Threshold Alarm Notification
16	Log Processing Error Alarm Notification
17	Log State Change Notification

4.1.35.1 Common Event Notification Information

{Requirement I. 068} All event notifications identified in Table 4.2 shall have the following attributes:

1) Identifier

   This attribute shall represent an identifier for the event. The value of this identifier is not guaranteed to be unique.

2) Type

   This attribute shall represent the type of the event as identified in Table 4.2.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

Table 4.3 identifies those event notifications that contain a common set of attributes.

Table 4.3: Event Types with common attributes

	Event Type
1	Object Creation Notification
2	Object Deletion Notification

Table 4.3: Event Types with common attributes

	Event Type
3	Attribute Value Change Notification
4	State Change Notification
5	Threshold Crossing Alert (TCA) Notification
6	Alarm Notification
7	Route Change Notification

{Requirement I. 093} All event notifications identified in Table 4.3 shall have the following attributes:

1) Object Name

This attribute shall represent the name of the object against which the event notification is generated.

2) Object Type

This attribute shall represent the type of the object against which the event notification is generated.

3) EMS timestamp

This attribute shall represent the time at which the event occurred at the EMS.

4) NE timestamp

This attribute shall represent the time at which the event occurred at the NE.

5) Edge Point

This attribute shall indicate whether the event is related to a Termination Point (TP) at the edge of a subnetwork. (Refer to {Requirement I. 008}).

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.35.1.1 AID

There are certain alarm conditions that the EMS may wish to report to an NMS for which there is no explicit object modeled across the NML-EML Interface (i.e. there is no specific object type defined). Under these conditions the EMS shall use the “AID” object type.

{Requirement I. 057} The NML-EML Interface shall allow the EMS to generate alarms against objects that are not explicitly modeled by the Interface by using the “AID” objectType.

The EMS shall ensure that all such entities have a unique value for the AID within the Network Element (NE).

Source: Version 2.0.

Mandatory/Optional: Optional.

4.1.35.2 Event Notification Types

4.1.35.2.1 Object Creation Notification

{Requirement I. 042}

An Object Creation Notification, is an event used across the NML-EML Interface to indicate that an object has been created.

An Object Creation Notification shall have in addition to the attributes identified in {Requirement I. 068} and {Requirement I. 093} the following attributes:

1) Object Information

This attribute shall represent all of the attribute names and their values for the created object.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.35.2.2 Object Deletion Notification

{Requirement I. 043}

An Object Deletion Notification, is an event used across the NML-EML Interface to indicate that an object has been deleted.

An Object Deletion Notification shall have the attributes identified in {Requirement I. 068} and {Requirement I. 093}.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.35.2.3 Attribute Value Change Notification

{Requirement I. 044}

An Attribute Value Change Notification, is an event used across the NML-EML Interface to indicate that one or more of the attribute values of an object have changed.

An Attribute Value Change Notification shall have in addition to the attributes identified in {Requirement I. 068} and {Requirement I. 093} the following attributes:

1) Attribute name

This attribute shall represent a list of the attribute name(s) that have changed their value.

2) Attribute value

This attribute shall represent a list of the new values of the attribute(s).

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.35.2.4 State Change Notification

{Requirement I. 045}

A State Change Notification, is an event used across the NML-EML Interface to indicate that a state transition has occurred.

An State Change Notification shall have in addition to the attributes identified in {Requirement I. 068} and {Requirement I. 093} the following attributes:

1) State attribute name

   This attribute shall represent the state attribute name(s) (identifier(s)) of the object that have changed their value.

2) State attribute value

   This attribute shall represent the new value of the state attribute(s) of the object.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.35.2.5 Protection Switch Notification

{Requirement I. 046} A Protection Switch Notification, is an event used across the NML-EML Interface to indicate that a protection switch has occurred.

A Protection Switch Notification shall have in addition to the attributes identified in {Requirement I. 068} the following attributes:

1) Type

   This attribute shall represent the type of the protection for which the switch has occurred.

2) Switch reason

   This attribute shall represent the reason for the switch.

3) Layer rate

   This attribute shall represent the layer at which the switch has occurred. (Refer to {Requirement I. 022}).

4) PG

   This attribute shall represent the name of the Protection Group (PG) in the case of a trail switch. Not used if the protection type is Subnetwork Connection Protection (SNCP).

5) Protected TP

   This attribute shall represent the name of the Termination Point (TP) being protected.

6) Switch away from TP

   This attribute shall represent the name of the TP being switched away from.

7) Switch to TP

   This attribute shall represent the name of the TP that is switched to.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.35.2.6 Equipment Protection Switch Notification

{Requirement I. 074} An Equipment Protection Switch Notification, is an event used across the NML-EML Interface to indicate that a equipment protection switch has occurred.

An Equipment Protection Switch Notification shall have in addition to the attributes identified in {Requirement I. 068} the following attributes:

1) Type

   This attribute shall represent the type of the protection for which the switch has occurred.

2) Switch reason

   This attribute shall represent the reason for the switch.

3) EPG

   This attribute shall represent the name of the Equipment Protection Group (EPG).

4) Protected Equipment

   This attribute shall represent the name of the Equipment being protected.

5) Switch away from Equipment

   This attribute shall represent the name of the Equipment being switched away from.

6) Switch to Equipment

   This attribute shall represent the name of the Equipment that is switched to.

7) EMS timestamp

   This attribute shall represent the time at which the event occurred at the EMS.

8) NE timestamp

   This attribute shall represent the time at which the event occurred at the NE.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.1.35.2.7 Threshold Crossing Alert (TCA) Notification

{Requirement I. 047} A Threshold Crossed Alert (TCA) Notification, is an event used across the NML-EML Interface to indicate that a performance monitoring parameter threshold has been crossed.

An Threshold Crossed Alert (TCA) Notification shall have in addition to the attributes identified in {Requirement I. 068} and {Requirement I. 093} the following attributes:

1) Native EMS name

   This attribute represents the name of the object whose threshold has been crossed as presented on the EMS GUI. The native EMS name attribute is owned by the EMS which may or may not support changing this value.

2) Clearable

   This attribute shall indicate whether the TCA shall have an associated clear or is a clear.

3) Perceived severity

   This attribute shall indicate (when the TCA is reported as an alarm) whether it is a raise (value shall be INDETERMINATE) or a clear (value shall be CLEARED) alarm.

4) Layer rate

   This attribute shall indicate the layer at which the threshold was crossed (Refer to {Requirement I. 022}).

5) Granularity

   This attribute shall represent the time granularity of the TCA, either 15 minutes or 24 hours. This attribute is not applicable for instantaneous measurements (i.e. gauge type measurements).

6) Parameter name

   Refer to {Requirement I. 035}.

7) Parameter location

   Refer to {Requirement I. 035}.

8) Threshold type

   Refer to {Requirement I. 035}.

9) Value

   Refer to {Requirement I. 035}. *OPTIONAL*

10) Measurement units

   Refer to {Requirement I. 035}. - *OPTIONAL*

11) Acknowledgement

   Refer to {Requirement I. 071}.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.35.2.8 Alarm Notification

{Requirement I. 048} An Alarm Notification, is an event used across the NML-EML Interface to indicate that a fault condition has occurred.

An Alarm Notification shall have in addition to the attributes identified in {Requirement I. 068} and {Requirement I. 093} the following attributes:

1) Clearable

   Refer to {Requirement I. 050}.

2) Layer rate

   Refer to {Requirement I. 022}.

3) Probable Cause

   Refer to {Requirement I. 051}.

4) Perceived severity

   Refer to {Requirement I. 052}.

5) Service affecting

   Refer to {Requirement I. 053}.

6) Probable Cause Qualifier

   Refer to {Requirement I. 049}. *OPTIONAL.*

7) Affected PTPs

   This attribute shall in the case of equipment related alarms represent the names of the affected Physical Termination Point (PTP) implemented by the alarmed equipment. *OPTIONAL.*

8) Additional text

   Refer to {Requirement I. 056}. *OPTIONAL*.

9) Native EMS Name

   Refer to {Requirement I. 060}. *OPTIONAL.*

10) Native Probable Cause

   This attribute shall represent the value of the probable cause shown on the EMS user interface. *OPTIONAL.*

11) Acknowledgement

   Refer to {Requirement I. 071}.

12) Root Cause Alarm indication

This attribute shall indicate whether the alarm is a raw (un-correlated) alarm or a root cause alarm indication.

13) X.733 Event Type

This attribute shall represent the classification of the alarm in terms of the categories specified in ITU-T X.733. This is consistent with the ITU-T X.733 definition. *OPTIONAL.*

14) X.733 Specific problems

This attribute shall represents a clarification of the Probable Cause of the alarm. This is similar to Probable Cause Qualifier, but this attribute is designed to be human readable and compatible with ITU usage. This is consistent with the ITU-T X.733 definition. *OPTIONAL*

15) X.733 Backed-up status

This attribute shall represent whether or not the object emitting the alarm has been backed-up, and services provided to the user have, therefore, not been disrupted. This is consistent with the ITU-T X.733 definition *OPTIONAL*

16) X.733 Back-up object

This attribute shall represent the object that is providing back-up services for the object to which the alarm notification pertains. This parameter shall be present when the X.733 Backed-up status attribute is present and indicates that the object has been backed up. This is consistent with the ITU-T X.733 definition. *OPTIONAL.*

17) X.733 Trend indication

This attribute shall represent the current severity trend of the object it indicates that there are one or more alarms (“outstanding alarms”) which have not been cleared, and pertain to the same object as that to which this alarm (“current alarm”) pertains. This is consistent with the ITU-T X.733 definition. *OPTIONAL*

18) X.733 Correlated notifications

This attribute shall represent the a set of Notification identifiers and, if necessary, their associated object names. This set is defined to be the set of all notifications to which this notification is considered to be correlated. The source object name shall be present if the correlated event report is from an object other than the one in which the Correlated Notifications parameter appears. Otherwise it shall be empty. This is consistent with the ITU-T X.733 definition. *OPTIONAL.*

19) X.733 Monitored attributes

This attribute shall represent the one or more attributes of the managed object and their corresponding values at the time of the alarm. This is consistent with the ITU-T X.733 definition. *OPTIONAL.*

20) X.733 Proposed repair actions

This attribute shall represent one or more possible solutions (such as switch in standby equipment, retry, replace media). This is consistent with the ITU-T X.733 definition. *OPTIONAL.*

21) X.733 Additional Information

This attribute shall represent a set of additional information in an alarm notification. The same information can be directly encoded as separate parameters of the notification. However, this parameter is retained for consistency with ITU-T X.733. *OPTIONAL.*

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.35.2.8.1 Probable Cause Qualifier

{Requirement I. 049} The optional Probable Cause Qualifier parameter, when present, identifies further refinements to the Probable cause of the alarm, so as to correlate the "raise" and "clear" notifications of the same fault condition in case of ambiguity (i.e., when several different fault conditions give rise to the same values). The parameter gives more detail about the alarm, e.g., it may further qualify the source. Refer to {Requirement I. 054}.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.1.35.2.8.2 Clearable

{Requirement I. 050} An indication is required as to whether an alarm raise event will have an associated alarm clear event. If an alarm clear event is generated then the alarm is defined to be clearable.

The same distinction is used in Threshold Crossing Alert (TCA) Notification.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.35.2.8.3 Probable Cause

{Requirement I. 051} The Probable Cause will allow the EMS to indicate the likely cause of the alarm. Refer to supporting document SD1-33_ProbableCauses.pdf for the currently specified probable cause names.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.35.2.8.4 Perceived Severity

{Requirement I. 052} Perceived Severity is defined as:

- Critical
- Major
- Minor
- Warning
- Cleared or
- Indeterminate

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.35.2.8.5 Service Affecting

{Requirement I. 053} The service affecting indication is defined as the EMS' determination of whether or not the condition affects service. The EMS shall indicate if the condition is service affecting, is not service affecting, or is unknown as to whether it is service affecting.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.35.2.8.6 Alarm Identification

{Requirement I. 054} An instance of an alarm shall be uniquely identifiable if:

1) The Source Name, Layer and Probable Cause can uniquely correlate clears with alarms, then the Probable Cause Qualifier is empty (default value).
2) The Source Name, Layer and Probable Cause are not sufficient to uniquely correlate clears with raises, then the Probable Cause Qualifier field is not empty. It contains information such that any clear of that alarm would correlate to the alarm.

This means that if the alarm is raised, cleared, raised again, and cleared again, if the original clear and second alarm were missed, the second clear would clear the first alarm.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.1.35.2.8.7 Additional Text

{Requirement I. 056} The optional Additional Text attribute allows a free form text description to be reported.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.1.35.2.8.8 Acknowledgement

{Requirement I. 071} The optional Acknowledgement attribute shall have the following possible values:.

1) Not applicable

   This indicates that the EMS does not support acknowledgement for this event or does not support acknowledgement at all

2) Acknowledged

   This indicates that the alarm has been acknowledged in the EMS. All alarm fields other than emsTime and acknowledge indication shall remain similar to the original alarm notification.(The emsTime is always provided as the time that the alarm acknowledgement notification has been reported by the EMS.)

3) Unacknowledged

   This indicates that the alarm has not been acknowledged but the EMS, or in the event that the alarm has been previously acknowledged and then unacknowledged. All alarm fields other than emsTime shall remain in that case similar to the original alarm notification.(The emsTime is always provided as the time that the alarm acknowledgement notification has been reported by the EMS.).

Source: Version 3.0.

Mandatory/Optional: Optional.

4.1.35.2.9 File Transfer Status Notification

{Requirement I. 058}

A File Transfer Status Notification, is an event used across the NML-EML Interface to indicate that status of the transfer of the performance monitoring data file.

A File Transfer Status Notification shall have in addition to the attributes identified in {Requirement I. 068} the following attributes:

1) File name

   This attribute shall represent the name of the file being transferred (this shall include the path name).

2) Transfer status

   This attribute shall represent the current state of the transfer (in progress, failed or completed)

3) Percentage complete

   This attribute shall indicate the percent complete of the file transfer it shall be in the range 0..100.

4) Reason for failure

   This attribute shall represent, in the event of a failure, reason for the failure.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.1.35.2.10 Route Change Notification

{Requirement I. 059} A Route Change Notification is an event used across the NML-EML Interface to indicate a change in a Subnetwork Connection (SNC) route.

A Route Change Notification shall have in addition to the attributes identified in {Requirement I. 068} and {Requirement I. 093} the following attributes:

1) Route change state

   This attribute shall represent the current state of the route change (started, completed, or failed)

2) Route

   This attribute shall represent the new route following a successful route change.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.1.35.2.11 Heartbeat Notification

{Requirement I. 077} A Heartbeat Notification is an event used across the NML-EML Interface to indicate the state of the notification delivery mechanism between the EMS and the NMS.

The Heartbeat Notification shall have in addition to the attributes identified in {Requirement I. 068} the following attributes:

1) Object Name

   This attribute shall represent the name of the object against which the event notification is generated. In this case it shall be the name of the EMS

2) Object Type

   This attribute shall represent the type of the object against which the event notification is generated. In this case it shall identify the EMS as the type of the object

3) EMS timestamp

   This attribute shall represent the time at which the event occurred at the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.1.35.2.12 Performance Monitoring Point (PMP) State Change Notification

{Requirement I. 083}

A Performance Monitoring Point (PMP) State Change Notification is a special type of State Change Notification event used across the NML-EML Interface to indicate the following:

- PM data has been cleared
- PM data collection has been disabled or enabled
- TCA generation has been enabled or disabled.

This notification is used to report only one type of change at a time. The EMS shall not use both this notification and individual PMP state change notifications to report the same event.

This notification and an attribute value change notification on a PMP may be used interchangeably by the EMS s on individual PMPs and other times on a list of PMPs.

A PMP State Change Notification shall have in addition to the attributes identified in {Requirement I. 068} the following attributes:

1) PMP name(s)

   This attribute shall represent a list of the PMP name(s) for which a state attribute has changed its value.

2) Attribute value(s)

   This attribute shall represent a list of the state attribute name(s) that have changed their value along with their new values.

3) EMS timestamp

   This attribute shall represent the time at which the event occurred at the EMS.

4) NE timestamp

   This attribute shall represent the time at which the event occurred at the NE.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.1.35.2.13 Software Backup Status Notification

{Requirement I. 086}

A Software Backup Status Notification, is an event used across the NML-EML Interface to indicate that status of the backup of the Managed Element (ME) data.

A Software Backup Status Notification shall have in addition to the attributes identified in {Requirement I. 068} the following attributes:

1) NE Name

   This attribute shall represent the name of the Network Element that is being backed up.

2) Backup status

   This attribute shall represent the current state of the transfer (idle. in progress, completed, aborted or failed)

3) EMS timestamp

   This attribute shall represent the time at which the event occurred at the EMS.

4) NE timestamp

   This attribute shall represent the time at which the event occurred at the NE.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.1.35.2.14 Log Attribute Value Change Notification

{Requirement I. 095} A Log Attribute Value Change Notification is a special type of Attribute Value Change Notification event used across the NML-EML Interface to indicate the following Log attributes have changed:

- capacity alarm threshold
- log full action
- maximum log size
- start time
- stop time
- week mask
- changing the discriminator constraint
- max record life

A Log Attribute Value Change Notification shall have in addition to the attributes identified in {Requirement I. 068} the following attributes:

1) Attribute identifier

   This attribute shall represent the attribute that has changed its value.

2) Old attribute value

   This attribute shall represent the old value of the attribute.

3) New attribute value

   This attribute shall represent the new value of the attribute.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.1.35.2.15 Log Capacity Threshold Alarm Notification

{Requirement I. 096}

A Log Capacity Threshold Alarm Notification is a special type of Threshold Crossing Alert (TCA) Notification event used across the NML-EML Interface to indicate the Log capacity alarm threshold attribute has been crossed.

A Log Capacity Threshold Alarm Notification shall have in addition to the attributes identified in {Requirement I. 068} the following attributes:

1) Observed value

   This attribute shall represent the current Log size, as a percentage of the maximum Log size.

2) Crossed value

   This attribute shall represent the threshold level that has been crossed.

3) Perceived severity

   This attribute shall represent the severity of the alarm. The perceived severity is minor if the Log is not full, and critical otherwise.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.1.35.2.16 Log Processing Error Alarm Notification

{Requirement I. 097}

A Log Processing Error Alarm Notification is a special type of Alarm Notification event used across the NML-EML Interface to indicate a processing error has occurred associated with the Log functionality.

A Log Processing Error Alarm Notification shall have in addition to the attributes identified in {Requirement I. 068} the following attributes:

1) Error number

   This attribute shall represent the error number associated with the problem.

2) Error reason

   This attribute shall represent a textual description of the problem.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.1.35.2.17 Log State Change Notification

{Requirement I. 098}

A Log State Change Notification is a special type of State Change Notification event used across the NML-EML Interface to indicate a change in the state of the Log.

A Log State Change Notification shall have in addition to the attributes identified in {Requirement I. 068} the following attributes:

1) State identifier

   This attribute shall represent the attribute that has changed its value.

2) New state value

   This attribute shall represent the new value of the state.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.2 Category II: Normal Sequences, Dynamic Requirements

4.2.1 Inventory Management

4.2.1.1 Inventory Retrieval

4.2.1.1.1 Element Management System (EMS) Inventory

{Requirement II. 001} The NML-EML Interface shall allow the NMS to retrieve all the attributes of the Element Management System (EMS).

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.1.1.2 Managed Element (ME) Inventory

{Requirement II. 002} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Managed Element (ME)s that are being managed by the EMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 003} The NML-EML Interface shall allow the NMS to retrieve the names of all the Managed Element (ME)s that are being managed by the EMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 004} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Managed Element (ME)s contained within an NMS specified MultiLayer Subnetwork (MLSN).

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 005} The NML-EML Interface shall allow the NMS to retrieve the names of all the Managed Element (ME)s contained within an NMS specified MultiLayer Subnetwork (MLSN)

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 006} The NML-EML Interface shall allow the NMS to retrieve the attributes of a Managed Element (ME) given an NMS specified ME name.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.1.1.3 MultiLayer Subnetwork (MLSN) Inventory

The NMS and EMS shall have a common view of the topology that is under the control of the EMS. The EMS will determine the MultiLayer Subnetwork (MLSN)s and shall provide the MLSNs information to the NMS.

{Requirement II. 008} The NML-EML Interface shall allow the NMS to retrieve the attributes of a MultiLayer Subnetwork (MLSN), given an NMS specified MLSN name.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 009} The NML-EML Interface shall allow the NMS to retrieve the names of the containing MultiLayer Subnetwork (MLSN)(s) for a given NMS specified Managed Element (ME) name.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 010} The NML-EML Interface shall allow the NMS to retrieve a the names of all the top-level MultiLayer Subnetwork (MLSN)(s).

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 011} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the top-level MultiLayer Subnetwork (MLSN)s that are being managed by the EMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.1.1.4 Topological Link (TL) Inventory

{Requirement II. 012} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the top-level Topological Link (TL)s between the MultiLayer Subnetwork (MLSN)s that are managed by the EMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 013} The NML-EML Interface shall allow the NMS to retrieve the names of all the top-level Topological Link (TL)s between the MultiLayer Subnetwork (MLSN)s that are managed by the EMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 014} The NML-EML Interface shall allow the NMS to retrieve the attributes of a top-level Topological Link (TL), given an NMS specified top-level TL name.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 015} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Topological Link (TL)s between NEs given an NMS specified MultiLayer Subnetwork (MLSN).

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 016} The NML-EML Interface shall allow the NMS to retrieve the names of all the Topological Link (TL)s between NEs given an NMS specified MultiLayer Subnetwork (MLSN).

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 017} The NML-EML Interface shall allow the NMS to retrieve the attributes of a Topological Link (TL), given an NMS specified TL name.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 380} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Topological Link (TL)s that are terminated at a specific Flow Domain (FD) given an NMS specified FD name.

Note:

This also includes the internal Topological Links.

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.1.1.5 Call Inventory

{Requirement II. 381}

{R II.013}
{R II. Call 1}

The NML-EML Interface shall allow the NMS to retrieve details of all Calls within the scope of an NMS specified top-level MultiLayer Routing Area (MLRA) name. The EMS shall return the following:

- the attributes of all the Calls.
- the attributes of all the top-level Connections for each Call.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 383}

{R II.091}
{R II. Call 2}

The NML-EML Interface shall allow the NMS to retrieve the details of a Call within the scope of an NMS specified top-level MultiLayer Routing Area (MLRA) name with an NMS specified Call ID or Call Name. The EMS shall return the following:

- all the attributes of the Call.
- all the attributes of the top-level Connections within the Call.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 415}

The NML-EML Interface shall allow the NMS to retrieve details of all Calls within the scope of an NMS specified MultiLayer Routing Area (MLRA) name. The EMS shall return the following:

- the attributes of all the Calls.
- the attributes of all the top-level Connections for each Call.
- the attributes of all the Subnetwork Connection (SNC)s for each Connection.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 416} The NML-EML Interface shall allow the NMS to retrieve the details of all Calls within the scope of an NMS specified MultiLayer Routing Area (MLRA) name given an NMS specified Managed Element (ME) name. The EMS shall return the following:

- the attributes of all the Calls.
- the attributes of all the top-level Connections for each Call.
- all the attributes of all the Subnetwork Connection (SNC)s within the Connection.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 417} The NML-EML Interface shall allow the NMS to retrieve the details of all Calls within the scope of an NMS specified MultiLayer Routing Area (MLRA) name given an NMS specified Termination Point (TP) name. The EMS shall return the following:

- the attributes of all the Calls.
- the attributes of all the top-level Connections for each Call - if they exist.
- the attributes of all the Subnetwork Connection (SNC)s within the Connection.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 386} The NML-EML Interface shall allow the NMS to retrieve the details of a Call within the scope of an NMS specified top-level MultiLayer Routing Area (MLRA) name given an NMS specified Call name. The EMS shall return the following:

- all the attributes of the Call.
- all the attributes of the top-level Connections.
- all the attributes of the Subnetwork Connection (SNC)s within the Connection.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 385}

{R II.023}
{R II. Call 3}

The NML-EML Interface shall allow the NMS to retrieve the IDs of all the Calls within the scope of an NMS specified top-level MultiLayer Routing Area (MLRA) name; associated to an NMS specified Termination Point (TP) name, or Subnetwork Point Pool (SNPP) name, or TNA.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 387}

{R II.015}
{R II. Call 6}

The NML-EML Interface shall allow the NMS to retrieve the attributes of a Call within the scope of an NMS specified top-level MultiLayer Routing Area (MLRA) name given an NMS specified Call Name.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 388}

{R II. Call 7}

The NML-EML Interface shall allow the NMS to retrieve the details of all Calls within the scope of an NMS specified MultiLayer Routing Area (MLRA) name given an NMS specified Managed Element (ME) name. The EMS shall return the following:

- all the attributes of the Calls.
- all the attributes of the top-level Connections.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 382}

{R II. Call 12}

The NML-EML Interface shall allow the NMS to retrieve details of all Calls within the scope of an NMS specified MultiLayer Subnetwork (MLSN) name. The EMS shall return the following:

- the attributes of all the Calls.
- the attributes of all the Subnetwork Connection (SNC)s for each Call.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 412}

The NML-EML Interface shall allow the NMS to retrieve the details of all Calls within the scope of an NMS specified MultiLayer Subnetwork (MLSN) name given an NMS specified Managed Element (ME) name. The EMS shall return the following:

- the attributes of all the Calls.
- all the attributes of all the Subnetwork Connection (SNC)s within the Call.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 413}

The NML-EML Interface shall allow the NMS to retrieve the details of all Calls within the scope of an NMS specified MultiLayer Subnetwork (MLSN) name given an NMS specified Termination Point (TP) name. The EMS shall return the following:

- the attributes of all the Calls.
- the attributes of all the Subnetwork Connection (SNC)s within the Call.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 384}

{R II. Call 13}

The NML-EML Interface shall allow the NMS to retrieve the details of a Call within the scope of an NMS specified MultiLayer Subnetwork (MLSN) name given an NMS specified Call name. The EMS shall return the following:

- all the attributes of the Call.
- all the attributes of the Subnetwork Connection (SNC)s within the Call.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 414}

The NML-EML Interface shall allow the NMS to retrieve the attributes of a Call within the scope of an NMS specified MultiLayer Subnetwork (MLSN) name given an NMS specified Call name.

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.1.1.6 Connection Inventory

{Requirement II. 389}

{R II.022}
{R II. Call 10}

The NML-EML Interface shall allow the NMS to retrieve all of the attributes of a (top-level) Connection (managed by an EMS) given an NMS specified (top-level) Connection name.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 391}

{R II. 036}
{R II. Connection 2}

The NML-EML Interface shall allow the NMS to retrieve, for a specified (top-level) Connection, the list of MultiLayer Routing Area (MLRA) identifiers (one level subordinate to the MLRA containing the specified Connection) in which the Connections that support the specified (top-level) Connection are contained.

The response shall indicate whether the list is a full list of all MLRA identifiers or whether the list is a partial list (a sparse response).

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.1.1.7 Subnetwork Connection (SNC) Inventory

4.2.1.1.7.1 Flexible Subnetwork Connections (SNC)

A Subnetwork Connection (SNC) shall be considered a flexible SNC unless explicitly stated otherwise.

{Requirement II. 018} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Subnetwork Connection (SNC)s contained within an NMS specified MultiLayer Subnetwork (MLSN) name.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 019} The NML-EML Interface shall allow the NMS to retrieve the names of all the Subnetwork Connection (SNC) contained within an NMS specified MultiLayer Subnetwork (MLSN).

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 020} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Subnetwork Connection (SNC)s contained within an NMS specified MultiLayer Subnetwork (MLSN) and for a specific SNC layer rate(s).

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 021} The NML-EML Interface shall allow the NMS to retrieve the names of all the Subnetwork Connection (SNC)s contained within an NMS specified MultiLayer Subnetwork (MLSN) and connection rate(s).

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 022} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Subnetwork Connection (SNC)s that contain the NMS specified Termination Point (TP) and connection rate(s).

1) If a Physical Termination Point (PTP) is specified, then all SNC(s) that pass through the contained Connection Termination Point (CTP)s are returned. The SNCs returned include the CTPs at either end or as part of their route.

2) If a CTP is specified, then all the SNC(s) that pass through the specified CTP are returned. The SNCs returned include the CTPs at either end or as part of the route.

   If the CTP provides the source of a point-to-multipoint SNC then all the SNCs of that multipoint configuration that connect to that CTP will be returned.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 023} The NML-EML Interface shall allow the NMS to retrieve the names of all the Subnetwork Connection (SNC)s that contain the NMS specified Termination Point (TP) and connection rate(s).

If the connection rate list is empty:

1) If a Physical Termination Point (PTP) is specified then all SNC(s) containing all of the PTP(s)' contained Connection Termination Point (CTP)s are returned.

2) If a CTP is specified, then all the SNC(s) that contain the specified CTP are returned.

For either a PTP or a CTP identified by the NMS, if the NMS specifies one or more connection rates, the list of returned SNC names are constrained based on the specified connection rates.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 024} The NML-EML Interface shall allow the NMS to retrieve the attributes of a Subnetwork Connection (SNC) given an NMS specified SNC name.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 025} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Subnetwork Connection (SNC)s given an NMS specified SNC user label.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.1.1.7.2 Fixed Subnetwork Connections (SNC)

A fixed Subnetwork Connection (SNC) shall be an SNC in which all of the Cross-Connect (XC)s are fixed.

{Requirement II. 179} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the fixed Subnetwork Connection (SNC)s contained within an NMS specified MultiLayer Subnetwork (MLSN).

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 180} The NML-EML Interface shall allow the NMS to retrieve the names of all the fixed Subnetwork Connection (SNC) contained within an NMS specified MultiLayer Subnetwork (MLSN).

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 181} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the fixed Subnetwork Connection (SNC)s that contain the NMS specified Termination Point (TP) and connection rate(s).

1) If a Physical Termination Point (PTP) is specified, then all SNC(s) that pass through the contained Connection Termination Point (CTP)s are returned. The SNCs returned include the CTPs at either end or as part of their route.

2) If a CTP is specified, then all the SNC(s) that pass through the specified CTP are returned. The SNCs returned include the CTPs at either end or as part of the route.

   If the CTP provides the source of a point-to-multipoint SNC then all the SNCs of that multipoint configuration that connect to that CTP will be returned.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 182} The NML-EML Interface shall allow the NMS to retrieve the names of all the fixed Subnetwork Connection (SNC)s that contain the NMS specified Termination Point (TP) and connection rate(s).

If the connection rate list is empty:

1) If a Physical Termination Point (PTP) is specified then all SNC(s) containing all of the PTP(s)' contained Connection Termination Point (CTP)s are returned.

2) If a CTP is specified, then all the SNC(s) that contain the specified CTP are returned.

For either a PTP or a CTP identified by the NMS, if the NMS specifies one or more connection rates, the list of returned SNC names are constrained based on the specified connection rates.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.1.1.8 Route Inventory

{Requirement II. 026} The NML-EML Interface shall allow the NMS to retrieve the attributes of a Route given an NMS specified Subnetwork Connection (SNC) name.

If the SNC has alternative routes, then the NML-EML Interface shall allow the NMS to retrieve the intended route (if the SNC is the Pending or Partial state), the active route otherwise.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 218} The NML-EML Interface shall allow the NMS to retrieve the attributes of a Route and the attributes of all the Topological Link (TL)s given an NMS specified Subnetwork Connection (SNC) name.

If the SNC has alternative routes, then the NML-EML Interface shall allow the NMS to retrieve the intended route (if the SNC is the Pending or Partial state), the active route otherwise.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 256} The NML-EML Interface shall allow the NMS to retrieve the attributes of the backup Route given an NMS specified Subnetwork Connection (SNC) or Connection name and Route identifier.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 260} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Routes (intended and backup) given an NMS specified Subnetwork Connection (SNC) name.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 261} The NML-EML Interface shall allow the NMS to retrieve the attributes of the intended Route given an NMS specified Subnetwork Connection (SNC) or Connection name.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 390}

{R II. Connection 1}
{R II. 034}

The NML-EML Interface shall allow the NMS to retrieve all the attributes of a Route, and optionally all the attributes of the MultiLayer Subnetwork Point Pool Link (MLSNPPLink)s, given an NMS specified (top-level) Connection name.

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.1.1.9 Termination Point (TP) Inventory

{Requirement II. 027} The NML-EML Interface shall allow the NMS to retrieve the attributes of a Termination Point (TP) given an NMS specified TP name.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 029} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Connection Termination Point (CTP)s that are associated with an NMS specified Traffic Descriptor (TD) name.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 030} The NML-EML Interface shall allow the NMS to retrieve the names of all the Termination Point Pool (TP Pool)s that are associated with an NMS specified Termination Point (TP).

Descriptions of the associations can be obtained from the respective TPPool attributes for the description of use.

In case of ATM VP TP Pool administration the specified TP is an ATM NI CTP representing a real user network interface that gets partitioned into virtual user network interfaces represented by TP pools. Each TP pool contains ATM VP CTPs that are clients of the specified ATM NI CTP.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 186} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Termination Point (TP)s that are associated with an NMS specified Transmission Descriptor (TMD) name.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 285} The NML-EML Interface shall allow the NMS to retrieve the names of all the Termination Point (TP)s that are associated with an NMS specified Threshold Crossing Alert (TCA) Parameter Profile name.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.1.1.10 Termination Point Pool (TP Pool) Inventory

{Requirement II. 268} The NML-EML Interface shall allow the NMS to retrieve the attributes of a Termination Point Pool (TP Pool) given an NMS specified TP Pool name.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 031} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Termination Point Pool (TP Pool)s given an NMS specified MultiLayer Subnetwork (MLSN) name.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 032} The NML-EML Interface shall allow the NMS to retrieve the names of all the Termination Point Pool (TP Pool)s given an NMS specified MultiLayer Subnetwork (MLSN) name.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 028} The NML-EML Interface shall allow the NMS to retrieve the names of all the Termination Point (TP)s and Group Termination Point (GTP)s that have been grouped by the NMS specified Termination Point Pool (TP Pool).

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.1.1.11 Physical Termination Point (PTP) Inventory

{Requirement II. 033} The NML-EML Interface shall allow the NMS to retrieve the attributes all the Physical Termination Point (PTP)s (ports) given an NMS specified Managed Element (ME) name and one or more layer rate(s).

The NML-EML Interface shall allow the NMS to specify one or more layer rates to set the scope of the retrieval of PTPs from the EMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

Example physical TP layer rates covering SONET and SDH are: Electrical STS1 / STM0; Electrical STS3 / STM1; Optical OC1 / STM0; Optical OC3 / STM1

{Requirement II. 034} The NML-EML Interface shall allow the NMS to retrieve the names of all the Physical Termination Point (PTP)s (ports) given an NMS specified Managed Element (ME) name and one or more layer rate(s).

The NML-EML Interface shall allow the NMS to specify one or more layer rates to set the scope of the retrieval of PTPs from the EMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

Example physical TP layer rates covering SONET and SDH are: Electrical STS1 / STM0; Electrical STS3 / STM1; Optical OC1 / STM0; Optical OC3 / STM1.

{Requirement II. 035} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Physical Termination Point (PTP)s (ports) given an NMS specified Managed Element (ME) name and connection layer rate(s).

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 036} The NML-EML Interface shall allow the NMS to retrieve the names of all the Physical Termination Point (PTP)s (ports) given an NMS specified Managed Element (ME) name connection layer rate(s).

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 214} The NML-EML Interface shall allow the NMS to retrieve a the attributes of all the Physical Termination Point (PTP)s (ports) and Floating Termination Point (FTP)s given an NMS specified Managed Element (ME) name and one or more layer rate(s).

The NML-EML Interface shall allow the NMS to specify one or more layer rates to set the scope of the retrieval of PTPs and FTPs from the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 215} The NML-EML Interface shall allow the NMS to retrieve the names of all the Physical Termination Point (PTP)s (ports) and Floating Termination Point (FTP)s given an NMS specified Managed Element (ME) name and one or more layer rate(s).

The NML-EML Interface shall allow the NMS to specify one or more layer rates to set the scope of the retrieval of PTPs and FTPs from the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 216} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Physical Termination Point (PTP)s (ports) and Floating Termination Point (FTP)s given an NMS specified Managed Element (ME) name that are capable of supporting the NMS specified connection layer rate(s).

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 217} The NML-EML Interface shall allow the NMS to retrieve the names of the Physical Termination Point (PTP)s (ports) and Floating Termination Point (FTP)s given an NMS specified Managed Element (ME) name that are capable of supporting the NMS specified connection layer rate(s).

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 289} The NML-EML Interface shall allow the NMS to retrieve the names of the Protection Group (PG)s containing the given an NMS specified Physical Termination Point (PTP) name.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.1.1.12 Edge Termination Point (Edge TP) Inventory

{Requirement II. 037} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Edge Termination Point (Edge TP)s (ports) given an NMS specified MultiLayer Subnetwork (MLSN) name and one or more layer rate(s).

Source: Version 2.0.

Mandatory/Optional: Optional.

Example physical TP layer rates covering SONET and SDH are: Electrical STS1 / STM0; Electrical STS3 / STM1; Optical OC1 / STM0; Optical OC3 / STM1.

{Requirement II. 038} The NML-EML Interface shall allow the NMS to retrieve the names of all the Edge Termination Point (Edge TP)s (ports) given an NMS specified MultiLayer Subnetwork (MLSN) name and one or more layer rate(s).

The NML-EML Interface shall allow the NMS to specify one or more layer rates to set the scope of the retrieval of Edge TPs from the EMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

Example physical TP layer rates covering SONET and SDH are: Electrical STS1 / STM0; Electrical STS3 / STM1; Optical OC1 / STM0; Optical OC3 / STM1.

{Requirement II. 039} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Edge Termination Point (Edge TP)s (ports) given an NMS specified MultiLayer Subnetwork (MLSN) name and that are capable of supporting the NMS specified connection layer rate(s).

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 040} The NML-EML Interface shall allow the NMS to retrieve the names of all the Edge Termination Point (Edge TP)s (ports) given an NMS specified MultiLayer Subnetwork (MLSN) name and that are capable of supporting the NMS specified connection layer rate(s).

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.1.1.13 Contained Termination Point (TP) Inventory

4.2.1.1.13.1 Potential Termination Point (TP)s

{Requirement II. 041} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the potentially present Termination Point (TP)(s) contained by (served by) a given TP identified by a NMS specified TP name and layer rate.

Source: Version 2.0.

Mandatory/Optional: Optional.

Some examples:

If the NMS specifies an OC3 physical TP (port) name that can be mapped to STS1s or VT1.5s, then the EMS response would include 3 STS1 TPs (CTPs) and 84 VT1.5 TPs (CTPs), respectively.

If the NMS specifies an STS1 TP name (physical or logical TP, i.e., a physical port or a logical channel (CTP) that can be mapped to VT1.5s), then the EMS response would include 28 VT1.5 TPs (CTPs)

{Requirement II. 042} The NML-EML Interface shall allow the NMS to retrieve the names of all the potentially present Termination Point (TP)(s) contained by (served by) a given TP identified by a NMS specified TP name and layer rate.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.1.1.13.2 Cross-connectible or cross-connected Termination Point (TP)s

{Requirement II. 043} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the currently cross-connectible or cross-connected Termination Point (TP)(s) that are contained by (served by) a given TP identified by a NMS specified TP name and layer rate.

Source: Version 2.0.

Mandatory/Optional: Optional.

Some examples:

If the NMS specifies an OC3 physical TP (port) name that can be mapped to STS1s or VT1.5s, then

If none of the STS1 CTPs are terminated and mapped the EMS response would include only the 3 STS1 TPs (CTPs).

If all 3 STS1 CTPs are terminated and mapped, then 84 VT1.5 TPs (CTPs), are returned.

If the NMS specifies an STS1 TP name (physical or logical TP, i.e., a physical port or a logical channel (CTP) that can be mapped to VT1.5s), then the EMS response would include:

28 VT1.5 TPs (CTPs) if the STS1 TP is terminated and mapped.

No TPs if the STS1 TP is neither terminated nor mapped.

{Requirement II. 044} The NML-EML Interface shall allow the NMS to retrieve the name(s) of all the currently cross-connectible or cross-connected Termination Point (TP)(s) that are contained by (served by) a given TP identified by a NMS specified TP name and layer rate.

Source: Version 2.0.

Mandatory/Optional: Optional.

Some examples:

If the NMS specifies an OC3 physical TP (port) name that can be mapped to STS1s or VT1.5s, then

- If none of the STS1 CTPs are terminated and mapped the EMS response would include only the names of 3 STS1 TPs (CTPs).
- If all 3 STS1 CTPs are terminated and mapped, then the names of 84 VT1.5 TPs (CTPs), are returned.

If the NMS specifies an STS1 TP name (physical or logical TP, i.e., a physical port or a logical channel (CTP) that can be mapped to VT1.5s), then the EMS response would include the names of:

- 28 VT1.5 TPs (CTPs) if the STS1 TP is terminated and mapped.

No TPs if the STS1 TP is neither terminated nor mapped.

4.2.1.1.13.3 In Use Termination Point (TP)s

An “in use” Termination Point (TP) is defined as a TP that is used by an Subnetwork Connection (SNC) in any state (including pending) or a TP that is terminated and mapped (either with or without assigned bandwidth).

{Requirement II. 045} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the “in use” (actual) Termination Point (TP)s that are contained by (served by) a given TP identified by a NMS specified TP name and layer rate.

Source: Version 2.0.

Mandatory/Optional: Optional.

This operation will be used when there are a large number of potential TPs (e.g., in ATM). All of the potential TPs are not returned. The TPs can be scoped on layer rate. If no layer rate is specified, then actual TP at all the contained layers are returned. If layer rate(s) are specified, then only actual TPs at the specified layer rates are returned.

{Requirement II. 046} The NML-EML Interface shall allow the NMS to retrieve the names of all the “in use” (actual) Termination Point (TP) that are contained by (served by) a given TP identified by a NMS specified TP name and layer rate.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.1.1.14 Containing Termination Point (TP) Inventory

{Requirement II. 047} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Termination Point (TP)(s) that contain (serves) the TP identified by the NMS specified TP name

Source: Version 2.0.

Mandatory/Optional: Optional.

Some examples:

a) If the NMS specifies an STS1 TP (CTP) name that is a channel of an OC3, then the EMS would respond with the OC3 physical TP.

b) If the NMS specifies an OC3 physical TP (port) name, then the EMS would respond with no containing TPs.

{Requirement II. 048} The NML-EML Interface shall allow the NMS to retrieve the name(s) of all the Termination Point (TP)(s) that contain the TP identified by the NMS specified TP name.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.1.1.15 Floating Termination Point (FTP) Inventory

{Requirement II. 210} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Floating Termination Point (FTP)s given an NMS specified Managed Element (ME) name and one or more layer rate(s).

The NML-EML Interface shall allow the NMS to specify one or more layer rates to set the scope of the retrieval of FTPs from the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 211} The NML-EML Interface shall allow the NMS to retrieve the names of all the Floating Termination Point (FTP)s (ports) given an NMS specified Managed Element (ME) name and one or more layer rate(s).

The NML-EML Interface shall allow the NMS to specify one or more layer rates to set the scope of the retrieval of TPs from the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 212} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Floating Termination Point (FTP)s given an NMS specified Managed Element (ME) name that are capable of supporting the NMS specified connection layer rate(s).

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 213} The NML-EML Interface shall allow the NMS to retrieve the names of all the Floating Termination Point (FTP)s given an NMS specified Managed Element (ME) name that are capable of supporting the NMS specified connection layer rate(s).

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.1.1.16 Associated Termination Point (TP) Inventory

{Requirement II. 049} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Termination Point (TP)s associated with a given TP identified by an NMS specified TP name.

Some examples of associated TPs to be returned by the EMS

In the case of an open Path Switched Ring (PSR) topology, if the NMS specifies a CTP (timeslot) on one end of the open PSR, the EMS shall return the CTP on the other end of the open PSR corresponding to the specified timeslot, associated with relationship, (as illustrated in Figure 4.3).

Source: Version 2.0.

Mandatory/Optional: Optional.

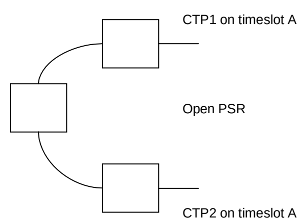


Figure 4.3: Associated CTP Relationship in an Open Ring

4.2.1.1.17 Supported Termination Point (TP) Inventory

{Requirement II. 050} The NML-EML Interface shall allow the NMS to retrieve the attributes of all of the Physical Termination Point (PTP)(s) that are supported by the Equipment identified by the NMS specified Equipment name.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 051} The NML-EML Interface shall allow the NMS to retrieve all the name(s) of the Physical Termination Point (PTP)(s) that are supported by the Equipment identified by the NMS specified Equipment name.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.1.1.18 Supporting Termination Point (TP) Inventory

{Requirement II. 052} MFD 11 The NML-EML Inerface shall allow the NMS to retrieve the attributes of all the Equipment(s) that support the Physical Termination Point (PTP) or Matrix Flow Domain (MFD) identified by the NMS specified PTP or MFD name.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 053} MFD 21 The NML-EML Interface shall allow the NMS to retrieve all the name(s) of the Equipment(s) that support the Physical Termination Point (PTP) or Matrix Flow Domain (MFD) identified by the NMS specified TP or MFD name.

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.1.1.19 Performance Management (PM) Inventory

4.2.1.1.19.1 Performance Management (PM) Parameters

{Requirement II. 054} The NML-EML Interface shall allow the NMS to retrieve the set of supported PM parameters for an NMS specified Managed Element (ME). Refer to {Requirement II. 118}.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.1.1.19.2 TCA Thresholds

{Requirement II. 055} The NML-EML Interface shall allow the NMS to retrieve the value of a specific Threshold Crossing Alert (TCA) Parameter for a Termination Point (TP) identified by an NMS specified TP name, layer rate, granularity and location.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.1.1.19.3 TCA Parameter Profiles

{Requirement II. 233} The NML-EML Interface shall allow the NMS to retrieve the names of all the Threshold Crossing Alert (TCA) Parameter Profiles that are being managed by the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 273} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Threshold Crossing Alert (TCA) Parameter Profiles that are being managed by the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 234} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Threshold Crossing Alert (TCA) Parameter Profile given an NMS specified TCA Parameter Profile name.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.1.1.19.4 Performance Monitoring Point (PMP) Inventory

{Requirement II. 221} The NML-EML Interface shall allow the NMS to retrieve the names of all the Performance Monitoring Point (PMP)s supported by a Managed Element (ME) or a Termination Point (TP) for an NMS specified ME or TP name.

In the case of the NMS supplied ME name the names of the PMPs associated with all the TPs contained within the ME shall be returned. In the case of the NMS supplied TP name then only the names of those PMPs associated with the named TP shall be returned.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 222} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Performance Monitoring Point (PMP) supported by a Managed Element (ME) or a Termination Point (TP) for an NMS specified ME or TP name.

In the case of the NMS supplied ME name the PMPs associated with all the TPs contained within the ME shall be returned. In the case of the NMS supplied TP name then only those PMPs associated with the named TP shall be returned.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.1.1.20 Equipment Inventory

{Requirement II. 056} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Equipments given an NMS specified Managed Element (ME) name.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 057} The NML-EML Interface shall allow the NMS to retrieve the names of all the Equipments given an NMS specified Managed Element (ME) name.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 058} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Equipment contained directly in the Equipment Holder identified by an NMS specified Equipment Holder name.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 279} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Equipments and Equipment Holders given an NMS specified Equipment Holder name.

The EMS shall return all Equipments and Equipment Holders at all levels of containment beneath the sepcified Equipment Holder.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 280} The NML-EML Interface shall allow the NMS to retrieve the names of all the Equipments given an NMS specified Equipment Holder name.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.1.1.21 Supporting Equipment Inventory

{Requirement II. 225} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Equipments that support an Equipment identified by an NMS specified Equipment name.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 226} The NML-EML Interface shall allow the NMS to retrieve names of all the Equipments that support an Equipment identified by an NMS specified Equipment name.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.1.1.22 Supported Equipment Inventory

{Requirement II. 227} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Equipments that are supported by an Equipment identified by an NMS specified Equipment name.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 228} The NML-EML Interface shall allow the NMS to retrieve the names of all the Equipments that are supported by an Equipment identified by an NMS specified Equipment name.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.1.1.23 Protection Inventory

{Requirement II. 059} The NML-EML Interface shall allow the NMS to retrieve all the Protection Group (PG)s available in the Managed Element (ME)s.

Source: Version 2.0.

Mandatory/Optional: Optional.

This shall be used by the NMS to manage protected trails between subnetworks. In the case of MSSPRing (BLSR), these protection groups also contain information about the, SPRING_NODE_ID which is needed at the time of subnetwork connection creation (i.e. the ingress/egress nodes of a ring).

{Requirement II. 274} The NML-EML Interface shall allow the NMS to retrieve the names of all the Connection Termination Point (CTP)s that support Non-Preemtible Unprotected Traffic (NUT) services associated with a given Protection Group (PG) identified by an NMS specified PG name.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 275} The NML-EML Interface shall allow the NMS to retrieve the names of all the Connection Termination Point (CTP)s that support protected services associated with a given Protection Group (PG) identified by an NMS specified PG name.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 276} The NML-EML Interface shall allow the NMS to retrieve the names of all the Connection Termination Point (CTP)s that support preemptively and unprotected services associated with a given Protection Group (PG) identified by an NMS specified PG name.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.1.1.24 Equipment Protection Inventory

{Requirement II. 174} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Equipment Protection Group (EPG)s available in the Managed Element (ME)s.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.1.1.25 Traffic Descriptor (TD) Inventory

{Requirement II. 060} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Traffic Descriptor (TD)s that are being managed by the EMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 061} The NML-EML Interface shall allow the NMS to retrieve the names of all the Traffic Descriptor (TD)s that are being managed by the EMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 062} The NML-EML Interface shall allow the NMS to retrieve the attributes of a Traffic Descriptor (TD) given an NMS specified TD name.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.1.1.26 Transmission Descriptor (TMD) Inventory

{Requirement II. 187}

The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Transmission Descriptor (TMD)s that are being managed by the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 188}

The NML-EML Interface shall allow the NMS to retrieve the names of all the Transmission Descriptor (TMD)s that are being managed by the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 189}

The NML-EML Interface shall allow the NMS to retrieve the attributes of a Transmission Descriptor (TMD) for an NMS specified TMD name.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 352}

TMD 1

The NML-EML Interface shall allow the NMS to retrieve the layered transmission parameters of a Transmission Descriptor (TMD) by using a filter that is based on the layer rates and the groupings defined in the supporting document SD1-16_LayeredParameters.pdf.

It shall also be possible to retrieve all layered transmission parameters defined in the TMD.

For a list of the currently defined set of supported transmission parameters refer to the supporting document SD1-16_LayeredParameters.pdf.

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.1.1.27 Cross-Connect (XC) Inventory

4.2.1.1.27.1 Flexible Cross-Connect (XC)s

A Cross-Connect (XC) shall be considered a flexible XC unless explicitly stated otherwise.

{Requirement II. 063} The NML-EML Interface shall allow the NMS to retrieve the attributes of all of the Cross-Connect (XC)s given an NMS specified Managed Element (ME) name.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.1.1.27.2 Fixed Cross-Connect (XC)s

A fixed Cross-Connect (XC) is an XC that cannot be deleted by the NMS.

{Requirement II. 183} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the fixed Cross-Connect (XC)s given an NMS specified Managed Element (ME) name.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 185} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the potentially fixed Cross-Connect (XC)s that are associated with the NMS specified Termination Point (TP) name.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.1.1.28 Group Termination Point (GTP) Inventory

{Requirement II. 171} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Group Termination Point (GTP)s that are being managed by the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 172} The NML-EML Interface shall allow the NMS to retrieve all the names of all the Group Termination Point (GTP)s that are being managed by the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 173} The NML-EML Interface shall allow the NMS to retrieve the attributes of a Group Termination Point (GTP) for an NMS specified GTP name.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 184} The NML-EML Interface shall allow the NMS to retrieve the attributes of a Group Termination Point (GTP) that contains a given NMS specified Connection Termination Point (CTP) name.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 272} The NML-EML Interface shall allow the NMS to retrieve the names of all the Termination Point Pool (TP Pool)s that are associated with the NMS specified Group Termination Point (GTP) name.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.1.1.29 Alarm Severity Assignment Profile (ASAP) Inventory

{Requirement II. 205} The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Alarm Severity Assignment Profile (ASAP)s that are being managed by the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 206} The NML-EML Interface shall allow the NMS to retrieve the names of all the Alarm Severity Assignment Profile (ASAP)s that are being managed by the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 207} The NML-EML Interface shall allow the NMS to retrieve the attributes of a Alarm Severity Assignment Profile (ASAP) for an NMS specified ASAP name.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 208} The NML-EML Interface shall allow the NMS to retrieve all the Alarm Severity Assignment Profile (ASAP)s that are assigned to an NMS specified object.

Only Termination Point (TP)s can refer to more than one ASAP, at most one ASAP per encapsulated layer rate.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 209} The NML-EML Interface shall allow the NMS to retrieve the name of all the objects assigned to an NMS specified Alarm Severity Assignment Profile (ASAP) name.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.1.2 Inventory Notifications

As subnetwork resources are updated, the EMS shall be required to notify its clients (e.g. NMS) about changes in inventory, e.g., link additions/deletions, Network Element (NE)s being added or deleted from a Subnetwork (for physical and logical resources).

Table 4.4: Objects for which create and delete notification shall be generated

	Object
1	Alarm Severity Assignment Profile (ASAP)
2	Call
3	Connection
4	Equipment
5	Equipment Holder
6	Equipment Protection Group (EPG)
7	Floating Termination Point (FTP)
8	Flow Domain (FD)
9	Flow Domain Fragment (FDFr)
10	Group Termination Point (GTP)
11	Log
12	Managed Element (ME)
13	Matrix Flow Domain (MFD)
14	MultiLayer Routing Area (MLRA)
15	MultiLayer Subnetwork (MLSN)
16	MultiLayer Subnetwork Point Pool (MLSNPP)
17	MultiLayer Subnetwork Point Pool Link (MLSNPPLink)
18	Performance Monitoring Point (PMP)
19	Physical Termination Point (PTP)
20	Protection Group (PG)

Table 4.4: Objects for which create and delete notification shall be generated

	Object
21	Subnetwork Connection (SNC)
22	Termination Point Pool (TP Pool)
23	Threshold Crossing Alert (TCA) Parameter Profile
24	Topological Link (TL)
25	Traffic Conditioning (TC) Profile
26	Traffic Descriptor (TD)
27	Transmission Descriptor (TMD)

4.2.1.2.1 Object Creation Notifications

{Requirement II. 064} The NML-EML Interface shall allow the NMS to subscribe to notifications from the EMS related to the creation of the objects specified in Table 4.4.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.1.2.2 Object Deletion Notifications

{Requirement II. 065} The NML-EML Interface shall allow the NMS to subscribe to notifications from the EMS related to the deletion of the objects specified in Table 4.4.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.1.2.3 Attribute Value Change Notification

{Requirement II. 066} The NML-EML Interface shall allow the NMS to subscribe to notifications from the EMS related to changes in the values of the attributes of the objects specified in Table 4.4 and the following objects:

1) Connection Termination Point (CTP)
2) Element Management System (EMS)

An Attribute Value Change Notification shall be emitted for an object where one or more of its' attribute values have changed.

Source: Version 2.0.

Mandatory/Optional: Optional.

Attribute Value Change notifications are emitted on a single object, multiple attribute basis.

4.2.1.2.4 State Change Notification

{Requirement II. 067} The NML-EML Interface shall allow the NMS to subscribe to notifications from the EMS related to changes in the values of the state attribute of the following objects:

1) Call
2) Connection
3) Connection Termination Point (CTP)
4) Equipment
5) Equipment Holderr
6) Equipment Protection Group (EPG)
7) Floating Termination Point (FTP)
8) Flow Domain (FD)
9) Log
10) Managed Element (ME)
11) Physical Termination Point (PTP)
12) Performance Monitoring Point (PMP)
13) Protection Group (PG)
14) Subnetwork Connection (SNC)

All objects with state attributes shall emit a State Change Notification when the value of the state attribute changes.

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.2 Provisioning

4.2.2.1 Managed Element/Subnetwork Administration and Configuration

4.2.2.1.1 Connection Termination Point (CTP) Termination and Mapping

{Requirement II. 068} The NML-EML Interface shall allow the NMS to request that an NMS specified Connection Termination Point (CTP) be terminated and mapped.

This request, if successful, will configure the NMS specified CTP such that it will then be capable of supporting lower rate connections. If the CTP is successfully configured such that it is capable of supporting lower rate connections, then the CTP's mapping mode should indicate as such.

Note that before carrying out the NMS' request, the EMS should confirm that the NMS specified CTP is capable of being terminated and mapped and that it is not involved in an active cross-connection at the CTP's rate. Some examples:

- Termination and mapping of an STS1 CTP such that it will support VT1.5 CTPs.
- Termination and mapping of a T3 CTP such that it will support T1 CTPs.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 069} The NML-EML Interface shall allow the NMS to request that an NMS specified Connection Termination Point (CTP) no longer be terminated and mapped.

This request, if successful, will configure the NMS specified CTP such that it will then be capable of supporting cross-connections at the CTP's rate. If the CTP is successfully configured such that it is capable of supporting cross-connections at the CTP's rate, then the CTP's mapping mode should indicate as such.

Note that before carrying out the NMS' request, the EMS shall confirm that the NMS specified CTP is not supporting an active Cross-Connect (XC) at a client layer rate (e.g., for an NMS specified STS1 CTP, EMS should confirm that no contained VT1.5 CTPs are involved in an active cross-connection).

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 071} The NML-EML Interface shall allow the NMS to retrieve the mapping mode of an NMS specified Connection Termination Point (CTP) name.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.2.1.2 Provisioning of Termination Point (TP) Parameters

{Requirement II. 072} The NML-EML Interface shall allow the NMS to provision Termination Point (TP) transmission parameters.

Refer to supporting document SD1-16_LayeredParameters.pdf for the currently defined set of supported TP parameters.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.2.1.2.1 Assignment of Transmission Descriptor (TMD)s

{Requirement II. 194} The NML-EML Interface shall allow the NMS to associate an ingress and/or egress Transmission Descriptor (TMD) with a given Termination Point (TP) identified by an NMS specified TP name.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.1.2.1.1 Transmission Descriptor (TMD) Assignment behavior

{Requirement II. 277} The assignment of a Transmission Descriptor (TMD) to a Termination Point (TP) or Matrix Flow Domain (MFD) by using the TMD's name amounts to an overwriting of the layered transmission parameters of the TP or MFD by the layered transmission parameters of the TMD and to an overwriting of the additional information parameters of the TP or MFD by the additional Object information parameters of the TMD.

Note that these parameters may also be set according to {Requirement II. 072}, {Requirement II. 223} and {Requirement II. 307} without using a TMD. Current parameters of the TP or MFD that are not present as parameters of the TMD are left unchanged by the TMD assignment.

The unassignment of a TMD from a TP or MFD (by using the empty TMD name) has no effect on the parameters of the TP or MFD, i.e. the layered transmission parameters and additional info parameters of the TP or MFD remain unchanged.

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.2.1.2.2 Verification of Transmission Descriptor (TMD) Assignment

{Requirement II. 278} The NML-EML Interface shall allow the NMS to validate the Transmission Descriptor (TMD)s assigned to a Termination Point (TP) or Matrix Flow Domain (MFD) given an NMS specified TP or MFD name.

The following is provided for clarification:

- The assignment of a Transmission Descriptor (TMD) to a Termination Point (TP) or Matrix Flow Domain (MFD) is called consistent, if whenever a TMD transmission parameter is also present as a TP transmission parameter or a TMD additional Object information parameter is also present as a TP or MFD additional information parameter and the common parameters of the TMD and TP or MFD have the same values.

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.2.1.2.3 Assignment of Threshold Crossing Alert (TCA) Parameter Profiles

{Requirement II. 235} The NML-EML Interface shall allow the NMS to associate a Threshold Crossing Alert (TCA) Parameter Profile with a Termination Point (TP).

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.1.3 Termination Point (TP) Alarm Reporting

{Requirement II. 108} The NML-EML Interface shall allow the NMS to activate (allow, or turn on) alarm reporting for a particular Termination Point (TP).

The alarm reporting must be turned on at the layer represented by the Termination Point (TP). Refer to {Requirement II. 072}.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 109} The NML-EML Interface shall allow the NMS to deactivate (inhibit, or turn off) alarm reporting for a particular Termination Point (TP).

The alarm reporting must be turned off at the layer represented by the Termination Point (TP). Refer to {Requirement II. 072}

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.2.1.4 ATM Virtual Path (VP) and Virtual Connection (VC) Connection Termination Point (CTP) Provisioning

Section 4.2.2.1.4 is only applicable to ATM technology.

{Requirement II. 073} The NML-EML Interface shall allow the NMS to configure terminated and mapped VP (or VC) Connection Termination Point (CTP)s at the end of a VP (or VC) trail.

A terminated (and available for mapping) VP or VC CTP can be turned to non terminated (nor available for mapping), (and can therefore be deleted on the NE if required), only if not used as a server by other lower level CTPs (e.g., a terminated VP CTP can be deleted only if does not carry any VC CTPs).

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 074} The NML-EML Interface shall allow the NMS to configure (assign) the ingress and egress Traffic Descriptor (TD) on a VP or VC CTP that is terminated.

Source: Version 2.0.

Mandatory/Optional: Optional.

This is used for the management of an explicit VP overlay.

4.2.2.1.5 Subnetwork Connection (SNC) / Connection Alarm Reporting

In order to provide this, the EMS has to correlate TP related information into “arc” related information.

Note: How to provide this correlation is behavior of the EMS and is therefore outside the scope of the MTNM interface. The activation / de-activation do not imply anything on the alarm reporting flag of any of the related TPs of the SNC / topological link.The NMS shall be able to retrieve the status of the activation / de-activation.

{Requirement II. 159} The NML-EML Interface shall allow the NMS to activate (allow, or turn on) alarm reporting for an NMS specified Subnetwork Connection (SNC) or Connection name.

.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 160} The NML-EML Interface shall allow the NMS to de-activate (inhibit, or turn off) alarm reporting for a for an NMS specified Subnetwork Connection (SNC) or Connection name.

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.2.1.6 Topological Link (TL) Alarm Reporting

{Requirement II. 161} The NML-EML Interface shall allow the NMS to activate (allow, or turn on) alarm reporting for a particular Topological Link (TL).

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 162} The NML-EML Interface shall allow the NMS to de-activate (inhibit, or turn off) alarm reporting for a particular Topological Link (TL).

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.1.7 Group Termination Point (GTP) Alarm Reporting

{Requirement II. 219} The NML-EML Interface shall allow the NMS to activate (allow, or turn on) alarm reporting for a particular Group Termination Point (GTP).

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 220} The NML-EML Interface shall allow the NMS to de-activate (inhibit, or turn off) alarm reporting for a particular Group Termination Point (GTP).

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.2 Common Attribute Management

The NML-EML Interface defines a number of objects which have common attributes (an attribute does not have to be defined from every object in order for it to be considered common). The NML-EML Interface shall allow the NMS to set the value of certain of the common attributes. The follow list identifies the common attributes for which the NMS shall be allowed to set the value.

- User label[1]
- Owner
- Additional information
- Native EMS name
- Network Access Domain

[1] The User label is required to be validated for uniqueness by the EMS.

4.2.2.2.1 User Label

{Requirement II. 075} The NML-EML Interface shall allow the NMS to provision the User label attribute for all objects identified in Table 4.1.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.2.2.2 Owner

{Requirement II. 076} The NML-EML Interface shall allow the NMS to provision the Owner attribute for all objects identified in Table 4.1.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.2.2.3 Additional information

{Requirement II. 223} The NML-EML Interface shall allow the NMS to provision the Additional information attribute for all objects identified in Table 4.1.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.2.4 Native EMS Name

{Requirement II. 077} The NML-EML Interface shall allow the NMS to provision the Native EMS name attribute for all objects identified in Table 4.1.

Refer to supporting document SD1-24_nativeEMSName.pdf for the rules for native EMS names.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.2.2.5 Network Access Domain (NAD)

Table 4.5: Objects containing the Network Access Domain attribute

	Object Name
1	Call
2	Connection
3	Flow Domain (FD)
4	Flow Domain Fragment (FDFr)
5	Group Termination Point (GTP)
6	Managed Element (ME)
7	Matrix Flow Domain (MFD)
8	MultiLayer Subnetwork (MLSN)
9	Subnetwork Connection (SNC)
10	Termination Point (TP)[a]
11	Topological Link (TL)

a. This includes all types of TP (i.e. PTPs, CTPs and FTPs)

{Requirement II. 193} The NML-EML Interface shall allow the NMS to provision the Network Access Domain (NAD) attribute for the objects identified in Table 4.5.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.3 Group Termination Point (GTP) Management

4.2.2.3.1 Creation of Group Termination Point (GTP)s

{Requirement II. 164} The NML-EML Interface shall allow the NMS to create a Group Termination Point (GTP) in the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.3.1.1 Group Termination Point (GTP) Create Data

{Requirement II. 165} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS create a Group Termination Point (GTP)

1) User label

   Refer to {Requirement I. 060}.

2) User label uniqueness

   This parameter shall indicate to the EMS to check whether the user label is unique amongst the GTPs within the EMS.

3) Owner

   Refer to {Requirement I. 060}.

4) Contained CTPs

   This parameter shall represent a list of the names of the Connection Termination Point (CTP) that are to be contained by the GTP.

5) Starting CTP name

   In cases where the CTPs are contiguous and of the same layer rate, this parameter shall indicate the first CTP in the group. This parameter is used in lieu of the Contained CTPs parameter.

6) Number of CTPs

   This parameter is used in conjunction with the starting CTP name parameter. It shall indicate the number of contiguous CTPs that follow the first CTP in the group. It equals 1 minus the total number of CTPs in the GTP.

7) Additional information.

   Refer to {Requirement I. 060}.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.2.2.3.2 Deletion of Group Termination Point (GTP)s

{Requirement II. 166} The NML-EML Interface shall allow the NMS to delete a Group Termination Point (GTP) given the NMS specified GTP name from the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.3.3 Modification of Group Termination Point (GTP)s

{Requirement II. 167} The NML-EML Interface shall allow the NMS to add or remove Connection Termination Point (CTP)s from a Group Termination Point (GTP) given the NMS specified CTP names and the GTP name from the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.3.4 Termination Point Pool (TP Pool) Management

4.2.2.3.4.1 Creation of Termination Point Pool (TP Pool)s

{Requirement II. 264} The NML-EML Interface shall allow the NMS to create a Termination Point Pool (TP Pool) in the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.3.4.2 Termination Point Pool (TP Pool) Create Data

{Requirement II. 265} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS create a Termination Point Pool (TP Pool)

1) User label

   Refer to {Requirement I. 060}.

2) User label uniqueness

   This parameter shall indicate to the EMS to check whether the user label is unique amongst the TP Pools within the EMS.

3) Owner

   Refer to {Requirement I. 060}.

4) Containing Subnetwork

   This parameter shall represent the name of MultiLayer Subnetwork (MLSN) containing the TP Pool.

5) Contained members

   This parameter shall represent a list of the names of the Termination Point (TP)s or Group Termination Point (GTP)s, all taken from MEs that belong to the above-specified Subnetwork, that are to be contained by the TP Pool.

6) Layered transmission parameters

   This parameter shall represent the common layers and transmission parameters the above-specified Contained TPs, or TPs contained in Contained GTPs, are required to have (e.g., ATM VP layer with prescribed traffic characteristics).

7) Description of use

   This attribute shall describe the specific use of the TP pool, in particular how its members are collected and administered.

8) Additional information.

   Refer to {Requirement I. 060}.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.2.2.3.5 Deletion of Termination Point Pool (TP Pool)s

{Requirement II. 266} The NML-EML Interface shall allow the NMS to delete a Termination Point Pool (TP Pool) given the NMS specified TP Pool name from the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.3.6 Modification of Termination Point Pool (TP Pool)s

{Requirement II. 267} The NML-EML interface shall allow the NMS to add or remove Termination Point (TP)s or Group Termination Point (GTP)s to or from a TP Pool given the NMS specified TP or GTP names and the TP Pool name.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.4 Topological Link Management

4.2.2.4.1 Creation of Topological Link (TL)s

{Requirement II. 168} The NML-EML Interface shall allow the NMS to create a Topological Link (TL) in the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.4.1.1 Topological Link (TL) Creation Data

{Requirement II. 169} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS create a Topological Link (TL)

1) User label

   Refer to {Requirement I. 060}.

2) User label uniqueness

   This parameter shall indicate to the EMS to check whether the user label is unique amongst the TLs within the EMS.

3) Owner

   Refer to {Requirement I. 060}.

4) Directionality

   This attribute shall represent the directionality of the TL (bidirectional or unidirectional).

5) aEnd Termination Point (TP)

   This parameter shall represent the name of the aEnd TP of the TL.

6) zEnd Termination Point (TP)

   This parameter shall represent the name of the zEnd TP of the TL.

7) Layer Rate

   This parameter shall represent the layer rate of the TL. Refer to {Requirement I. 022}.

8) Network Access Domain

   This attribute represents the Network Access Domain to which this TL has been assigned.

9) Alarm reporting

   This attribute shall indicate whether alarm reporting for the TL is enabled or disabled.

10) Alarm severity assignment profile

   This attribute shall represent the name of the Alarm Severity Assignment Profile (ASAP) that has been assigned to the TL.

11) Additional information.

   Refer to {Requirement I. 060}.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.2.2.4.2 Deletion of Topological Link (TL)s

{Requirement II. 170} The NML-EML Interface shall allow the NMS to delete a Topological Link (TL) given the NMS specified TL name from the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.5 Traffic Descriptor (TD) Management

4.2.2.5.1 Creation of Traffic Descriptor (TD)s

{Requirement II. 097} The NML-EML Interface shall allow the NMS to create a Traffic Descriptor (TD) in the EMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.2.5.1.1 Traffic Descriptor (TD) Creation Data

{Requirement II. 098} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS create a Traffic Descriptor (TD):

1) User label

   Refer to {Requirement I. 060}.

2) User label uniqueness

   This attribute shall indicate to the EMS that the value of the user label attribute must be unique amongst the TDs within the EMS.

3) Owner

   Refer to {Requirement I. 060}.

4) Service Category

   Refer to {Requirement I. 027}.

5) Conformance Profile.

   Refer to {Requirement I. 028}.

   - Standard traffic descriptor parameters for this Conformance profile. Refer to {Requirement I. 029}.

6) Traffic Parameters

   This attribute shall represent the parameters (sets of name/value pairs) used in the traffic descriptor. Refer to supporting document SD1-16_LayeredParameters.pdf for a list of possible traffic parameters.

7) Additional information

   Refer to {Requirement I. 060}.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.2.5.2 Deletion of Traffic Descriptor (TD)s

{Requirement II. 099} The NML-EML Interface shall allow the NMS to delete a Traffic Descriptor (TD) given the NMS specified TD name from the EMS.

The EMS shall refuse/fail this request if any CTPs are associated with this Traffic Descriptor.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.2.6 Transmission Descriptor (TMD) Management

4.2.2.6.1 Creation of Transmission Descriptor TMD)s

{Requirement II. 190} The NML-EML Interface shall allow the NMS to create a Transmission Descriptor (TMD) in the EMS; given the NMS specified data listed in {Requirement II. 191}.

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.2.6.1.1 Transmission Descriptor (TMD) Creation Data

{Requirement II. 191} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS create a Transmission Descriptor (TMD):

1) User label

   Refer to {Requirement I. 060}.

2) User label uniqueness

   This attribute shall indicate to the EMS that the value of the user label attribute must be unique amongst the TMDs within the EMS.

3) Owner

   Refer to {Requirement I. 060}.

4) Layered transmission parameters

   This attribute shall represent a list of transmission parameters which can be set at a specified layer on a Termination Point (TP) or Matrix Flow Domain (MFD) by association with this TMD. Specific parameters include, for example, frame format, line code, alarm reporting control (enable/ disable), TP service state (In Service, Out Of Service etc.).

5) Additional Object information

   This attribute shall represent any additional information parameters which can be set on the Termination Point (TP) or on a Matrix Flow Domain (MFD) by association with this TMD.

6) containing TMD name

   The attribute contains the name of another TMD which is considered to contain this TMD. The semantics of the containment is that the TMD to be created shall inherit the layered transmission parameters and additional Object information from the containing TMD.

7) External representation

   This attribute shall represent a reference to the external representation of the TMD (e.g., an XML file name). The content of this information is opaque at the NML-EML Interface and not utilized.

Note:

   It provides a useful means to guarantee the consistency between the TMD definition as used at the NML-EML interface (all information except the external representation reference) and the TMD definition as used outside the NML-EML interface. It is up to the NMS, however, to use externally ONLY the representation referenced by the TMD and to check the consistency.

8) Additional information

   Refer to {Requirement I. 060}.

Source: Version 3.5.

Mandatory/Optional: Mandatory.

4.2.2.6.2 Modification of Transmission Descriptor (TMD)s

{Requirement II. 353}

TMD 2

The NML-EML Interface shall allow the NMS to modify a Transmission Descriptor (TMD) in the EMS given the NMS specified data listed in the updated {Requirement II. 191} for the creation of a TMD.

When transmission parameters are modified, this will automatically modify the corresponding parameters in all associated TPs/MFDs on a best effort basis. TPs/MFDs whose parameters could not be modified shall be returned by the EMS.

Note: Only the modified parameters will be updated in the TPs/MFDs; i.e., the TMD parameter/value list may be inconsistent with the corresponding parameter/value list of the associated TPs/MFDs.

Source: Version 3.5.

Mandatory/Optional: Optiona.

4.2.2.6.3 Deletion of Transmission Descriptor (TMD)s

{Requirement II. 192}

The NML-EML Interface shall allow the NMS to delete a Transmission Descriptor (TMD) given the NMS specified TMD name from the EMS.

The EMS shall reject this request if any Termination Point (TP)s or Matrix Flow Domain (MFD)s are associated with this TMD.

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.2.7 Alarm Severity Assignment Profile (ASAP) Management

4.2.2.7.1 Creation of Alarm Severity Assignment Profile (ASAP)s

{Requirement II. 196}

The NML-EML Interface shall allow the NMS to create an Alarm Severity Assignment Profile (ASAP) in the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.7.1.1 Alarm Severity Assignment Profile (ASAP) Creation Data

{Requirement II. 197} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS create an Alarm Severity Assignment Profile (ASAP):

1) User label

   Refer to {Requirement I. 060}.

2) User label uniqueness

   This attribute shall indicate to the EMS that the value of the user label attribute must be unique amongst the ASAPs within the EMS.

3) Owner

   Refer to {Requirement I. 060}.

4) Alarm severity assignments

   This attribute shall represent the set of alarm severity assignments. Refer to {Requirement I. 082}.

5) Additional information

   Refer to {Requirement I. 060}.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.7.2 Modification of Alarm Severity Assignment Profile (ASAP)s

{Requirement II. 198} The NML-EML Interface shall allow the NMS to modify an Alarm Severity Assignment Profile (ASAP) in the EMS.

The EMS shall refuse/fail this request if the ASAP is fixed, i.e. it can neither be modified or deleted by the NMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.7.2.1 Alarm Severity Assignment Profile (ASAP) Modification Data

{Requirement II. 199} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS modify an Alarm Severity Assignment Profile (ASAP):

1) ASAP name

   This parameter shall represent the name of the ASAP that is to be modified.

2) User label

   Refer to {Requirement I. 060}.

3) User label uniqueness

   This attribute shall indicate to the EMS that the value of the user label attribute must be unique amongst the ASAPs within the EMS.

4) Owner

   Refer to {Requirement I. 060}.

5) Alarm severity assignments

   This attribute shall represent the new set of alarm severity assignments that are to be applied to the ASAP. Refer to {Requirement I. 082}.

6) Additional information

   Refer to {Requirement I. 060}.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.7.3 Deletion of Alarm Severity Assignment Profile (ASAP)s

{Requirement II. 200} The NML-EML Interface shall allow the NMS to delete an Alarm Severity Assignment Profile (ASAP) given the NMS specified ASAP name from the EMS.

The EMS shall refuse/fail this request if at least one object is pointing to this ASAP instance, or the ASAP cannot be deleted, i.e. neither can be modified nor deleted by NMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.7.4 Assignment of Alarm Severity Assignment Profile (ASAP)s

{Requirement II. 201} The NML-EML Interface shall allow the NMS to assign an Alarm Severity Assignment Profile (ASAP) to any of the objects identified in Table 4.6.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.7.4.1 Alarm Severity Assignment Profile (ASAP) Assignment Data

{Requirement II. 202} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS assign an Alarm Severity Assignment Profile (ASAP) to an object:

1) ASAP name

   This parameter shall represent the name of the ASAP that is to be assigned.

2) Object name

   This parameter shall represent the name of the object to which the ASAP is to be assigned. Refer to Table 4.6 for a list of the objects.

3) Layer rate

   This parameter shall represent the layer rate to which the ASAP is applicable. This shall be need when the addressed object is a Termination Point (TP).

4) .Additional information

   Refer to {Requirement I. 060}.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.2.2.7.5 De-assignment of Alarm Severity Assignment Profile (ASAP)s

{Requirement II. 203} The NML-EML Interface shall allow the NMS to de-assign an Alarm Severity Assignment Profile (ASAP) to any of the objects identified in Table 4.6.

The EMS shall refuse/fail this request if the ASAP is assigned in a fixed way to the object.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.7.5.1 Alarm Severity Assignment Profile (ASAP) De-assignment Data

{Requirement II. 204} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS de-assign an Alarm Severity Assignment Profile (ASAP) to an object:

1) Object name

   This parameter shall represent the name of the object to which the ASAP is to be de-assigned. Refer to Table 4.6 for a list of the objects.

2) Layer rate

   This parameter shall represent the layer rate to which the ASAP is applicable. This shall be need when the addressed object is a Termination Point (TP).

3) .Additional information

   Refer to {Requirement I. 060}.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.2.2.8 Threshold Crossing Alert (TCA) Parameter Profile Management

4.2.2.8.1 Creation of Threshold Crossing Alert (TCA) Parameter Profiles

{Requirement II. 236} The NML-EML Interface shall allow the NMS to create an Threshold Crossing Alert (TCA) Parameter Profile in the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.8.1.1 Threshold Crossing Alert (TCA) Parameter Profile Creation Data

{Requirement II. 237} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS create an Threshold Crossing Alert (TCA) Parameter Profile:

1) ME name

   This parameter shall represent the name of the Managed Element (ME) in which the profile is to be created.

2) User label

   Refer to {Requirement I. 060}.

3) User label uniqueness.

   This parameter shall indicate to the EMS that the value of the user label attribute must be unique amongst the TCA Parameter Profiles within the EMS.

4) Owner

   Refer to {Requirement I. 060}.

5) Layer rate

   This parameter shall represent the layer to which the TCA parameter threshold values apply. (Refer to {Requirement I. 022}).

6) TCA Parameter assignments

   This parameter shall represent the set of TCA parameter assignments. Refer to {Requirement I. 035}.

7) Additional information

   Refer to {Requirement I. 060}.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.8.2 Modification of Threshold Crossing Alert (TCA) Parameter Profiles

{Requirement II. 238} The NML-EML Interface shall allow the NMS to modify an Threshold Crossing Alert (TCA) Parameter Profile in the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.8.2.1 Threshold Crossing Alert TCA) Parameter Profile Modification Data

{Requirement II. 239} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS modify an Threshold Crossing Alert (TCA) Parameter Profile:

1) TCA Parameter Profile name

   This parameter shall represent the name of the TCA Parameter Profile that is to be modified.

2) TCA Parameter assignments

   This parameter shall represent the set of TCA parameter assignments. Refer to {Requirement I. 035}.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.2.8.3 Deletion of Threshold Crossing Alert (TCA) Parameter Profiles

{Requirement II. 240} The NML-EML Interface shall allow the NMS to delete an Threshold Crossing Alert (TCA) Parameter Profile given the NMS specified TCA Parameter Profile name from the EMS.

The EMS shall refuse/fail this request if at least one object is pointing to this TCA Parameter Profile instance.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.3 Call Management

4.2.3.1 Creation of Calls

{Requirement II. 392}

{Requirement II. Call 100}

The NML-EML Interface shall allow the NMS to establish a Call within an NMS specified top-level MultiLayer Routing Area (MLRA) name or MultiLayer Subnetwork (MLSN) name given the NMS specified data listed in {Requirement II. 393}.

Note: At least one end point of the Call has to be within the EMS domain. The other end points may be "off-network" Termination Point (TP)s (i.e., addressed by a remote address) or an Subnetwork Point (SNP), Subnetwork Point Pool (SNPP) or Transport Network Assigned (TNA) address not directly managed by the local EMS.

If the successful establishment of the Call provides connectivity in a different client layer (e.g., Call is on the SDH server layer and client layer is Ethernet) the EMS is required to automatically create a Topological Link (TL) between the end points in the client layer.

On success, the EMS shall return the created Call and Connections and/or Subnetwork Connection (SNC)s, the current diversity information and also the TP data that was set if any.

Source: Version 3.5.

ITU-T Recommendation G.7718 (2/2005) Requirement R 48 and R 49

Mandatory/Optional: Optional.

4.2.3.1.1 Call Creation Data

{Requirement II. 393}

{Requirement II. Call 101}

The NML-EML Interface shall allow the NMS to specify the following parameters when it requests an EMS to establish a Call, including data that are related to the Connections to be established for the Call:

1) Name

   This parameter defines the identifier of the new Call which will be used over the interface. The EMS has to make sure that the name of the Call is unique within the EMS domain. If no name is provided by the NMS, the EMS has to define a unique name.

   Note: The name of the Call is not changeable after the Call is created.

   Refer to {Requirement I. 060}.

2) User label

   Refer to {Requirement I. 060}.

3) User label uniqueness

   This parameter shall indicate to the EMS that the value of the User Label attribute must be unique amongst the Calls within the EMS.

4) Owner

   Refer to {Requirement I. 060}.

5) Network Access Domain

   Refer to {Requirement I. 134}.

6) Operation effort

   This parameter indicates if the operation is atomic or best-effort.

   Atomic means that the operation is only successful when all requested (top-level) Connections and/or SNCs could be created. If this is not possible the EMS has to roll-back.

   The atomic value is mandatory for an EMS that supports restoration (i.e., in a Control Plane environment).

   Best-effort means that the operation is also successful when not all (or even no) requested top-level Connections or SNCs could be created.

7) A-end TP

This parameter provides the A-end TP of the Call. It may be of the following name/id type (Note: In a Control Plane environment, the type is independent of the type of the Z-end TP):

- SNPP ID
  which may be resolved to the SNP level
- SNP
- CPTP/FTP/CTP/PTP Name
- TNA Name
  which identifies either a TNA or a Group-TNA Name
- Null name.

8) Z-end TP

This parameter provides the Z-end TP of the Call. It may be of the following name/id type (Note: In a Control Plane environment, the type is independent of the type of the A-end TP):

- SNPP ID
  which may be resolved to the SNP level
- SNP
- CPTP/FTP/CTP/PTP Name
- TNA Name
  which identifies either a TNA or a Group-TNA Name
- Null name.

9) Call Parameter (best effort)

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

- Severely Degraded Threshold
- Degraded Threshold.

10) Connection-related data - *OPTIONAL*

This parameter allows the NMS to specify the data to be used for creating the (top-level) Connections.

For Connection releated data refer to {Requirement II. 398} and for Subnetwork Connection releated data refer to {Requirement II. 084}.

11) TP-related data - *OPTIONAL*

This parameter shall identify a list of TPs that are to be modified as part of the establish request. Each item in the list shall contain the following:

- TP name (generic TP names are not allowed)
- Transmission parameters (one list per layer rate)
- Ingress transmission descriptor name
- Egress transmission descriptor name
- Mapping mode.

Note:

Parameters 12) - 17) are diversity-related.

12) Co-routing level of effort

Refer to {Requirement I. 134}.

13) Link Diversity level of effort

Refer to {Requirement I. 134}.

14) Node Diversity level of effort

Refer to {Requirement I. 134}.

15) Link SRG Type

Refer to {Requirement I. 134}.

16) Node SRG Type

Refer to {Requirement I. 134}.

17) Number of Route Groups

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

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

Value “0” means that the NMS doesn't care about the number of Route Groups.

Value “NA” means that this parameter can be ignored.

18) Additional information

Refer to {Requirement I. 060}.

Source: Version 3.5.

ITU-T Recommendation G.7718 (2/2005) Requirement R 48 and R 49

Mandatory/Optional: Mandatory.

4.2.3.2 Call Deletion

{Requirement II. 394}

{Requirement II. Call 102}

The NML-EML Interface shall allow the NMS to release a Call within an NMS specified MultiLayer Routing Area (MLRA) name or MultiLayer Subnetwork (MLSN) name given the NMS specified Call name.

The request shall also delete all (top-level) Connections and Subnetwork Connection (SNC)s supporting the Call.

If the Call has an associated Topological Link (TL) in the client technology, this link will also be deleted.

The NMS shall be able to configure transmission parameters (TP data) on associated TPs by identifying for each TP to be modified:

- TP name (generic TP names are not allowed)
- Transmission parameters (one list per layer rate)
- Mapping mode.

On success, the EMS shall return the TP data that was set if any.

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.3.3 Call Modification

{Requirement II. 395}

{Requirement II. Call 103}

The NML-EML Interface shall allow the NMS to modify a Call within an NMS specified MultiLayer Routing Area (MLRA) name or MultiLayer Subnetwork (MLSN) name given the NMS specified data listed in {Requirement II. 396}.

Note: Call bandwidth modification is not covered by this requirement.

Source: Version 3.5.

ITU-T Recommendation G.7718 (2/2005) Requirement R 48 and R 49

Mandatory/Optional: Optional.

4.2.3.3.1 Call Modification Data

{Requirement II. 396}

{Requirement II. Call 104}

The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS modifies a Call:

1) User label

   Refer to {Requirement I. 060}.

2) User label uniqueness

   This parameter shall indicate to the EMS that the value of the new user label attribute must be unique amongst the Calls within the EMS.

3) Owner

   Refer to {Requirement I. 060}.

4) Network access domain

   This parameter shall indicate the new Network Access Domain to which this Call will be assigned to.

5) Additional information

   Refer to {Requirement I. 060}.

Note:

This request is best effort; except the change of the User Label with Force uniqueness.

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.3.3.2 Addition of Connections

{Requirement II. 397}

{Requirement II. Call 105}

The NML-EML Interface shall allow the NMS to add one or more Connections / Subnetwork Connections to a Call given an NMS specified Call name and the data specified in {Requirement II. 398}.

On success, the EMS shall return the created Connections/SNCs, the current diversity information and also the TP data that was set if any.

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.3.3.2.1 Connection Addition Data

{Requirement II. 398}

{Requirement II. Call 106}

The NML-EML Interface shall allow the NMS to specify the following (top-level) Connection-related / Subnetwork Connection (SNC) - related data when it requests an EMS to add one or more (top-level) Connections / SNCs to an existing Call:

1) Name

   Refer to {Requirement I. 060}.

2) User Label

   Refer to {Requirement I. 060}.

3) Force uniqueness

   This parameter shall indicate to the EMS that the value of the new user label attribute must be unique amongst the (top-level) Connections within the EMS.

4) Owner

   Refer to {Requirement I. 060}.

5) Layer Rate

   Refer to {Requirement I. 131}.

6) A-end ID

   The attribute specifies the A-end of the connection in terms of an SNP or SNPP. Refer to {Requirement I. 131}.

7) Z-end ID

   The attribute specifies the Z-end of the connection in terms of an SNP or SNPP. Refer to {Requirement I. 131}.

8) A-end TP

   Refer to {Requirement I. 131}.

9) Z-end TP

   Refer to {Requirement I. 131}.

10) A-end TNA Name

   Refer to {Requirement I. 131}.

11) Z-end TNA Name

Refer to {Requirement I. 131}.

12) Directionality

Refer to {Requirement I. 131}.

13) Reroute Allowed

Refer to {Requirement I. 131}.

14) Static Protection Level

Refer to {Requirement I. 131}.

15) Maximum Cost

Refer to {Requirement I. 131}.

16) Protection Effort

Refer to {Requirement I. 131}.

17) Routing Constraint Effort

Refer to {Requirement I. 131}.

Note:

For the top level Connection only simple will be used in this version.

Add-drop-A and Add-drop-Z are only for subordinate Connections in this version.

18) Connection Setup Type

Refer to {Requirement I. 131}.

19) Revertive

Refer to {Requirement I. 131}.

20) Network Access Domain

Refer to {Requirement I. 131}.

21) Priority

Refer to {Requirement I. 131}.

22) Alarm Reporting Indication

Refer to {Requirement I. 131}.

23) Alarm Severity Assignment Profile Pointer

Refer to {Requirement I. 131}.

24) Route Group Label

Refer to {Requirement I. 131}.

25) TP-related data - *OPTIONAL*

This parameter identifies a list of TPs that are to be modified as part of the request. Each item in the list shall contain the following:

- TP name (generic TP names are not allowed)
- Transmission parameters (one list per layer rate)
- Mapping mode.

26) Additional information

Refer to {Requirement I. 060}.

27) Routing Constraints

The NMS may provide routing constraint information as part of the creation and activation of a Connection. The routing constraint information in a specific Connection create shall include:

EITHER:

Resources that must not be part of the route chosen as a result of the request (i.e. excluded resources). The resources may be:

- MultiLayer Routing Area (MLRA)
- MultiLayer Subnetwork Point Pool Link (MLSNPPLink)
- Subnetwork Point Pool (SNPP)
- Subnetwork Point (SNP)
- Managed Element (ME)
- Termination Point (TP)
- Subnetwork Connection (SNC)
- Topological Link (TL)
- Group Termination Point (GTP)

OR:

Resources that must form part of the route chosen as a result of the request (i.e. included resources). The resources may be:

- MultiLayer Routing Area (MLRA)
- MultiLayer Subnetwork Point Pool Link (MLSNPPLink)
- Subnetwork Point Pool (SNPP)
- Subnetwork Point (SNP)
- Cross-Connect (XC)
- Termination Point (TP)
- Managed Element (ME)
- Topological Link (TL)
- Group Termination Point (GTP)

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.3.3.3 Removal of Connections

{Requirement II. 399}

{Requirement II. Call 107}

The NML-EML Interface shall allow the NMS to remove one or more top-level Connection / Subnetwork Connection (SNC) from a Call given an NMS specified Call name and top-level Connection name(s) or Subnetwork Connection name(s).

The NMS shall be able to configure transmission parameters (TP data) on associated TPs by identifying for each TP to be modified:

- TP name (generic TP names are not allowed)
- Transmission parameters (one list per layer rate)
- Mapping mode.

On success, the EMS shall return the current diversity information and also the TP data that was set if any.

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.3.3.4 Diversity and Co-routing Management

{Requirement II. 400}

{Requirement II. L-II.26}

The NML-EML interface shall allow the NMS to associate an existing Connection to a Route Group within the existing Call.

Note:

This requirement applies to support the need to add diversity characteristics to an existing Connection within a Call, i.e., changing the Connection's Route Group association from none to a particular Route Group.

The Route Group can be a new Route Group.

Route separacy of the Connections of a Call is indicated by using different “route group” values for the Connections. In MTNM release 3.5, Separacy can be Best Effort or Mandatory.

Co-routing of Connections of a Call is indicated by using the same "route group" value for the Connections. In MTNM release 3.5, Co-routing is Best Effort only. Therefore, in MTNM release 3.5, even though two Connections have the same Route Group value, it doesn't guarantee they are really co-routed.

Source: Version 3.5.

at&t, Deutsche Telekom, Verizon

Mandatory/Optional: Optional.

{Requirement II. 401}

{Requirement II. L-II.29}

The NML-EML interface shall support the ability for the NMS to request diversity amongst route groups within a Call with a specified level-of-effort, namely, best effort or mandatory.

In the case of best effort, if 100% diverse route cannot be found, then the EMS shall return the maximally diverse route and the list of compromises (such as resources being shared) shall be reported in the return notification.

Source: Version 3.5.

at&t, Deutsche Telekom, Verizon

Mandatory/Optional: Optional.

{Requirement II. 402}

{Requirement II. Call 108}

The NML-EML Interface shall allow the NMS to set/modify diversity and co-routing parameters of a Call given the NMS specified data listed in {Requirement II. 403}.

The EMS shall return SC and AVC notifications for the Call and Connections to reflect the changes (indicating the appropriate states of the Call/Connections and the current diversity information).

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 403}

{Requirement II. Call 109}

The NML-EML Interface shall allow the NMS to specify the following parameters when it requests an EMS to set/modify diversity and co-routing parameters of a Call:

1) Call name

   This parameter defines the identifier of the Call whose diversity and co-routing parameters shall be changed.

2) Diversity parameters (between route groups):

   - Link Diversity level of effort
     Allowed values are: None, Best Effort, Mandatory.
   - Link Diversity SRG Type
     (may be null indicating basic MLSNPP Link diversity)
   - Node Diversity level of effort
     Allowed values are: None, Best Effort, Mandatory.
   - Node Diversity SRG Type
     (may be null indicating basic Routing Node diversity)
   - Number of Route Groups

3) Co-routing parameters (within route groups):

   - Co-routing level of effort
     Allowed values are: None, Best Effort, Mandatory.

4) List of Connections

   Each Connection has a Route Group Label.

5) Connection Route Rearrangement Allowed

This parameter indicates whether rerouting is allowed or not when the network endeavours to satisfy the request.

6) Additional information

Refer to {Requirement I. 060}.

Note:

If any of the following is true:

- "Co-routing level of effort" is not "None"
- "Link Diversity level of effort" is not "None"
- "Node Diversity level of effort" is not "None" and the "Number of Route Groups" is "NA"

then each Connection of the Call must be provided with a "Route Group Label" specified (i.e. of non-Null value).

Source: Version 3.5.

at&t, Deutsche Telekom, Verizon

Mandatory/Optional: Optional.

4.2.4 Connection Management

{Requirement II. 404}

{Requirement II.13}

For Soft Permanent Connection setup requests, the NML-EML interface shall provide the capability to specify the strict routing constraints, in terms of a complete list of nodes.

Source: Version 3.5.

ITU-T Recommendation G.7718 (2/2005) Requirement R 63

Mandatory/Optional: Optional.

{Requirement II. 405}

{Requirement II.16}

The NML-EML interface shall provide the ability to distinguish between the Permanent Connections (PC), Soft Permanent Connections (SPC), and Switched Connections (SC).

Source: Version 3.5.

ITU-T Recommendation G.7718 (2/2005) Requirement R 50, but does not mention Permanent Connections

Mandatory/Optional: Optional.

{Requirement II. 406}

{Requirement L-II.32}

The NML-EML interface shall support the ability to retrieve the relationships between Connections, Route Groups, and Calls.

Note: This requirement is supported indirectly by Call and Connection retrievals, which allow the NMS to determine the relationship.

Source: Version 3.5.

AT&T, Deutsche Telekom, Verizon

Mandatory/Optional: Optional.

{Requirement II. 407}

{Requirement II.43}

The NML-EML interface shall provide the ability to query the Control Plane for the status of a Soft Permanent Connection (SPC).

Source: Version 3.5.

ITU-T Recommendation G.7718 (2/2005) Requirement R 62 - e

Mandatory/Optional: Optional.

{Requirement II. 408}

{Requirement II.44}

The NML-EML interface shall provide the ability to query the Control Plane for the Connection attributes of a Soft Permanent Connection.

Source: Version 3.5.

ITU-T Recommendation G.7718 (2/2005) Requirement R 62 - f

Mandatory/Optional: Optional.

{Requirement II. 409}

{Requirement II.50}

The NML-EML interface shall provide the ability to query the Control Plane for the status of a Switched Connection.

The EMS shall support the capabilities to autonomously notify addition or removal of a switched connection, connection route, connection attributes, and connection status. The NMS shall be able to re-route a switched connection and tier-down a switch connection.

Source: Version 3.5.

ITU-T Recommendation G.7718 (2/2005) Requirement R 62 - d

Mandatory/Optional: Optional.

{Requirement II. 410}

{Requirement II.51}

The NML-EML interface shall provide the ability to query the Control Plane for the Connection attributes of a Switched Connection (SC).

The EMS shall support the capabilities to autonomously notify addition or removal of an SC, Connection route, Connection attributes, and Connection status. The NMS shall be able to re-route an SC and tier-down an SC.

Source: Version 3.5.

ITU-T Recommendation G.7718 (2/2005) Requirement R 62 - e

Mandatory/Optional: Optional.

{Requirement II. 411}

{Requirement II.57}

The NML-EML interface shall support the ability to manage Calls with 1 or more Control Plane Connections. For each Call, it shall be possible to retrieve a Control Plane Connection for the following information:

- Call id
- Information on individual Control Plane Connections.

Source: Version 3.5.

ITU-T Recommendation G.7718 (2/2005) Requirement R 48 and R 49

Mandatory/Optional: Optional.

4.2.5 Subnetwork Connection (SNC) Management (Simple and Point-to-Multi-Point Connections)

4.2.5.1 Static Route Selection

{Requirement II. 080} The NML-EML Interface shall allow the NMS to create a Subnetwork Connection (SNC) with a particular route. Refer to {Requirement I. 018}

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.5.2 Creation of Subnetwork Connection (SNC)s

{Requirement II. 082} The NML-EML Interface shall allow the NMS to create a planned Subnetwork Connection (SNC) in an EMS given the NMS specified data listed in {Requirement II. 084}.

NMS specified SNC data lAs a result of the (successful) completion of this request, the EMS shall create an object representing the SNC, but shall not attempt to establish (on NEs) any of the cross-connections associated with the SNC. The successfully created SNC will be in pending state.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 083} The NML-EML Interface shall allow the NMS to create point to multi-point configurations.

Each leg of a point to multi-point configuration shall be represented by its own Subnetwork Connection (SNC) which allows for individual management of each leg. (e.g. add or remove). All SNCs of a point to multi-point configuration share the same aEnd Termination Point (TP).

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 153} The NML-EML Interface shall allow the NMS to create a Subnetwork Connection (SNC) without specifying the ending Termination Point (TP) instances, but only the ending Managed Element (ME) instances or containing TP instances.

The end TP instances will be chosen by the EMS, at SNC creation time. These ME or TP instances are therefore identified by the name value “EMS_assigned”.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.5.2.1 Subnetwork Connection (SNC) Creation Data

{Requirement II. 084} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS create a Subnetwork Connection (SNC):

1) User label

   Refer to {Requirement I. 060}.

2) User label uniqueness

   This parameter shall indicate to the EMS that the value of the user label attribute must be unique amongst the SNCs within the EMS.

3) Owner

   Refer to {Requirement I. 060}.

4) Directionality

   This parameter shall represent the directionality of the SNC (bidirectional or unidirectional). The EMS shall set the directionality to unidirectional if either an ingress or egress Traffic Descriptor is zero for an ATM Subnetwork Connection.

5) Static protection level

   This parameter shall indicate the degree of internal resilience/protection of the SNC e.g., to indicate whether the Subnetwork Connection should be Protected, Preemtible, or Unprotected. The EMS will be required to create a SNC with the specified protection level. Refer to the supporting document SD1-36_SNCTypes.pdf for details on the static protection level parameter.

6) Protection effort

   This parameter shall indicate whether the resilience requested must be achieved of not. Refer to the supporting document SD1-36_SNCTypes.pdf for details on the protection effort parameter.

7) SNC Type

   Refer to {Requirement I. 040}.

8) Layer Rate

   Refer to {Requirement I. 022}.

9) Routing constraint data

   Refer to {Requirement I. 018}.

10) Complete route

    This parameter shall indicate whether the routing constraint data attribute specifies a complete route.

11) Network routed

This parameter shall indicates whether the route shall be computed by the network, by the EMS or either.

12) Reroute allowed

This parameter shall indicate if the SNC may be rerouted.

13) Network reroute

This parameter shall indicate if the reroute (if allowed) shall be computed by the network, by the EMS, or by either.

14) Revertive

This parameter shall indicate whether the SNC shall always attempt to return to it's intended Route.

15) Priority

This parameter shall represent the priority of the SNC (i.e. highest (0) to lowest).

16) Exclusive intended route

This parameter shall indicate if the intended route that is provided as part of this creation data is an exclusive route.

17) aEnd TP(s)

This parameter represents a list of the names of the aEnd Termination Point (TP) (s) that shall be the aEnd points of the SNC. The names may be either specific or generic. The TPs may of the following type:

- Connection Termination Point (CTP)
- Group Termination Point (GTP)
- Floating Termination Point (FTP)

18) zEnd TP(s)

This parameter represents a list of the names of the zEnd Termination Point (TP) (s) that shall be the zEnd points of the SNC. The names may be either specific or generic. The TPs may of the following type:

- Connection Termination Point (CTP)
- Group Termination Point (GTP)
- Floating Termination Point (FTP)

19) Additional information

Refer to {Requirement I. 060}.

20) Bundled SNC

This parameter shall indicate if the SNC to be created is a bundled SNC

21) GTP deletion

The parameter shall only be used when creating bundled SNCs. It shall indicate that the EMS has to delete all the interior GTPs supporting the bundled SNC when the SNC is deleted.

22) Alarm reporting

This parameter shall indicate whether alarm reporting for the SNC is to be enabled or disabled.

23) Alarm severity assignment profile

This attribute shall represent the name of the Alarm Severity Assignment Profile (ASAP) that is to be assigned to the SNC.

24) aEnd point role

This attribute shall represent the role of the aEnd Termination Point (TP)s of the SNC. Refer to {Requirement I. 092}.

25) zEnd point role

This attribute shall represent the role of the zEnd Termination Point (TP)s of the SNC. Refer to {Requirement I. 092}.

26) Network Access Domain

This attribute represents the Network Access Domain (NAD) to which this SNC has been assigned.

27) Grade of impact

This parameter shall indicate the degree to which the creation of the SNC may impact the Subnetwork. Refer to the supporting document SD1-36_SNCTypes.pdf for details on the grade of impact parameter.

28) EMS freedom level

This parameter shall indicate the level of freedom given to the EMS in determining the effect on the Subnetwork when performing SNC operations.

29) Route Group Label

Refer to {Requirement I. 015}.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.5.3 Validation of Subnetwork Connection (SNC)s

{Requirement II. 085}

The NML-EML Interface shall allow an NMS to check if a valid Subnetwork Connection (SNC) can be created given the NMS specified SNC creation data specified in {Requirement II. 084}.

The NMS may also request the EMS to consider the SNC resources to determine whether activation of the SNC, if applied, will succeed. The validity check will take into consideration the subnetwork's equipment capabilities and the current resource states.

This is a best effort guarantee as the resources may not be available when the NMS tries to actually activate the SNC.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.5.4 Activation of Subnetwork Connection (SNC)s

{Requirement II. 086}

The NML-EML Interface shall allow the NMS to request that an EMS activate an NMS specified Subnetwork Connection (SNC) given the NMS specified SNC data listed in {Requirement II. 087}

As a result of the (successful) completion of this request, the EMS will have issued the required commands such that all Cross-Connect (XC)s associated with (comprising) the SNC are in place (i.e., are provisioned on NE(s)). The state of the successfully activated SNC will transition to active.

If the NMS specified SNC has more than one route, this operation unlocks all the routes, delegating the EMS and/or the network (e.g. restoration process) the actual activation of the appropriate route.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.5.4.1 Subnetwork Connection (SNC) Activation Data

{Requirement II. 087} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS activate a Subnetwork Connection (SNC).

1) SNC name

   This parameter shall indicate the name of the previously created SNC that is to be activated.

2) Grade of impact

   This parameter shall indicate the degree to which the activation of the SNC may impact the Subnetwork. Refer to the supporting document SD1-36_SNCTypes.pdf for details on the grade of impact parameter.

3) EMS freedom level

   This parameter shall indicate the level of freedom given to the EMS in determining the effect on the Subnetwork when performing SNC operations.

4) Termination Point (TP)s to modify

   This parameter shall identify a list of TPs that are to be modified as part of the activation request. Each item in the list shall contain the following:

   - TP name (generic TP names are not allowed)
   - Transmission parameters
   - Ingress traffic descriptor name
   - Egress traffic descriptor name
   - Mapping mode

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.5.5 Creation and Activation of Subnetwork Connection (SNC)s

{Requirement II. 088} The NML-EML Interface shall allow the NMS to request that a Subnetwork Connection (SNC) be both created and activated (as a result of a single request from the NMS) given the NMS specified SNC data listed in {Requirement II. 084}.

As a result of (successful) completion of this request, the EMS shall create an object representing the SNC and shall have issued required commands such that all cross-connections associated with (comprising) the SNC are in place (i.e., are provisioned on NE(s)). The state of the successfully created and activated SNC will transition to active.

If the NMS specified SNC has more than one route, this operation unlocks all the routes, delegating the EMS and/or the network (e.g. restoration process) the actual activation of the appropriate route.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.5.5.1 Subnetwork Connection (SNC) Creation and Activation Data

{Requirement II. 089} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS create and activate a Subnetwork Connection (SNC):

1) The parameters required to create an SNC (Refer to {Requirement II. 084}).
2) Termination Point (TP)s to modify

   This parameter shall identify a list of TPs that are to be modified as part of the activation request. Each item in the list shall contain the following:

   - TP name (generic TP names are not allowed)
   - Transmission parameters
   - Ingress transmission descriptor name
   - Egress transmission descriptor name
   - Mapping mode

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.2.5.6 Addition of a Route to a Subnetwork Connection (SNC) or a Connection

{Requirement II. 241} The NML-EML interface shall allow the NMS to add a protection route to an NMS specified Subnetwork Connection (SNC) or Connection in an EMS given the NMS specified data listed in {Requirement II. 242}.

As a result of (successful) completion of this request, the EMS shall add the new route to the SNC or Connection, but shall not attempt to establish (on NEs) any cross connections as side effect of this operation, because the route is created in locked state.

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.5.6.1 Subnetwork Connection (SNC) Route Addition Data

{Requirement II. 242} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS add a route to a Subnetwork Connection (SNC) or Connection.

1) SNC or Connection name

   This parameter shall represent the name of the SNC or Connection to which the Route is to be assigned that is to be activated.

2) Grade of impact

   This parameter shall indicate the degree to which the activation of the added route may impact the Subnetwork. (Refer to the supporting document SD1-36_SNCTypes.pdf for details on the grade of impact parameter.

3) EMS freedom level

   This parameter shall indicate the level of freedom given to the EMS in determining the effect on the Subnetwork when performing SNC or Connection operations.

4) Intended

   This parameter shall indicate if this route is the new intended route for the SNC or Connection, as opposed to a back-up route.

5) Exclusive

   This parameter shall indicate if this route is an exclusive route for this SNC or Connection. An exclusive route is a route that no other SNCs or Connections can share any of its XCs or CTPs.

6) Routing constraint data

   Refer to {Requirement I. 018}.

7) Complete route

   This parameter shall indicate whether the routing constraint data attribute specifies a complete route.

8) Additional information

   Refer to {Requirement I. 060}.

Source: Version 3.5.

Mandatory/Optional: Mandatory.

4.2.5.7 Removal of a Route from a Subnetwork Connection (SNC)

{Requirement II. 243} The NML-EML Interface shall allow the NMS to remove a protection route from an NMS specified Subnetwork Connection (SNC) given the NMS specified SNC data listed in {Requirement II. 244}.

As a result of (successful) completion of this request, the EMS shall delete the protection route of specified SNC. The specified route must not be in the unlocked state, and must not be the intended route. Of course it is possible to delete a locked backup route which is "in use" by other SNC route, because this operation has no side effect on routes of any other SNCs, even if sharing Cross-Connect (XC)s/Connection Termination Point (CTP)s.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.5.7.1 Subnetwork Connection (SNC) Route Removal Data

{Requirement II. 244} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS remove a route from a Subnetwork Connection (SNC).

1) SNC name

   This parameter shall represent the name of the SNC from which the Route is to be removed.

2) Route identifier

   This parameter shall represent the identifier of the route that is to be removed.

3) EMS freedom level

   This parameter shall indicate the level of freedom given to the EMS in determining the effect on the Subnetwork when performing SNC operations.

4) Additional information

   Refer to {Requirement I. 060}.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.2.5.8 Creation-Modification of a Subnetwork Connection (SNC)

{Requirement II. 245} The NML-EML Interface shall allow the NMS to modify (a route of) an NMS specified Subnetwork Connection (SNC) in the EMS given the NMS specified SNC data listed in {Requirement II. 246}.

As a result of (successful) completion of this request, the EMS shall update the route description, but shall not attempt to establish (in the network) any of the Cross-Connect (XC)s associated with the modified route. The state of the successfully modified route will be locked.

If the EMS can preserve the name of the NMS specified SNC then the route is modified. If the EMS cannot preserve the name of the NMS specified SNC then the EMS shall create a new pending SNC from an existing pending or active SNC.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.5.8.1 Subnetwork Connection (SNC) Modification Data

{Requirement II. 246} The NML-EML Interface shall allow the NMS to specify in addition to the creation parameters specified in {Requirement II. 084}, the following parameters when it requests that an EMS modify a Subnetwork Connection (SNC):

1) SNC name

   This parameter shall represent the name of the SNC to be modified.

2) Route identifier

   This parameter shall represent the identifier of the route that is to be modified. If not specified, the intended Route shall be used

3) Modification type

   This parameter shall indicate the type of the of modification to be performed (i.e. rerouting or add/remove protection)

4) Retain SNC

   This parameter shall indicate if when modifying an SNC whether the original SNC shall be deleted or put into the pending state.

5) Modify server layers

   This parameter shall indicate whether the EMS is allowed to modify the server layers to fulfil the protection constraints identified by this request.

6) Added or new route

   This parameter shall represent (depending on the modification type), the route of a new protection leg or the whole SNC. When it describes a segment to be added, either the SNCP cross-connects or the switch TPs that will be changed in the segment may be specified by the NMS. The EMS then chooses the missing segments. Alternatively, the NMS may specify the full route.

7) Removed route

   This parameter shall represent the protection leg that is to be removed from the SNC. Either the last cross-connects (that contain the SNCP) are specified by the NMS or the full route may be specified. This parameter can be used in conjunction with Added Or New Route only to reroute a segment

8) Termination Point (TP)s to modify

   This parameter shall identify a list of TPs that are to be modified as part of the modification request. Each item in the list shall contain the following:

   - TP name (generic TP names are not allowed)
   - Transmission parameters
   - Ingress transmission descriptor name
   - Egress transmission descriptor name
   - Mapping mode

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.2.5.9 Modification of a Subnetwork Connection (SNC)

{Requirement II. 257} The NML-EML Interface shall allow the NMS to modify an NMS specified Subnetwork Connection (SNC) in the EMS given the NMS specified SNC data listed in {Requirement II. 246}.

As a result of (successful) completion of this request, the EMS shall modify and activate the SNC.

The end result of this operation is equivalent to an NMS using the create-modify operation followed by the swap operation.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.5.10 Swap a Subnetwork Connection (SNC)

{Requirement II. 258} The NML-EML Interface shall allow the NMS to activate a pending Subnetwork Connection (SNC) and deactivate an active SNC in a single atomic operation given the NMS specified SNC data listed in {Requirement II. 259}.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.5.10.1 Subnetwork Connection (SNC) Swap Data

{Requirement II. 259} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS swap two Subnetwork Connection (SNC)s.

1) SNC name to be deactivated

   This parameter shall indicate the name of the active SNC that is to be deactivated.

2) SNC name to be activated

   This parameter shall indicate the name of the previously created SNC that is to be activated.

3) Grade of impact

   This parameter shall indicate the degree to which the activation of the SNC may impact the Subnetwork. Refer to the supporting document SD1-36_SNCTypes.pdf for details on the grade of impact parameter.

4) EMS freedom level

   This parameter shall indicate the level of freedom given to the EMS in determining the effect on the Subnetwork when performing this operation.

5) TPs to modify

   This parameter shall identify a list of Termination Point (TP)s that are to be modified as part of the swap request. Each item in the list shall contain the following:

   - TP name (generic TP names are not allowed)
   - Transmission parameters
   - Ingress traffic descriptor name
   - Egress traffic descriptor name
   - Mapping mode

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.5.11 Switch the Route of a Subnetwork Connection (SNC)

{Requirement II. 247} The NML-EML Interface shall allow the NMS to switch a route of a NMS specified Subnetwork Connection (SNC) given the NMS specified SNC data listed in {Requirement II. 248}.

As a result of (successful) completion of this request, the EMS shall activate in the network the input route, and deactivate the currently active route, plus all the partial routes, if any. The operation is refused if performed on a pending SNC, or on a locked route. The operation does not affect the administrative state of any route. The restoration process is still allowed to re-route again, e.g. in case of failures.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.5.11.1 Subnetwork Connection (SNC) Route Switch Data

{Requirement II. 248} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS switched the route of a Subnetwork Connection (SNC).

1) SNC name

   This parameter shall represent the name of the SNC for which the Route is to be switched.

2) Route identifier

   This parameter shall represent the identifier of the route that is to be switched.

3) Grade of impact

   This parameter shall indicate the degree to which the route switch of the SNC may impact the Subnetwork. Refer to the supporting document SD1-36_SNCTypes.pdf for details on the grade of impact parameter.

4) EMS freedom level

   This parameter shall indicate the level of freedom given to the EMS in determining the effect on the Subnetwork when performing SNC operations.

5) Termination Point (TP)s to modify

   This parameter shall identify a list of TPs that are to be modified as part of the switch route request. Each item in the list shall contain the following:

   - TP name (generic TP names are not allowed)
   - Transmission parameters
   - Ingress transmission descriptor name
   - Egress transmission descriptor name
   - Mapping mode

6) Additional information

   Refer to {Requirement I. 060}.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.2.5.12 Set the Administrative state of a Subnetwork Connection (SNC) or Connection Route

{Requirement II. 249} The NML-EML Interface shall allow the NMS to set the administrative state of one or more of the routes of an NMS specified Subnetwork Connection (SNC) or Connection given the NMS specified data listed in {Requirement II. 250}.

As a result of (successful) completion of this request, the EMS shall update the administrative state of addressed routes. The unlocked routes of an SNC or Connection are the set of resources the restoration process is allowed to work with.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 250} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS set the administrative state of one or more of the routes of a Subnetwork Connection (SNC) or Connection.

1) SNC or Connection name

   This parameter shall represent the name of the SNC or Connection for which the Route is to be switched.

2) Route identifiers

   This parameter shall represent a list of the identifiers of the routes that are to have their administrative state set.

3) Administrative state values

   This attribute shall represent a list of the values for the administrative states of the routes specified.

4) Additional information

   Refer to {Requirement I. 060}.

Source: Version 3.5.

Mandatory/Optional: Mandatory.

4.2.5.13 Set the intended Route of a Subnetwork Connection (SNC) or Connection

{Requirement II. 251} The NML-EML Interface shall allow the NMS to set a route as the intended one of an NMS specified Subnetwork Connection (SNC) or Connection given the NMS specified data listed in {Requirement II. 248}.

As a result of (successful) completion of this request, the addressed route is the intended one, and the formerly intended route is a backup one

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.5.13.1 Subnetwork Connection (SNC) set intended Route Data

{Requirement II. 252} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS set the administrative state of one or more of the routes of a Subnetwork Connection (SNC) or Connection.

1) SNC or Connection name

   This parameter shall represent the name of the SNC or Connection for which the Route is to be switched.

2) Route identifier

   This parameter shall represent the identifier of the route that is to be set to the intended route of the SNC or Connection.

3) Additional information

   Refer to {Requirement I. 060}.

Source: Version 3.5.

Mandatory/Optional: Mandatory.

4.2.5.14 Deactivation of Subnetworks Connections

{Requirement II. 090}

The NML-EML Interface shall allow the NMS to request that an EMS deactivate an NMS specified Subnetwork Connection (SNC). Refer to {Requirement II. 091} for the data provide as part of the deactivation request.

As a result of (successful) completion of this request, the EMS will have issued required commands such that all cross-connections associated with (comprising) the Subnetwork Connection have been removed (i.e., are no longer provisioned on NE(s)), but the EMS shall preserve the object representing the Subnetwork Connection. The successfully deactivation of the specified SNC will transition to pending.

If the NMS specified SNC has more then one route, this operation locks all the routes, which means that EMS and/or the network (e.g. restoration process) have no more control over these routes. All the currently active Cross-Connect (XC)s for this SNC shall be removed, of any (active or partial) route.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.5.14.1 Subnetwork Connection (SNC) Deactivation Data

{Requirement II. 091} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS de-activate a Subnetwork Connection (SNC):

1) SNC name

   This attribute shall indicate the name of the previously created SNC that is to be activated.

2) Grade of impact

   This attribute shall indicate the degree to which the activation of the SNC may impact the Subnetwork. Refer to the supporting document SD1-36_SNCTypes.pdf for details on the grade of impact parameter.

3) EMS freedom level

   This attribute shall indicate the level of freedom given to the EMS in determining the effect on the Subnetwork when performing SNC operations.

4) Termination Point (TP)s to modify

   This attribute shall identify a list of TPs that are to be modified as part of the activation request. Each item in the list shall contain the following:

   - TP name (generic TP names are not allowed)
   - Transmission parameters
   - Ingress transmission descriptor name
   - Egress transmission descriptor name
   - Mapping mode

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.5.15 Deletion of Subnetwork Connection (SNC)s

{Requirement II. 092} The NML-EML Interface shall allow the NMS to request that an EMS delete an NMS specified Subnetwork Connection (SNC). Refer to {Requirement II. 093} for the data provide as part of the delete request.

As a result of the (successful) completion of this request, the EMS shall delete the object representing the SNC. The EMS shall refuse/fail this request if any of the cross-connections associated with (comprising) the SNC were in place (i.e., are provisioned on NE(s)) at the time of the request (i.e., the SNC must be successfully deactivated before the EMS will allow the SNC to be deleted).

If the SNC has more than one route, then the operation deletes the SNC, its intended and all back-up route(s).

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.5.15.1 Subnetwork Connection (SNC) Deletion Data

{Requirement II. 093} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS delete a Subnetwork Connection (SNC):

1) SNC name

   This attribute shall indicate the name of the previously created SNC that is to be activated.

2) EMS freedom level

   This attribute shall indicate the level of freedom given to the EMS in determining the effect on the Subnetwork when performing SNC operations.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.5.16 Deactivate and Delete Subnetwork Connection (SNC)s and Point to MultiPoint Connections

{Requirement II. 094}

The NML-EML Interface shall allow the NMS to request that an EMS both deactivate and delete (as a result of a single request) an NMS specified Subnetwork Connection (SNC). Refer to {Requirement II. 095} for the data provided as part of the deactivate and delete request.

As a result of (successful) completion of this request, the EMS will have issued required commands such that all cross-connections associated with (comprising) the Subnetwork Connection have been removed (i.e., are no longer provisioned on NE(s)), and the EMS shall delete the object representing the Subnetwork Connection.

If the NMS specified SNC has more than one route, this operation locks all the routes, which means that EMS and/or the network (e.g. restoration process) have no more control over these routes. All the currently active Cross-Connect (XC)s for this SNC shall be removed, of any (active or partial) route. Then the operation deletes the SNC, its intended and all back-up route(s).

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.5.16.1 Subnetwork Connection (SNC) Deactivation and Deletion Data

{Requirement II. 095} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS de-activate and delete a Subnetwork Connection (SNC):

1) SNC name

   This attribute shall indicate the name of the previously created SNC that is to be activated.

2) Grade of impact

   This attribute shall indicate the degree to which the activation of the SNC may impact the Subnetwork. Refer to the supporting document SD1-36_SNCTypes.pdf for details on the grade of impact parameter.

3) EMS freedom level

   This attribute shall indicate the level of freedom given to the EMS in determining the effect on the Subnetwork when performing SNC operations.

4) Termination Point (TP)s to modify

   This attribute shall identify a list of TPs that are to be modified as part of the activation request. Each item in the list shall contain the following:

   - TP name (generic TP names are not allowed)
   - Transmission parameters
   - Ingress transmission descriptor name
   - Egress transmission descriptor name
   - Mapping mode

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 096} The NML-EML Interface shall allow an NMS to de-activate and delete the leg of a point to multipoint configuration.

Each leg of a point to multipoint configuration shall be a separate Subnetwork Connection (SNC).

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.5.17 Subnetwork Connection (SNC) Management Mode

{Requirement II. 100} The NML-EML Interface shall allow the NMS to retrieve the subnetwork connection management mode of operation.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.6 Fault Management

4.2.6.1 Alarm Surveillance

4.2.6.1.1 Alarm Administration

4.2.6.1.1.1 Alarm Subscription

The NML-EML Interface shall support the reporting of alarm conditions and clear conditions.

{Requirement II. 101} The NML-EML Interface shall allow the NMS to subscribe to non-EMS system related alarms from the EMS, (e.g. Managed Element (ME), MultiLayer Subnetwork (MLSN), Subnetwork Connection (SNC), Equipment and Environmental alarms, etc.).

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 102} The NML-EML Interface shall allow the NMS to subscribe to EMS system related alarms from the EMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 105} The NML-EML Interface shall allow of one or more NMSs to subscribe to EMS alarms.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 106} The NML-EML Interface shall allow the NMS to un-subscribe from a successful previous subscription for alarms.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 107} The NML-EML Interface shall allow the NMS to un-subscribe based on one or more filtering criteria from a successful previous subscription for alarms.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.6.1.1.2 Alarm Filtering

{Requirement II. 103} The NML-EML Interface shall allow the NMS to specify zero or more alarm filters to be applied by the EMS as part of the NMS' alarm subscription request/update.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 104} The NML-EML Interface shall allow the NMS to specify whether an alarm that matches a filter criterion be forwarded to the NMS by the EMS (subscription mechanism) or whether an alarm/event that matches filter criterion be excluded from being forwarded up to the NMS by the EMS (filtering mechanism).

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.6.1.1.3 Alarm Loss

{Requirement II. 177} The NML-EML Interface shall allow the EMS to inform the NMS about the status of event forwarding and whether lifecycle events and/or alarms have been discarded.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 178} The NML-EML Interface shall allow the EMS to inform the NMS about the availability of the event channel.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.6.1.2 Alarm Summary

{Requirement II. 110} The NML-EML Interface shall allow the NMS to retrieve active Network Element (NE) alarms from the EMS for a given NMS specified NE name and filter criteria (filtering may be performed using the probable cause and/or severity).

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 111} The NML-EML Interface shall allow the NMS to retrieve all active EMS related alarms from the EMS for a given NMS specified filter criteria (filtering may be performed using the probable cause and/or severity).

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 270} The NML-EML Interface shall allow the NMS to retrieve all active EMS related alarms and NE related alarms from the EMS for a given NMS specified filter criteria (filtering may be performed using the probable cause and/or severity).

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 287} The NML-EML Interface shall allow the NMS to retrieve unacknowledged active Network Element (NE) alarms from the EMS for a given NMS specified NE name and filter criteria (filtering may be performed using the probable cause and/or severity).

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 154} The NML-EML Interface shall allow the NMS to retrieve all unacknowledged active EMS related alarms and NE related alarms from the EMS for a given NMS specified filter criteria (filtering may be performed using the probable cause and/or severity).

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 288} The NML-EML Interface shall allow the NMS to retrieve all unacknowledged active EMS related alarms from the EMS for a given NMS specified filter criteria (filtering may be performed using the probable cause and/or severity).

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.6.1.3 Alarm Acknowledgement

{Requirement II. 155} The NML-EML Interface shall allow the NMS to acknowledge an alarm.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 156} The NML-EML Interface shall allow the NMS to un-acknowledge an alarm.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 157} The NML-EML Interface shall allow the NMS to subscribe to notifications from the EMS to indicate that an alarm has been acknowledged. (i.e. the EMS, the same NMS, or another NMS).

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 158} The NML-EML Interface shall allow the NMS to subscribe to notifications from the EMS to indicate that an alarm has been un-acknowledged. (i.e. the EMS, the same NMS, or another NMS).

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.6.1.4 Alarm Severity Assignment

To control the severities of alarms that may be reported by the EMS the NML-EML Interface shall allow an NMS to assign an Alarm Severity Assignment Profile (ASAP) to certain of the object visible across the interface. Table 4.6 identifies the objects to which an ASAP may be assigned.

Table 4.6: Objects to which an ASAP may be assigned

	Object Name
1	Element Management System (EMS)
2	Equipment
3	Equipment Holder
4	Equipment Protection Group (EPG)
5	Group Termination Point (GTP)
6	Managed Element (ME)
7	Protection Group (PG)
8	Subnetwork Connection (SNC)
9	Termination Point (TP)
10	Topological Link (TL)

4.2.6.1.5 Alarm Root Cause Indication

{Requirement II. 298} The NML-EML Interface shall allow the EMS to indicate whether an alarm is a raw (un-correlated) alarm or a root cause alarm indication.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

{Requirement II. 224} The NML-EML Interface shall allow the EMS to change its diagnosis of a root cause and send an appropriate update to the NMS

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.2.7 Protection

4.2.7.1 Trail and Subnetwork Connection (SNC) Protection

This section addresses the NML-EML Interface requirements that enable an NMS to discover and manage trail and subnetwork connection protection and the switching of both trails and the subnetwork connection protection.

The basic principle is one of discovery of trail protection than to manage protection switching via the interface.

This section only applies to SONET/SDH.

4.2.7.1.1 Determine all ring and linear trail protection schemes

{Requirement II. 112} The NML-EML Interface shall allow the NMS to discover all trail protection schemes (both linear and ring configurations), to the extent that the EMS knows of the configuration that exist in the network managed by the EMS.

Refer to {Requirement II. 059}

Source: Version 2.0.

Mandatory/Optional: Optional.

It is possible that the resources of a ring (or a linear system) are split among more than one EMS.

The NML-EML Interface shall not indicate if the ring is a complete ring, a portion of a complete ring or a open ring that is still in the process of being provisioned (or any linear system).

The ordering of Network Element (NE)s within a ring is not explicitly indicated across the NML-EML Interface. This information may be inferred from the Topological Link (TL)s passed across the NML-EML Interface.

4.2.7.1.2 Determine traffic source for trail and subnetwork connection protection schemes

{Requirement II. 114} The NML-EML Interface shall allow the NMS to determine the traffic source of a Protection Group (PG) or a Subnetwork Connection (SNC). In addition the NMS needs to determine:

1) The current protection switch state (whether protection switching is locked, automatic or forced). switch reason
2) The protection attributes (e.g. whether the scheme is unidirectional or bi directional (also known as single or dual ended) or the protocol used for MSSPRING). PGP parameter
3) If the switching is revertive or not.
4) Support for 1+1 (with no extra traffic capability) and1:N which does support extra traffic on the protection resources.

Source: Version 2.0.

Mandatory/Optional: Optional.

This shall provide operator the information on a end-to-end trail basis.

4.2.7.1.3 Protection Switching Notifications

{Requirement II. 115} The NML-EML Interface shall allow the EMS to send notifications in case of switching events related to trail and subnetwork connection protection (SNCP).

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.7.1.4 Protection Switching Commands

4.2.7.1.4.1 Perform Protection Switch

{Requirement II. 116} The NML-EML Interface shall allow the NMS to perform protection switch commands that are supported by a Connection Termination Point (CTP) or a Protection Group (PG) that is currently connected as being able to perform a protection switch.

CTPs are used only for protection switch commands that cannot be performed via the PG object. For example for SNCP no PG object exists and the protection switch operation is applied directly to a CTP.

The following are the known values for SDH APS and VC Trail Protection schemes:

1) Lockout
2) Clear
3) Forced Switch
4) Manual Switch
5) Exerciser

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.7.1.4.2 Query Protection Switch Status

{Requirement II. 117}

The NML-EML Interface shall allow the NMS to query the EMS to determine if any persistent protection switch commands have been invoked.

This query shall be supported on a Connection Termination Point (CTP) and on a Protection Group (PG).

Refer to {Requirement II. 195} for the details if the data that shall be returned by the EMS.

The query on CTP is only applicable for protection schemes that do not employ a PG. For example for SNCP protection no protection group object exists and the protection switch operation and query is applied directly on a CTP.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 195} The NML-EML Interface shall allow the NMS to retrieve the following switch status information for a Protection Group (PG) or a Connection Termination Point (CTP) identified by an NMS specified PG or CTP name:

1) Type

   This attribute shall represent the type of the protection for which the switch has occurred.

2) Switch reason

   This attribute shall represent the reason for the switch.

3) Layer rate

   This attribute shall represent the layer at which the switch has occurred. (Refer to {Requirement I. 022}).

4) PG

   This attribute shall represent the name of the Protection Group (PG) in the case of a trail switch. Not used if the protection type is Subnetwork Connection Protection (SNCP).

5) Protected TP

   This attribute shall represent the name of the Termination Point (TP) being protected.

6) Switch away from TP

   This attribute shall represent the name of the TP being switched away from.

7) Switch to TP

   This attribute shall represent the name of the TP that is switched to.

8) Additional information

   Refer to {Requirement I. 060}.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.7.2 Equipment Protection

4.2.7.2.1 Determine active Equipment

{Requirement II. 175} The NML-EML Interface shall allow the NMS to determine the active Equipment instances within an Equipment Protection Group (EPG). In addition the NMS needs to determine:

1) The current protection switch state (whether protection switching is locked, automatic or forced).
2) The protection attributes.
3) If the switching is revertive or not.

Refer to {Requirement II. 269} for the details of the data that shall be returned by the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 269} The NML-EML Interface shall allow the NMS to retrieve the following switch status information for a Equipment Protection Group (EPG) identified by an NMS specified EPG name:

1) Type

   This attribute shall represent the type of the protection for which the switch has occurred.

2) Switch reason

   This attribute shall represent the reason for the switch.

3) EPG

   This attribute shall represent the name of the Equipment Protection Group (EPG).

4) Protected Equipment

   This attribute shall represent the name of the Equipment being protected.

5) Switch to Equipment

   This attribute shall represent the name of the Equipment that is switched to.

6) Additional information

   Refer to {Requirement I. 060}.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.7.2.2 Protection Switching Notifications

{Requirement II. 176} The NML-EML Interface shall allow the EMS to send notifications in case of an Equipment protection switch.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.8 Performance Management

The NML-EML Interface shall support Performance Monitoring functions for Termination Point (TP)s.

4.2.8.1 Supported PM Parameters

{Requirement II. 118} The NML-EML Interface shall allow the NMS to monitor and collect PM parameters (counters and analogue gauge measurements, nearEnd and/or farEnd) for 24 hours (uni- or bidirectional) and 15 minutes (unidirectional) measurement intervals.

Refer to the supporting document SD1-28_PerformanceParameters.pdf for the currently defined PM parameters.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.8.1.1 Parameter Measurement Points

{Requirement II. 119} The NML-EML Interface shall allow the NMS to monitor and collect the PM parameters defined in {Requirement II. 118} on per TP basis.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 120} The NML-EML Interface shall allow the NMS to enable and disable PM on endpoints of a SNC.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.8.2 Enabling/Disabling of PM data collection

{Requirement II. 121} The NML-EML Interface shall allow the NMS to enable and disable the collection of PM data on a list of TP names basis or for a list of Managed Element (ME)s.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.8.3 Enabling / Disabling of TCA notification

{Requirement II. 122} The NML-EML Interface shall allow the NMS to enable and disable the notification of Threshold Crossed Alerts on a list of TPs or for a list of Managed Element (ME)s.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.8.4 PM Capabilities

{Requirement II. 124} The NML-EML Interface shall allow the NMS to query the EMS for the number of supported previous or history day-registers and 15minute-registers.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.8.5 TCA Threshold Retrieval

4.2.8.6 TCA Threshold Setting

{Requirement II. 125} The NML-EML Interface shall allow the NMS to set the value of specific Threshold Crossing Alert (TCA) Parameters for a Termination Point (TP) identified by an NMS specified TP name, layer rate, granularity and location.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.8.7 Threshold Crossed Notifications

{Requirement II. 126} The NML-EML Interface shall allow the EMS to send Threshold Crossed Alerts (TCAs) upon a threshold violation spontaneously (immediately after getting the event form the ME).

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.8.8 Historical PM Data

4.2.8.8.1 File Transfer

{Requirement II. 128} The NML-EML Interface shall allow the NMS to request the delivery of a historical PM data file for a specified set of Termination Point (TP)s and PM parameters to an external data server for further evaluation.

Refer to {Requirement II. 284} for the contents of the data file.

The file shall have a defined format that is defined in the supporting document SD1-30_PMFileFormat.pdf.

It shall be readable by spreadsheet programs. When reading the register values, the EMS shall inform the NMS if the determination of PM data has not been activated for a specified TP.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.8.8.1.1 PM Data File Format

{Requirement II. 284} The historical PM data file returned by the EMS shall contain the following information

1) Start time

   This attribute shall represent the start of time frame for contained PM records.

2) End time

   This attribute shall represent the end of time frame for contained PM records.

3) Layer rate

   This attribute shall represent the layer rate of the PM data. The layer specified must be supported by the containing TP. Refer to {Requirement I. 022}.

4) Granularity

   This parameter shall represent the time granularity of the PM measurement data, either 15 minutes or 24 hours. This attribute is not applicable for instantaneous measurements (i.e. gauge type measurements)

5) User label

   Refer to {Requirement I. 060}.

6) Period end time

   This attribute shall represent the period end time of the interval for which the data is being reported.

7) Monitored time

   This attribute shall represent the number of seconds monitored within the measurement period. *OPTIONAL*

8) Number of periods

   This attribute shall represent the number of measurement periods with the same value and status starting at the monitored time. This can be used for zero suppression and unavailable periods. *OPTIONAL*

9) PM parameter measurements

   This attribute shall represent a list of the PM parameter measurements. Refer to {Requirement II. 283}.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

{Requirement II. 129} The NML-EML Interface shall allow the NMS to request (using a data file as defined in {Requirement II. 128}) the values of all (or a specific number of) historical 24hour registers of all the PM measurement points within the EMS controlled domain that have been activated.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 130} The NML-EML Interface shall allow the NMS to request (using a data file as defined in {Requirement II. 128}) the values of all (or a specific number of) historical 15minute registers of all the PM measurement points within the EMS controlled domain that have been activated.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.8.8.2 On Demand

{Requirement II. 163} The NML-EML Interface shall allow the NMS to request for a specific period of time the historical PM measurements for a specified set of Termination Point (TP)s and PM parameters.

When reading the history bucket values, the EMS shall inform the NMS if the determination of PM data has not been activated for a specified TP.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.8.9 Current PM Data Retrieval

{Requirement II. 131} The NML-EML Interface shall allow the NMS to request the current PM measurement for a list of Termination Point (TP)s.

When reading the register values, the EMS shall inform the NMS if the determination of PM data has not been activated for a specified TP.

Refer to {Requirement II. 282} for the details if the data that shall be returned by the EMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 281} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests the current PM measurement data for a list of Termination Point (TP)s.

1) TP name

   This parameter shall represent the name of the TP from which to retrieve the PM measurement data.

2) Layer rates

   This parameter shall represent a list of the layer rates of the PM measurement data to be retrieved. The layer specified must be supported by the containing TP. Refer to {Requirement I. 022}.

3) Granularities

   This parameter shall represent a list of the time granularities of the PM measurement data, either 15 minutes or 24 hours.

4) Locations

   This parameter shall represent a list of the locations of the PM measurement data.

5) PM Parameters

   This parameter shall represent a list of the names of the PM parameters for which measurement data shall be retrieved.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.2.8.9.1 PM Measurement Data

{Requirement II. 282} The NML-EML Interface shall allow the NMS to retrieve the following PM Data for a Termination Point (TP) identified by an NMS specified TP name:

1) Layer rate

   This parameter shall represent the layer rate of the PM data. The layer specified must be supported by the containing TP. Refer to {Requirement I. 022}.

2) Granularity

   This parameter shall represent the time granularity of the PM measurement data, either 15 minutes or 24 hours. This attribute is not applicable for instantaneous measurements (i.e. gauge type measurements)

3) Retrieval time

   This parameter shall represent the NE time at which the PM measurement data was obtained from the NE.

4) PM parameter measurements

   This attribute shall represent a list of the PM parameter measurements. Refer to {Requirement II. 283}.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.8.9.1.1 PM Parameter Measurement

{Requirement II. 283} A PM Parameter Measurement represents a specific PM parameter data measurement associated with a Termination Point (TP):

1) PM parameter

   This parameter shall represent the name of the PM parameter.

2) Location

   This attribute shall represent the location of the performance monitoring measurement.

3) Value

   This attribute shall represent the layer rate of the PM data. The layer specified must be supported by the containing TP. Refer to {Requirement I. 022}.

4) Measurement unit

   This attribute shall represent the unit of the measurement (e.g. seconds, milli-amps, hertz etc.)

5) Status

   This attribute shall represent the status of a measurement interval.(i.e. valid, incomplete, invalid, unavailable or zero-suppressed).

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.8.10 Clearing PM Data

{Requirement II. 132} The NML-EML Interface shall allow the NMS ability to clear (reset) the PM Data for a list of TPs or for a list of Network Element (NE)s.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.9 Equipment Management

An equipment model shall be supported at the NML-EML Interface to provide the following capabilities to the network operator and management systems.

4.2.9.1 Acquire Actual Equipment

{Requirement II. 133} The NML-EML Interface shall allow the NMS to retrieve the actual equipment configuration in the network to allow an operator to perform the following:

1) To be able to locate in the instances of equipment of a particular type/ version/serial-number.
2) The information shall be to the level of field replaceable entity and shall be structured such that the replaceable unit is identifiable.
3) The information shall be structured such that searches may be performed on the data from the specific vendors equipment with a familiar degree of detail to that available at the vendors own system (i.e. at the same level of granularity).
4) To allow the network management system to convey an accurate view of the actual equipping in the network.
5) A slot shall be able to identify the equipage (equipped or unequipped) of a circuit pack.
6) The interface shall be able to retrieve specific equipment data and shall be able to report notification of addition/removal of a equipment.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.9.2 Actual versus Expected Equipment

{Requirement II. 134} The NML-EML Interface shall allow the NMS the ability to identify a mismatch between the “actual” and “expected” equipment.

This shall allow the NMS to aid in fault correction and service impact interpretation (e.g. Mean Time to Repair). A slot shall identify the actual deployed circuit pack and a list of supported circuit packs or equipment’s.

1) The card that has been physically installed in a holder
2) The card that was created (i.e. set as expected) to be in the specific equipment holder.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.9.3 Installed Equipment

{Requirement II. 135} The NML-EML Interface shall allow the NMS to retrieve information related to installed Equipment.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.9.4 Provisioning of Equipment

{Requirement II. 136} The interface shall allow the NMS to provision the expected Equipment for each Equipment Holder.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.9.4.1 Equipment Creation Data

{Requirement II. 263} The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS provision an Equipment

1) User label

   Refer to {Requirement I. 060}.

2) User label uniqueness

   This parameter shall indicate to the EMS to check whether the user label is unique amongst the equipments within the EMS.

3) Owner

   Refer to {Requirement I. 060}.

4) Expected equipment type

   This parameter shall represent the type of the expected equipment.

5) Alarm reporting

   This parameter shall indicate whether alarm reporting for this equipment is to be enabled or disabled.

6) Alarm severity assignment profile

   This parameter shall represent the name of the Alarm Severity Assignment Profile (ASAP) that is to be assigned to the Equipment.

7) Manufacturer

   This parameter shall represent the name of the equipment vendor.

8) Protection role

   This parameter shall represent the protection role (e.g. primary or secondary) that the equipment plays in case it takes part in an equipment protection scheme

9) Protection scheme state

   This parameter shall indicate the state of the protection scheme (i.e. whether it is to be active or locked).

10) Additional information.

    Refer to {Requirement I. 060}.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

4.2.9.5 Unprovisioning of Equipment

{Requirement II. 262} The interface shall allow the NMS to unprovision the expected Equipment for each Equipment Holder.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.9.6 Provisioning Equipment Alarm Reporting

{Requirement II. 078} The NML-EML Interface shall allow the NMS to enable and disable alarm reporting on an Equipment.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.9.7 Provisioning Equipment Holder Alarm Reporting

{Requirement II. 079} The NML-EML Interface shall allow the NMS to enable and disable alarm reporting on an Equipment Holder.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.10 Software Management

4.2.10.1 Database Backup

{Requirement II. 229} The NML-EML Interface shall allow the NMS to request the EMS to backup the data of NMS specified Managed Element (ME).

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 230} The NML-EML Interface shall allow the NMS to retrieve the current status of a NMS specified Managed Element (ME) database backup.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 231} The NML-EML Interface shall allow the NMS to request the EMS abort the database backup for an NMS specified Managed Element (ME).

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 232} The NML-EML Interface shall allow the NMS to retrieve the names of all the database backups for a list of NMS specified Managed Element (ME) names that are available on the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.11 Log Management

{Requirement II. 253} The NML-EML Interface shall allow the NMS to retrieve all of the MTNM notifiable events and alarms available on the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 254} The NML-EML Interface shall support the following log capabilities defined in ITU Recommendation X.735.

1) delete Log Records
2) get Log Records
3) retrieve Log Records
4) set Administrative State
5) set Capacity Alarm Threshold
6) set Discriminator Construct
7) set Log Full Action
8) set Max Log Size
9) set Max Record Life
10) set Week Mask

Source: Version 3.0.

Mandatory/Optional: Optional.

{Requirement II. 255} There shall be one instance of the Log Service per EMS.

Source: Version 3.0.

Mandatory/Optional: Mandatory.

{Requirement II. 271} The NML-EML Interface shall allow the NMS to retrieve the attributes of the Log from the EMS.

Source: Version 3.0.

Mandatory/Optional: Optional.

4.2.12 Maintenance Commands

{Requirement II. 137} The NML-EML Interface shall allow the NMS to set and release the maintenance commands that are supported by a Termination Point (TP).

The following is a list of maintenance of operations that shall be supported:

1) Facility Loopback
2) Terminal Loopback
3) Facility Forced AIS
4) Terminal Forced AIS
5) Force RDI
6) Set as segment end point (Note: unset is provided by the already-included release action)
7) End-to-end loopback OAM cell (ATM)
8) Segment loopback OAM cell (ATM)
9) Local Loop Qualification (DSL)
10) DSL Line Supervision (DSL)

A distinct error message will be returned to distinguish between a command rejected because of the current state of the object and between the command not being supported

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 138} The NML-EML Interface shall allow the NMS to query the EMS to determine if any persistent maintenance commands have been invoked.

This query is support by the Managed Element (ME) and Termination Point (TP) objects.

Source: Version 2.0.

Mandatory/Optional: Optional.

It will be possible to add additional commands, as required for other technologies or to support more robust maintenance needs.

4.2.13 Interface Version

{Requirement II. 286} The NML-EML Interface shall support the abiltity to allow an NMS to retrieve the interface version from the EMS.

Source: Version 2.0.

Mandatory/Optional: Mandatory.

4.2.14 Security Management

4.2.14.1 Identification

{Requirement II. 139} The NML-EML Interface shall support the identification of an NMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.14.2 Authentication

{Requirement II. 140} The NML-EML Interface shall allow the NMS to be authenticated before access to the EMS is allowed.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.14.3 Session

{Requirement II. 141} The NML-EML Interface shall allow the NMS to establish a communication session following the successful authentication of the NMS by the EMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 142} The NML-EML Interface shall allow the NMS to establish multiple communication sessions.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 143} The NML-EML Interface shall allow the NMS or EMS to terminate a communication session.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.14.3.1 Detection

{Requirement II. 144} The NML-EML Interface shall allow the EMS, in the event that it can detect a security violation or realize a failed access attempt, to generate an appropriate alarm.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.15 System Requirements

4.2.15.1 NML-EML Communication Status Monitoring

{Requirement II. 145} The NML-EML Interface shall allow for both the NMS and the EMS to detect an NMS/EMS communication failure.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.2.15.2 NML-EML Interface Capabilities

{Requirement II. 146} The NML-EML Interface shall allow the NMS to retrieve a list of interface operations with information whether they are supported or not supported by the EMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 290}

Cap 1

The NML-EML Interface shall allow the NMS to retrieve the connectivity awareness capability (with regard to the connectionless technologies) provided by the EMS.

The EMS works either in the “connectivity-aware” or in the “connectivity-unaware” mode:

- “connectivity aware” indicates that the EMS has some capability to detect the connectivity between the MFDs within an FD
- “connectivity-unaware” indicates that the EMS has no capability at all to detect connectivity between the MFDs within an FD.

Source: Version 3.5.

Mandatory/Optional: Optional.

4.2.16 GUI Cut-Through (GCT)

{Requirement II. 147} The NML-EML Interface shall allow the NMS to access the EMS user interface.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 148} The GCT feature must be supported by a generic cross-platform interface.

The NML-EML Interface shall be supplemented by a client-server window system that would facilitate the actual launch of the GCT.

Source: Version 2.0.

Mandatory/Optional: Optional.

The window system protocol (e.g. X-protocol) providing the solution for the actual launch of the GCT is outside the MTNM interface solution set. In order for the MTNM GCT interface to be truly platform independent it should not be based on the implementation details relevant to any specific window system.

{Requirement II. 149} EMS GCT functionality on NMS is the same as the one available to an EMS client user when invoked within the EMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

The security issues, however, are not currently addressed in this interface (at this release bilateral agreement is required, in the future security issues may be defined as a part of the interface).

When there is a bilateral agreement regarding the definition of users the following applies:

The EMS allows control of the various EMS features depending on capability level of the user. Within a given window context, user access should be limited only to those operations, which are allowed given the (EMS) user info.

User identification may have the following behavior (based on bilateral agreement):

As long as at least one GUI cut-through session is active, then the user remains logged in.

For a less secure, seamless cut-through the EMS user login and password is maintained at the NMS (outside the scope of the Interface). When the cut-through session is invoked, then the login and password is passed across the Interface

When there is a bilateral agreement regarding the definition of user capabilities the following applies:

The EMS shall allow the restriction of user access by allowing an NMS to explicitly specify the user capability. (When there is no specified user id the capability value alone determines the functionality of the GCT.) Typical user capabilities that are envisioned are read-only and read-write.

When both user identification and user capability are both agreed upon and both are specified then the user capability is to be applied as a further restriction to the capability implicit in the user information.

{Requirement II. 150} The GUI Cut-through shall allow the context to be specified. The user can make a request for a specific object in different window contexts and the user's entry point into the EMS should be in the same context or window type as requested by the NMS. The only window context that must be supported is the top-level context. If the EMS does not provide data for a window type, the NMS will use the data for the top-level window. The suggested optional window contexts are Fault, Configuration (software management/connection management), Accounting, Performance, Security and Systems Management. When the desired window context is unavailable for the given object scope, then a window context that contains and/or allows navigation into the requested context is provided. The interface will allow the NMS to retrieve all the GCT contexts that are supported by the EMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement II. 151} The GCT feature may apply to different objects managed within each EMS, i.e. the GCT request must have a scope (either EMS or ME). The scope field in the GCT data record reflects the supported scope of the GCT operation.

Source: Version 2.0.

Mandatory/Optional: Optional.

When the GCT is not implemented for the requested object scope, the EMS will launch the GCT of the closest superior object available (this information is available through the hierarchy of the object name). For example, if a certain GCT is unavailable for the requested ME then a GCT (of the same context) should be launched for the EMS instead. The NMS should request the narrowest scope desired. Alternatively the NMS shall only request a scope that is known to be supported according to the profile/GCT capability information received.

{Requirement II. 152} If possible, the EMS should be able to actively manage the GCT application windows.

1) Integrate the EMS window within NMS window hierarchy (e.g. associate it to the NMS main window on creation of the GCT window) if made possible by the window system protocol.
2) The EMS should be able to close all of the GCT windows upon request by the NMS, or notify the NMS that closing of the GCT is disabled. Optional based on EMS.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.3 Category III: Abnormal or Exception Conditions, Dynamic Requirements

{Requirement III. 001} The NML-EML Interface shall allow the NMS to detect when the EMS is no longer available.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirement III. 002} The NML-EML Interface shall allow the EMS to detect when the NMS is no longer available.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.4 Category IV: Expectations and Non-Functional Requirements

{Requirements IV. 001} The NML-EML Interface must take into account scalability and performance issues. Real-world scenarios shall be applied to check for object instance scale (MIB instances) and operations intensity

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirements IV. 002} Physical entities shall be supported in a concise manner. It is the intention of this model to be a practical scalable solution.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirements IV. 003} The NML-EML Interface shall allow an EMS to support multiple NMSs simultaneously.

Source: Version 2.0.

Mandatory/Optional: Optional.

{Requirements IV. 004} The NML-EML Interface shall allow an NMS to support multiple EMSs simultaneously.

Source: Version 2.0.

Mandatory/Optional: Optional.

4.5 Category V: System Administration Requirements

No System Administration requirements have been identified at this time.

5 Connectionless Management Requirements

This is an Addendum to the requirements found in Section 4 that addresses the Connectionless Technology requirements, in particular the Etherent requirements for the NML-EML Interface.

5.1 Category I: Static and Structural Requirements

Note:

The Connectionless Port TP (CPTP) and Flow Point (FP), discussed in this document, are not actual objects. They are special cases of PTP/FTP and CTP, respectively. PTPs/FTPs which represents CPTPs are identified by the layered parameter “ConnectionlessPort” and CTPs which represents FPs are distinguished by their connectionless layer rate.

5.1.1 Termination Point (TP)

Refer to Section 4.1.4.

5.1.1.1 Connectionless Port TP (CPTP)

{Requirement I. 100}

TP 3

A Connectionless Port Termination Point (CPTP) represents a potential port capability for connectionless technologies, i.e. the CPTP has potential client TPs which are Flow Points (FP).

A CPTP is not an object. The term CPTP is used in this specification for defining the characteristics of a port at a connectionless matrix.

Note: the clients of a CPTP, i.e. FPs, are connected via the matrix.

A CPTP can be a Physical Termination Point (PTP), a Floating Termination Point (FTP) or a Connection Termination Point (CTP). A CPTP is modelled as a PTP object if the port is an external port, or as an FTP object if the port is an internal encapsulation port or as a CTP object if the port is an external encapsulation port.

For examples for CPTPs implemented as PTP, FTP and CTP refer to the supporting document SD1-44_ConnectionlessTechnologyManagement.pdf.

A Boolean layered parameter for PTPs, FTPs and CTPs at connectionless layers (e.g. Ethernet) shall identify the TP as a CPTP.

Note in the case of Ethernet client layer:

- a CPTP corresponds to an IEEE bridge port, which can be an UNI port (Network Access Port) or an NNI port (Network Port)
- the CPTP will always have Directionality set to “bidirectional”.

Source: Version 3.5.

Mandatory/Optional: Mandatory (if EMS supports connectionless layers)

5.1.1.2 Flow Domain Edge CPTP (FD EdgeCPTP)

{Requirement I. 101}

TP 1

A Flow Domain Edge Connectionless Port TP (CPTP) (FD Edge CPTP) is a TP that is at an entrance or exit point of a Flow Domain (FD). It provides endpoints (Flow Point (FP)s) for provisioning of Flow Domain Fragment (FDFr)s at a connectionless client layer. It is characterized by the value “fd Edge” of the PortTPRoleState layered parameter (see supporting document SD1-16_LayeredParameters.pdf) in the PTP/FTP/CTP object.

Note: In the case of Ethernet, Flow Points are always bidirectional and so FD Edge CPTPs are entrance and exit points of FDs.

Source: Version 3.5.

Mandatory/Optional: Mandatory (if EMS supports connectionless layers)

5.1.1.3 Flow Point (FP)

{Requirement I. 102}

TP 2

A Flow Point (FP) is a point of an Flow Domain Fragment (FDFr), where the FDFr enters or exits an Matrix Flow Domain (MFD). An edge FP represents the endpoint of a FDFr.

An FP is contained by and is the client of a Connectionless Port TP (CPTP). When the server CPTP is configured as “fdEdge”, all client FPs become also edge FPs.

Flow Points are created as CTP objects when the associated FDFr is created, and are deleted when the associated FDFr is deleted. Flow Points do not exist without an associated FDFr.

Notes:

- Only “in use” FPs are represented as CTP objects at the interface and therefore only “in use” FPs can be inventoried.
- FPs with layer rate Ethernet will always have Directionality set to “bi-directional”.
- The connectionless layered parameters are contained in the layered transmission parameters attribute inherited from the TP object. This attribute shall represent the technology-specific parameters associated with the different connectionless layers (e.g. Ethernet, DVB, IPTV) that are supported by the FP. Refer to chapter “Connectionless Technology Parameters” of the supporting document SD1-16_LayeredParameters.pdf for details of the currently defined connectionless parameters.
- It is recommended (not required) that when an NMS does not provide a name for the FP the EMS use the FDFr VLAN-ID if VLAN is used; otherwise use a string with a “P” as a prefix followed by a unique number.

Source: Version 3.5.

Mandatory/Optional: Mandatory (if EMS supports connectionless layers)

5.1.2 Matrix Flow Domain (MFD)

{Requirement I. 103}

MFD 1

A Matrix Flow Domain (MFD) is contained within a Managed Element (ME) and represents a grouping of assigned Connectionless Port TP (CPTP). These assigned CPTPs adapt to TPs at a client layer (Flow Points) that support connectionless switching.

MFDs are indirectly (via its assigned CPTPs) interconnected by Calls.

The MFDs are used to describe the Flow Domain (FD)s. An MFD shall be related to at most one FD.

Note: If the layer the assigned CPTPs adapts to is the Ethernet MAC layer, the MFD corresponds to an IEEE 802.1 bridge.

Source: Version 3.5.

Mandatory/Optional: Mandatory (if EMS supports connectionless layers).

{Requirement I. 104}

MFD 2

A Matrix Flow Domain (MFD) object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Layered transmission parameters

   This attribute shall represent the technology-specific parameters associated with the different connectionless layers (e.g. Ethernet, DVB) that are supported by the MFD. Refer to chapter “Connectionless Technology Parameters” of the supporting document SD1-16_LayeredParameters.pdf for details of the currently defined connectionless parameters.

2) Network access domain

   This attribute represents the Network Access Domain (NAD) to which this MFD has been assigned to.

3) Flexible

   This attribute shall indicate whether the MFD is fixed (i.e., cannot be configured by the NMS) or is flexible.

4) TMD state

   This attribute shall indicate the state of consistency between the MFD and its associated Transmission Descriptor (TMD).

Source: Version 3.5.

Mandatory/Optional: Mandatory (if EMS supports connectionless layers).

5.1.3 Flow Domain (FD)

{Requirement I. 105}

FD 1

A Flow Domain (FD) is contained within the EMS and represents a network grouping of zero or more Matrix Flow Domains (MFDs) and zero or more CPTPs that are assigned to these MFDs.

FD Edge CPTPs represent the external access points to the FD.

FD Internal CPTPs represent the internal points within an FD.

A “singleton” FD is an FD which contains only one MFD.

A “singleton” FD does not have FD Internal CPTPs, hence, only has FD Edge CPTPs.

The main services provided by an FD are to serve as container for Flow Domain Fragments (FDFrs) and for their set-up, modify and tear-down operations.

Note:

FDFrs represent connectionless (e.g., Ethernet) services. Additionally, an FD represents the network view of a region with potential connectionless services. Such a region may be associated with a given service provider's customer or internal network application.

Source: Version 3.5.

Mandatory/Optional: Mandatory (if EMS supports connectionless layers).

{Requirement I. 106}

FD 2

A Flow Domain (FD) object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Layered transmission parameters

   This attribute shall represent the technology-specific parameters associated with the different connectionless layers (e.g., Ethernet, DVB, Fiber Channel) that are supported by the FD for connectionless switching. Refer to chapter “Connectionless Technology Parameters” of the supporting document SD1-16_LayeredParameters.pdf for details of the currently defined connectionless parameters (the same FD may support switching for different connectionless layers).

2) Network Access Domain

   This attribute represents the Network Access Domain (NAD) to which this FD has been assigned to.

3) FD connectivity state

   This attribute shall provide an indication to the NMS about the server layer connectivity between the MFDs associated to an FD. It shall have one of the following values:

   - fully connected
     (i.e., all FD Edge CPTPs are reachable to each other)
   - not fully connected
     (i.e., at least one FD Edge CPTP is not reachable by another FD Edge CPTP).
   - unknown
     (i.e., the connectivity state of the FD is not known by the EMS). An EMS providing the “connectivity-aware” mode is not allowed to set the value to “unknown”.

4) FD type

   This (read-only) attribute shall provide an indication of the type of the FD. It may have one of the following values:

   - singleton
     The FD contains at most one MFD.
   - network
     The FD may contain more than one MFD.

Source: Version 3.5.

Mandatory/Optional: Mandatory (if EMS supports connectionless layers).

5.1.4 Traffic Conditioning (TC) Profile

{Requirement I. 107}

TC 1

A Traffic Conditioning (TC) Profile contains the parameters that policies the traffic at the ingress and or egress (in this version only ingress is supported) of a connectionless layer (e.g., Ethernet) network. The policies can be applied to

- a whole UNI / NNI port (CPTP)
  (e.g., for Ethernet per VLAN-Id, Priority and CoS)
- a point of an FDFr (FP)
  (e.g., for Ethernet per VLAN-Id, Priority and CoS)

TC Profiles are associated to CPTPs and FPs via the "Traffic Mapping Table" contained in the "Layered transmission parameters" attribute. Refer to the supporting document SD1-16_LayeredParameters.pdf for details of the "Traffic Mapping Table".

The association of a TC Profile to an FP or CPTP is defined in a "cascaded" manner; i.e., parameters associated to traffic units at an CPTP can be overwritten by parameters associated to the same traffic units at an FP.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement I. 108}

TC 2

A Traffic Conditioning (TC) Profile object represents a collection of attributes, which are used to define the traffic conditioning parameters (e.g., bandwidth characteristics) on a Termination Point (TP).

The TC Profile object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Network access domain

   This attribute represents the Network Access Domain (NAD) to which this TC Profile has been assigned to.

2) Default profile

   This Boolean attribute shall define the TC Profile as a non-deleteable profile (modification may be possible or rejected). An EMS may contain more than one default profiles which condition different TPs.

3) Layered traffic conditioning parameters

   This attribute shall represent a list of traffic conditioning parameters which can be set and/or retrieved at a specified connectionless layer on a Termination Point (TP) having this TC Profile assigned. Refer to chapter "Traffic conditioning parameters" of the supporting document SD1-16_LayeredParameters.pdf for details of the currently defined traffic conditioning parameters.

Source: Version 3.5.

Mandatory/Optional: Mandatory (if EMS supports connectionless layers).

{Requirement I. 109}

TC 4

The EMS has to make sure that the traffic conditioning and traffic mapping configured in the NE is always represented by appropriate traffic mapping tables at the CPTPs, FPs or at both. In the case where the NMS does not provide a traffic mapping table (or not a complete one), the EMS has to provide the appropriate traffic mapping tables.

For details of the Traffic Mapping Table refer to the supporting document SD1-16_LayeredParameters.pdf.

The Traffic Mapping Table shall represent, over the interface, the current configuration of the TPs.

Source: Version 3.5.

Mandatory/Optional: Optional

5.1.5 Flow Domain Fragment (FDFr)

{Requirement I. 110}

FDFr 1

A Flow Domain Fragment (FDFr) is a logical entity that represents a transparent end-to-end connectivity between two or more Flow Point (FP)s (at the same connectionless layer and with compatible directionality) within a Flow Domain (FD).

As a consequence all FPs of the FDFr belong to this FD in the sense that they are client FPs of Connectionless Port TP (CPTP)s of this FD.

Source: Version 3.5.

Mandatory/Optional: Mandatory (if EMS supports connectionless layers).

{Requirement I. 111}

FDFr 2

A Flow Domain Fragment (FDFr) object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Directionality

   This attribute shall represent the directionality of the FDFr (bidirectional or unidirectional).
   Note: In the case of Ethernet, Directionality is always bidirectional.

2) Layered transmission parameters

   This attribute shall represent the technology-specific parameters associated with the layer that the FDFr is connecting (e.g., Ethernet, DVB). Refer to supporting document SD1-16_LayeredParameters.pdf for details of the currently defined connectionless parameters.

3) aEnd TPs

   This attribute shall represent a list of Flow Points that delimit the FDFr and characterize its edges (entrance and/or exit points). They are clients of the FD Edge CPTPs.

   In case of a unidirectional FDFr this attribute contains the list of source Flow Points. In case of a bidirectional FDFr this attribute may be combined with the zEnd TPs attribute to obtain all the Flow Points that are associated to the FDFr.

Note:

For a bidirectional point to point FDFr it is suggested, but not mandatory, to put one TP in the aEnd and one in the zEnd, as with SNCs and TLs. For a multipoint FDFr, or a point-to-point FDFr that may be expanded to multipoint, it is suggested to put all the TPs in the aEnd.

4) zEnd TPs

This attribute shall represent a list of Flow Points that delimit the FDFr and characterize its edges (entrance and/or exit points). They are clients of the FD Edge CPTPs.

In case of a unidirectional FDFr this attribute contains the list of sink Flow Points that delimit the FDFr and characterize its edges (exit points). They are clients of the FD Edge CPTPs.
In case of a bidirectional FDFr this attribute may be combined with the aEnd TPs attribute to obtain **all** the Flow Points that are associated to the FDFr.

5) Network Access Domain

This attribute represents the Network Access Domain to which this FDFr has been assigned.

6) Flexible

This attribute shall indicate whether the FDFr is fixed or is flexible. (Fixed means it cannot be modified or deleted by the NMS, in particular FPs cannot be added or removed.).

7) Administrative state

This attribute shall indicate whether the FDFr is locked (i.e., traffic units cannot flow through the FDFr) or unlocked (i.e., traffic units are allowed to flow through the FDFr).

8) FDFr state

The FDFr state attribute indicates one of the following values:

- Active
  This state identifies that all MFDFrs and all (edge and internal) FPs for the FDFr are active in the network.
- Partial
  This state identifies that not all parts (MFDFrs or FPs) of the FDFr were created during the creation operation, or that not all parts of the FDFr were deleted during the deletion operation.

9) FDFr type

This attribute represents the type of the FDFr. Possible values are:

- point-to-point
- point-to-multipoint (E-tree)
- multipoint

Source: Version 3.5.

Mandatory/Optional: Mandatory (if EMS supports connectionless layers).

5.1.6 Flow Domain Fragment (FDFr) Route

{Requirement I. 112}

FDFr 3

An FDFr Route object shall represent the route of a Flow Domain Fragment (FDFr). An FDFr route shall be represented as a partially ordered series of Matrix Flow Domain Fragment (MFDFr)s through which the FDFr traverses. Only MFDFrs on the FDFr's layer rate are part of the route.

In case of a unidirectional route, the MFDFrs that are part of the route shall be listed from the Matrix Flow Domain (MFD) on which the FDFr starts (first entry) to the MFD on which the FDFr ends (last entry).

A route belongs to only one FDFr. MFDFrs cannot be shared by routes of different FDFrs.

Source: Version 3.5.

Mandatory/Optional: Optional

{Requirement I. 113}

FDFr 4

An FDFr Route object shall have the following attributes:

1) Contained MFDFrs

   This attribute shall represent the partially ordered list of Matrix Flow Domain Fragments (MFDFrs) that constitute the FDFr route.

2) Actual state

   This attribute shall represent a summary state of the actual states of the MFDFrs in the network, regardless of the FDFr that the MFDFrs are currently serving. It can have only the following values:

   - Inactive - none of the contained MFDFrs are active in the network
   - Active - all of the contained MFDFrs are active in the network
   - Partial - one or more, but not all the MFDFrs are active in the network.

3) Additional information

   Refer to {Requirement I. 060}.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.1.7 Matrix Flow Domain Fragment (MFDFr)

{Requirement I. 114}

MFDFr 1

A Matrix Flow Domain Fragment (MFDFr) object shall represent the portion of a Flow Domain (FD) within a Matrix Flow Domain (MFD) inside a Network Element (NE).

An MFDFr is atomic and is similar to a Flow Domain Fragment (FDFr) in a singleton FD.

An MFDFr is primarily used in the specification of a Flow Domain Fragment (FDFr) Route, in cases where the route must be specified by the NMS.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement I. 115}

MFDFr 2

An MFDFr object shall have the following attributes:

1) aEnd TPs

This attribute shall represent a list of Flow Points (FPs) that delimit the MFDFr and characterize its edges (entrance and/or exit points).

In case of a unidirectional MFDFr this attribute contains the list of source Flow Points. In case of a bidirectional MFDFr this attribute may be combined with the zEnd TPs attribute to obtain all the Flow Points that are associated to the MFDFr.

2) zEnd TPs

This attribute shall represent a list of sink Flow Points that delimit the MFDFr and characterize its edges (exit points).

In case of a bidirectional MFDFr this attribute may be combined with the aEnd TPs attribute to obtain all the Flow Points that are associated to the MFDFr.

3) Directionality

This attribute shall represent the directionality of the MFDFr (bidirectional or unidirectional).

Note:

In the case of Ethernet, the directionality is always bidirectional.

4) Flexible

This attribute shall indicate whether the MFDFr is fixed or is flexible. (Fixed means it cannot be modified or deleted by the NMS, in particular FPs cannot be added or removed.).

5) Active

This attribute shall indicate if the MFDFr has been successfully activated in the Network Element (NE).

6) MFDFr type

This attribute represents the type of the FDFr. Possible values are:

- point-to-point
- point-to-multipoint (E-tree)
- multipoint

7) Additional information

Refer to {Requirement I. 060}.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2 Category II: Normal Sequences, Dynamic Requirements

5.2.1 Inventory Management

5.2.1.1 Inventory Retrieval

5.2.1.1.1 Termination Point (TP) Inventory

{Requirement II. 291}

TP 2

The NML-EML Interface shall allow the NMS to retrieve the layered transmission parameters of a Termination Point (TP) by using a filter that is based on the Layer Rates and the groupings defined in the supporting document SD1-16_LayeredParameters.pdf.

It shall also be possible to retrieve all layered transmission parameters defined in the TP.

For a list of the currently defined set of supported TP transmission parameters refer also to the SD1-16_LayeredParameters.pdf supporting document.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 292}

TP 4

The NML-EML Interface shall allow the NMS to retrieve the attributes of the Matrix Flow Domain (MFD) that is assigned to an NMS specified Connectionless Port TP (CPTP).

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.1.1.2 Matrix Flow Domain Inventory

{Requirement II. 295} MFD 2	The NML-EML Interface shall allow the NMS to retrieve the attributes of all Matrix Flow Domain (MFD)s contained within an NMS specified Managed Element (ME). It is irrelevant whether or not the MFDs are actually associated to a Flow Domain (FD). Source: Version 3.5. Mandatory/Optional: Optional.
{Requirement II. 296} MFD 3	The NML-EML Interface shall allow the NMS to retrieve the attributes of a Matrix Flow Domain (MFD) given an NMS specified MFD name. Source: Version 3.5. Mandatory/Optional: Optional.
{Requirement II. 297} MFD 4	The NML-EML Interface shall allow the NMS to retrieve the attributes of all Connectionless Port TP (CPTP)s assigned to a given Matrix Flow Domain (MFD). Source: Version 3.5. Mandatory/Optional: Optional.
{Requirement II. 300} MFD 12	The NML-EML Interface shall allow the NMS to retrieve the attributes of the Matrix Flow Domain (MFD)s that are supported by a specific Equipment; given an NMS specified Equipment name. Source: Version 3.5. Mandatory/Optional: Optional.

{Requirement II. 301}

MFD 15

The NML-EML Interface shall allow the NMS to retrieve all the Matrix Flow Domain (MFD)s that are associated with an NMS specified Transmission Descriptor (TMD).

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 302}

MFD 18

The NML-EML Interface shall allow the NMS to retrieve the layered transmission parameters of a Matrix Flow Domain (MFD) by using a filter that is based on the Layer Rate and the groupings defined in the supporting document SD1-16_LayeredParameters.pdf.

It shall also be possible to retrieve all layered transmission parameters defined in the MFD.

For a list of the currently defined set of supported MFD transmission parameters refer also to the SD1-16_LayeredParameters.pdf supporting document.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 303}

MFD 19

The NML-EML Interface shall allow the NMS to retrieve the attributes of the Flow Domain (FD) that is associated to an NMS specified Matrix Flow Domain (MFD).

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 304}

MFD 20

The NML-EML Interface shall allow the NMS to retrieve the attributes of all Connectionless Port TP (CPTP)s which are potentially able to be assigned to an NMS specified Matrix Flow Domain (MFD).

Notes:

Potentially means: The CPTPs are on the same equipment or same rack with backplane connectivity as the NMS specified MFD.

It is irrelevant whether the CPTPs are already assigned to an MFD or not.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.1.1.3 Flow Domain Inventory

{Requirement II. 314}

FD 2

The NML-EML Interface shall allow the NMS to retrieve the attributes of all Flow Domain (FD)s managed by the EMS.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 315}

FD 5

The NML-EML Interface shall allow the NMS to retrieve the attributes of all Connectionless Port TP (CPTP)s associated to the Flow Domain (FD). It shall be possible to filter this request to retrieve only “fdEdge” (UNI and E-NNI sides), only “fdInternal” only or all CPTPs.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 316}

FD 6

The NML-EML Interface shall allow the NMS to retrieve the attributes of all Matrix Flow Domain (MFD)s associated with an NMS specified Flow Domain (FD).

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 317}

FD 7

The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Flow Domain (FD)s given an NMS specified FD user label.

Note:

The user label may or may not be unique within the EMS domain. Non-unique user labels can be used by the NMS for example to associate several FDs from the same EMS domain, typically to indicate that those FDs participate in the same service provider's application (corporate customer, etc.). This situation may occur for example in case the EMS supports only singleton FDs.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 318}

FD 13

The NML-EML Interface shall allow the NMS to retrieve the attributes of a Flow Domain (FD) given an NMS specified FD name.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 319}

FD 19

The NML-EML Interface shall allow the NMS to retrieve the layered transmission parameters of a Flow Domain (FD) by using a filter that is based on the layer rates and the groupings defined in the supporting document SD1-16_LayeredParameters.pdf.

It shall also be possible to retrieve all layered transmission parameters defined in the FD.

For a list of the currently defined set of supported FD transmission parameters refer also to the SD1-16_LayeredParameters.pdf supporting document.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.1.1.4 Traffic Conditioning Inventory

{Requirement II. 330}

TC 1

The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Traffic Conditioning (TC) Profiles that are being managed by the EMS.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 331}

TC 3

The NML-EML Interface shall allow the NMS to retrieve the attributes of a Traffic Conditioning (TC) Profile for an NMS specified TC Profile name.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 332}

TC 4

The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Termination Point (TP)s that are associated with an NMS specified Traffic Conditioning (TC) Profile name.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.1.1.5 Flow Domain Fragment (FDFr) Inventory

{Requirement II. 338}

FDFr 2

The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Flow Domain Fragment (FDFr)s contained within an NMS specified Flow Domain (FD). It shall be possible to filter the FDFrs to be retrieved based on their connectivity rates.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 339}

FDFr 4

The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Flow Domain Fragment (FDFr)s that connect or pass through client Flow Point (FP)s of the NMS specified Connectionless Port TP (CPTP).

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 340}

FDFr 5

The NML-EML Interface shall allow the NMS to retrieve the attributes of a Flow Domain Fragment (FDFr) given an NMS specified FDFr name.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 341}

FDFr 6

The NML-EML Interface shall allow the NMS to retrieve the attributes of all the Flow Domain Fragment (FDFr) given an NMS specified FDFr user label.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 342} FDFr 14	The NML-EML Interface shall allow the NMS to retrieve the attributes of the Flow Domain Fragment (FDFr) connecting a Flow Point (FP) given an NMS specified FP name. Source: Version 3.5. Mandatory/Optional: Optional.
{Requirement II. 343} FDFr 19	The NML-EML Interface shall allow the NMS to retrieve the Matrix Flow Domain Fragment (MFDFr)s of a Flow Domain Fragment (FDFr) given an NMS specified FDFr name. Source: Version 3.5. Mandatory/Optional: Optional.
{Requirement II. 344} FDFr 16	The NML-EML Interface shall allow the NMS to retrieve the layered transmission parameters of a Flow Domain Fragment (FDFr) by using a filter that is based on the groupings defined in the supporting document SD1-16_LayeredParameters.pdf. It shall also be possible to retrieve all layered transmission parameters defined in the FDFr. For a list of the currently defined set of supported FDFr transmission parameters refer also to the SD1-16_LayeredParameters.pdf supporting document. Source: Version 3.5. Mandatory/Optional: Optional.

5.2.2 Provisioning

5.2.2.1 Floating Termination Point Management

5.2.2.1.1 Creation of Floating Termination Point (FTP)s

{Requirement II. 293}

TP 5

The NML-EML Interface shall allow the NMS to create a Floating Termination Point (FTP) given the NMS specified data listed in {Requirement II. 299}.

The EMS will attempt to fulfill the request including the creation of the appropriate fragment CTPs if applicable (e.g., VCAT or LAG).

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.1.1.1 Floating Termination Point (FTP) Creation Data

{Requirement II. 299}

TP 5

The NML-EML Interface shall allow the NMS to specify the following parameters when it requests the creation of a Floating Termination Point (FTP) object.

1) User label

   Refer to {Requirement I. 060}.

2) User label uniqueness

   This parameter shall indicate to the EMS that the value of the user label attribute must be unique amongst the FTPs within the EMS domain.

3) Owner

   Refer to {Requirement I. 060}.

4) Network Access Domain

   This attribute represents the Network Access Domain (NAD) to which the new FTP shall be assigned to.

5) Equipment name

   Equipment hosting the new FTP in the ME.

6) Ingress TMD

   This attribute shall represent the name of the ingress Transmission Descriptor (TMD) associated with this FTP.

7) Egress TMD

   This attribute shall represent the name of the egress Transmission Descriptor (TMD) associated with this FTP.

8) Layered transmission parameters

   This parameter represents the client layer rate (e.g., LR_Ethernet), the layer rate (e.g., LR_Encapsulation, LR_Fragment, LR_LAG_Fragment) and the server layer rate (e.g., VC12, any supported concatenated layer rate, including LR_Fragment) of the new CPTP. A list of technology-specific transmission parameters is associated to every layer rate. Refer to the supporting documents SD1-18_layers.pdf and SD1-16_LayeredParameters.pdf for details of the currently defined layer rates and transmission parameters.

Note:

   Virtual Concatenation (VCAT) is indicated by the presence of the LR_Fragment layer rate and its associated layered transmission parameters (e.g., number and rate of the server CTPs), and that Link Aggregation (LAG) is indicated by the presence of the LR_LAG_Fragment layer rate and its associated layered transmission parameters.

9) Additional information

   Refer to {Requirement I. 060}.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.1.2 Deletion of Floating Termination Point (FTP)s

{Requirement II. 294}

TP 6

The NML-EML Interface shall allow the NMS to delete a Floating Termination Point (FTP) given the NMS specified FTP name from the EMS.

The deletion request shall fail if:

- the FTP is a Call endpoint, or
- the FTP or any of its contained server CTPs (VCAT and LAG case) is an Subnetwork Connection (SNC) endpoint, or
- if the FTP cannot be explicitly deleted (e.g., was automatically created by the EMS).
- if the FTP is a CPTP whose role is "*assigned*" or "*fdEdge*".

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.2 Matrix Flow Domain Management

5.2.2.2.1 Creation of Matrix Flow Domain (MFD)s

{Requirement II. 305}

MFD 7

The NML-EML Interface shall allow the NMS to create a Matrix Flow Domain (MFD) within an Managed Element (ME); given the NMS specified data listed in {Requirement II. 306}.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.2.1.1 Matrix Flow Domain (MFD) Creation Data

{Requirement II. 306}

MFD 8

The NML-EML Interface shall allow the NMS to specify the following parameters when it requests the creation of a Matrix Flow Domain (MFD):

1) User label

   Refer to {Requirement I. 060}.

2) User label uniqueness

   This parameter shall indicate to the EMS that the value of the user label attribute must be unique amongst the MFDs within the EMS domain.

3) Owner

   Refer to {Requirement I. 060}.

4) Network Access Domain

   This attribute represents the Network Access Domain (NAD) to which the new MFD shall be assigned to.

5) List of unassigned CPTPs

   This parameter contains the list of unassigned CPTPs which shall be assigned to the MFD to be created.

6) Connectionless Layered Parameters

   This parameter shall represent the connectionless technology parameters associated with the different layers (e.g. Ethernet, DVB) that are supported by the MFD. Refer to chapter “Connectionless Technology Parameters” of the supporting document SD1-16_LayeredParameters.pdf for details of the currently defined connectionless parameters.

7) Additional information

   Refer to {Requirement I. 060}.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.2.2 Modification of Matrix Flow Domain (MFD)s

{Requirement II. 307}

MFD 9

The NML-EML Interface shall allow the NMS to modify a Matrix Flow Domain (MFD); given the NMS specified data listed in {Requirement II. 308}.

The modification of the MFD shall be done on a best effort basis. All attributes that could not be modified shall be returned in the reply.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.2.2.1 Matrix Flow Domain (MFD) Modification Data

{Requirement II. 308}

MFD 10

The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS modify an Matrix Flow Domain (MFD):

1) User label

   Refer to {Requirement I. 060}.

2) User label uniqueness

   This parameter shall indicate to the EMS that the value of the user label attribute must be unique amongst the MFDs within the EMS.

3) Owner

   Refer to {Requirement I. 060}.

4) Network Access Domain

   This attribute represents the new Network Access Domain (NAD) to which the MFD shall be assigned to.

5) Connectionless Layered Parameters

   This parameter shall represent the connectionless technology parameters associated with the different layers (e.g. Ethernet, DVB) that are to be changed in the MFD. Refer to chapter “Connectionless Technology Parameters” of the supporting document SD1-16_LayeredParameters.pdf for details of the currently defined connectionless parameters.

6) Additional information

   Refer to {Requirement I. 060}.

Source: Version 3.5.

Mandatory/Optional: Mandatory (if EMS supports connectionless layers).

5.2.2.2.3 Deletion of Matrix Flow Domain (MFD)s

{Requirement II. 309}

MFD 6

The NML-EML Interface shall allow the NMS to delete a Matrix Flow Domain (MFD) from a Managed Element (ME) given the NMS specified MFD name.

The request shall be denied if the MFD to be deleted is still associated to a Flow Domain (FD).

The request shall be denied if the MFD to be deleted is fixed.

The Connectionless Port TP (CPTP)s shall be automatically de-assigned from the MFD (i.e., change the TP role to unassigned) before the MFD is deleted.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.2.4 Connectionless PTP (CPTP) Management

{Requirement II. 310}

MFD 13

The NML-EML Interface shall allow the NMS to assign Connectionless Port TP (CPTP)s to a Matrix Flow Domain (MFD).

The NMS shall provide a list of unassigned CPTP names to be associated to the MFD. The provided CPTPs must be potential CPTPs for this MFD (e.g. have to be on the same equipment or same rack with backplane connectivity).

The request shall be denied if the MFD is “fixed”.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 311}

MFD 14

The NML-EML Interface shall allow the NMS to un-assign Connectionless Port TP (CPTP)s from a Matrix Flow Domain (MFD).

The NMS shall provide a list of assigned CPTPs to be un-assigned from the MFD.

The request shall be denied if the MFD is “fixed”.

A request to un-assign an FD Edge CPTP from a MFD shall be denied if a Flow Domain Fragment uses this CPTP.

The CPTPs which are also associated as Flow Domain Edge CPTP (FD EdgeCPTP)s to a Flow Domain (FD) shall automatically be de-associated from the FD; refer to Figure “State diagram for Port TP Role State” of the supporting document SD1-44_ConnectionlessTechnologyManagement.pdf.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.2.5 Transmission Descriptor (TMD) Management

{Requirement II. 312}

MFD 16

The NML-EML Interface shall allow the NMS to associate a Transmission Descriptor (TMD) to a given Matrix Flow Domain (MFD) identified by an NMS specified MFD name.

The association will overwrite specific transmission parameters of the MFD with the corresponding parameter values contained in the TMD. The Behaviour is defined in {Requirement II. 277}.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 313}

MFD 17

The NML-EML Interface shall allow the NMS to de-associate a Transmission Descriptor (TMD) from a given Matrix Flow Domain (MFD) identified by an NMS specified MFD name.

The de-association shall not influence any transmission parameter configured in the MFD. The Behaviour is defined in {Requirement II. 277}.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.3 Flow Domain (FD) Management

5.2.2.3.1 Creation of Flow Domain (FD)s

{Requirement II. 321} FD 8	The NML-EML Interface shall allow the NMS to create a Flow Domain (FD) within the EMS; given the NMS specified data listed in {Requirement II. 322}. The association of the Connectionless Port TP (CPTP)s to the FD shall be done on a best effort basis. Source: Version 3.5. Mandatory/Optional: Optional.

5.2.2.3.1.1 Flow Domain (FD) Creation Data

{Requirement II. 322}

FD 9

The NML-EML Interface shall allow the NMS to specify the following parameters when it requests the creation of a Flow Domain (FD):

1) Name

   This parameter defines the identifier of the new FD which will be used over the interface. The EMS has to make sure that the name of the FD is unique within the EMS domain. If no name is provided by the NMS, the EMS has to define a unique name.

2) User label

   Refer to {Requirement I. 060}.

3) User label uniqueness

   This parameter shall indicate to the EMS that the value of the user label attribute must be unique amongst the FDs within the EMS domain.

4) Owner

   Refer to {Requirement I. 060}.

5) Network Access Domain

   This parameter shall indicate the Network Access Domain (NAD) to which this FD will be associated to.

6) List of Matrix Flow Domains

   This parameter identifies the list of MFDs to be associated to the new FD. The MFDs to be associated must exist and must not be associated to another FD, i.e., they have to be un-associated.

7) Connectionless layered parameters

   This parameter shall represent the connectionless technology parameters associated with the different layers (e.g. Ethernet, DVB, Fiber Channel) that are supported by the FD. Refer to chapter "Connectionless Technology Parameters" of the supporting document SD1-16_LayeredParameters.pdf for details of the currently defined connectionless parameters.

8) Additional information

   Refer to {Requirement I. 060}.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.3.2 Modification of Flow Domain (FD)s

{Requirement II. 323}

FD 11

The NML-EML Interface shall allow the NMS to modify a Flow Domain (FD); given the NMS specified data listed in {Requirement II. 324}.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.3.2.1 Flow Domain (FD) Modification Data

{Requirement II. 324}

FD 12

The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS modify an Flow Domain (FD):

1) User label

   Refer to {Requirement I. 060}.

2) User label uniqueness

   This parameter shall indicate to the EMS that the value of the user label attribute must be unique amongst the MFDs within the EMS.

3) Owner

   Refer to {Requirement I. 060}.

4) Network Access Domain

   This parameter shall indicate the Network Access Domain (NAD) to which this FD will be associated to.

5) Connectionless layered parameters

   This parameter shall represent the connectionless technology parameters associated with the different layers (e.g. Ethernet, DVB, Fiber Channel) that are to be changed in the FD. Refer to chapter "Connectionless Technology Parameters" of the supporting document SD1-16_LayeredParameters.pdf for details of the currently defined connectionless parameters.

6) Additional information

   Refer to {Requirement I. 060}.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.3.3 Deletion of Flow Domain (FD)s

{Requirement II. 325}

FD 10

The NML-EML Interface shall allow the NMS to request the deletion of a Flow Domain (FD) from the EMS given the NMS specified FD name.

The request shall de-associate all Matrix Flow Domain (MFD)s and Flow Domain Edge CPTP (FD EdgeCPTP)s from the FD and then delete the FD. The Port TP role state indication (see {Requirement I. 100}) of the CPTPs which corresponds to the de-associated FD Edge CPTPs shall be set to “assigned”.

The request shall be denied if Flow Domain Fragment (FDFr)s are provisioned on any of the FD Edge CPTPs associated to the FD.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.3.4 Connectionless PTP (CPTP) Management

{Requirement II. 326} FD 14	The NML-EML Interface shall allow the NMS to associate Connectionless Port TP (CPTP)s to a Flow Domain (FD). The NMS shall provide a list of assigned CPTPs to be associated to the FD. These CPTPs have to be already assigned to one of the Matrix Flow Domain (MFD)s that are associated to the FD. The successful association of the CPTPs shall update the Port TP role state indication (see {Requirement I. 100}) of the CPTPs (i.e., newly associated CPTPs become "FD Edge"). The association of the CPTPs to the FD shall be done on a best effort basis. Source: Version 3.5. Mandatory/Optional: Optional.
{Requirement II. 327} FD 15	The NML-EML Interface shall allow the NMS to dissociate Flow Domain Edge CPTP (FD EdgeCPTP)s from a Flow Domain (FD). The NMS shall provide a list of FD Edge CPTPs to be dissociated from the FD. The request shall be denied if a Flow Domain Fragment (FDFr) uses this FD Edge CPTP. The successful dissociation of the CPTPs shall update the Port TP role state indication (see {Requirement I. 100}) of the CPTPs (i.e., newly dissociated CPTPs lose their "FD Edge" property, i.e., become "assigned"). Source: Version 3.5. Mandatory/Optional: Optional.

5.2.2.3.5 Matrix Flow Domain (MFD) Management

{Requirement II. 328}

FD 16

The NML-EML Interface shall allow the NMS to associate Matrix Flow Domain (MFD)s to a Flow Domain (FD).

The NMS shall provide a list of existing MFDs to be associated to the FD.

The MFDs to be associated must not be associated to another FD.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 329}

FD 17

The NML-EML Interface shall allow the NMS to dissociate Matrix Flow Domain (MFD)s from a Flow Domain (FD).

The NMS shall provide a list of associated MFDs to be dissociated from the FD.

The request shall be denied if a Flow Domain Fragment (FDFr) uses Flow Point (FP)s that are served by an Flow Domain Edge CPTP (FD EdgeCPTP) which is assigned to this (edge) MFD.

An EMS working in the “connectivity-aware” mode shall also deny the request if the MFD to be dissociated is not at the edge of an FD but is used by an FDFr.

The request dissociates all FD Edge CPTPs which are associated to the MFD to be dissociated from the FD and sets the Port TP role state indication (see {Requirement I. 100}) of the CPTPs to “assigned”.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.4 Traffic Conditioning (TC) Profile Management

5.2.2.4.1 Creation of Traffic Conditioning (TC) Profiles

{Requirement II. 333}

TC 8

The NML-EML Interface shall allow the NMS to create a Traffic Conditioning (TC) Profile in the EMS given the NMS specified data listed in {Requirement II. 334}.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.4.1.1 Traffic Conditioning (TC) Profile Creation Data

{Requirement II. 334}

TC 9

The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS creates or modifies a Traffic Conditioning (TC) Profile:

1) User label

   Refer to {Requirement I. 060}.

2) User label uniqueness

   This parameter shall indicate to the EMS that the value of the user label attribute must be unique amongst the Traffic Conditioning Profiles within the EMS domain.

3) Owner

   Refer to {Requirement I. 060}.

4) Layered traffic conditioning parameters

   This parameter shall represent a list of traffic conditioning parameters which can be set and/or retrieved at a specified connectionless layer on a Termination Point (TP) having this TC Profile associated. Refer to chapter "Traffic Conditioning Parameters" of the supporting document SD1-16_LayeredParameters.pdf for details of the currently defined connectionless parameters.

5) Additional information

   Refer to {Requirement I. 060}.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.4.2 Modification of Traffic Conditioning (TC) Profiles

{Requirement II. 335}

TC 15

The NML-EML Interface shall allow the NMS to modify a Traffic Conditioning (TC) Profile in the EMS given the NMS specified data listed in {Requirement II. 334}.

Note: When traffic conditioning parameters are modified, this will automatically modify the traffic conditioning in all associated TPs.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.4.3 Deletion of Traffic Conditioning (TC) Profiles

{Requirement II. 336}

TC 7

The NML-EML Interface shall allow the NMS to delete a Traffic Conditioning (TC) Profile in the EMS given an NMS specified TC Profile name.

The EMS shall reject the deletion request if any Flow Domain Edge CPTP (FD EdgeCPTP) or Flow Point (FP) is still associated with this TC Profile.

The EMS shall reject the deletion request if the NMS wants to delete the default TC Profile

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.4.4 Traffic Conditioning (TC) Profile Configuration

{Requirement II. 337}

TC 10

The NML-EML Interface shall allow the NMS to configure the mappings of specific Traffic Classes and Traffic Conditioning (TC) Profiles to specific groups of traffic units (e.g., Ethernet frames identified by VLAN-Id and Priority) at the following points:

a) a UNI / NNI port (Connectionless Port TP (CPTP))

b) a FDFr point (Flow Point (FP)).

The NMS shall provide a complete (except default) set of new mappings which overwrite all (except default) existing mappings in the Traffic Mapping Table. The provisioning of an empty mapping set removes all (except default) existing mappings; i.e., the traffic units are conditioned according to the default mapping.

Note:

The traffic units may still be conditioned specifically by another TP on this port.

As a result of (successful) completion of this request, the EMS shall condition the traffic units flowing through the resource as defined by the Traffic Class and the TC Profile, or as defined by the default Traffic Class and default TC Profile respectively.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.5 Flow Domain Fragment (FDFr) Management

5.2.2.5.1 Creation of Flow Domain Fragment (FDFr)s

{Requirement II. 345}

FDFr 7

The NML-EML Interface shall allow the NMS to create a Flow Domain Fragment (FDFr) given the NMS specified data listed in {Requirement II. 346}.

This includes FDFrs for flows of untagged frames.

The EMS shall create all necessary Flow Point (FP)s and Matrix Flow Domain Fragment (MFDFr)s (not visible as separate named objects at the interface) comprising the FDFr.

The request is successful if at least two (edge) FPs can be created. In case not all requested (edge) FPs can be connected when the FDFr is created, the EMS returns the list of the not connectable FPs in the success reply.

If internal Connectionless Port TP (CPTP)s are provided by the NMS and if it is not possible to include all provided internal CPTPs to the route of the FDFr, the operation will be rejected.

If the EMS works in the "connectivity-aware" mode, the NMS can request one of two creation results when not all FPs have potential connectivity to one another:

1) reject the creation request
2) add all Flow Points regardless of potential connectivity.

The EMS shall check if the new requested FDFr already exists, either entirely or partially or differently. The following cases have to be respected:

1) An existing FDFr matches the FDFr being requested ("matches" means that the set of flow points resulting from the activation of the FDFr is the same. Creation of two successive FDFr's on the same set of CPTPs, but with different VLAN IDs do NOT match).
    a) If the name specified by the NMS is the same as the name of the existing FDFr, or no name is specified:

       The operation will succeed (if all goes well). In this case (same name, or no name) the FDFr will be returned with the original name.
    b) If the name specified by the NMS is not the same as the name of the existing FDFr, the operation is rejected with exception Object In Use.

2) The existing FDFr is a subset or superset of the requested FDFr.

   a) If the NMS did not give a name, or if the NMS name is different from the name of the existing FDFr the operation is rejected with exception Object In Use.

   b) If the NMS name is the same as the name of the existing FDFr, the operation is rejected (because this operation does not allow to change the endpoints).

Note:

The above does not depend on the “ACTIVE” or “PARTIAL” state of the previous FDFr, but on the set of endpoints and where applicable the set of MFDFrs, of each FDFr.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.5.1.1 Flow Domain Fragment (FDFr) Creation Data

{Requirement II. 346}

FDFr 8

The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS creates a Flow Domain Fragment (FDFr):

1) Name

This parameter defines the identifier of the new FDFr which will be used over the interface. The EMS has to make sure that the name of the FDFr is unique within the containing Flow Domain. If no name is provided by the NMS, the EMS has to define a unique name.

Note:

The name of the FDFr is not changeable after the FDFr is created.

2) User label

Refer to {Requirement I. 060}.

3) User label uniqueness

This parameter shall indicate to the EMS that the value of the user label attribute must be unique amongst the FDFrs within the EMS.

4) Owner

Refer to {Requirement I. 060}.

5) Network Access Domain

This parameter indicates the Network Access Domain to which this FDFr has to be assigned.

6) Connectionless layered parameters

This parameter shall represent the connectionless technology parameters associated with the different layers (e.g. Ethernet, DVB, Fiber Channel) that are to be changed in the FD. Refer to chapter "Connectionless Technology Parameters" of the supporting document SD1-16_LayeredParameters.pdf for details of the currently defined connectionless parameters.

7) Directionality

This parameter shall indicate the directionality of the new FDFr (bidirectional or unidirectional).

Note:

In the case of Ethernet, Directionality is always bidirectional.

8) aEnd TPs

This parameter shall represent a list of of CPTP names that delimit the FDFr and characterize its edges (entrance and/or exit points). As a result of creating the FDFr, FPs are created as clients of the FD Edge CPTPs.

In case of a unidirectional FDFr this attribute contains the list of source FD Edge CPTPs. In case of a bidirectional FDFr this attribute may be combined with the zEnd TPs attribute to obtain all the FD Edge CPTPs that are associated to the FDFr.

Note:

For a bidirectional point to point FDFr it is suggested, but not mandatory, to put one TP in the aEnd and one in the zEnd, as with SNCs and TLs. For a multipoint FDFr, or a point-to-point FDFr that may be expanded to multipoint, it is suggested to put all the TPs in the aEnd.

9) zEnd TPs

In case of a unidirectional FDFr this attribute contains the list of sink FD Edge CPTPs that delimit the FDFr and characterize its edges (exit points). As a result of creating the FDFr, FPs are created as clients of the FD Edge CPTPs.

In case of a bidirectional FDFr this attribute may be combined with the aEnd TPs attribute to obtain all the FD Edge CPTPs that are associated to the FDFr.

10) Internal TPs

An optional (possibly empty) list of internal CPTP names that must be included in the route of the FDFr. As a result of creating the FDFr, FPs are created as clients of the internal CPTPs.

11) MFDFrs

An optional (possibly empty) list of MFDFrs that make up the route of the FDFr. This attribute may be omitted if the FDFr is routed by the network. As a result of creating the FDFr, MFDFrs are created in the various MFDs.

12) Full route

This parameter shall identify if the internal TPs and MFDFrs describe the full route of the FDFr (as opposed to only a partial constraint). When no routing constraints are specified,the value of false must be used.

13) Termination Point(s) to configure

This parameter shall identify a list of CPTPs and FPs that are to be configured as part of the creation request. Only CPTPs and FPs related to the new FDFr can be configured. Each item in the list shall contain the following:

- termination point name
- transmission parameters (incl. Traffic Mapping Table, Alarm Severity Assignment Profile names and TCA Parameter Profile names)
- ingress/egress transmission descriptor names

14) FDFr type

This parameter shall identify the type of the new FDFr (point-to-point, point-to-multipoint (E-tree), multipoint).

15) Connectivity requirement

This parameter shall identify (for a “connectivity-aware” EMS) the requested operation mode in case not all FPs have potential connectivity to one another:

- reject the creation request
- add all Flow Points regardless of potential connectivity.

If the EMS is not connectivity-aware, this parameter is ignored.

16) Administrative state

This parameter shall indicate whether the FDFr shall be locked (i.e., traffic units cannot flow through the FDFr) or unlocked (i.e., traffic units are allowed to flow through the FDFr).

17) Additional information

Refer to {Requirement I. 060}.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.5.2 Modification of Flow Domain Fragment (FDFr)s

{Requirement II. 347}

FDFr 11

The NML-EML interface shall allow the NMS to modify a Flow Domain Fragment (FDFr); given the NMS specified data listed in {Requirement II. 348}.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.5.2.1 Flow Domain Fragment (FDFr) Modification Data

{Requirement II. 348}

FDFr 12

The NML-EML Interface shall allow the NMS to specify the following parameters when it requests that an EMS modify a Flow Domain Fragment (FDFr):

1) User label

   Refer to {Requirement I. 060}.

2) User label uniqueness

   This parameter shall indicate to the EMS that the value of the new user label attribute must be unique amongst all FDFrs within the EMS domain.

3) Owner

   Refer to {Requirement I. 060}.

4) Network Access Domain

   This parameter indicates the new Network Access Domain to which this FDFr has to be assigned.

5) Connectionless layered parameters

   This parameter shall represent the connectionless technology parameters associated with the layer (e.g., Ethernet, DVB) that the FDFr is connecting. Refer to chapter “Connectionless Technology Parameters” of the supporting document SD1-16_LayeredParameters.pdf for details of the currently defined connectionless parameters.

6) TPs to remove

   This parameter shall represent a list of Connection Port Termination Point (CPTP) names that must be removed from the Flow Domain Fragment (FDFr). As a result of modifying the FDFr, the client Flow Points are deleted.

7) aEnd TPs

   This parameter shall represent a list of additional Connection Port Termination Point (CPTP) names that delimit the Flow Domain Fragment (FDFr) and characterize its edges (entrance and/or exit points). As a result of modifying the FDFr, Flow Points are created as clients of the FD Edge CPTPs.

8) zEnd TPs

This parameter shall represent a list of additional Connection Port Termination Point (CPTP) names that delimit the Flow Domain Fragment (FDFr) and characterize its edges (exit points). As a result of modifying the FDFr, Flow Points are created as clients of the FD Edge CPTPs.

9) Internal TPs

This parameter shall represent a list of additional internal Connection Port Termination Point (CPTP) names that must be added to the route of the Flow Domain Fragment (FDFr). As a result of modifying the FDFr, Flow Points are created as clients of the internal CPTPs.

10) Termination Point(s) to modify

This parameter shall identify a list of Flow Points (FPs) that are to be modified as part of the modification request. Only FPs related to the FDFr can be modified. Each item in the list shall contain the following:

- termination point name
- transmission parameters (incl. Traffic Mapping Table, Alarm Severity Assignment Profile names and TCA Parameter Profile names)
- ingress/egress transmission descriptor names

11) Administrative state

This parameter shall indicate whether the FDFr shall be locked (i.e., traffic units cannot flow through the FDFr) or unlocked (i.e., traffic units are allowed to flow through the FDFr).

12) Additional information

Refer to {Requirement I. 060}.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.5.3 Deletion of Flow Domain Fragment (FDFr)s

{Requirement II. 349}

FDFr 9

The NML-EML interface shall allow the NMS to delete a Flow Domain Fragment (FDFr) given the NMS specified FDFr name.

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.5.4 Flow Point (FP) Addition

{Requirement II. 350}

FDFr 13

The NML-EML Interface shall allow the NMS to add Flow Point (FP)s, identified by their server Flow Domain Edge CPTP (FD EdgeCPTP)s, or by their FD Internal CPTPs, to a Flow Domain Fragment (FDFr).

The NMS provides a list of FD Edge CPTPs to be added to the FDFr as additional end points. As a result of adding the CPTPs, FPs are created as clients of the CPTPs.

The NMS provides a list of FD Internal CPTPs to be added to the FDFr as additional internal points.

It shall be possible for the NMS to configure each Flow Point to be added (created) with the following Parameters:

- transmission parameters (incl. Traffic Mapping Table, Alarm Severity Assignment Profile names and TCA Parameter Profile names)
- ingress/egress transmission descriptor names

If the EMS works in the “connectivity-aware” mode, the NMS can request one of two results when not all FPs have potential connectivity to all FPs already in the FDFr:

- reject the request
- add all Flow Points regardless of potential connectivity

Source: Version 3.5.

Mandatory/Optional: Optional.

5.2.2.5.5 Flow Point Removal

{Requirement II. 351}

FDFr 15

The NML-EML Interface shall allow the NMS to remove Flow Point (FP)s from a Flow Domain Fragment (FDFr).

The NMS provides a list of existing Connectionless Port TP (CPTP)s to be removed from the FDFr. The Flow Points who are clients of the removed CPTPs and are associated with the FDFr are deleted.

The removal of CPTPs shall be rejected if less than two end Flow Points would remain after successful completion of the request.

Source: Version 3.5.

Mandatory/Optional: Optional.

6 Control Plane Requirements

This section lists the single concept, requirement statements to be fulfilled by the Business Requirement Model.

6.1 Category I: Static and Structural Requirements

6.1.1 MultiLayer Subnetwork Point Pool (MLSNPP)

{Requirement I. 118}

{R I. 003}

The MultiLayer Subnetwork Point Pool (MLSNPP) object is used to represent a list of Layered Subnetwork Point Pool (Layered SNPP), one Layered SNPP per supported layer rate.

The Subnetwork Point (SNP)s in the Layered SNPP of the MLSNPP shall represent Connection Termination Point (CTP)s that are G.805 TCPs (refer to the supporting document SD1-18_layers.pdf).

Source: Version 3.5.

Mandatory/Optional: Optional

{Requirement I. 119}

{R I. 004}

The MLSNPP object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) Directionality

   This attribute shall represent the directionality of its SNP components. All SNPs within a MLSNPP shall have the same direction. The direction of the SNP is inherited from the TP (CTP/PTP/FTP).

2) Layered SNPP List

   This attribute shall represent the list of Layered Subnetwork Point Pool (Layered SNPP), i.e., one Layered SNPP per supported layer rate.

Source: Version 3.5.

Mandatory/Optional: Mandatory

6.1.2 MultiLayer Subnetwork Point Pool Link (MLSNPPLink)

{Requirement I. 120}

{R I. 005}

The MultiLayer Subnetwork Point Pool Link (MLSNPPLink) object represents a set (>0) of control plane SNPP Links in different layered networks. The SNPP link interconnects a pair of SNPPs in different subnetworks that are part of the same layer network. The MLSNPPLink is delimited by a pair of Subnetwork Point Pool Link (SNPP Link)s.

Source: Version 3.5.

Mandatory/Optional: Optional

{Requirement I. 121}

{R I. 006}

The MultiLayerSNPPLink object shall have, in addition to the attributes identified in {Requirement I. 060}, the following attributes:

1) aEnd MLRA Name

   This attribute shall represent the name of the MultiLayer Routing Area (MLRA) to which the MultiLayerSNPPLink is connected.

2) zEnd MLRA Name

   This attribute shall represent the name of the MultiLayer Routing Area (MLRA) to which the MultiLayerSNPPLink is connected.

3) Layered SNPP Link List

   This attribute shall represent the list of Layered Subnetwork Point Pool Link (Layered SNPPLink)s. In the list, there shall be one entry for each supported layer rate.

4) aTNA Group Name

   This attribute represents the TNA Group name of the a-End of the MultiLayerSNPPLink.

5) zTNA Group Name

   This attribute represents the TNA Group name of the a-End of the MultiLayerSNPPLink.

6) aTNA Name

   This attribute represents the TNA name of the a-End of the MultiLayerSNPPLink.

7) zTNA Name

   This attribute represents the TNA name of the z-End of the MultiLayerSNPPLink.

8) Capacity

   This attribute shall represent the available capacity of the prescribed MultiLayerSNPPLink, in terms of the capacity in each of the layer rates supported (i.e., the layer rate of the constituent Subnetwork Point Pool Link (SNPP Link)s)

9) Direction

   This attribute shall represent the directionality of the link (i.e., unidirectional, bi-directional)

10) Interface Type

   This attribute shall represent the type of the link (i.e., "UNI", "I-NNI", "Internal E-NNI", "External E-NNI", and "unspecified"; where External E-NNI means that the link is at the edge of the NMS domain).

11) Signaling protocol

This attribute shall represent the signaling protocol used for the link. The default value for this attribute shall be “not applicable”.

12) Cost

This attribute shall represent a vector of one or more metrics, each of which indicates the relative desirability of a particular link over another during path selection.

13) Discovered

This attribute indicates whether the link is discovered or not (i.e., provisioned).

14) Availability

This attribute shall represent a vector of one or more availability factors for the MultiLayerSNPPLink. Availability may be represented in different ways between domains and within domains. Within domains, it may be used to represent a survivability capability of the MultiLayerSNPPlink.

15) Link Shared Risk Group

This attribute represents the risks factors assigned to the MultiLayerSNPPLink. The string shall represent a sequence of {risk type, and sequence of values}.

Note:

The behavior when routing a connection that specifies a risk type that is not present in a link is dictated by the control plane implementation.

16) Signaling enabled

This attribute shall specify whether signaling is enabled on the link.

17) Signaling controller Identifier

This attribute shall specify the identifier of the signaling controller to which the link is assigned.

18) Signaling parameters

This attribute specifies the parameters to be used for signaling.

Source: Version 3.5.

Mandatory/Optional: Mandatory

6.1.2.1 Edge MultiLayer Subnetwork Point Pool Link (Edge MLSNPPLink)

{Requirement I. 122}

{R I. 019}

An edge MLSNPPLink of a MultiLayer Routing Area (MLRA) is a MultiLayer Subnetwork Point Pool Link (MLSNPPLink) that is connecting the prescribed MLRA with another peer MLRA.

OR

An edge MLSNPPLink of an EMS-managed network domain is a MLSNPPLink external and connecting to the EMS-managed network domain.

Source: Version 3.5.

Mandatory/Optional: Optional

6.1.3 Layered Subnetwork Point Pool Link (Layered SNPPLink)

{Requirement I. 123}

{R I. 018}

The Layered SNPP Link shall represent the SNPP Link at a particular layer rate.

The Layered SNPP Link shall have the following attributes:

1) Layer rate

   This attribute shall represent the layer rate of the SNPP Link.

2) SNPP Link list

   This attribute shall identify the names of the SNPP Links.

Source: Version 3.5.

Mandatory/Optional: Optional

6.1.4 Subnetwork Point Pool Link (SNPP Link)

{Requirement I. 124} The Subnetwork Point Pool Link (SNPP Link) object shall represent a link that interconnects two Subnetwork Point Pool (SNPP)s .

Source: Version 3.5.

Mandatory/Optional: Optional

{Requirement I. 125} The Subnetwork Point Pool Link (SNPP Link) object shall have the following attributes:

1) SNPP Link ID

   This attribute shall represent the identifier of the SNPP Link.

2) aEnd SNPP

   This attribute shall represent the Subnetwork Point Pool (SNPP) at the A end of the SNPP Link.

3) zEnd SNPP

   This attribute shall represent the Subnetwork Point Pool (SNPP) at the Z end of the SNPP Link.

Source: Version 3.5.

Mandatory/Optional: Mandatory

6.1.5 Layered Subnetwork Point Pool (Layered SNPP)

{Requirement I. 126}

{R I. 013}

The Layered SNPP shall represent a list of Subnetwork Point Pool (SNPP) of the same layer rate.

The Layered SNPP shall have the following attributes:

1) Layer rate

   This attribute shall represent the layer rate of the SNPPs.

2) SNPP list

   This attribute shall identify the names of the alias SNPP (one for each coincident routing area edge).

Source: Version 3.5.

Mandatory/Optional: Optional

6.1.6 Subnetwork Point Pool (SNPP)

{Requirement I. 127}

{R I. 014}

The SNPP is a set of Subnetwork Point (SNP)s that are grouped together for the purposes of routing.

The SNPP shall have the following attributes:

1) SNPP ID

   This attribute shall represent the identifier of the SNPP.

2) SNP list

   This attribute shall identity the SNPs that form the SNPP.

3) Routing Area ID

   This attribute shall identity the Routing Area to which the SNPP is associated.

4) TNA Name

   This attribute shall represent the TNA of the SNPP if the link is a UNI link.

5) TNAGroup Name

   This attribute shall represent the TNA of the group of link ends to which this link end belongs.

Source: Version 3.5.

Mandatory/Optional: Optional

6.1.7 Subnetwork Point (SNP)

{Requirement I. 128}

{R I. 017}

The SNP shall be the control plane representation of the same resource of Connection Termination Point (CTP).

The SNP shall have the following attributes:

1) SNP ID

This attribute shall represent the identifier of the SNP.

2) TP Name

This attribute shall identify the CTP that the SNP represents.

3) TNA Name

This attribute shall represent the TNA name of SNP for the purpose of Call management.

Source: Version 3.5.

Mandatory/Optional: Optional

6.2 Category II: Normal Sequences, Dynamic Requirements

6.2.1 Inventory Management

6.2.1.1 Inventory Retrieval

6.2.1.1.1 MultiLayer Routing Area (MLRA) Inventory

{Requirement II. 354}

{R II. 23}
{R II. 028}

The NML-EML Interface shall allow the NMS to retrieve the attributes of all the MultiLayer Routing Area (MLRA)s that are being managed by the EMS.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 355}

{R II. 001}

The NML-EML Interface shall allow the NMS to retrieve the attributes of a MultiLayer Routing Area (MLRA) given an NMS specified MLRA name.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 356}

{R II. 002}

The NML-EML Interface shall allow the NMS to retrieve the attributes of all the subordinate MultiLayer Routing Area (MLRA)s that are one level down given an NMS specified MLRA name.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 357}

{R II. 051}

The NML-EML Interface shall allow the NMS to retrieve the attributes of the top level MultiLayer Routing Area (MLRA).

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 358}

{R II. 082}

The NML-EML Interface shall allow the NMS to retrieve the name of the top level MultiLayer Routing Area (MLRA).

Source: Version 3.5.

Mandatory/Optional: Optional.

6.2.1.1.2 MultiLayer SNPP Link (MLSNPPLink) Inventory

{Requirement II. 359}

{R II. 004}

The NML-EML Interface shall allow the NMS to retrieve the attributes of all the internal (one level down) MultiLayer Subnetwork Point Pool Link (MLSNPPLink)s given an NMS specified MultiLayer Routing Area (MLRA) name.

In the request, it shall be allowed to specify whether the Subnetwork Point (SNP) information is to be included in the response.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 360}

{R II. 083}

The NML-EML Interface shall allow the NMS to retrieve the attributes of all the MultiLayer Subnetwork Point Pool Link (MLSNPPLink)s that are being managed by the EMS.

In the request, it shall be allowed to specify whether the Subnetwork Point (SNP) information is to be included in the response.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 361}

{R II. 006}

The NML-EML Interface shall allow the NMS to retrieve the attributes of a MultiLayer Subnetwork Point Pool Link (MLSNPPLink) given an NMS specified MLSNPPLink name.

In the request, it shall be allowed to specify whether the Subnetwork Point (SNP) information is to be included in the response.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 362}

{R II. 007}

The NML-EML Interface shall allow the NMS to retrieve the attributes of all of the MultiLayer Subnetwork Point Pool Link (MLSNPPLink)s given an NMS specified Termination Point (TP) name.

In the request, it shall be allowed to specify whether the Subnetwork Point (SNP) information is to be included in the response.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 363}

{R II. 052}

The NML-EML Interface shall allow the NMS to retrieve all the Edge MultiLayer Subnetwork Point Pool Link (Edge MLSNPPLink)s given an NMS specified MultiLayer Routing Area (MLRA) name.

In the request, it shall be allowed to specify whether the Subnetwork Point (SNP) information is to be included in the response.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 364}

{R II. 084}

The NML-EML Interface shall allow the NMS to retrieve all the MultiLayer Subnetwork Point Pool Link (MLSNPPLink)s given an NMS specified Transport Network Assigned (TNA) address.

In the request, it shall be allowed to specify whether the Subnetwork Point (SNP) information is to be included in the response.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 365}

{R II. 053}

The NML-EML Interface shall allow the NMS to retrieve all the MultiLayer Subnetwork Point Pool Link (MLSNPPLink)s interconnecting a pair of NMS specified MultiLayer Routing Area (MLRA) names.

In the request, it shall be allowed to specify whether the Subnetwork Point (SNP) information is to be included in the response.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 366}

{R II. 054}

The NML-EML Interface shall allow the NMS to retrieve the MultiLayer Subnetwork Point Pool Link (MLSNPPLink) capacity (in terms of available capacity for each layer rate) given an NMS specified MLSNPPLink name.

Source: Version 3.5.

Mandatory/Optional: Optional.

6.2.1.1.3 MultiLayer SNPP (MLSNPP) Inventory

{Requirement II. 367}

{R II. 057}

The NML-EML Interface shall allow the NMS to retrieve all the MultiLayer Subnetwork Point Pool (MLSNPP)s given an NMS specified MultiLayer Routing Area (MLRA) name.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 368}

{R II. 085}

The NML-EML Interface shall allow the NMS to retrieve the attributes of a MultiLayer Subnetwork Point Pool (MLSNPP) given an NMS specified MLSNPP name.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 369}

{R II. 086}

The NML-EML Interface shall allow the NMS to retrieve the attributes of all the MultiLayer Subnetwork Point Pool (MLSNPP)s given an NMS specified Termination Point (TP) name.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 370}

{R II. 087}

The NML-EML Interface shall allow the NMS to retrieve all the MultiLayer Subnetwork Point Pool (MLSNPP)s given an NMS specified Transport Network Assigned (TNA) address.

Source: Version 3.5.

Mandatory/Optional: Optional.

{Requirement II. 371}

{R II. 088}

The NML-EML Interface shall allow the NMS to retrieve all the MultiLayer Subnetwork Point Pool (MLSNPP)s that are being managed by the EMS.

Source: Version 3.5.

Mandatory/Optional: Optional.

6.2.2 Provisioning

6.2.2.1 MultiLayer SNPP Link (MLSNPPLink) Management

6.2.2.1.1 Signalling Controller Assignment

{Requirement II. 373}

{R II. 029}

The NML-EML Interface shall allow the NMS to assign a MultiLayer Subnetwork Point Pool Link (MLSNPPLink) of type UNI to a signalling controller managed by an EMS given an NMS specified MultiLayer Subnetwork Point Pool (MLSNPP) name.

Source: Version 3.5.

Mandatory/Optional: Optional.

6.2.2.1.2 Signalling Controller De-assignment

{Requirement II. 374}

{R II. 030}

The NML-EML Interface shall allow the NMS to de-assign a MultiLayer Subnetwork Point Pool Link (MLSNPPLink) of type UNI from a Signalling Controller managed by an EMS given an NMS specified MultiLayer Subnetwork Point Pool (MLSNPP) name.

Source: Version 3.5.

ITU-T Recommendation G.7718 (2/2005) Requirement R 36

Mandatory/Optional: Optional.

6.2.2.1.3 Configuring Signalling Protocol Parameters

{Requirement II. 375}

{R II. 58}
{R II. 031}

The NML-EML Interface shall allow the NMS to configure the signalling protocol parameters of a MultiLayer Subnetwork Point Pool Link (MLSNPPLink) of type UNI managed by an EMS given an NMS specified MLSNPPLink name.

Source: Version 3.5.

ITU-T Recommendation G.7718 (2/2005) Requirement R 36

Mandatory/Optional: Optional.

6.2.2.1.4 Enable Signalling

{Requirement II. 376}

{R II. 032}

The NML-EML Interface shall allow the NMS to enable signalling on a MultiLayer Subnetwork Point Pool Link (MLSNPPLink) of type UNI managed by an EMS given an NMS specified MLSNPPLink name.

Source: Version 3.5.

ITU-T Recommendation G.7718 (2/2005) Section 7.2 #5

Mandatory/Optional: Optional.

6.2.2.1.5 Disable Signalling

{Requirement II. 377}

{R II. 033}

The NML-EML Interface shall allow the NMS to disable signalling on a MultiLayer Subnetwork Point Pool Link (MLSNPPLink) of type UNI managed by an EMS given an NMS specified MLSNPPLink name.

Source: Version 3.5.

ITU-T Recommendation G.7718 (2/2005) Section 7.2 #5

Mandatory/Optional: Optional.

6.2.2.1.6 TNA Assignment

{Requirement II. 378}

{R II. 090}

The NML-EML Interface shall allow the NMS to request an EMS to assign TNAs to MultiLayer Subnetwork Point Pool Link (MLSNPPLink).

Source: Version 3.5.

Mandatory/Optional: Optional.

6.2.2.2 MultiLayer SNPP (MLSNPP) Management

6.2.2.2.1 TNA Assignment

{Requirement II. 379}

{R II. 089}

The NML-EML Interface shall allow the NMS to request an EMS to assign TNAs to MultiLayer Subnetwork Point Pool (MLSNPP).

Source: Version 3.5.

Mandatory/Optional: Optional.

7 Use Cases

7.1 Actor-System Context Diagram

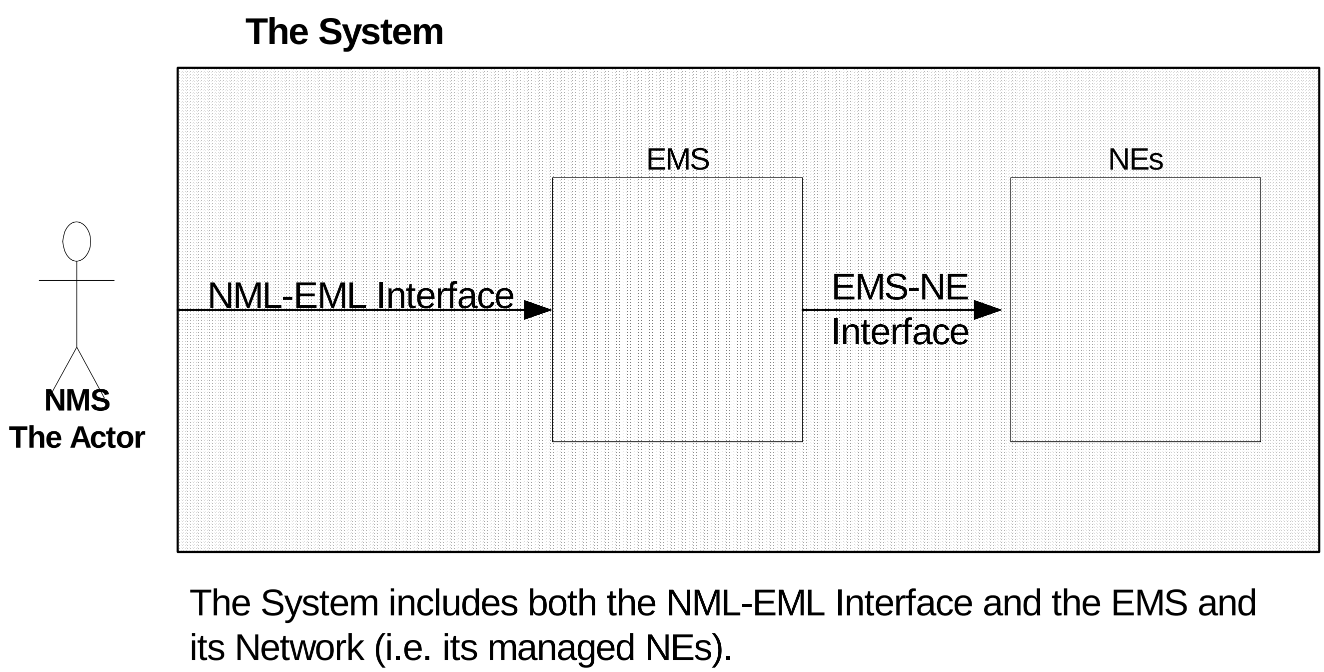


Figure 7.1: Actor-System Context Diagram

7.1.1 Exceptions

Table 7.1 identifies the complete list of exceptions that may be raised by an EMS in response to a request from an NMS. In the "Description" section of each Use Case the specific exceptions that may be raised are identified. However in addition to these specific exceptions an EMS may also raise the following "general" exceptions:

- Internal Error
- Not Implemented
- Unable To Comply
- NE Comm Loss

The Unable To Comply exception may be raised whenever the EMS cannot respond to a request, some Use Cases may identify specific conditions that will result in this exception.

When communication with at least one NE is necessary to fulfil the requests in a use case and communication to the NE(s) is lost, exception "NE Comm Loss" is raised. this also applies to the case where communication to a "Control Plane" entity is required to fulfil a request. A specific step for this exception will not be specified in the description of the use cases.

Table 7.1: Use Case Exceptions

ID	Name	General Description
1	Internal Error	The request has resulted in an EMS internal error.
2	Not Implemented	The entire request is not supported by the EMS or the request with the specified input parameters is not supported.
3	Invalid Input	The request contains an input parameter that is syntactically incorrect or identifies an object of the wrong type or is out of range.
4	Entity Not Found	The specified object instance does not exist.
5	Object In Use	The object identified in the request is currently in use.
6	User Label In Use	The user label uniqueness constraint can not be met; the specified user label is currently being used.
7	Unable To Comply	The EMS cannot respond to the the request.
8	Unsupported Routing Constraints	The EMS is unable to satisfy the requested routing constraints.
9	Access Denied	The NMS is not permitted to perform the request.
10	Capacity Exceeded	The request will result in resources being created or activated beyond the capacity supported by the NE or EMS.
11	Not In Valid State	The state of the specified object is such that the EMS cannot perform the request.
12	Protection Effort Not Met	The level of protection effort in the request cannot be met by the EMS.
13	Timeslot In Use	A timeslot is already in use.
14	TP Invalid Endpoint	The specified TP does not exist or cannot be created.
15	NE Comm Loss	The EMS is unable to communicate with the NE and communication is required to complete the request.

7.2 NMS-EMS Session Management Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.2.1:	EMS (Re)starts	{Requirement I. 041}, {Requirements IV. 003}
Use Case 7.2.2:	NMS creates a session with EMS	{Requirement I. 041}, {Requirement II. 139}, {Requirement II. 140}, {Requirement II. 144}, {Requirement II. 145}
Use Case 7.2.3:	NMS retrieves the interface version used by the EMS	{Requirement I. 030}, {Requirement I. 031}
Use Case 7.2.4:	NMS closes a session with an EMS	{Requirement II. 139}, {Requirement II. 140}
Use Case 7.2.5:	EMS closes a session with an NMS	{Requirement II. 139}, {Requirement II. 140}
Use Case 7.2.6:	NMS detects that an EMS is unavailable	{Requirement II. 145}, {Requirement III. 001}, {Requirement III. 002}
Use Case 7.2.7:	EMS detects that an NMS is unavailable	{Requirement I. 041}, {Requirement II. 145}, {Requirement III. 001}, {Requirement III. 002}

7.2.1 EMS (Re)starts

Use Case 7.2.1: EMS (Re)starts

Name	EMS (Re)starts
Summary	The EMS starts up and creates an entry point (root) EMS object. It also prepares everything to send out notifications. This use case contains some implementation details about the usage of the CORBA Naming and Notification Services.
Actor(s)	None
Pre-Conditions	Any required communication services are running on the EMS and Notification Service hosts. Naming Services can be federated to link an NMS and EMS Naming Service at a given “context”.
Begins When	The EMS starts.
Description	1) When the EMS starts it puts an entry point object in a location that is available to an NMS. • The entry point object is a CORBA object reference (i.e. an Interoperable Object Reference (IOR)). • The CORBA Naming Service can be used to store the CORBA object references. Exchanging CORBA object references via another mechanism is acceptable if agreed to by both EMS and NMS implementation groups. 2) If not already done at a previous startup, the EMS contacts the Notification Service for creation of an event channel. A reference to this event channel can therefore be provided to an NMS upon request (refer to Use Case 7.2.2: NMS creates a session with EMS). 3) If not already done at a previous startup, the EMS registers as a supplier to this event channel. 4) The EMS (re)connects to the event channel and is thus able to send out notifications. Note: The EMS may not be completely initialized when the entry point object is made available. This is because the EMS will have to do internal initialization as well as initializing the interface. It is an EMS implementation decision on how to handle requests arriving during initialization.
Ends When	The EMS has made the entry point EMS object available and is able to send out notifications.
Post-Conditions	The EMS is ready to receive requests on its interface and to send out notifications.

Use Case 7.2.1: EMS (Re)starts

Exceptions	1) Internal EMS errors – may not initialize some internal components. 2) May not be able to register Object. 3) May not be able to register or to connect to the notification service. 4) Communication services may not be running – installation or administration problem.
Traceability	{Requirement I. 041}, {Requirements IV. 003}

7.2.2 NMS creates a session with EMS

Use Case 7.2.2: NMS creates a session with EMS

Name	NMS creates a session with EMS
Summary	The NMS client finds the EMS entry point object of the required interface version. It also registers and connects to the event channel as a consumer of notifications. The NMS may then synchronize its network database. This use case contains some implementation details about the usage of CORBA Naming and Notification Services. It also assumes that there exists one Notification Service per EMS (as recommended, but not mandated).
Actor(s)	NMS
Pre-Conditions	Post-conditions of Use Case 7.2.1: EMS (Re)starts.
Begins When	NMS (re)starts - in which case this use case is performed for every EMS to be enrolled or detects that there is a new EMS that it wishes to enroll.

Use Case 7.2.2: NMS creates a session with EMS

<table>
<tr><td>Description</td><td>
1) The NMS locates the EMS entry point object.
2) The NMS retrieves the version of the EMS for validation as described in Use Case 7.2.3: NMS retrieves the interface version used by the EMS.
3) The NMS sets up a session with the EMS.
The EMS may perform identification and authentication by verifying the user id and the password provided by the NMS. It is then able to detect security violations and to send out appropriate alarms.
4) The NMS sends a request to retrieve the capabilities from the EMS.
5) The EMS replies with its supported capabilities.
6) The NMS retrieves the reference of the event channel to be used to receive notifications from the EMS.
7) If not already done in a previous session setup, the NMS registers as a consumer to this event channel and creates any filters that it requires. For examples of the type of information that the NMS may register for refer to Use Case 7.4.1: NMS registers to receive network updates information from the EMS and Use Case 7.9.3: NMS registers to receive alarms or threshold crossing alerts from an EMS.
8) If the NMS restarts, i.e. if it was registered earlier, the notification service sends out all notifications that occurred during the NMS' downtime (this is only the case if persistence has been enabled for the notification service). In this case the NMS can synchronize its database by evaluating these notification.
Note:
There are time-out values recommended for alarms and other events. After these time-outs expire the notification service discards notifications. If an NMS downtime had exceeded these times the NMS should synchronize the active alarms or the whole configuration, respectively. If event loss (synchronization) notifications are supported by the EMS then the NMS will synchronize the alarms and/or the entire configuration once retrieving such a notification.
If the EMS supports the Log Service then the NMS may use the capabilities of the Log Service to resynchronize.
9) If the NMS starts for the first time or if it decides to discard all notifications sent out in step 6, it (re)discovers the EMS' network inventory as described in the Use Case 7.4.3: NMS discovers the EMS network inventory.
10) The NMS periodically checks the availability of the EMS.
11) The EMS periodically checks the availability of the NMS
</td></tr>
<tr><td>Ends When</td><td>
Either NMS or EMS calls one of the following use cases:
• Use Case 7.2.4: NMS closes a session with an EMS.
• Use Case 7.2.5: EMS closes a session with an NMS.
• Use Case 7.2.6: NMS detects that an EMS is unavailable.
• Use Case 7.2.7: EMS detects that an NMS is unavailable.
</td></tr>
</table>

Use Case 7.2.2: NMS creates a session with EMS

Post-Conditions	1) There exists a valid session between the NMS and the EMS which is supervised by both. 2) The NMS is initialized and ready to communicate with the EMS as well as to receive notifications.
Exceptions	1) No compatible version of the entry point object. 2) Access denied: The EMS detects a security violation. 3) Invalid input: The NMS session specified is invalid. 4) Processing failure: The requested operation could not be performed.
Traceability	{Requirement I. 041}, {Requirement II. 139}, {Requirement II. 140}, {Requirement II. 144}, {Requirement II. 145}

7.2.3 NMS retrieves the interface version used by the EMS

Use Case 7.2.3: NMS retrieves the interface version used by the EMS

Name	NMS retrieves the interface version used by the EMS
Summary	The NMS checks the version of the EMS interface with which it is interacting.
Actor(s)	NMS
Pre-Conditions	The NMS successfully locates the EMS entry point object.
Begins When	The NMS sends a request to retrieve the interface version.
Description	1) NMS sends a request for the interface version to the EMS. 2) EMS responds with the interface version.
Ends When	EMS sends the interface version to the NMS.
Post-Conditions	1) Same as pre-conditions. 2) NMS has retrieved the interface version supported by the EMS.
Exceptions	None
Traceability	{Requirement I. 030}, {Requirement I. 031}

7.2.4 NMS closes a session with an EMS

Use Case 7.2.4: NMS closes a session with an EMS

Name	NMS closes a session with an EMS
Summary	The NMS closes communication to the EMS as well as to the notification service, e.g. the NMS performs a controlled shutdown.
Actor(s)	NMS
Pre-Conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) Session between EMS and NMS has been established.
Begins When	NMS starts to close the session.
Description	1) The NMS signs out for notifications for which it was previously registered and disconnects from the notification service. There is no confirmation from the EMS on this request. 2) The NMS informs the EMS that it wishes to end communications.
Ends When	The NMS stops communicating with the EMS.
Post-Conditions	1) The NMS has released all resources associated with the EMS. 2) The EMS has released all resources associated with the NMS. 3) The NMS process is no longer communicating with the EMS. 4) Notifications are no longer sent to the NMS.

Use Case 7.2.4: NMS closes a session with an EMS

Exceptions	None
Traceability	{Requirement II. 139}, {Requirement II. 140}

7.2.5 EMS closes a session with an NMS

Use Case 7.2.5: EMS closes a session with an NMS

Name	EMS closes a session with an NMS.
Summary	The EMS closes communication to the NMS, e.g. because of a controlled EMS shutdown. The NMS therefore closes the communication to the notification service.
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	EMS starts to delete the session.
Description	1) The EMS informs the NMS that it wishes to end communications. There is no confirmation from the NMS on this request. 2) The NMS may sign out for notifications for which it was previously registered and may disconnect from the notification service. Note that there is no guarantee to find the same notification service when a new session between NMS and EMS is created later on.
Ends When	The NMS stops communicating with the EMS.
Post-Conditions	1) The NMS has taken the appropriate internal action and released all resources associated with the EMS. 2) The EMS has released all resources associated with the NMS. 3) The NMS process is no longer communicating with the EMS. 4) The notification service is not trying to send notifications from this EMS to the NMS any longer if the NMS decided to sign out.
Exceptions	None
Traceability	{Requirement II. 139}, {Requirement II. 140}

7.2.6 NMS detects that an EMS is unavailable

Use Case 7.2.6: NMS detects that an EMS is unavailable

Name	NMS detects that an EMS unavailable
Summary	The NMS detects that an EMS it had a session with before, or the notification service became unavailable. This might be detected due to a failure on a request or on a heartbeat. The NMS therefore releases the session with the EMS. This may be because of an EMS process crash, because the EMS hardware being powered off before the NMS process has done a controlled shut down, because of a connection breakdown or a notification service failure.
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.

Use Case 7.2.6: NMS detects that an EMS is unavailable

Begins When	The NMS receives a failure on a heartbeat or on a request to the EMS or the notification service.
Description	1) The NMS receives a failure on a heartbeat or on a request to the EMS or the notification service which means that the EMS or the notification service isn't available any longer. 2) The NMS signs out for notifications for which it was previously registered and disconnects from the notification service if the notification service is still available. 3) The NMS releases all resources associated with the EMS. 4) The NMS takes appropriate internal action and frees up any resources associated with the EMS connection..
Ends When	The NMS has cancelled the session with the EMS.
Post-Conditions	The NMS process is no longer communicating with the EMS. The NMS needs to create a new session to restore the communication.
Exceptions	None
Traceability	{Requirement II. 145}, {Requirement III. 001}, {Requirement III. 002}

7.2.7 EMS Detects that an NMS is unavailable

Use Case 7.2.7: EMS detects that an NMS is unavailable

Name	EMS detects that an NMS is unavailable
Summary	The EMS detects that an NMS it had a session with before became unavailable. This might be detected due to a failure on a heartbeat. The EMS therefore releases the session with the NMS. This may be because of an NMS process crash, because the NMS hardware being powered off before the EMS process has done a controlled shut down or because of a connection breakdown.
Actor(s)	EMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The EMS receives a failure on a heartbeat.
Description	1) The EMS receives a failure on a heartbeat. 2) The EMS releases all resources associated with the NMS.
Ends When	The EMS has cancelled the session with the NMS.
Post-Conditions	1) The EMS process is no longer communicating with the NMS. The NMS needs to create a new session to restore the communication. 2) If there wasn't a failure with the notification service, the notification service may (if persistence is enabled) continue to store all notifications which cannot be delivered to the NMS. Thus, it is possible to deliver them after an NMS restart.
Exceptions	None
Traceability	{Requirement I. 041}, {Requirement II. 145}, {Requirement III. 001}, {Requirement III. 002}

7.3 EMS-NE Session Management Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.3.1:	EMS loses communication to a Network Element	{Requirement I. 003}, {Requirement II. 067}

7.3.1 EMS loses communication to a Network Element

Use Case 7.3.1: EMS loses communication to a Network Element

Use Case Name	EMS loses communication to a Network Element
Summary	The EMS detects that communication to an NE has been lost (e.g. a craft has reset the NE)
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	EMS detects loss of communication with the Network Element (NE)
Description	1) The EMS detects that communication to an NE has been lost. 2) EMS sends a state change notifications to the Notification Service indicating a change in the communication state to the NE. 3) Once the NE returns to a fully operational state and communication between it and the EMS is reinstated, EMS sends state change notifications to the Notification Service. 4) If the EMS automatically resynchronizes with the NE then the EMS shall send the appropriate notifications to the Notification Service. 5) If the EMS does not automatically resynchronize with the NE then an state change notification shall be sent to the Notification Service indicating that the EMS is out of synchronization with the NE (NE synchronization state attribute is set to false). Another state change notification shall be sent to the notification service when the EMS has resynchronized with the NE (NE synchronization state attribute is set to true). While the EMS is out of synchronization with the NE the EMS might not send notifications to the Notification Service.
Ends when	The EMS sends applicable notifications to the Notification Service
Post-conditions	The EMS is aware of the communication state changes
Exceptions	No Interface exceptions as no NMS-initiated operations
Traceability	{Requirement I. 003}, {Requirement II. 067}

7.4 Discovery and Inventory Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.4.1:	NMS registers to receive network updates information from the EMS	{Requirement II. 064}, {Requirement II. 065}, {Requirement II. 066}, {Requirement II. 067}
Use Case 7.4.2:	NMS resynchronizes its database with the EMS	Refer to Use Case 7.4.3: and Use Case 7.4.4:
Use Case 7.4.3:	NMS discovers the EMS network inventory	{Requirement II. 001}, {Requirement II. 002}, {Requirement II. 011}, {Requirement II. 012}, {Requirement II. 015}, {Requirement II. 018}, {Requirement II. 031}, {Requirement II. 033}, {Requirement II. 041}, {Requirement II. 314}, {Requirement II. 338}, {Requirement II. 295}, {Requirement II. 343} or {Requirement II. 043}, {Requirement II. 056}, {Requirement II. 059}, {Requirement II. 060}

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.4.4:	NMS queries EMS concerning inventory	{Requirement II. 002}, {Requirement II. 004}, {Requirement II. 006}, {Requirement II. 011}, {Requirement II. 008}, {Requirement II. 012}, {Requirement II. 015}, {Requirement II. 018}, {Requirement II. 020}, {Requirement II. 022}, {Requirement II. 024}, {Requirement II. 026}, {Requirement II. 033}, {Requirement II. 040}, {Requirement II. 047}, {Requirement II. 050}, {Requirement II. 052}, {Requirement II. 056}, {Requirement II. 058}, {Requirement II. 292}, {Requirement II. 296}, {Requirement II. 340}, {Requirement II. 300}, {Requirement II. 301}, {Requirement II. 303}, {Requirement II. 318}, {Requirement II. 295}, {Requirement II. 316}, {Requirement II. 297}, {Requirement II. 304}, , {Requirement II. 314}, {Requirement II. 317}, {Requirement II. 315}, {Requirement II. 330}, {Requirement II. 338}, {Requirement II. 342}, {Requirement II. 339}, {Requirement II. 341}, {Requirement II. 332}, {Requirement II. 343}
Use Case 7.4.5:	EMS notifies NMS of inventory change	{Requirement II. 064}, {Requirement II. 065}, {Requirement II. 066}, {Requirement II. 067}

7.4.1 NMS registers to receive network inventory updates from the EMS

Use Case 7.4.1: NMS registers to receive network updates information from the EMS

Use Case Name	NMS registers to receive network inventory information from the EMS.
Summary	The NMS registers with the notification service related to the EMS, sets the appropriate filter to receive network inventory update notifications, and connects to the notification service.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS has a reference for the notification service used by the EMS.
Begins when	The NMS sends a request to register itself at the notification service related to the EMS.
Description	1) The NMS registers at the notification service related to the EMS as a consumer of notifications (if this has not been done earlier). 2) The NMS sets the filter criteria needed to receive inventory updates from the EMS via the notification service. 3) The NMS connects to the notification service and thus is able to receive notifications matching the filter conditions specified (if this has not been done earlier). Note: The NMS can request that the inventory notifications be filtered based on resource type (e.g., PTPs) and/or notification type (i.e., creations, attribute value changes, deletions). All the inventory notifications are time stamped. Refer to {Requirement II. 064} for object creation notifications, {Requirement II. 065} for object deletion notifications, {Requirement II. 066} for attribute change notifications and {Requirement II. 067} for state change notifications.
Ends when	In case of success: The NMS receives a positive acknowledgement to its connection request to the notification service. In case of failure: The NMS receives a negative acknowledgement to its registration request, an invalid filter specified or the request times out.
Post-conditions	In case of success: The NMS is connected to the Notification Service with filtering criteria to receive network inventory update notifications. In case of failure: The NMS is not connected to the Notification Service with filtering criteria to receive network inventory update notifications.

Use Case 7.4.1: NMS registers to receive network updates information from the EMS

Exceptions	Filter creation: • Invalid grammar Filter building • Invalid constraint Connection phase: • Illegal consumer type • Consumer already connected.
Traceability	{Requirement II. 064}, {Requirement II. 065}, {Requirement II. 066}, {Requirement II. 067}

7.4.2 NMS resynchronizes its database with the EMS

Use Case 7.4.2: NMS resynchronizes its database with the EMS

Use Case Name	NMS resynchronizes its database with the EMS
Summary	The NMS sends a series of queries to the EMS, with the intent of re synchronizing its understanding of the EMS's network inventory with the actual EMS inventory.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS has determined that it wants to resynchronize its database with that of the EMS.
Begins when	The NMS sends the first inventory query to the EMS.
Description	1) This Use Case is basically Use Case 7.4.3: NMS discovers the EMS network inventory in a different context. 2) It is also possible to do a partial re-synchronization In this case Use Case 7.4.4: NMS queries EMS concerning inventory is re-used (multiple times).
Ends when	In case of success: The NMS has received the last response to the requests that it sent the EMS. In case of failure: The NMS has determined all requests have been acknowledged or have timed out, and at least one of the requests has timed out or was acknowledged in the negative (the EMS could not or would not return the requested information).
Post-conditions	In case of success: The NMS has collected all the requested inventory information from the EMS. In case of failure: The NMS has not collected all the inventory information that it requested.
Exceptions	1) Communication failure between the NMS and the EMS. 2) The NMS may include incorrect or unknown information in its queries to the EMS. 3) Entity not found: when an input parameter references an object that does not exist 4) Query unknown or not supported. 5) Unable to comply
Traceability	Refer to Use Case 7.4.3: and Use Case 7.4.4:.

7.4.3 NMS discovers the EMS network inventory

Use Case 7.4.3: NMS discovers the EMS network inventory

Use Case Name	NMS discovers the EMS network inventory

Use Case 7.4.3: NMS discovers the EMS network inventory

Summary	The NMS sends a series of queries to the EMS, with the intent of discovering the network inventory managed by the EMS.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS has not yet discovered the inventory managed by the EMS.
Begins when	The NMS sends the first inventory query to the EMS.
Description	There are many possible ways for the NMS to obtain the EMS's inventory. Two possibilities are given below, however the queries can be sent in many different combinations. Although the requests themselves are synchronous, the NMS may send subsequent inventory requests to the EMS before receiving a response to a pending request. Although not mandated by this use case, the NMS may want to start receiving EMS inventory updates at the time the discovery process begins. Once the NMS has received responses to all the inventory queries, the NMS can analyze the inventory update notifications that it received during the discovery process (if any) and updates its inventory accordingly. **(Approach A: Discovery of all subnetworks' details)** 1) The NMS sends a request to retrieve **EMS** information from the EMS. The EMS returns the name, user label, native EMS name, owner, type and software version of the EMS. 2) The NMS sends a request to retrieve all **Subnetworks** from the EMS. The EMS returns the names and associated information for all the subnetworks that it manages. 3) The NMS sends a request to retrieve all **Managed Elements** from the EMS. For each request, the EMS returns the names and associated information concerning the Managed Elements. 4) For each Managed Element, the NMS sends a request to retrieve the **Physical Termination Points** (including FTPs) from the EMS. For each request, the EMS returns the names of the Physical Termination Points (PTPs) and all information associated with the PTP, including a indication of whether the PTP is on the edge of the containing subnetwork. Alternatively for each termination point (PTP or CTP), the EMS sends a request to retrieve all the contained **Performance Monitoring Points** (PMPs) within a specified PTP or CTP from the EMS. For each request, the EMS returns the names of the contained PMPs and all associated information. 5) For each subnetwork, the NMS sends a request to retrieve all the **Topological Links** within a specified subnetwork from the EMS. The EMS returns all the names and associated information for all the topological links associated with a subnetwork. 6) The NMS sends a request to retrieve all the **top-level Topological Links** from the EMS. The EMS returns the names and associated information for each of the top-level topological links.

Use Case 7.4.3: NMS discovers the EMS network inventory

7) For each PTP, the EMS sends a request to retrieve all the **contained CTPs** within a specified PTP or CTP from the EMS.
For each request, the EMS returns the names of the contained CTPs and all associated information. If the CTP contains other CTPs, this information will also be returned.

8) For each identified subnetwork, the NMS sends a request to retrieve all the **Subnetwork Connections** from the EMS.
For each request, the EMS returns the names of the subnetwork connections and all associated information.

9) For each subnetwork connection, the NMS sends a request to retrieve all the **Routes**.

10) For every Managed Element the NMS sends a request to retrieve all the **Protection Groups** from the EMS.
For each request, the EMS returns the names of the protection groups and all associated information for all the protection groups it manages within the Managed Element specified.

11) For every Managed Element the NMS sends a request to retrieve all **Cross-Connects** from the EMS.
For each request, the EMS returns the associated information for all the cross-connects it manages within the Managed Element specified.

12) For every subnetwork the NMS sends a request to retrieve all **TP Pools** from the EMS.
For each request, the EMS returns the names and associated information for all the TP Pools it manages within the subnetwork specified.

13) If supported by the EMS, the NMS sends a request to retrieve all **Traffic Descriptors** from the EMS.
The EMS returns the names and associated information for all the traffic descriptors it manages.

14) If supported by the EMS, for every Managed Element the NMS sends a request to retrieve all **Equipment Holders** and **Equipment** from the EMS.
For each request, the EMS returns the names and associated information for all the equipment holders and equipment it manages in the ME specified.

15) Refer to Use Case 7.12.5: NMS retrieves PM capabilities of a Managed Element (ME).

16) The NMS sends a request to retrieve all **Flow Domains** from the EMS.
The EMS returns the names and associated information for all the Flow Domains that it manages.

17) For each identified Flow Domain, the NMS sends a request to retrieve all the **Flow Domain Fragments** from the EMS.
For each request, the EMS returns the names of the Flow Domain Fragments and all associated information.

18) For each Flow Domain Fragment, the NMS sends a request to retrieve its **Flow Domain Fragment Route**.

Use Case 7.4.3: NMS discovers the EMS network inventory

	19) For every Managed Element the NMS sends a request to retrieve all **Matrix Flow Domains** from the EMS. For each request, the EMS returns the names and associated information for all Matrix Flow Domains it manages within the ME specified. 20) For each Matrix Flow domain, the NMS sends a request to retrieve all **Matrix Flow Domain Fragments**. For each request, the EMS returns the names and associated information for all Matrix Flow Domain Fragments it manages within the ME specified. **(Approach B: Discovery of subnetworks with their edge points only)** 1) The NMS sends a request to retrieve all the **EMS** information from the EMS. The EMS returns the name, user label, native EMS name, owner, type and software version of the EMS. 2) The NMS sends a request to retrieve all **Subnetworks** from the EMS. The EMS returns the names and associated information for all the subnetworks that it manages. 3) For each subnetwork, the NMS sends a request to retrieve from the EMS the **Physical Termination Points** (PTPs) which are the edges of a specified subnetwork. For each request, the EMS returns the names of the edge points and all information associated with it. 4) The NMS sends a request to retrieve all the **top-level Topological Links** from the EMS. The EMS returns the names and associated information for each of the top-level topological links. 5) For each PTP, the EMS sends a request to retrieve all the **contained CTPs** within a specified PTP or CTP from the EMS. For each request, the EMS returns the names of the contained CTPs and all associated information. If the CTP contains other CTPs, this information will also be returned. 6) For each identified subnetwork, the NMS sends a request to retrieve all the **Subnetwork Connections** from the EMS. For each request, the EMS returns the names of the subnetwork connections and all associated information. 7) If supported by the EMS, the NMS sends a request to retrieve all **Traffic Descriptors** from the EMS. The EMS returns the names and associated information for all the traffic descriptors it manages.
Ends when	In case of success: The NMS has received the last response to the requests that it sent the EMS. In case of failure: The NMS has determined all requests have been acknowledged or have timed out, and at least one of the requests has timed out or was acknowledged in the negative (the EMS could not or would not return the requested information).

Use Case 7.4.3: NMS discovers the EMS network inventory

Post-conditions	In case of success: The NMS has collected all the requested inventory information from the EMS. In case of failure: The NMS has not collected all the inventory information that it requested.
Exceptions	1) Communication failure between the NMS and the EMS. 2) The NMS may include incorrect or unknown information in its queries to the EMS. 3) Entity not found: when an input parameter references an object that does not exist. See Use Case 7.4.2: NMS resynchronizes its database with the EMS. 4) Query unknown or not supported. 5) Unable to comply.
Traceability	{Requirement II. 001}, {Requirement II. 002}, {Requirement II. 011}, {Requirement II. 012}, {Requirement II. 015}, {Requirement II. 018}, {Requirement II. 031}, {Requirement II. 033}, {Requirement II. 041}, {Requirement II. 314}, {Requirement II. 338}, {Requirement II. 295}, {Requirement II. 343} or {Requirement II. 043}, {Requirement II. 056}, {Requirement II. 059}, {Requirement II. 060}

7.4.4 NMS queries the EMS concerning inventory

Use Case 7.4.4: NMS queries EMS concerning inventory

Use Case Name	NMS queries EMS concerning inventory

Use Case 7.4.4: NMS queries EMS concerning inventory

Summary	The NMS sends a request to the EMS concerning a particular network inventory item. The following network inventory queries are possible: 1) List of all **Subnetworks** managed by the EMS (the names or all associated attributes). 2) List of all **Managed Elements** within a subnetwork (the names or all associated attributes). 3) List of all the **Managed Elements** that are managed by the EMS (the names or all associated attributes). 4) List of all **PTPs** (including FTPs) associated with a Managed Element (just the names, or with all associated attributes). 5) List of all **CTPs** supported by a PTP (just the names, or with all associated attributes). 6) List of all **CTPs** supported by a CTP (just the names, or with all associated attributes). 7) List of all **Topological Links** within a subnetwork (the names or all associated attributes). 8) List of all **top-level Topological Links** (the names or all associated attributes), i.e., the links between subnetworks. 9) List of all **Subnetwork Connection** associated with a subnetwork (just the names, or with all associated attributes). 10) The present value of the attributes associated with a particular Subnetwork, Managed Element, Topological Link, PTP, CTP, or Subnetwork Connection, or Matrix Flow Domain, or Flow Domain or Encapsulation Layer Link, or Flow Domain Fragment, or Traffic Conditioning Profile. 11) The name of the containing Subnetwork for a specified Managed Element. 12) List of all Subnetwork Connections at a specified rate (just the names, or with all associated attributes). Example rates are VT1.5/TU11 and VT2/TU12. 13) List of Subnetwork Connections that use a specified TP (just the names, or with all associated attributes). The TP can be a CTP or a PTP. 14) The route(s) of a specified subnetwork connection. 15) List of all the PTPs (including FTPs) at a specified layer or set of layers (e.g., Electrical STS1/STM0, Optical OC3/STM1) and associated with a particular Managed Element. The request can be for just the names of the PTP, or all the associated attributes. 16) List of the all the PTPs at the edge of a subnetwork (just the names, or with all associated attributes). The request can be scoped to get only the PTPs at a specified layer or set of layers. 17) List of all TPs that contain a specified CTP (just the names, or with all associated attributes). 18) List of TPs associated with a specified TP (just the names, or with all associated attributes).

Use Case 7.4.4: NMS queries EMS concerning inventory

19) List of all cross-connections for a specified Managed Element. The request can be filtered to get only the cross-connections at a specified layer rate or set of layer rates.

20) List of Equipment and Equipment Holder objects contained in a specified Managed Element, or Equipment Holder, for all levels of the containment hierarchy (just the names, or with all associated attributes).

21) List of all Equipment and Equipment Holder objects directly contained in a specified Equipment Holder.

   - This method differs from the previous one in that it only looks at the next level of the containment hierarchy.

22) List of all PTPs (including FTPs) supported by an Equipment (just the names, or with all associated attributes).

   - The PTPs that are returned are those that share their physical layer with the primary equipment (i.e. that represent a port on the equipment or are connected by a fibre, wire, etc.).
   - When there is equipment protection, this operation reports PTPs for the primary equipment only. That is, when called on a protecting equipment (as opposed to the primary equipment), this operation returns an empty list, irrespective of the current switch status.

23) List of all Equipment objects which implement a PTP (or an FTP). (just the names, or with all associated attributes).

   - The equipments that are returned are those which support the physical layer of the PTP (i.e. have the port on them or are connected by a fibre, wire, etc.).
   - For a particular PTP the Tx port and Rx port may be on different cards and in this case both should be returned. Equipment that are used by the PTPs, but that do not support them directly (such as a shared DEMUX card) are not reported.

24) List of all protection groups. Refer to Use Case 7.8.1: NMS retrieves all the Protection Groups of a Managed Element.

25) List of names of CTPs that can carry protected traffic (protected TPs).

26) List of names of CTPs that can carry preemptible extra traffic (preemptible TPs).

27) List of names of CTPs that are configured to carry Non-preemptible Unprotected extra Traffic (NUT TPs).

28) List of all ASAPs of the EMS.

29) List of all Matrix Flow Domains contained in a Managed Element (the names or all associated attributes).

30) List of all Matrix Flow Domains within a Flow Domain.

31) The Flow Domain that a Matrix Flow Domain is associated to.

32) List of all CPTPs assigned to a Matrix Flow Domain.

33) List of all CPTPs which are potentially able to be assigned to a Matrix Flow Domain.

34) The Matrix Flow Domain to which a CPTP has been assigned.

Use Case 7.4.4: NMS queries EMS concerning inventory

	35) List of all Equipments supporting a Matrix Flow Domain. 36) List of all Matrix Flow Domains supported by an Equipment. 37) List of all Matrix Flow Domain names associated to a Transmission Descriptor. 38) List of all Flow Domains managed by the EMS (the names or all associated attributes). 39) List of all Flow Domains having a specific User Label. 40) List of all Edge CPTPs associated to a Flow Domain. 41) List of all Matrix Flow Domains supporting a Flow Domain (the names or all associated attributes). 42) List of all Bandwidth Profiles managed by the EMS (the names or all associated attributes). 43) List of all Flow Domain Fragments contained in a Flow Domain (the names or all associated attributes). 44) The Flow Domain Fragment connecting a specific Flow Point (CTP). 45) List of all Flow Domain Fragments connecting Flow Points of a specific CPTP. 46) List of all Flow Domain Fragments having a specific User Label. 47) List of all TP names associated to a Traffic Conditioning Profile. 48) List of all fdEdge FPs (the names or all associated attributes) used by a specified FDFr. 49) List of all fdInternal FPs (the names or all associated attributes) used by a specified FDFr. 50) List of all FPs (the names or all associated attributes) used by a specified FDFr.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS determines that it needs to query the EMS concerning a particular network inventory item.
Begins when	The NMS sends an inventory query to the EMS.
Description	1) The NMS sends a network inventory query to the EMS. 2) The EMS provides the requested network inventory information to the NMS.
Ends when	In case of success: The NMS receives the requested network inventory information. In case of failure: The NMS receives a negative response from the NMS or the request times out.

Use Case 7.4.4: NMS queries EMS concerning inventory

Post-conditions	In case of success: The NMS has received the requested network inventory information from the EMS. In case of failure: The EMS has not received the requested network inventory information.
Exceptions	1) Communication failure between the NMS and the EMS. 2) The NMS may include incorrect or unknown information in its queries to the EMS. See Use Case 7.4.2: NMS resynchronizes its database with the EMS. 3) Query unknown or not supported. 4) Unable to comply.
Traceability	{Requirement II. 002}, {Requirement II. 004}, {Requirement II. 006}, {Requirement II. 011}, {Requirement II. 008}, {Requirement II. 012}, {Requirement II. 015}, {Requirement II. 018}, {Requirement II. 020}, {Requirement II. 022}, {Requirement II. 024}, {Requirement II. 026}, {Requirement II. 033}, {Requirement II. 040}, {Requirement II. 047}, {Requirement II. 050}, {Requirement II. 052}, {Requirement II. 056}, {Requirement II. 058}. {Requirement II. 296}, {Requirement II. 318}, {Requirement II. 292}, {Requirement II. 303}, {Requirement II. 340}, {Requirement II. 300}, {Requirement II. 301}, , {Requirement II. 295}, {Requirement II. 316}, {Requirement II. 297}, {Requirement II. 304}, {Requirement II. 314}, {Requirement II. 317}, {Requirement II. 315},{Requirement II. 330}, {Requirement II. 338}, {Requirement II. 342}, {Requirement II. 339}, {Requirement II. 341}, {Requirement II. 332}, {Requirement II. 343}

7.4.5 EMS notifies NMS of inventory change

Use Case 7.4.5: EMS notifies NMS of inventory change

Use Case Name	EMS notifies NMS of inventory change
Summary	The EMS detects a change in the monitored network and notifies the NMS.
Actors	EMS
Pre-conditions	The NMS has executed Use Case 7.4.1: NMS registers to receive network updates information from the EMS.
Begins when	The EMS detects a change in the monitored network.
Description	1) The EMS detects a change in the monitored network and generates a notification to inform the NMS. Refer to {Requirement II. 064} for object creation notifications, {Requirement II. 065} for object deletion notifications, {Requirement II. 066} for attribute change notifications and {Requirement II. 067} for state change notifications. 2) The NMS receives the notification from the Notification Service.
Ends when	In case of success: The NMS receives the notification. In case of failure: The NMS does not receive the notification.
Post-conditions	In case of success: The NMS database remains aligned with the EMS's database. In case of failure: The NMS database is misaligned with the EMS's database.
Exceptions	
Traceability	{Requirement II. 064}, {Requirement II. 065}, {Requirement II. 066}, {Requirement II. 067}

7.5 Provisioning Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.5.1:	NMS provisions the mapping mode of a CTP	{Requirement II. 068}
Use Case 7.5.2:	NMS un-maps a server layer CTP	{Requirement II. 069}
Use Case 7.5.3:	NMS provisions the User Label	{Requirement II. 075}
Use Case 7.5.4:	NMS provisions the Owner	{Requirement II. 076}
Use Case 7.5.5:	NMS provisions the Additional Information	{Requirement II. 223}
Use Case 7.5.6:	NMS provisions the Native EMS Name	{Requirement II. 077}
Use Case 7.5.7:	NMS Provisions the TP Transmission Parameters	{Requirement II. 072}

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.5.8:	NMS provisions alarm reporting on for a TP	{Requirement II. 108}
Use Case 7.5.9:	NMS provisions alarm reporting off for a TP	{Requirement II. 109}
Use Case 7.5.10:	NMS creates a Topological Link (TL)	{Requirement II. 168}
Use Case 7.5.11:	NMS deletes a Topological Link (TL)	{Requirement II. 170}
Use Case 7.5.12:	NMS creates a Transmission Descriptor (TMD)	{Requirement II. 190}
Use Case 7.5.13:	NMS modifies a Transmission Descriptor (TMD) on a TP	{Requirement II. 194}
Use Case 7.5.14:	NMS sets the Transmission Descriptor (TMD) Profile Pointer	{Requirement II. 194}, {Requirement II. 277}, {Requirement II. 312}, {Requirement II. 313}
Use Case 7.5.15:	NMS modifies a Transmission Descriptor (TMD)	{Requirement II. 353}
Use Case 7.5.16:	NMS retrieves scoped Transmission Parameters	{Requirement I. 102}, {Requirement II. 302}, {Requirement II. 319}, {Requirement II. 344}, {Requirement II. 352}
Use Case 7.5.17:	NMS deletes a Transmission Descriptor (TMD)	{Requirement II. 192}
Use Case 7.5.18:	NMS creates a Group Termination Point (GTP)	{Requirement II. 164}, {Requirement II. 165}
Use Case 7.5.19:	NMS modifies a Group Termination Point (GTP)	{Requirement II. 167}
Use Case 7.5.20:	NMS deletes a Group Termination Point (GTP)	{Requirement II. 166}
Use Case 7.5.21:	NMS creates a Termination Point Pool (TP Pool)	{Requirement II. 264}, {Requirement II. 265}
Use Case 7.5.22:	NMS modifies a Termination Point Pool (TP Pool)	{Requirement II. 267}
Use Case 7.5.23:	NMS deletes a Termination Point Pool (TP Pool)	{Requirement II. 266}
Use Case 7.5.24:	NMS assigns an Alarm Severity Assignment Profile (ASAP) to a CTP	{Requirement II. 201}
Use Case 7.5.25:	NMS locks (in a forced response deferred/graceful fashion) or unlocks a number of IMA links to modify the transport capacity of the corresponding fixed IMA group	
Use Case 7.5.26:	NMS requests dynamic provisioning of an IMA group by the EMS subject to a prescribed bandwidth that is communicated as number and connectable layer rate of the IMA links	
Use Case 7.5.27:	NMS provisions the IMA virtual link between two peer IMA groups as a topological link	
Use Case 7.5.28:	NMS unprovisions an IMA virtual link between IMA groups	
Use Case 7.5.29:	NMS provisions or re-provisions a single DSL line	

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.5.30:	NMS provisions multiple DSL lines by using TMDs	

7.5.1 NMS provisions the mapping mode of a CTP

Use Case 7.5.1: NMS provisions the mapping mode of a CTP

Use Case Name	NMS provisions the mapping mode of a CTP
Summary	The NMS sets the mapping mode of a CTP (e.g., DS3, STS1, or VC4) to support client layer rate connections.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS knows the CTP it wishes to terminate and map, e.g. has a handle for the CTP. 3) CTP has to be able to be terminated and mapped.
Begins when	The NMS sends a request to the EMS to terminate and map the CTP (server layer CTP).
Description	1) The EMS validates the request against the server layer CTP specified as described in steps 2 through 8. 2) The EMS validates if specified CTP exists. If not exception 1) is thrown. 3) The EMS validates if the specified CTP is capable of being mapped. 4) The EMS validates if the specified CTP is involved in an existing server layer cross-connection. If yes, the CTP cannot be terminated and mapped. 5) The EMS validates ME containing the specified CTP is not accessible, if not then exception 4) is thrown. 6) The EMS ignores any pending SNCs involving this CTP at either the server layer rate or a client layer rate. 7) Note that it may be possible to set a CTP that is involved in a partial subnetwork connection to terminated and mapped (even if the partial SNC is at the server layer rate) if this CTP is not involved in an existing active cross-connection. 8) The EMS validates if the CTP was already terminated and mapped, then the operation shall be considered a success. 9) The EMS terminates and maps the CTP specified. An AVC will be sent to the Notification Service. 10) If in the specific implementation of NE/EMS the client layer CTPs are actually created as a result of this operation, then it is assumed that the alarm reporting state of the newly created client layer CTPs will be disabled by default. Object creation notifications will not be generated for the newly created client layer CTPs. The EMS sends a response to the NMS
Ends when	In case of success: The NMS receives a response. In case of failure: The NMS receives a negative response or exception.

Use Case 7.5.1: NMS provisions the mapping mode of a CTP

Post-conditions	In case of success: 1) The server layer trail termination functions are available. 2) The CTP (including server layer trail termination functions) alarm reporting state remains unchanged from its setting prior to use of this operation. If the CTP had not previously existed on the NE, then it is assumed that the alarm reporting state of the newly created CTP will be disabled by default. 3) If in the specific implementation of NE/EMS the client layer CTPs are actually created then it is assumed that the alarm reporting state of the client layer CTPs will be disabled by default on application of the terminate and map operation. 4) AVC notifications are sent to the notification service. In case of failure: None.
Exceptions	1) Entity not found: The specified CTP does not exist. 2) Processing failure: The requested operation could not be performed. 3) Unable to comply: The CTP is involved in an existing, active cross connection at the CTP's native rate (CTP's layer rate); or the CTP is not in a valid state. 4) NE communication loss
Traceability	{Requirement II. 068},

7.5.2 NMS un-maps a server layer CTP

Use Case 7.5.2: NMS un-maps a server layer CTP

Use Case Name	NMS un-maps a server layer CTP
Summary	The NMS sets the mapping mode of a server layer CTP (e.g., DS3, STS1, or VC4) to no longer be mapped to client layer capacity (e.g., VT Group/TUG/VT1.5/VC12 etc.) and to no longer terminate the corresponding server layer G.805 trail.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS has a handle for a CTP that it wishes to modify. 3) CTP has to be able to be un-terminated and un-mapped.
Begins when	The NMS sends a request to the EMS to un-terminate and un-map a specified server layer CTP
Description	1) The EMS validates the request against the server layer CTP specified 2) The EMS validates if the specified server layer CTP exists 3) The EMS validates if the specified server layer CTP is capable of being un-mapped 4) The EMS validates if specified server layer CTP is involved in (is not supporting) an existing Active client layer cross-connection (i.e., the client CTPs (if any exist) are in a not connected connection state). If yes, the server layer CTP cannot be un-mapped. 5) Note that it is possible to adjust a Server layer CTP that is involved in a pending subnetwork connection, whether there are pending SNCs at a client layer rate or at the server layer rate. In other words, this operation ignores pending subnetwork connections involving the Server layer CTP or any client CTPs. 6) Note that it is possible to adjust a CTP that is involved in a partial subnetwork connection (even if the partial SNC is at a client layer rate) if none of the client CTPs are involved in an existing active client layer cross-connection. 7) If the Server layer CTP was already un-terminated and unadapted, then the operation shall be considered a success. 8) The EMS sets the mapping mode on the Server layer CTP to un-channelized/un-terminated. AVC will be sent to the Notification Service. 9) The EMS applies any necessary operations on the NE to deactivate traffic and management functions on the G.805 TTP, e.g., disables path trace processing, disables multiplexing/channelizing/mapping functions (makes the server layer CTP available for cross-connection at the server layer), removes VT visibility, etc. The alarm reporting state of the server layer CTP (and aggregated G.805 TTP) remains unchanged from its setting prior to use of this operation. 10) The EMS sends a response to the NMS.

Use Case 7.5.2: NMS un-maps a server layer CTP

Ends when	In case of success: The NMS receives a response. In case of failure: The NMS receives a negative response.
Post-conditions	In case of success: 1) The appropriate structure is set on the NE so that the Server layer CTP (e.g., an STS1 CTP) is then available to be part of an SNC at the CTP rate (e.g., STS1). 2) The Server layer CTP (and aggregated G.805 TTP) alarm reporting state remains unchanged from its setting prior to use of this operation. 3) AVC notifications are sent to the Notification Service. In case of failure: None. NMS has received a negative response.
Exceptions	1) Entity not found: The specified CTP does not exist 2) Processing failure: The requested operation could not be performed. 3) Unable to comply: The CTP is supporting (containing) one or more lower order (client) CTPs that are involved in existing, active cross connections at the lower order CTP rates. 4) NE communication loss
Traceability	{Requirement II. 069}

7.5.3 NMS provisions the User Label

Use Case 7.5.3: NMS provisions the User Label

Use Case Name	NMS provisions the User Label.
Summary	The NMS provisions the User Label of an object.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS knows the object for which it wishes to set the user label.
Begins when	The NMS sends a request to the EMS to set the user label of an object and specifies whether the user label has to be unique among all objects of the same class as this object.
Description	1) If the NMS has specified that the user label has to be unique, the EMS checks if any other object of the same class has the given user label. If yes, exception 3) is thrown. 2) EMS sets the user label to the specified value. 3) If there has been a change in the value of the user label attribute, then the EMS forwards an AVC to the Notification Service. 4) EMS replies with a success indication.
Ends when	The NMS receives a response or a failure exception.
Post-conditions	In case of success: User label of the specified object is set/modified and the corresponding AVCs are sent to the notification service. In case of failure: None
Exceptions	1) Entity not found: The specified object does not exist. 2) Processing failure: The requested operation could not be performed. 3) Unable to comply: The User Label can not be set. 4) User label in use: The given user label is already being used. 5) Invalid input: The specified name is invalid. 6) NE communication loss.
Traceability	{Requirement II. 075}

7.5.4 NMS provisions the Owner

Use Case 7.5.4: NMS provisions the Owner

Use Case Name	NMS provisions the Owner.
Summary	The NMS provisions the Owner of an object.
Actors	NMS

Use Case 7.5.4: NMS provisions the Owner

Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS knows the object for which it wishes to set the owner.
Begins when	The NMS sends a request to the EMS to set the owner of an object.
Description	1) EMS sets the owner attribute of the specified object to the specified value. 2) If there has been a change in the value of the owner attribute, then the EMS forwards an AVC to the notification service. 3) EMS replies with a success indication
Ends when	The NMS receives a response or a failure exception.
Post-conditions	In case of success: Owner of specified object is set/modified and the corresponding AVC is sent to the notification service. In case of failure: None
Exceptions	1) Entity not found: The specified object does not exist. 2) Processing failure: The requested operation could not be performed.
Traceability	{Requirement II. 076}

7.5.5 NMS provisions the Additional Information

Use Case 7.5.5: NMS provisions the Additional Information

Use Case Name	NMS provisions the Additional Information.
Summary	The NMS provisions the Additional Information of an object.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS knows the object for which it wishes to set the additional information.
Begins when	The NMS sends a request to the EMS to set the additional information of an object.
Description	1) EMS sets the additional information attribute of the specified object to the specified value. As an input only the list of parameters to be changed, removed, or added is provided. If an entry is to be removed a special value is applied. If a parameter is specified that is currently not part of the additional information attribute of the specified object that parameter is added by the EMS with the specified value. The EMS may reject removal and addition requests. 2) If there has been a change in the value of the additional information attribute, then the EMS forwards an AVC to the notification service. 3) EMS replies with a success indication
Ends when	The NMS receives a response or a failure exception.
Post-conditions	In case of success: Additional information of the specified object is set/modified and the corresponding AVC is sent to the notification service. In case of failure: None
Exceptions	1) Entity not found: The specified object does not exist. 2) Processing failure: The requested operation could not be performed. 3) Unable to comply: The EMS is unable or unwilling to execute the request. 4) Invalid input: The input parameters are syntactically incorrect. 5) NE communication loss.
Traceability	{Requirement II. 223}

7.5.6 NMS provisions the Native EMS Name

Use Case 7.5.6: NMS provisions the Native EMS Name

Use Case Name	NMS provisions the Native EMS Name
Summary	The NMS provisions the native EMS name of an object.
Actors	NMS

Use Case 7.5.6: NMS provisions the Native EMS Name

Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS knows the object for which it wishes to set the native EMS name. 3) The EMS supports setting of native EMS name by NMS.
Begins when	The NMS sends a request to the EMS to set the native EMS name of an object.
Description	1) EMS sets the native EMS name of the specified object to the specified value. 2) If there has been a change in the native EMS name, then the EMS forwards an AVC to the notification service. 3) EMS replies with a success indication.
Ends when	The NMS receives a response or a failure exception.
Post-conditions	In case of success: Native EMS Name of specified object is set/modified and the corresponding AVC is sent to the notification service. In case of failure: None
Exceptions	1) Entity not found: The specified object does not exist. 2) Processing failure: The requested operation could not be performed. 3) Invalid input: The specified name is invalid. 4) Unable to comply: The native EMS name can not be set. 5) NE communication loss.
Traceability	{Requirement II. 077}

7.5.7 NMS provisions the TP Transmission Parameters

Use Case 7.5.7: NMS Provisions the TP Transmission Parameters

Use Case Name	NMS Provisions the TP Transmission Parameters
Summary	1) The NMS requests that a termination point's (TP) transmission parameters be provisioned. For example, the types of TP transmission parameters that can be provisioned include for SONET/SDH/DWDM: • For DS1: frame format (e.g. SF, ESF, and Unframed) and line coding (e.g. B8ZS and AMI) • For DS3: frame format (e.g. M23, C-bit parity, and Unframed) • For STS-1: expected incoming and outgoing path trace • For DWDM TP: TunnedFequency, etc. 2) The NMS must be aware that the provisioning of a TP transmission parameter when the TP is actively involved in a SNC may cause service disruption. The provisioning of a connected TP does not involve in the tearing down of the associated cross-connect. Note: The specific TP transmission parameters and values varies across different vendors and technologies.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends the request to set a transmission parameter(s) on the TP.
Description	1) The NMS sends the request to set transmission parameter(s) on the specified TP. 2) The EMS validates the TP reference (e.g. name). 3) The EMS validates the request (e.g. transmission parameter name(s) exists and value(s) are valid) and supported. 4) The EMS sets the specified TP's transmission parameter(s) to the specified value(s). 5) If there has been a transmission parameter value change(s), then the EMS forwards attribute value change notification(s) to all subscribing NMSs. 6) The EMS replies with a success indication.
Ends when	In case of success: The NMS receives an indication of success of the action. In case of failure: The NMS receives an indication of failure of the action.

Use Case 7.5.7: NMS Provisions the TP Transmission Parameters

Post-conditions	In case of success: 1) The TP's transmission parameter(s) have been set to the specified value. 2) The EMS has forwarded attribute value change notification(s) if the TP's transmission parameter(s) have changed in value. In case of partial success: If the request to set the transmission parameter(s) succeeded and the TP's transmission parameter(s) have changed in value, then the EMS has forwarded attribute value change notification(s) to all subscribing NMSs. In case of failure: Appropriate exception is reported to the NMS.
Exceptions	1) Processing failure: The requested operation could not be performed. 2) Invalid input: The TP reference is invalid. 3) NE communication loss.
Traceability	{Requirement II. 072}

7.5.8 NMS provisions alarm reporting on for a TP

Use Case 7.5.8: NMS provisions alarm reporting on for a TP

Use Case Name	NMS provisions alarm reporting on for a TP
Summary	The NMS requests to activate all alarm reporting on a termination point (TP).
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends the request to activate alarm reporting to the EMS.
Description	1) The NMS sends the request to activate alarm reporting on for a specified TP. 2) The EMS validates the TP reference (e.g. name). 3) The EMS enables alarm reporting on the specified TP. The alarm reporting state of the contained TP(s) may also be enabled. 4) The EMS replies with a success indication. 5) Attribute Value Change notification(s) for the specified TP and the contained TP(s), if any, are forwarded to the notification service indicating that alarm reporting has been activated for these TP(s).
Ends when	In case of success: The NMS receives an indication of success of the action. In case of failure: The NMS receives an indication of failure of the action.
Post-conditions	In case of success: 1) Alarm monitoring is enabled on the specified TP. This does not mean that alarm is reported anyway, because Alarm Severity Assignment Profile may perform further filtering. 2) The EMS has forwarded an attribute value change notification if there was an attribute value change with the enabling of alarm monitoring on the TP. In case of failure: None.
Exceptions	1) Processing failure: The requested operation could not be performed. 2) Invalid input: The TP reference is invalid. 3) NE communication loss.
Traceability	{Requirement II. 108}

7.5.9 NMS provisions alarm reporting off for a TP

Use Case 7.5.9: NMS provisions alarm reporting off for a TP

Use Case Name	NMS provisions alarm reporting off for a TP

Use Case 7.5.9: NMS provisions alarm reporting off for a TP

Summary	The NMS requests to deactivate alarm reporting on a specified termination point (TP). Notes: There are no side effects upon transmission behavior (propagated alarm signals e.g. AIS) associated with the TP.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends the request to deactivate alarm reporting to the EMS.
Description	1) The NMS sends the request to deactivate alarm reporting on the specified TP. 2) The EMS validates the TP reference (e.g. name). 3) The EMS disables alarm reporting on the specified TP. The alarm reporting state of the contained TP(s) may also be disabled. This disables alarm reporting even if Alarm Severity Assignment Profile would allow it. 4) The EMS replies with a success indication. 5) Attribute Value Change notification(s) for the specified TP and the contained TP(s), if any, are forwarded to the notification service indicating that alarm reporting has been deactivated for these TP(s).
Ends when	In case of success: The NMS receives an indication of success of the action. In case of failure: The NMS receives an indication of failure of the action.
Post-conditions	In case of success: 1) Alarm reporting is disabled on the specified TP and all the contained TP(s).. 2) The EMS has forwarded an attribute value change notification. In case of failure: None.
Exceptions	1) Processing failure: The requested operation could not be performed. 2) Invalid input: The TP reference is invalid. 3) NE communication loss.
Traceability	{Requirement II. 109}

7.5.10 NMS creates a Topological Link (TL)

Use Case 7.5.10: NMS creates a Topological Link (TL)

Use Case Name	NMS creates a Topological Link (TL)
Summary	The NMS creates a Topological Link (TL).
Actors	NMS.
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The Topological Link end TPs have to exist
Begins when	The NMS sends the request to the EMS to create a Topological Link.
Description	1) The EMS validates the request syntax. If not exception 1) is thrown. 2) The EMS validates the TL name is not already in use. As owner of TPs and TLs, the EMS may perform this validation against data that was manually entered at the EMS or auto-discovered. This EMS authority to use auto-discovered information also applies to the following data validations. If not exception 2) is thrown. 3) The EMS validates that the A and Z end TPs referenced in the request exist in its domain of control. If not exception 3) is thrown. 4) The EMS validates that neither TP is already assigned to a TL. If not exception 2) is thrown. 5) The EMS validates that the rate specified in the request, as well as any other additional creation information, is consistent with its own data. If not exception 2) is thrown. 6) As owner of TPs and TLs, the EMS validates the request against its own criteria. If not exception 4) is thrown. 7) The EMS creates the TL. 8) The EMS sends a response to the NMS.
Ends when	In case of success: The NMS receives an indication of success of the action. In case of failure: The NMS receives an indication of failure of the action.
Post-conditions	In case of success: The TL is available. In case of failure: None. NMS has received an indication of failure of the action or exception

Use Case 7.5.10: NMS creates a Topological Link (TL)

Exceptions	1) Invalid input: request syntax is not valid. 2) Object in use: the request attempts to create a TL that already exists, or the request references TPs that are already associated with TLs. 3) Entity not found: the request references an entity that does not exist. 4) Unable to comply: the EMS rejects the request based on its own criteria. 5) Processing failure: The requested operation could not be performed. 6) User label in use: The user label supplied by the NMS is already in use.
Traceability	{Requirement II. 168}

7.5.11 NMS deletes a Topological Link (TL)

Use Case 7.5.11: NMS deletes a Topological Link (TL)

Use Case Name	NMS deletes a Topological Link (TL)
Summary	The NMS deletes a Topological Link (TL).
Actors	NMS.
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS has identified an existing Topological Link for deletion.
Begins when	The NMS sends the request to the EMS to delete a Topological Link.
Description	1) The EMS validates the request syntax. If not exception 1) is thrown. 2) The EMS validates that the TL exists in its span of control. If not, exception 2 is thrown. 3) As owner of TLs, the EMS validates the request against its own criteria. If not exception 3) is thrown. 4) The EMS deletes the TL. 5) The EMS sends a response to the NMS.
Ends when	In case of success: The NMS receives an indication of success of the action. In case of failure: The NMS receives an indication of failure of the action.
Post-conditions	In case of success: The TL is deleted. In case of failure: None.
Exceptions	1) Invalid input: request syntax is not valid. 2) Entity not Found: the request references a TL that does not exist. 3) Unable to Comply: the EMS rejects the request based on its own criteria. 4) Processing failure: The requested operation could not be performed.
Traceability	{Requirement II. 170}

7.5.12 NMS creates a Transmission Descriptor (TMD)

Use Case 7.5.12: NMS creates a Transmission Descriptor (TMD)

Use Case Name	NMS creates a Transmission Descriptor (TMD)

Use Case 7.5.12: NMS creates a Transmission Descriptor (TMD)

Summary	The use case describes how the NMS requests that an EMS create a Transmission Descriptor (TMD). The NMS passes the TMD information to the EMS. The resulting TMD is returned as a result to the NMS.
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS sends the create TMD request to EMS.
Description	1) The NMS sends a create TMD request to EMS with the TMD creation information. The NMS may request that the EMS enforce a unique user label (the user label is selected by the NMS and the EMS makes sure the user label is not already in use). 2) The EMS validates the request. The EMS creates the TMD, assigns a unique name to the TMD and stores it persistently. 3) The EMS replies with a success indication. 4) A TMD object creation event is sent by EMS to the notification service. Note: It is up to the internal implementation of EMS which data are stored in EMS persistently and which ones will be queried from ME as required.
Ends When	The EMS sends a reply to the NMS.
Post-Conditions	The TMD has been created in the EMS
Exceptions	1) Not implemented: The EMS does not support this service. 2) Invalid input: The NMS has supplied invalid input data. 3) User label in use: The user label supplied by the NMS is already in use. 4) Capacity exceeded: The Input data is valid but the EMS cannot create the TMD because the EMS has already exceeded its maximum number of TMDs. 5) Processing failure: The requested operation could not be performed.
Traceability	{Requirement II. 190}

7.5.13 NMS modifies a Transmission Descriptor (TMD) on a TP

Use Case 7.5.13: NMS modifies a Transmission Descriptor (TMD) on a TP

Use Case Name	NMS modifies a Transmission Descriptor (TMD) on a TP
Summary	The NMS changes the Transmission Descriptor (TMD) (ingress and/or egress) for a TP.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) TransmissionDescriptor to be associated has to exist.
Begins when	The NMS sends the modify TP request to the EMS.
Description	1) The NMS sends the modify TP request to the EMS. 2) The EMS validates the TP identifier. 3) The EMS validates the TMD Name(s). 4) The EMS communicates with the Network Elements (NE). 5) The EMS changes the TMD(s) if possible. This will fail under the following conditions: • The TP is terminating, but was not explicitly created. • The input data is valid but the EMS cannot change the TMD because resources are not available on the NE. • The input data is valid but the EMS cannot change the TMD because the new TMD does not provide enough resources for the existing SNCs running over the TP. 6) The EMS replies with a success indication. 7) The EMS will generate an AVC for the TP with the new TMD name.
Ends when	The EMS sends a reply to the NMS.
Post-conditions	The TMD(s) on the TP has been updated.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Internal Error: The request has resulted in an EMS internal error. 3) Invalid Input: The request contains an input parameter that is syntactically incorrect or identifies an object of the wrong type or is out of range. 4) Entity Not Found: The specified object instance does not exist. 5) Unable To Comply: The EMS cannot perform the the request. 6) NE communication loss: The EMS is unable to communicate with the NE and communication is required to complete the request.
Traceability	{Requirement II. 194}

7.5.14 NMS sets the Transmission Descriptor (TMD) Profile Pointer

Use Case 7.5.14: NMS sets the Transmission Descriptor (TMD) Profile Pointer

Use Case Name	NMS sets the Transmission Descriptor (TMD) Profile Pointer
Summary	The purpose of this use case is to configure the Layered Transmission Parameters of a Termination Point (TP) or a Matrix Flow Domain (MFD) by modifying the assignment of the Transmission Descriptor (TMD) to this TP/MFD.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends the request to the EMS.

Use Case 7.5.14: NMS sets the Transmission Descriptor (TMD) Profile Pointer

Description	1) The NMS sends the request to the EMS to set the TMD pointer. 2) The NMS may provide: • an ingress TMD name (in case of a TP) or • an egress TMD name (in case of a TP)or • a TMD name (in case of an MFD) If a TMD has to be removed, the NMS has to provide an “empty TMD name” instead of the TMD name. 3) The EMS checks the syntax of the request. If the request is not valid, an Invalid Input exception is raised. 4) If a TMD has to be removed: The EMS checks the existence of the TMD to be removed. If the TMD is not associated, an Entity Not Found exception is raised. Only the pointer attribute in the TP/MFD which contains the name of this TMD has to be updated and an AVC notification has to be send. Note: The parameter values in the TP/MFD remain unchanged. 5) If a TMD has to be added: The EMS checks the existence of the TMD to be added. If the TMD does not exist, an Entity Not Found exception is raised. If the TMD to be added is already associated to the TP/MFD, the EMS shall overwrite all corresponding parameters in the TP/MFD with the values contained in the TMD. Parameters which are not already in the TP/MFD will be added to the TP/MFD. This request ensures consistency between the parameter values of the TMD and the parameter values of the TP/MFD. (Note: Individual values may have been changed in the TP/MFD before via the use cases "NMS provisions the TP Transmission Parameters" and "NMS modifies a Matrix Flow Domain".) If the parameter values can not be set on the TP/MFD, an Unable To Comply exception is raised. If the TMD is not already assigned to the TP/MFD the EMS shall set all parameter values as defined in the TMD profile. If the parameter values can not be set on the TP/MFD, an Unable To Comply exception is raised. The pointer attribute in the TP/MFD which contains the name of this TMD has to be updated and an AVC notification has to be send.
Ends when	In case of success: The parameter values of the TP/MFD are consistent with the parameter values of the (newly) associated TMD profile. In case of failure: The NMS receives an exception as an indication of the failure of the request.

Use Case 7.5.14: NMS sets the Transmission Descriptor (TMD) Profile Pointer

Post-conditions	In case of success: The parameter values of the TP/MFD are consistent with the parameter values of the (newly) associated TMD profile. In case of failure: Nothing has changed in the System.
Exceptions	Refer to Table 7.1
Traceability	{Requirement II. 194}, {Requirement II. 277}, {Requirement II. 312}, {Requirement II. 313}

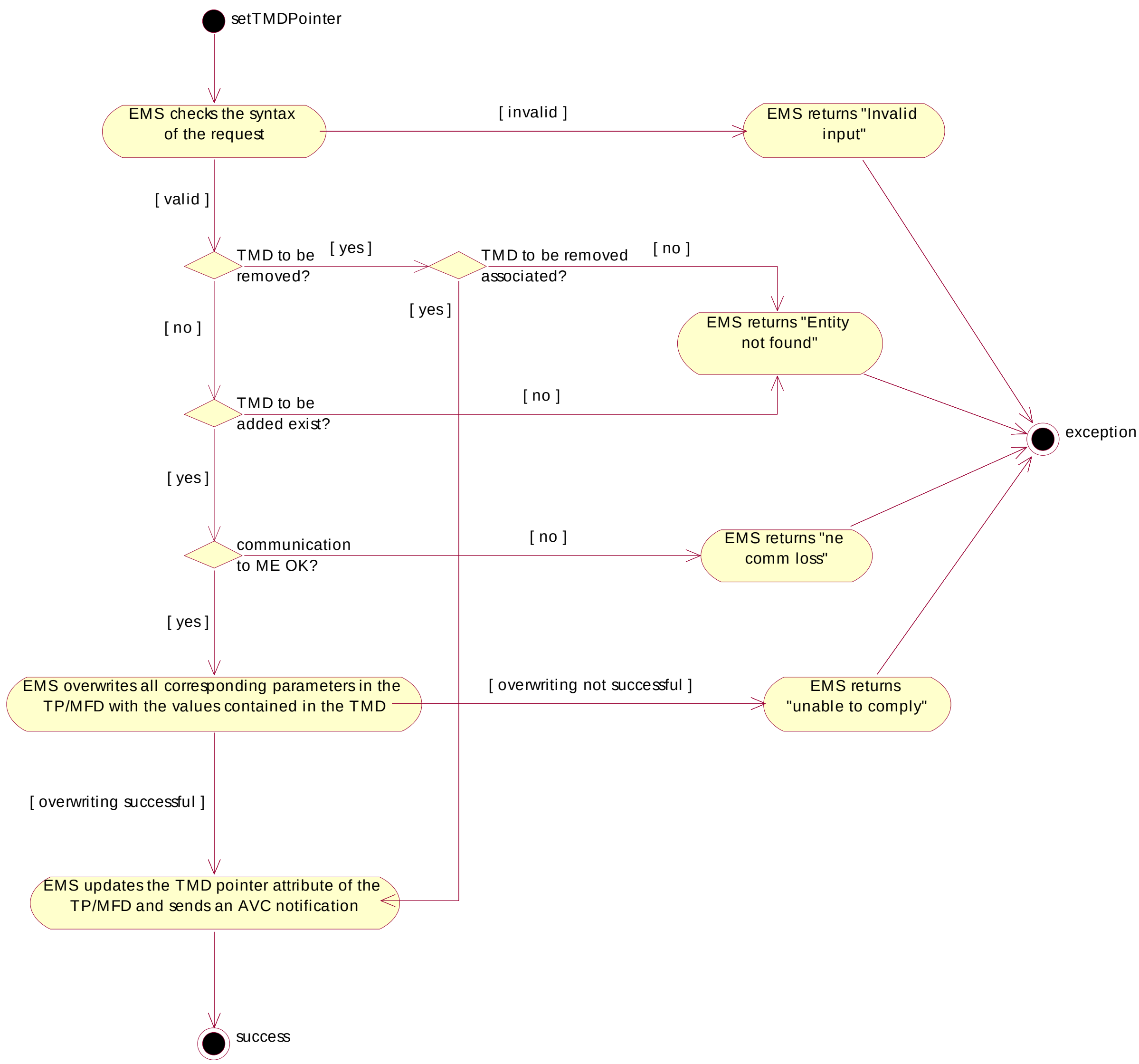


Figure 7.2: Activity Diagram for NMS sets the Transmission Descriptor (TMD) Profile Pointer

7.5.15 NMS modifies a Transmission Descriptor (TMD)

Use Case 7.5.15: NMS modifies a Transmission Descriptor (TMD)

Use Case Name	NMS modifies a Transmission Descriptor (TMD)
Summary	The purpose of this use case is to modify the attributes and transmission parameters in an already created Transmission Descriptor (TMD). This operation overwrites specific parameter values of the TMD with the new provided parameter values. Existing Parameters can also be removed from the TMD. The corresponding parameter values of all associated TPs/MFDs are changed accordingly.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The TMD to be modified must exist. 3) A communication between the EMS and the relevant MEs has to be active.
Begins when	The NMS sends the request to the EMS.
Description	1) The NMS sends the request to the EMS to modify the TMD parameters. 2) The NMS provides the list of attribute values and transmission parameter values which have to be modified in the TMD. 3) The EMS checks the syntax of the request. If the request is not valid, an Invalid Input exception is raised. 4) The EMS checks the existence of the TMD to be modified. If the TMD does not exist, an Entity Not Found exception is raised. 5) The EMS overwrites the attributes and parameters of the TMD with the provided ones. This includes the deletion of parameters. If it is not possible to change any attribute/parameter value, an Unable To Comply exception is raised. 6) The EMS checks the connection to all MEs that contains associated TPs or MFDs. If connection to all MEs is lost, an NE Comm Loss exception is raised. Otherwise the EMS has to return the names of the MEs which are not connected. 7) The EMS changes all parameter values in all TPs and MFDs associated to this TMD according to the modified values; i.e., only the modified parameters will be updated. Note: Parameters which have been deleted in the TMD will not be changed in the associated TPs and MFDs. 8) The EMS has to return the names of all TPs and MFDs that could not be changed to the new parameter values due to some error reasons.
Ends when	In case of success: The EMS returns the distinguishing information of the modified TMD. In case not all associated TPs/MFDs could be updated to the modified attribute and parameter values, the EMS returns the list of these TPs/MFDs. In case of failure: The NMS receives an exception as an indication of the failure of the request.

Use Case 7.5.15: NMS modifies a Transmission Descriptor (TMD)

Post-conditions	In case of success: The TMD contains the new provided attribute and parameter values. Every TP/MFD (except the TPs/MFDs returned as failed TPs/MFDs) associated to this TMD has been set to the new parameter values and will use the new values. In case of failure: Nothing has changed in the System.
Exceptions	Refer to Table 7.1
Traceability	{Requirement II. 353}

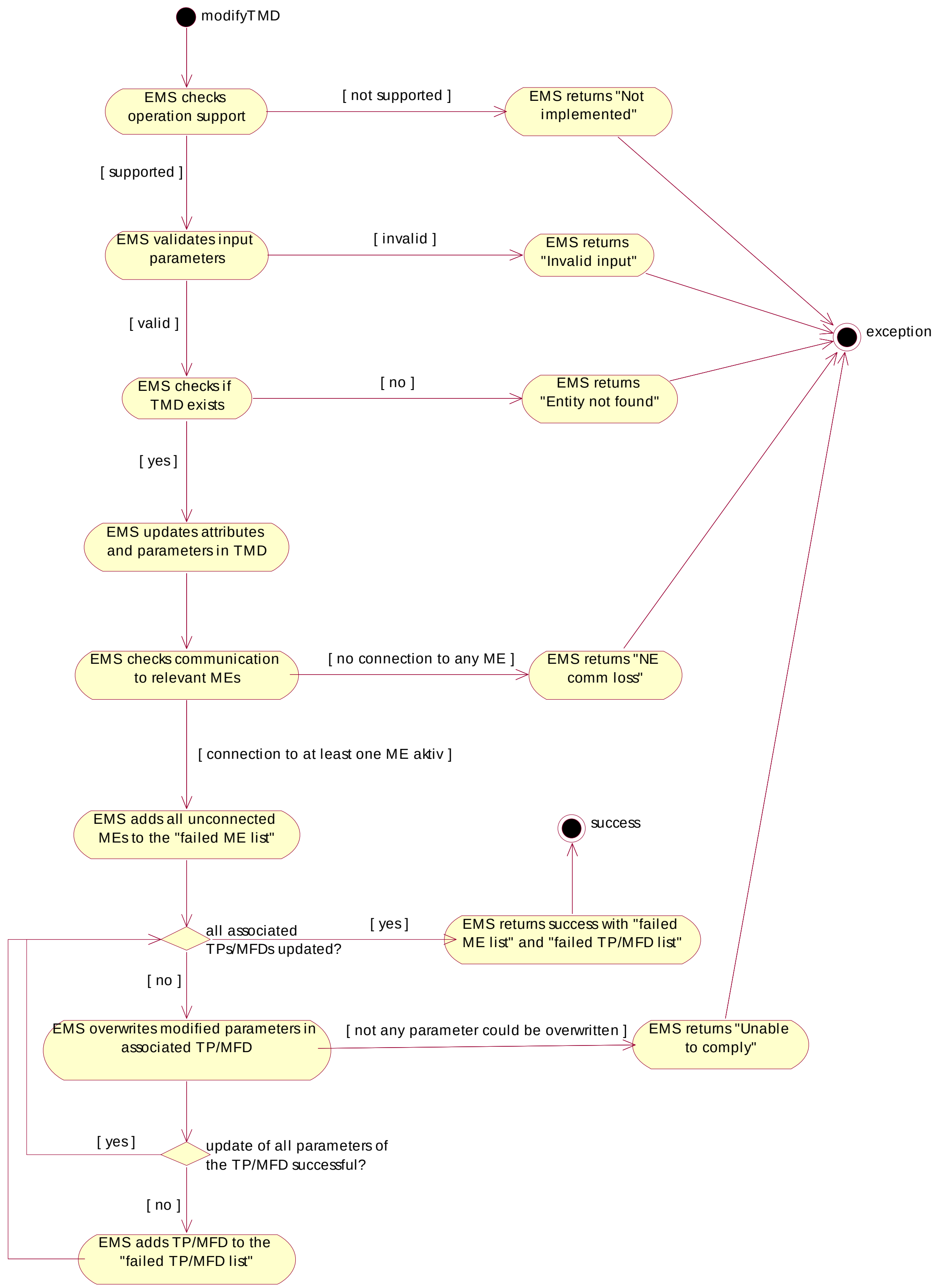


Figure 7.3: Activity Diagram for NMS modifies a Transmission Descriptor (TMD)

7.5.16 NMS retrieves scoped Transmission Parameters

Use Case 7.5.16: NMS retrieves scoped Transmission Parameters

Use Case Name	NMS retrieves scoped Transmission Parameters
Summary	The purpose of this use case is to retrieve a scoped set of Transmission Parameters.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a scoped request to the EMS.
Description	1) The NMS sends the request to the EMS to retrieve the Transmission Parameters. 2) The NMS provides the object containing the Transmission Parameters to be retrieved. 3) The NMS provides the filter criteria defining the layer rates and the groups of Transmission Parameters to be retrieved. 4) The EMS checks the syntax of the request. If the request is not valid, an Invalid Input exception is raised. 5) The EMS checks the existence of the object containing the Transmission Parameters. If the object does not exist, an Entity Not Found exception is raised. 6) The EMS returns the requested set of Transmission Parameters.
Ends when	In case of success: The EMS has returned all requested Transmission Parameters. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: Nothing has changed in the System. In case of failure: Nothing has changed in the System.
Exceptions	Refer to Table 7.1
Traceability	{Requirement I. 102}, {Requirement II. 302}, {Requirement II. 319}, {Requirement II. 344}, {Requirement II. 352},

7.5.17 NMS deletes a Transmission Descriptor (TMD)

Use Case 7.5.17: NMS deletes a Transmission Descriptor (TMD)

Use Case Name	NMS deletes a Transmission Descriptor (TMD)
Summary	The NMS deletes a Transmission Descriptor (TMD). Note: • The NMS is responsible to maintain the consistency of TMD across multiple administrative subnetworks. Routine integrity checking may be required. NMS must also check that the TMD is not being used in any EMS under its network domain, before the TMD can be deleted from the persistent storage. • This operation is idempotent. If the service is called with the name of a non-existent TMD, it will succeed.
Actor(s)	NMS
Pre-Conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The Transmission Descriptor to be deleted has to exist. 3) The Transmission Descriptor has no association to a TP or an MFD.
Begins When	The NMS sends the delete TMD request to the EMS.
Description	1) The NMS sends a delete TMD request to the EMS. 2) The EMS validates the TMD identifier and checks that this TMD is not being used on the EMS. 3) If no TPs or MFDs are using this TMD, the EMS deletes the TMD. 4) The EMS replies with a success indication. 5) A TMD object deletion event is sent by EMS to the notification service.
Ends When	The EMS sends a reply to the NMS.
Post-Conditions	The TMD has been deleted in the EMS.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Communication Failure with NE. 3) Object in use: The TMD is being used. 4) Entity not found: The TMD does not exist in the EMS. 5) Invalid input: The NMS has supplied invalid input data (The TMD Name is invalid).
Traceability	{Requirement II. 192}

7.5.18 NMS creates a Group Termination Point (GTP)

Use Case 7.5.18: NMS creates a Group Termination Point (GTP)

Use Case Name	NMS creates a Group Termination Point (GTP)

Use Case 7.5.18: NMS creates a Group Termination Point (GTP)

Summary	The use case describes how the NMS requests that an EMS create a new GTP. The NMS requests the creation of the GTP by either 1) listing the CTPs that comprise the GTP, or 2) in the case of contiguous CTPs of the same layer rate, the NMS may list the first CTP and the number of following (contiguous) CTPs. The resulting GTP is returned as a result to the NMS.
Actors	NMS.
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS has defined a Topological Link
Begins when	The NMS sends the request to the EMS to create a Group Termination Point (GTP).
Description	1) The NMS sends a create GTP request to EMS with the GTP distinguishing information. The NMS can provide either a listing of the specific CTPs that comprise the GTP, or the NMS may list the first CTP and the number of following CTPs (this second approach is only valid for contiguous CTPs of the same layer rate). The NMS also needs to determine the gtpEffort level. If gtpEffort is set to EFFORT_SAME, then the EMS must create the GTP with the exact same list of CTPs as provided with the GTP creation request. If EFFORT_WHATEVER is specified, then the EMS may comply with the total bandwidth requirement by using a different set of CTPs. It should be noted that this mode (i.e., EFFORT_WHATEVER) allows for the GTP components to be instantiated at a later time by the ME (e.g., upon detection of user's signal). Therefore the operation may successfully return a new GTP with an empty listOfTPs attribute (to be updated at a later time once the component CTPs are created in the ME). 2) The EMS validates the request. The EMS creates the GTP, assigns a unique name to the GTP and stores it persistently. If the NMS has specified EFFORT_WHATEVER, the EMS may create the GTP with a different list of CTPs than specified by the NMS, or create the GTP with an empty list of CTPs. In the latter case, the EMS supplies the CTPs at some later point in time (typically upon detection of a user signal connecting to the GTP). 3) 3)The EMS replies with a success indication. 4) A Create GTP Notification is sent by EMS to the notification service. Note: It is up to the internal implementation of EMS which data are stored in EMS persistently and which ones will be queried from ME as required.
Ends when	In case of success: The NMS receives an indication of success of the action. In case of failure: The NMS receives an indication of failure of the action.

Use Case 7.5.18: NMS creates a Group Termination Point (GTP)

Post-conditions	In case of success: The GTP is available. In case of failure: None. NMS has received an indication of failure of the action or exception
Exceptions	1) Service not implemented by the EMS 2) The NMS has supplied invalid input data. 3) The user label supplied by the NMS is already in use. 4) The Input data is valid but the EMS cannot create the GTP because there are insufficient resources in the network to create the GTP.
Traceability	{Requirement II. 164}, {Requirement II. 165}

7.5.19 NMS modifies a Group Termination Point (GTP)

Use Case 7.5.19: NMS modifies a Group Termination Point (GTP)

Use Case Name	NMS modifies a Group Termination Point (GTP)
Summary	The use case describes how the NMS requests that an EMS modify an existing GTP that has a non-empty list of CTPs. The modify GTP operation is used to add or delete CTPs from a GTP. For a given request, the NMS can only add or delete CTPs, not both. It is not possible to add a CTP that is already involved in a cross connection or SNC, or that is part of another GTP. Attempts to modify a GTP that is involved in a cross connection (or SNC) should be rejected by the EMS. The operation is best-effort, i.e., the EMS will add or delete as many of the identified CTPs as possible. If the service is called with the name of a non-existent GTP or CTP, it will fail. If the GTP was initially created with gtpEffort equal to EFFORT_SAME, the EMS should reject the modification request.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The GTP to be modified already exists, and the EMS has assigned a set of CTPs to the GTP.
Begins when	The NMS sends the modify GTP request to the EMS.
Description	1) The NMS sends a modify GTP request to EMS with the GTP distinguishing information. 2) If the request involves CTP additions, the EMS will attempt to add as many of the CTPs as possible. If the request involves CTP deletions, the EMS will attempt to delete as many of the requested CTPs as possible. 3) The EMS replies with a success indication and an updated listing of the CTPs that comprise the modified GTP. 4) The EMS sends an AVC notification to the notification service, indicating that the list of CTPs for the GTP has changed.
Ends when	The EMS sends a reply to the NMS.
Post-conditions	The GTP has been modified in the EMS.
Exceptions	1) Service not implemented by the EMS. 2) The NMS has supplied invalid input data. 3) The user label supplied by the NMS is already in use. 4) The EMS has not yet assigned a set of CTPs to the GTP. 5) The GTP was initially created with gtpEffort equal to EFFORT_SAME. 6) The Input data is valid but the EMS cannot modify the GTP because there are insufficient resources in the network to make the requested modification.
Traceability	{Requirement II. 167}

7.5.20 NMS deletes a Group Termination Point (GTP)

Use Case 7.5.20: NMS deletes a Group Termination Point (GTP)

Use Case Name	NMS deletes a Group Termination Point (GTP)
Summary	The NMS deletes a GTP. This operation is idempotent. If the service is called with the name of a non-existent GTP, it will succeed.
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS sends the delete GTP request to the EMS.
Description	1) The NMS sends a delete GTP request to the EMS. 2) The EMS validates the GTP identifier and checks that this GTP is not being used in a XC or SNC. 3) If no XCs or SNCs are using this GTP, the EMS deletes the GTP. 4) The EMS replies with a success indication. 5) A GTP object deletion event is sent by EMS to the notification service.
Ends When	The EMS sends a reply to the NMS.
Post-Conditions	The GTP has been deleted in the EMS.
Exceptions	1) Service not implemented 2) Communication Failure with NE. 3) The GTP is being used by an XC or an SNC. 4) The NMS has supplied invalid input data (The TMD Name is invalid).
Traceability	{Requirement II. 166}

7.5.21 NMS creates a Termination Point Pool (TP Pool)

Use Case 7.5.21: NMS creates a Termination Point Pool (TP Pool)

Use Case Name	NMS creates a Termination Point Pool (TP Pool)
Summary	The use case describes how the NMS requests that an EMS create a TP Pool. The NMS passes the TP Pool create information to the EMS. The resulting TP Pool is returned as a result to the NMS. Remark: • It is up to the NMS whether it maintains TP Pools across multiple administrative subnetworks and hence EMSs.
Actors	NMS.
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends the request to the EMS to create a TP Pool.

Use Case 7.5.21: NMS creates a Termination Point Pool (TP Pool)

Description	1) The NMS sends a create TPPool request to EMS with the TPPool creation information. The NMS may request that the EMS enforce a unique user label (the user label is selected by the NMS and the EMS makes sure the user label is not already in use). 2) The EMS validates the request. The EMS creates the TP Pool, assigns a unique name to the TPPool and stores it persistently. 3) The EMS replies with a success indication. 4) A TPPool object creation event is sent by EMS to the notification service.
Ends when	In case of success: The NMS receives an indication of success of the action. In case of failure: The NMS receives an indication of failure of the action.
Post-conditions	In case of success: The TPPool is available. In case of failure: None. NMS has received an indication of failure of the action or exception
Exceptions	1) Service not implemented by the EMS 2) The NMS has supplied invalid input data. 3) The user label supplied by the NMS is already in use. 4) The Input data is valid but the EMS cannot create the TPPool because the EMS has already exceeded its maximum number of TP Pools or the requested TPPool is too large.
Traceability	{Requirement II. 264}, {Requirement II. 265}

7.5.22 NMS modifies a Termination Point Pool (TP Pool)

Use Case 7.5.22: NMS modifies a Termination Point Pool (TP Pool)

Use Case Name	NMS modifies a Termination Point Pool (TP Pool)
Summary	The NMS changes the contents of a TPPool.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The TP pool to be modified has to exist. 3) TPs or GTPs to be removed have to be idle. 4) TPs or GTPs to be added have to exist and have to belong to the TPPool's Subnetwork and must not be contained in a GTP or another TP pool.
Begins when	The NMS sends the modify TPPool request to the EMS.
Description	1) The NMS sends the modify TPPool request to the EMS, i.e. requests to add or delete TPs or GTPs to or from the TPPool. 2) The EMS validates the passed TP identifiers. 3) The EMS validates the TPPool identifier. 4) The EMS changes the TPPool attributes if possible. This will fail under the following conditions: • One or more passed TPs are "in use". • One or more passed TPs are contained in a GTP. 5) The EMS replies with a success indication. 6) The EMS will generate an AVC for the TPPool with the changed attributes..
Ends when	The EMS sends a reply to the NMS.
Post-conditions	The TPPool has been modified in the EMS.
Exceptions	1) Service not implemented by the EMS. 2) Non-specific EMS internal failure. 3) The NMS has supplied invalid input data (i.e., the TPPool identifier of one or more TP identifiers are invalid). 4) One or more TPs or GTPs do not exist or do not belong to the TPPool's Subnetwork, and so the addressed TPPool can not be modified. 5) An input parameter references an object that does not exist. 6) One or more TPs or GTPs to be removed are not idle, or one or more TPs or GTPs to be added are contained in a GTP or another TPPool.7)The operation would result in resources being created or activated beyond the capacity supported by the NE/EMS.
Traceability	{Requirement II. 267}

7.5.23 NMS deletes a Termination Point Pool (TP Pool)

Use Case 7.5.23: NMS deletes a Termination Point Pool (TP Pool)

Use Case Name	NMS deletes a Termination Point Pool (TP Pool)
Summary	The NMS deletes a TPPool. Remark: • The NMS is responsible to maintain the consistency of TPPool across multiple administrative subnetworks and hence EMSs. Routine integrity checking may be required. NMS must also check that the TPPool is not being used in any EMS under its network domain, before the TPPool can be deleted from the persistent storage. • This operation is idempotent. If the service is called with the name of a non-existent TPPool, it will succeed.
Actor(s)	NMS
Pre-Conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The TP pool to be deleted must not be in use, i.e. all of its members must be idle.
Begins When	The NMS sends the delete TPPool request to the EMS.
Description	1) The NMS sends a delete TPPool request to the EMS. 2) The EMS validates the TPPool identifier and checks that this TPPool is not being used on the EMS, i.e. has all its member TPs or GTPs idle. 3) If no TPs or GTPs are using this TPPool, the EMS deletes the TPPool.4)The EMS replies with a success indication. 4) A TPPool object deletion event is sent by EMS to the notification service.
Ends When	The EMS sends a reply to the NMS.
Post-Conditions	The TPPool has been deleted in the EMS. (Note. The TPPool can be deleted in the NMS if and only if no EMS uses this TD.)
Exceptions	1) Service not implemented 2) Non-specific EMS internal failure. 3) The TPPool is being used (i.e., not all contained TPs or GTPs are idle). 4) The NMS has supplied invalid input data (i.e., the TPPool name is invalid).
Traceability	{Requirement II. 266}

7.5.24 NMS assigns an Alarm Severity Assignment Profile (ASAP) to a CTP

Use Case 7.5.24: NMS assigns an Alarm Severity Assignment Profile (ASAP) to a CTP

Use Case Name	NMS assigns an Alarm Severity Assignment Profile (ASAP) to a CTP
Summary	The NMS assigns an ASAP, either previously created by the NMS, or created by EMS, to a CTP, at an NMS specified layer rate.

Use Case 7.5.24: NMS assigns an Alarm Severity Assignment Profile (ASAP) to a CTP

Actor(s)	NMS
Pre-Conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The identified ASAP already exists in the EMS. 3) In case the resource is a TP: the provided layer rate is an encapsulated layer rate of the TP. 4) The identified object has to exist. 5) The identified object has to support the ASAP pointer feature.
Begins When	The NMS sends the assign ASAP request to the EMS with the specified CTP.
Description	1) The NMS sends the request to assign the ASAP to the addressed CTP. 2) The EMS validates the assignment request. 3) if EMS does not support assignment of ASAPs via this interface, an exception is thrown. 4) If the ASAP name does not refer to an ASAP object, or the specified layer rate is invalid for the addressed resource, i.e. it is not an encapsulated layer rate, then an exception is thrown. 5) If the ASAP name or the resource name reference to non existent object, then an exception is thrown. 6) If there is a currently assigned ASAP, and this assignment is fixed on EMS side, then an exception is thrown. 7) If the resource name refers to an object not supporting ASAP pointer feature then an exception is thrown. 8) The NMS connects to the notification service and thus is able to receive notifications matching the filter conditions specified (if this has not been done earlier). Note: The main filtering criteria is on the notification type (i.e., alarm and/or threshold crossing alert). In addition, the NMS can request other filtering criteria. Any of the parameters of the filterable body of the alarm or threshold crossing alert notification can be used. The commonest filtering criteria for alarms are the Probable Cause and Perceived Severity parameters. See Use Case 7.9.1: NMS reconciles active alarms from an EMS and Use Case 7.9.2: NMS reconciles active alarms for a specified Managed Element
Ends When	The EMS sends a reply to the NMS.

Use Case 7.5.24: NMS assigns an Alarm Severity Assignment Profile (ASAP) to a CTP

Post-Conditions	In case of success: This operation causes an alarm re-evaluation of the already detected defects according to the following rules.If alarms are reportable (*): • if the severity changes from any of critical, major, minor, warning, to not alarmed, then an alarm notification with cleared is sent and the alarm is no longer available for any alarm retrieval operation. • if the severity changes from not alarmed to any of critical, major, minor, warning, then an alarm notification with the new perceivedSeverity is sent (with the current EMS/NE time) and the alarm is available for any alarm retrieval operation. • if the severity changes from any of critical, major, minor, warning, to any of critical, major, minor, warning, then the alarm re-evaluation process is not performed. (*) an alarm is reportable by ME/EMS when • AlarmReporting = “on” (for PTP, CTP, FTP), • alarmReportingIndicator = true (for SNC, TopologicalLink, Equipment, EquipmentHolder, GTP), • always reportable for all other objects which do not have any alarm reporting attribute. Moreover, once an alarm becomes reportable by ME/EMS then the following procedure is performed: • If the managed object has a valid aSAPpointer, then it is searched the entry with <same probableCause> AND <same probableCauseQualifier> AND <same nativeProbableCause> i) probableCauseQualifier value may be empty, which means <TRUE> condition ii) nativeProbableCause value may be empty, which means <TRUE> condition E.g. if the reportable alarm has LOS probableCause and an ASAP entry is found with LOS probableCause and both probableCauseQualifier and nativeProbableCause are empty strings, then the search is successful, and the associated severities are assigned: • the alarm is service affecting: it is assigned the severity specified in the serviceAffecting attribute; if not explicitly assigned, i.e. ANY, see below (*) • the alarm is service non affecting: it is assigned the severity specified in the serviceNonAffecting attribute; if not explicitly assigned, i.e. ANY, see below (*)

Use Case 7.5.24: NMS assigns an Alarm Severity Assignment Profile (ASAP) to a CTP

	• the alarm is service independent or EMS does not know whether the alarm actually affects the service or not: it is assigned the severity specified in the serviceIndependentOrUnknown attribute; if not explicitly assigned, i.e. ANY, see below (*) • If the corresponding probableCause is not found in the ASAP, or the managed object has not a (valid) aSAPpointer then: (*) it is assigned the default / native severity at EMS/NE side, if any, otherwise the INDETERMINATE severity is assigned. • Once a severity (including the INDETERMINATE) has been assigned, the alarm notification is emitted, but in the case of “NOTALARMED” - cleared severity, which causes the non emission of the alarm notification.Any operation of alarm retrieval will not include such “NOTALARMED” alarms. In case of failure: Either the currently assigned ASAP is maintained, e.g. because the assignment is fixed on EMS side, or no ASAP is assigned.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed. 3) Invalid input: The aSAPName does not refer to an ASAP object, or layerRate is invalid for the addressed resource, i.e. it is not an encapsulated layerRate. 4) Entity not found: The aSAPName or resourceName reference an object that does not exist. 5) Unable to comply: The currently assigned ASAP object cannot be de-assigned, or resourceName refers to object not supporting ASAP pointer feature.
Traceability	{Requirement II. 201}

7.5.25 NMS locks (in a forced response deferred/graceful fashion) or unlocks a number of IMA links to modify the transport capacity of the corresponding fixed IMA group

Use Case 7.5.25: NMS locks (in a forced response deferred/graceful fashion) or unlocks a number of IMA links to modify the transport capacity of the corresponding fixed IMA group

Use Case Name	NMS locks (in a forced response. deferred/graceful fashion) or unlocks a number of IMA links to modify the transport capacity of the corresponding fixed IMA group
Summary	Depending on the respective equipment support the bandwidth capacity of a ready-to-use or in-use configured IMA group can be adjusted by locking previously unlocked IMA links or unlocking previously locked IMA links or both.The NMS must first discover the server CTPs of the fixed IMA group FTP (because these are the associated potential or actual IMA links) and determine which of them are currently locked (or shutting down) and which are unlocked. Depending on the capacity requirement to be met it then locks or unlocks the appropriate IMA links. This Use Case is an application of the following use cases depending on how the EMS supports the locking and unlocking of a TP: • Use Case 7.4.3: NMS discovers the EMS network inventory • Use Case 7.5.7: NMS Provisions the TP Transmission Parameters • Use Case 7.5.5: NMS provisions the Additional Information
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS sends a request to the EMS for the server CTPs of the IMA group FTP
Description	1) The NMS queries the EMS regarding the CTPs supported by the IMA group FTP and figures out the server CTPs, i.e. the fragments. 2) The NMS verifies if it can meet the requirements of the bandwidth demand using the available fragments (number and transmission rate of fragments). 3) If the bandwidth demand can be fulfilled then the NMS selects the series of CTPs that needs to be locked (in a forced or graceful manner) or unlocked for adjusting the bandwidth. 4) The NMS sends requests to provision each CTP needed to support the bandwidth adjustment order (refer to Use Case 7.5.7: NMS Provisions the TP Transmission Parameters in case of using "ServiceState" response Use Case 7.5.5: NMS provisions the Additional Information in case of using "AdminState"). 5) The bandwidth demand is met successfully if the NMS receives a success indication from the EMS for each CTP provisioning and the new bandwidth is stored in the IMA group as transmission parameter "Bandwidth" of LR_ATM_NI.6. If one or several CTPs could not be provisioned properly then the NMS may choose to meet the bandwidth demand only partially or to re-provision the CTPs that were successfully provisioned (i.e., rollback the successful work).

Use Case 7.5.25: NMS locks (in a forced response deferred/graceful fashion) or unlocks a number of IMA links to modify the transport capacity of the corresponding fixed IMA group

Ends When	In case of success: The NMS receives the indication that all CTPs are provisioned properly. In case of failure: The NMS receives an indication of provisioning failure for at least one CTP.
Post-Conditions	In case of success: 1) All CTPs required to fulfill the bandwidth demand have been provisioned. 2) The EMS has forwarded TP modification notifications to all subscribing NMSs In case of failure: Some or all CTPs have not been newly provisioned.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed.
Traceability	

7.5.26 NMS requests dynamic provisioning of an IMA group by the EMS subject to a prescribed bandwidth that is communicated as number and connectable layer rate of the IMA links

Use Case 7.5.26: NMS requests dynamic provisioning of an IMA group by the EMS subject to a prescribed bandwidth that is communicated as number and connectable layer rate of the IMA links

Use Case Name	NMS requests dynamic provisioning of an IMA group by the EMS subject to a prescribed bandwidth that is communicated as number and connectable layer rate of the IMA links.
Summary	Using appropriate transmission parameters for the IMA group the NMS requests by the EMS to load the IMA group with a load of its choice but with prescribed bandwidth. This Use Case is an application of the following use case:· • Use Case 7.5.7: NMS Provisions the TP Transmission Parameters
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS splits the required bandwidth into a layer rate supported by the ME and a multiplicity factor.
Description	1) The NMS recalculates the required bandwidth according to the formula “FragmentServerLayer” * “AllocatedNumber” thereby taking care that “FragmentServerLayer” represents a transmission layer rate supported by the ME (refer to the transmission parameters for layer rate LR_Fragment). 2) The NMS executes the Use Case 7.5.7: NMS Provisions the TP Transmission Parameters for the IMA group TP with these two transmission parameters. 3) The bandwidth demand is met successfully if the NMS receives a success indication from the EMS for each CTP provisioning and the new bandwidth is stored in the IMA group as transmission parameter “Bandwidth” of LR_ATM_NI.
Ends When	In case of success: The NMS receives the indication that the IMA group has been re configured, namely AVCs for the two passed transmission parameters. In case of failure: The NMS receives an indication of configuration failure.
Post-Conditions	In case of success: 1) All SNCs required to reload the IMA group have been created implicitly. 2) The EMS has forwarded AVC notifications to all subscribing NMSs In case of failure: The NMS receives an indication of provisioning failure for the IMA group TP.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed.
Traceability	

7.5.27 NMS provisions the IMA virtual link between two peer IMA groups as a topological link

Use Case 7.5.27: NMS provisions the IMA virtual link between two peer IMA groups as a topological link

Use Case Name	NMS provisions the IMA virtual link between two peer IMA groups as a topological link
Summary	The NMS determines the number and layer rate of the IMA links required to support the bandwidth demand on either side of the IMA virtual link. It then creates a bidirectional TL between the two peer IMA groups at the ATM NI layer that provides the bandwidth. This Use Case is an application of the following use case: • Use Case 7.5.10: NMS creates a Topological Link (TL)
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS splits the required bandwidth into a layer rate supported by the ME and a multiplicity factor.
Description	1) The NMS recalculates the required bandwidth according to the formula “FragmentServerLayer” * “AllocatedNumber” thereby taking care that “FragmentServerLayer” represents a transmission layer rate supported by the ME (refer to the transmission parameters for layer rate LR_Fragment). 2) The NMS executes the Use Case 7.5.10: NMS creates a Topological Link (TL) with the following TL create data: bidirectional, LR_ATM_NI as layer rate, the two IMA groups as end points, and the two transmission parameters of step 1 as additional creation info. 3) The bandwidth demand is met successfully if the NMS receives a success indication from the EMS for the TL creation, the new bandwidth is stored in both IMA groups as transmission parameter “Bandwidth” of LR_ATM_NI, and the two transmission parameters are stored as additional info parameters of the TL.
Ends When	In case of success: The NMS receives the indication that the TL has been created. In case of failure: The NMS receives an indication of creation failure (of some SNC or the TL).
Post-Conditions	In case of success: 1) All SNCs required to load both IMA groups, and the TL, have been created. 2) The EMS has forwarded SNC and TL creation notifications to all subscribing NMSs. In case of failure: Some or all SNCs or the TL itself could not be created.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed.
Traceability	

7.5.28 NMS unprovisions an IMA virtual link between IMA groups

Use Case 7.5.28: NMS unprovisions an IMA virtual link between IMA groups

Use Case Name	NMS unprovisions an IMA virtual link between IMA groups
Summary	The NMS deletes the TL between the IMA groups it has previously created. This Use Case is an application of the following use case: • Use Case 7.5.11: NMS deletes a Topological Link (TL)
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS and either Use Case 7.5.27: NMS provisions the IMA virtual link between two peer IMA groups as a topological link or Use Case 7.7.15: NMS creates a flexible IMA group for both considered IMA groups (i.e., the NMS can only delete TLs that it owns).
Begins When	The NMS determines the name of the TL between the two IMA groups in question
Description	1) The NMS determines the name of the TL between the two considered IMA groups, optionally locks the two IMA groups (depending on EMS requirements), and executes the Use Case 7.5.11: NMS deletes a Topological Link (TL). 2) The IMA virtual link is deleted successfully if the NMS receives a success indication from the EMS for the TL deletion. 3) If the TL or one or several SNCs could not be deactivated and deleted (e.g., because they carry traffic and cannot be put out-of-service) then either the NMS needs to try again at a later time (IMA groups are only partially unloaded) or the NMS may set the Administrative state of the involved CTPs to “shutting down”, when supported by the EMS, to indicate its intention of SNC and TL deletion.
Ends When	In case of success: The NMS receives the indication that the TL has been deleted. In case of failure: The NMS receives an indication of deletion failure (of some SNC or the TL).
Post-Conditions	In case of success: 1) All SNCs constituting both IMA groups and the TL have been deleted. 2) The EMS has forwarded SNC and TL deletion notifications to all subscribing NMSs. In case of failure: Some or all SNCs or the TL itself could not be deleted.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed.
Traceability	

7.5.29 NMS provisions or re-provisions a single DSL line

Use Case 7.5.29: NMS provisions or re-provisions a single DSL line

Use Case Name	NMS provisions or re-provisions a single DSL line
Summary	The NMS identifies the DSL line to be provisioned as either a TL between the TU-C/O PTP and the TU-R PTP or a single PTP representing both TU PTPs (e.g., since the TU-R is unmanaged).The NMS (re-)provisions the DSL line by provisioning its one or two end points. It locks the end point(s) administratively when required by the responsible EMS, sets the desired DSL parameters (which are transmission parameters of the respective DSL PTPs), and unlocks the end point(s) of the DSL line if they were locked. If administrative locking and unlocking are required by an EMS this means that DSL re-provisioning cannot be done in-service. This Use Case is an application of the following use cases depending on how the EMS(es) support(s) the locking and unlocking of a TP: • Use Case 7.4.3: NMS discovers the EMS network inventory • Use Case 7.5.5: NMS provisions the Additional Information • Use Case 7.5.7: NMS Provisions the TP Transmission Parameters
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS has identified the DSL line to be (re-)provisioned as either a TL or a single PTP.
Description	1) The NMS identifies the DSL line to be (re-)provisioned as either a TL or a single PTP (refer to Use Case 7.4.3: NMS discovers the EMS network inventory). 2) The NMS may assume that the downstream configuration parameters are stored at the TU-C/O PTP. The NMS knows whether the upstream configuration parameters are stored at the TU-C/O PTP as well (usual case) or at the TU-R PTP. 3) The NMS locks the PTP(s) where the DSL parameters are stored when required. 4) The NMS writes the desired downstream DSL parameters to the TU-C/O PTP and the desired upstream DSL parameters to the PTP where they are stored (refer to Use Case 7.5.7: NMS Provisions the TP Transmission Parameters). 5) The NMS unlocks the (re-)provisioned PTP(s) if it was (they were) locked.
Ends When	In case of success: The NMS receives the indication that all PTPs are provisioned properly. In case of failure: The NMS receives an indication of provisioning failure for at least one PTP.
Post-Conditions	In case of success: 1) All PTPs constituting the DSL line have been provisioned. 2) The EMS has forwarded PTP modification notifications to all subscribing NMSes. In case of failure: Some or all PTPs have not been newly provisioned.

Use Case 7.5.29: NMS provisions or re-provisions a single DSL line

Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed.
Traceability	

7.5.30 NMS provisions multiple DSL lines by using TMDs

Use Case 7.5.30: NMS provisions multiple DSL lines by using TMDs

Use Case Name	NMS provisions multiple DSL lines by using TMDs
Summary	The NMS identifies each DSL line to be provisioned as either a TL between the TU-C/O PTP and TU-R PTP or a single PTP representing both TU PTPs (e.g., since the TU-R is unmanaged). The NMS provisions each DSL line by provisioning its one or two end points by means of the same DSL TMDs. It locks all end points administratively when required by the EMS, assigns the desired TMDs to all end points, and unlocks the end points if they were locked. If administrative locking and unlocking are required by an EMS this means that DSL mass provisioning cannot be done in-service. This Use Case is an application of the following use cases depending on how the EMS(es) support(s) the locking and unlocking of a TP: • Use Case 7.4.3: NMS discovers the EMS network inventory • Use Case 7.5.5: NMS provisions the Additional Information • Use Case 7.5.7: NMS Provisions the TP Transmission Parameters • Use Case 7.5.14: NMS sets the Transmission Descriptor (TMD) Profile Pointer
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS has identified the DSL line to be mass provisioned as either a TL or a single PTP.
Description	1) The NMS identifies the DSL line to be mass provisioned as either TLs or a single PTPs (refer to Use Case 7.4.3: NMS discovers the EMS network inventory). 2) The NMS identifies the TMDs to be used. There may be a TMD for downstream/ egress direction and a TMD for upstream/ingress direction (as seen from the TU-C/O PTP resp. aEndTP of the DSL line), or a TMD for both directions. 3) The NMS locks all PTPs where the DSL downstream and upstream configuration parameters are stored when required. 4) The NMS assigns the desired downstream TMD to all TU-C/O PTPs that store downstream parameters and the desired upstream TMD (which could be the same as the downstream TMD) to all PTPs that store upstream parameters (refer to Use Case 7.5.14: NMS sets the Transmission Descriptor (TMD) Profile Pointer). 5) The NMS unlocks the mass provisioned PTPs if they were locked.
Ends When	In case of success: The NMS receives the indication that all PTPs are provisioned properly. In case of failure: The NMS receives an indication of provisioning failure for at least one PTP.

Use Case 7.5.30: NMS provisions multiple DSL lines by using TMDs

Post-Conditions	In case of success: 1) All PTPs constituting an end point of the DSL lines have been provisioned. 2) The EMS has forwarded PTP modification notifications to all subscribing NMSes. In case of failure: Some or all PTPs have not been newly provisioned.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed.
Traceability	

7.6 Call Management Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.6.1:	(UC_20) NMS establishes a Call	{Requirement II. 392}, {Requirement II. 393}
Use Case 7.6.2:	(UC_21) NMS releases a Call	{Requirement II. 394}
Use Case 7.6.3:	(UC_22) NMS adds one or more Connections to a Call	{Requirement II. 397}, {Requirement II. 398}
Use Case 7.6.4:	(UC_23) NMS removes one or more Connections from a Call	{Requirement II. 399}
Use Case 7.6.5:	NMS modifies a Call	{Requirement II. 395}, {Requirement II. 396}
Use Case 7.6.6:	(UC_7a) NMS retrieves all the Connections and SNCs for a specific list of Calls	{Requirement II. 404}, {Requirement II. 405}, {Requirement II. 406}, {Requirement II. 407}, {Requirement II. 408}, {Requirement II. 409}, {Requirement II. 410}, {Requirement II. 411}
Use Case 7.6.7:	(UC_33) NMS requests EMS to set/modify diversity and co-routing parameters of a Call	{Requirement II. 400}, {Requirement II. 401}, {Requirement II. 402}, {Requirement II. 403}

7.6.1 NMS establishes a Call

Use Case 7.6.1: (UC_20) NMS establishes a Call

Name	NMS establishes a Call

Use Case 7.6.1: (UC_20) NMS establishes a Call

<table>
<tr><td>Summary</td><td>
This operation allows an NMS to establish a point-to-point Call with:

zero or more top-level Connections across one or more MultiLayer Routing Areas (MLRAs) or
directly zero or more Subnetwork Connections (SNCs) across one MultiLayer Subnetwork (MLSN).

In a non-restoration network the NMS can decide if the operation is atomic or best-effort.
The EMS establishes a Call that matches the parameters requested.
When the Call is established the returned structure provides the complete Call object.
Note: The Call Name which is the name by which the Control Plane identifies the call is included.
When supporting Connections (i.e., top-level Connections or Subnetwork Connections) are created the returned structure provides the complete Connection object for each created supporting Connection.
Note: The Connection Name which is the name by which the Control Plane or the EMS based network router identifies the Connection is included.
The returned structure also provides a list of TP data that was successfully set.
General notes:

It is valid for one of the end TPs of the Call to be an “off-network” TP (i.e., outside the EMS domain and therefore addressed by a remote address) or an SNP/SNPP/TNA not directly managed by local EMS.
The bandwidth of the Call can be provided by a single Connection, by a Virtual Concatenation Group (VCG), or by Link Aggregation (LAG).
When the EMS is requested to create the Connections, criteria for diverse routing (Route Groups) can be given. Refer to the supporting document SD1-45_ASONControlPlaneManagement-Primer.pdf.
If best-effort is required and if it is not possible to create all requested Subnetwork Connections, the EMS will return an indication of which Subnetwork Connections could not be created.
For an EMS based router implementation the top-level Connection, when successfully routed, will be supported by SNCs in MLSNs and the SNCs will be supported by cross connections in Managed Elements.

Control Plane specific notes:

This case applies where there is a full Control Plane solution and where the solution has been implemented with an EMS based router:

It is expected that SNPP and TNA names do not exist and that the ends of the Calls and Connections will be specified in terms of TPs. However, where the Call/Connection ends outside the scope of the EMS an abstract name may be used that could be equated to a TNA name.

</td></tr>
</table>

Use Case 7.6.1: (UC_20) NMS establishes a Call

	• It is expected that the MLSNPP Link will not be a relevant entity but instead top-level Topological Links will be used. • One or more of the three endpoint name types (SNPP, TP or TNA) can be used in the response to the Call creation request and to any depth (e.g. SNPP only or SNPP and SNP). • In the Connection returned, the name of the local end point will be the fully resolved name of the channel for all name types that are known (i.e. TP, SNPP and TNA names). For the remote end point, the returned name will be whatever the Control Plane provided and will be at least the TNA Name. • For a Call to be successfully established all requested Connections must be successfully created (although they may be in searching state - refer to state model), i.e., only exact match (for number of Connections), if any Connection fails the entire Call fails (i.e., operation is always atomic). • The Connection end point should be in the scope of the corresponding Call endpoint. For example if the connection TNA Name does not correspond to the Call Group TNA Name the Control Plane may reject the Call/Connection. • For a particular end if the Call is specified with a null value the Connection end must be specified. If the Call has a specified end value the corresponding Connection end value may be null. • The top-level Connection, when successfully routed, will be supported by Connections represented in subordinate MLRAs and these Connections will be supported by Cross Connections in MLRAs that represent Routing Nodes.
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS sends the establish Call request to the EMS.

Use Case 7.6.1: (UC_20) NMS establishes a Call

Description	1) The NMS sends the request to establish a Call to the EMS. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If a specified object is not of the right type (PTP, CTP, FTP, CPTP), then an Invalid Input exception is raised. c) If the operation is used in an MLRA and it is not a top-level MLRA, then an Unable To Comply exception is raised. d) If a Call object with the Name specified already exists, then an Object In Use exception is raised. e) If a Connection object with the Name specified already exists, then an Object In Use exception is raised. f) If uniqueness of the user label is required, the EMS checks the user label for uniqueness; i.e., if a Call with the same user label exists already, a User Label In Use exception is raised. g) If uniqueness of the user label is required for the Connections, the EMS checks the user label for uniqueness; i.e., if a Connection with the same user label exists already, a User Label In Use exception is raised. h) If a specified object (PTP, CTP, FTP, CPTP) is not known to the EMS, then an Entity Not Found exception is raised. i) If a specified SNP or SNPP ID is not known to the EMS, then an Entity Not Found exception is raised. j) If a specified TNA Name or Group TNA name is not known to the EMS, then an Entity Not Found exception is raised. k) If the layer parameters are inconsistent, for example where two parameters at one TP conflict or where a parameter at one end of the Connection has a value that is illegal with respect to a parameter value at the other end of the Connection (e.g., LCAS without VCAT or LAG is requested or the LCAS specification at one end contradicts that at the other), an Unable To Comply exception is raised. l) If the EMS determines that the TNA Name structure is not valid for any TNA Name, then an Invalid Input exception is raised. m) In a Control Plane environment, if the EMS does not control the node associated with the Call originating point, then an Unable To Comply exception is raised. n) If one of the TPs identified in a Connection is in use, then an Object In Use exception is raised. o) If the diversity requirements contradict each other (e.g., • when the NMS provides a “Number of Route Groups” and “Route Group Labels” are provided in the Connection related data or • a “Number of Route Groups” is not provided and the NMS has not provided a “Route Group Label” for all Connections to be created), an Unable To Comply exception is raised.

Use Case 7.6.1: (UC_20) NMS establishes a Call

p) If the EMS/Control Plane cannot support the type of routing constraints (capability not available in the EMS or network), then an Unsupported Routing Constraints exception is raised.

q) If the EMS/Control Plane does not support specified diversity criteria, then an Unsupported Routing Constraints exception is raised.

r) If the EMS/Control Plane does not support specified Co-routing criteria, then an Unsupported Routing Constraints exception is raised.

s) If the EMS/Control Plane cannot satisfy the requested routing criteria, within the boundaries of the effort level requested, then an Unsupported Routing Constraints exception is raised.

t) If one or more resources requested to be used for the Connections associated with the request, are already in use, then an Object In Use exception is raised

3) If the request is valid the EMS sets the requested TP data.

4) If the request is valid and if **NO** Connections are requested:

a) the Call object is created in the “Established” state.

b) the EMS sends an indication of completion of the request to the NMS indicating the TP data that was set if any.

c) the EMS sends a Call creation notifications to reflect the changes.

5) In a non Control Plane environment, if the request is valid and if Subnetwork Connections are requested:

a) For each requested Subnetwork Connection the EMS creates an SNC.

For all successfully created SNCs the EMS creates a corresponding Connection with Connection state set to “Not applicable”.

For all SNCs that could only be created in “Partial” state no Connection object is created by the EMS.

The EMS sends an indication of the completion of the request to the NMS indicating the created Connection objects also the TP data that was set if any.

b) If best-effort is required and if it is not possible to create all requested Connections, the EMS will return the SNCs which could not be created as requested.

c) The EMS sends Call, Connection and SNC creation notifications to reflect the changes (indicating the appropriate states of the Call/Connections).

6) If the request is valid and if Connections are requested:

a) If the routes are not completely provided by the NMS, the EMS/Control Plane determines the routes of the Connections, according to the routing constraints associated with the Call request.

Use Case 7.6.1: (UC_20) NMS establishes a Call

	b) If routes for all the Connections were found within timer T0, the Call state is set to “Established” and the Connection state is set to “Complete”, the EMS sends an indication of the completion of request to the NMS indicating the diversity information and also the TP data that was set if any. c) If routes could not be found for all of the Connections within timer T0, the Call state is set to “In Progress” and zero or more Connections have their state set to “Complete” and one or more Connections have their state set to “Searching” and the EMS sends an indication of the completion of the request to the NMS indicating also the TP data that was set if any. The EMS sends Call and Connection creation notifications to reflect the changes (indicating the appropriate states of the Call/Connections). The EMS/Control Plane will keep trying to find routes within the constraints. For each Connection that is routed after timer T0 its state will change to “Complete” and the EMS sends a state change notification to reflect the change in the state of the Connection. d) When the final Connection of the Call changes state to “Complete” (i.e. all Connections are “Complete”) the Call state changes to “Established” and the EMS sends notifications to reflect the change in the state of the Call, the diversity information and also the TP data that was set if any. e) If prior to the final Connection of the Call being routed timer T1 expires all Connections will be deleted, the Call will be deleted and the EMS sends OD notifications to reflect the removal of the Call and all of its Connections. The TP data that was set will be restored to the previous values.

Use Case 7.6.1: (UC_20) NMS establishes a Call

	Control Plane environment specific notes: • Each Connection created in the Control Plane is supported by Connection Segments (represented through the interface as Connections) in subordinate MLRAs throughout the network (between the source and destination of the Call). A Connection Segment creation/deletion or change is detected by the EMS managing the MLRA. The EMS sends the appropriate OC, OD, SC or AVC notifications to reflect the change (refer to Use Case 8.2.7: (UC_8) EMS notifies of a new Call and its Connection(s)). • Notifications are NOT reported for Connection Segments in Routing Nodes. • The EMS that controls the Call originating point shall raise the OC or OD Notifications and SC for the Call and for all top-level Connections. • In the case where timer T1 expires the Call associated Connection Segments will be removed from the network (coordinated by Control Plane signalling). Where the EMS does not control the entire Call other EMSs in the network will detect deletion of Connection Segments. When a Connection Segment deletion is detected by the EMS managing the MLRA, the EMS sends the appropriate object deletion and AVC notification to reflect the change (refer to Use Case 8.2.7: (UC_8) EMS notifies of a new Call and its Connection(s)). • Notifications are NOT reported for Connections in Routing Nodes. • In the case where timer T1 expires the EMS that controls the Call originating point shall raise the OD notifications for the Call and for all top-level Connections. • In the case where timer T1 expires the order of Call OD notifications and Connection OD notifications is not specified. Call deletion notifications may follow Connection deletion notifications or vice-versa.
Ends When	In case of success: 1) The NMS receives an indication of success of the action. 2) The EMS returns the distinguishing information of the created Call. 3) The EMS returns the distinguishing information of each created (top-level) Connection. 4) The EMS returns the distinguishing information of each created SNC. 5) In case of best-effort and not all (top-level) Connections and SNCs could be created as requested, the EMS returns the failed (top-level) Connections and SNCs. In case of failure: The NMS receives an exception as an indication of the failure of the request.

Use Case 7.6.1: (UC_20) NMS establishes a Call

Post-Conditions	In case of success: 1) The Call object has been created. 2) The corresponding top-level Connection objects have been created. 3) The corresponding SNC objects have been created. 4) In a Control Plane environment: 5) Upon completion of each connection, connections (or for the case of an EMS based router SNC) at lower partitioning levels (for the EMS based router MLSN) have been created. 6) For each change of state in the top-level, subordinate connection, SNCs or call objects, appropriate state change notifications have been issued to the NMS. 7) For each connection or SNC, in case of unsuccessful connection set-up, the connection object or SNC is deleted and the NMS is notified via object deletion notification. A connection set-up may fail because one or more of the subordinate connections (or SNCs) at lower partitioning levels (or MLSN) have failed. 8) In case of unsuccessful call set-up, the call object is deleted and the NMS is notified via object deletion notification. In case of failure: Nothing has changed in the System, i.e., no Call and Connections have been created.
Exceptions	Refer to Table 7.1
Traceability	{Requirement II. 392}, {Requirement II. 393}

7.6.2 NMS releases a Call

Use Case 7.6.2: (UC_21) NMS releases a Call

Name	NMS releases a Call
Summary	This operation allows the NMS to release a point-to-point Call with zero or more supporting Connections (i.e., top-level Connections or Subnetwork Connections). This use case applies for SC and SPC type Calls (i.e. Calls established through the UNI as well as those established by the NMS). The NMS is allowed to release a Call that was established through UNI signalling. The EMS is required to delete the specified Call, delete all supporting Connections (except “fixed” SNCs) and to modify a set of TPs. The returned structure provides a list of TP data that was successfully modified. Notes: • In the Control Plane case the successful EMS response is an acceptance of the Call release, not an indication that all resources in the network have been released. Actual release of resources will be given by deletion notifications. • It is assumed that the Control Plane will remove the Connections from the network, along all partitioning levels, delete the Connection objects, remove the Call, delete the Call object and send appropriate notifications to the NMS via the EMS. As the Control Plane is not overseen by a single entity it is possible that there will be some delay before the Connection objects related to intermediate MLRAs are removed by their respective EMSs. There is no Partial state for top-level Connections. The Call object and all Connection objects related to the Call known by the EMS that controls the originating point of the Call will be deleted as a result of the request. • In the non Control Plane case the EMS has to make the effort to remove any partial SNCs resulting from this operation. If this is not possible, the EMS will return the list of SNCs which could not be deleted.
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS sends the release a Call request to the EMS.

Use Case 7.6.2: (UC_21) NMS releases a Call

Description	1) The NMS sends the request to release a Call to the EMS. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If a specified object is not of the right type (Call, TP), then an Invalid Input exception is raised. c) If a specified object (Call, TP) is not known to the EMS, then an Entity Not Found exception is raised. d) If the EMS is able to determine that the Call is carrying traffic I can reject the operation and raise an Unable To Comply exception. 3) If the request is valid the EMS: a) Sets the requested TP data (best effort). b) Deletes all Connections supporting the Call (except the “fixed” SNCs). The EMS has to make the effort to remove any Partial SNCs resulting from this operation. If this is not possible, the EMS will return the list of SNCs which could not be deleted. c) Deletes the Topological Links which were supported by the Call. d) Deletes the Call. e) Sends an indication of completion of request to the NMS. f) Indicates the TP data that was set. g) Sends deletion notifications to reflect the deletion of the Call, the Topological Link and all supporting Connections. Control Plane related Notes: • Each top-level Connection in the Control Plane is supported by Connection Segments (represented through the interface as Connections) in MLRAs throughout the network (between the source and destination of the Call). As a result of the request to release a Call all associated Connection Segments will be removed from the network (coordinated by Control Plane signaling). Where the EMS does not control the entire Call other EMSs in the network will detect deletion of Connection Segments. When a Connection Segment deletion is detected by the EMS managing the MLRA, the EMS sends the appropriate object deletion and AVC notifications to reflect the change (refer to Use Case 8.2.7: (UC_8) EMS notifies of a new Call and its Connection(s)). • Connections notifications are NOT reported for Routing Nodes. • The EMS that controls the Call originating point shall raise the object deletion notifications for the Call and for all top-level Connections. • The order of Call object deletion notifications and Connection object deletion notifications is not specified. Call deletion notifications may follow Connection deletion notifications or vice-versa.

Use Case 7.6.2: (UC_21) NMS releases a Call

	• The operation is considered atomic. Connections will not exist in the network without Calls. If the Control Plane signalling fails after the request is received, the Control Plane and EMS will deal with it, and the EMS will notify the NMS of successful completion. The Call and its Connections will be deleted when the signalling network becomes available. It is the responsibility of the EMS and the Control Plane to clean up the resources in the network.
Ends When	In case of success: 1) The NMS receives an indication of the success of the request. 2) In case not all Subnetwork Connections could be deleted, the EMS returns a list of the Subnetwork Connection names which could not be deleted. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The Call, the Topological Link and all supporting Connections of the Call have been deleted by the EMS. In case of failure: No change occurred to the system, i.e., the Call, the Topological Link and all supporting Connections have not been deleted.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 394}

7.6.3 NMS adds one or more Connections to a Call

Use Case 7.6.3: (UC_22) NMS adds one or more Connections to a Call

Use Case Name	NMS adds one or more Connections to a Call
Summary	This operation provides a way to add one or more point-to-point supporting Connections (i.e., top-level Connections or Subnetwork Connections) to an existing Call and set any necessary TP data. The parameters allow for supporting Connections with no constraints or for providing the complete creation data and everything in between. The EMS creates the supporting Connections that match the parameters requested. If the NMS specifies routing constraints for the associated supporting Connections in the request they should match the constraints based to the degree mandated by the appropriate effort statement. If the EMS cannot satisfy the NMS routing criteria, within the boundaries of the effort level requested, an exception is raised. Note: When a supporting Connection is created the returned structure provides the Connection id which is the name by which the Control Plane identifies the Connection. The returned structure also provides a list of TP data that was successfully set. In a Control Plane environment it is assumed that the Control Plane will either set up the entire top-level Connection or not; there is no Partial state for top-level Connections. This use case is an atomic transaction (i.e., not best effort). Control Plane related notes: • In the Connection returned, the name of the local end point will be the fully resolved name of the channel for all name types that are known (i.e. TP, SNPP and TNA names). For the remote end point, the returned name will be whatever the Control Plane provided and will be at least the TNA Name. • For a Call, all requested top-level Connections must be successfully created (although they may be in the Searching state - refer to the Call state transition diagram in TMF608) . That is only exact match (for number of top-level Connections), if any top-level Connection fails all other top-level Connections that have been added as part of this request will be deleted and the Call will return to the state prior to the request to add the top-level Connections. • The Connection end point should be in the scope of the corresponding Call endpoint. For example if the Connection TNA name does not correspond to the Call Group TNA name the Control Plane may reject the Call/Connection. The Call may be specified with Null endpoints.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the NMS to add connections to an existing Call.

Use Case 7.6.3: (UC_22) NMS adds one or more Connections to a Call

Description	1) The NMS sends the request to add one of more Connections (top-level Connections or Subnetwork Connections) to an existing Call. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If a specified object is not of the right type (e.g., Call, PTP, CTP, FTP, etc.), then an Invalid Input exception is raised. c) If a specified object (e.g., Call, PTP, CTP, FTP, etc.) is not known to the EMS, then an Entity Not Found exception is raised. d) If the specified Call is not in an acceptable state as determined by the EMS, an Unable To Comply exception is raised. e) If a specified SNP or SNPP ID is not known to the EMS, then an Entity Not Found exception is raised. f) If a specified TNA name or Group TNA name is not known to the EMS, then an Entity Not Found exception is raised. g) If the EMS determines that the TNA name structure is not valid for any TNA name, then an Invalid Input exception is raised. h) If a Connection with the NMS specified name already exists, then an Object In Use exception is raised. i) If one of the resources (e.g., TPs) identified in a Connection is in use, then an Object In Use exception is raised. j) If the NMS requests Connection user label uniqueness and the specified user label already exists in the EMS, then User Label In Use exception is raised. k) If the number of additional Connections exceeds the maximum provisionable, Capacity Exceeded exception is raised. l) If the EMS/Control Plane cannot support the type of routing constraints (e.g., capability not available in the EMS or network), then an Unsupported Routing Constraints exception is raised. m) If the EMS/Control Plane does not support specified diversity criteria, then an Unsupported Routing Constraints exception is raised. n) If the EMS/Control Plane does not support specified Co-routing criteria, then an Unsupported Routing Constraints exception is raised. o) If diversity or Co-routing are requested but not all Connections to be added do specify a "Route Group Label", then an Unable To Comply exception is raised. p) If the EMS/Control Plane cannot satisfy the requested routing criteria, within the boundaries of the effort level requested, then an Unsupported Routing Constraints exception is raised. q) If the layer parameters are inconsistent for example where two parameters at one TP conflict or where a parameter at one end of the Connection has a value that is illegal with respect to a parameter value at the other end of the Connection, then an Unable To Comply exception is raised. 3) If the request is valid, the EMS sets the requested TP data (best effort).

Use Case 7.6.3: (UC_22) NMS adds one or more Connections to a Call

4) In a Control Plane environment and if the request is valid:

    a) If the EMS/network can not determine the route without Connection rearrangement and Connection Route Rearangement is not allowed, an Unsupported Routing Constraints exception is raised.

    b) The EMS/network determines the route(s) of the Connections, including the routing constraints associated with the Call.

    c) If routes for all requested Connections were found within timer T0, the Connection state is set to “Complete” and the EMS sends an indication of completion of the request to the NMS indicating also the TP data that was set, if any.

    d) If routes were not found for all requested Connections within timer T0, then zero or more Connections have their state set to “Complete” and one or more Connections have their state set to “Searching” (the Call state remains unchanged) and the EMS sends an indication of completion of the request to the NMS indicating the diversity information and also the TP data that was set, if any.

    e) The EMS sends Connection creation notifications to reflect the changes (indicating the appropriate states of the Call/Connections).

    f) If the Call had no Connections prior to this request the EMS sends an object change notification to update the Call id if it changes as a result of the addition of the first connection (a Call id may not be available prior to routing the first Connection).

    g) The EMS/network will keep trying to find route(s) within the constraints.

    h) If a Connection is routed its state will change to “Complete” and the EMS sends a state change notification to reflect the change in state of the Connection.

    i) If prior to the final Connection being routed timer T1 expires all newly added Connections will be deleted. If the Call id is no longer known an attribute value change notification will be sent to indicate the change in the value of the Call id (i.e. the Call will be restored to its value prior to the request). The TP data that was set will be restored to the previous values.

5) In a non Control Plane environment and if the request is valid:

    a) The EMS/network determines the route(s) of the Connections, considering the routing constraints associated with the Call.

    b) For each requested Connection the EMS tries to create an SNC:

    - For all successfully created SNCs the EMS creates a corresponding Connection with a Connection state set to “Not Applicable”.
    - For all SNCs that could only be created in “Partial” state no Connection object is created by the EMS.
    - The EMS sends an indication of the completion of the request to the NMS indicating the created Connections, the “Partial” SNCs and also the TP data that was set, if any.

Use Case 7.6.3: (UC_22) NMS adds one or more Connections to a Call

<table>
<tr><td></td><td>
The EMS sends Connection and SNC creation notifications to reflect the changes (indicating the appropriate states of the new Connections and SNCs) the diversity information and also the TP data that was set if any.

Note:

Each Connection created in the Control Plane is supported by Connection Segments (represented through the interface as Connections) in MLRAs throughout the network (between the source and destination of the Call). A Connection Segment creation, deletion or change is detected by the EMS managing the MLRA. The EMS sends the appropriate object creation, deletion or attribute value change notification to reflect the change (refer to Use Case 8.2.7: (UC_8) EMS notifies of a new Call and its Connection(s)).
Connection notifications are NOT reported for Routing Nodes.
The EMS that controls the Call originating point shall raise the object creation or deletion notifications and state change notifications for the Call and for all top-level Connections.
In the case where timer T1 expires the Connection Segments will be removed from the network (coordinated by Control Plane signaling).

Where the EMS does not control the entire Call other EMSs in the network will detect deletion of Connection Segments. When a Connection Segment deletion is detected by the EMS managing the MLRA, the EMS sends the appropriate object deletion and attribute value change notification to reflect the change (refer to Use Case 8.2.7: (UC_8) EMS notifies of a new Call and its Connection(s)).
Where the EMS controls the Call originating point it shall raise the deletion notifications for all top-level Connections added during the operation.

In the case where timer T1 expires the order of Connection deletion notifications is not specified.

</td></tr>
<tr><td>Ends when</td><td>
In case of success:

The NMS receives an indication of success of the action.
The EMS returns the complete information of the added Connections.
The EMS sends appropriate attribute value change and state change notifications for the Call, and the appropriate object reation notifications for the Connections which were created.
In a non Control Plane environment and in case not all SNCs could be created as requested, the EMS returns the failed SNCs.

In case of failure:
The NMS receives an exception as an indication of the failure of the request.
</td></tr>
</table>

Use Case 7.6.3: (UC_22) NMS adds one or more Connections to a Call

Post-conditions	In case of success: 1) The corresponding Connection and SNC objects have been created and the NMS received the object creation notifications. 2) If timer T1 is still running the next conditions may also occur: • Upon completion of each Connection, the corresponding state change notifications are sent. • If the final requested Connection has been created, a state change notification for the Call is issued to the NMS and timer T1 is cancelled. 3) If timer T1 expires, for each Connection, the Connection object is deleted and the NMS is notified via object deletion notification. In case of failure: Nothing has changed in the System, i.e., the Call is unchanged.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 397}, {Requirement II. 398}

7.6.4 NMS removes one or more Connections from Call

Use Case 7.6.4: (UC_23) NMS removes one or more Connections from a Call

Use Case Name	NMS removes one or more Connections from a Call
Summary	This operation provides a way to remove one or more supporting Connections (i.e., top-level Connections or Subnetwork Connections) from an existing call and set any necessary TP data. This operation applies for all types of Calls (i.e. Calls established through the UNI as well as those established by the NMS). The NMS is allowed to remove top-level Connections that support a Call that was established through UNI signalling. The EMS is required to delete a number of specified supporting Connections associated with a specified Call and to modify a set of TPs. The returned structure provides a list of TP data that was successfully set. In the non Control Plane case the EMS has to make the effort to remove any partial SNCs resulting from this request. If this is not possible, the EMS will return the list of SNCs which could not be deleted. Control Plane related Notes: • The EMS response is an acceptance of the Connection delete request, not an indication that all resources in the network have been released. Actual release of resources will be given by deletion notifications. • It is assumed that the Control Plane will remove the Connections from the network, along all partitioning levels, delete the associated Connection objects and send appropriate notifications to the NMS via the EMS. As the Control Plane is potentially not managed by a single EMS it is possible that there will be some delay before the Connection objects related to intermediate MLRAs are removed by their respective EMSs. There is no Partial state for top-level Connections. The Connection objects related to the Call known by the EMS that controls the originating point of the Call will be deleted as a result of the request.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends the remove Connections from a Call request to the EMS.

Use Case 7.6.4: (UC_23) NMS removes one or more Connections from a Call

Description	1) The NMS sends the request to remove one or more Connections from an existing Call. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If a specified object is not of the right type (e.g., Call, Connection, TP), an Invalid Input exception is raised. c) If a specified object (e.g., Call, Connection, TP) is not known to the EMS, then an Entity Not Found exception is raised. d) If the EMS does not control the originating end of the Call specified, then an Unable To Comply exception is raised. e) If a Connection specified is not supporting the Call specified, then an Unable To Comply exception is raised. f) If the layer parameters are inconsistent, for example where two parameters at one TP conflict, then an Invalid Input exception is raised. 3) If the request is valid: a) The EMS sets the requested TP data (best effort). b) If the Call is in the “In Progress” state and the Connections to be deleted are the only Connections in the “Searching” state then the Call will transition to the “Established” state and an appropriate notification will be sent. c) The EMS deletes each identified SNC. The EMS has to make the effort to remove any partial SNCs resulting from this operation. If this is not possible, the EMS will return the list of SNCs which could not be deleted. d) The appropriate attribute value change and state change notifications, the diversity information and also the TP data that was set, if any, are sent by the EMS. e) The appropriate Connection deletion notification(s) is(are) sent by the EMS.
Ends when	In case of success: 1) The NMS receives an indication of success of the operation. 2) The EMS sends appropriate attribute value change and state change notifications for the Call, and the appropriate deletion notifications for the Connections which were deleted. 3) In case not all SNCs could be deleted, the EMS returns the list of the SNCs which could not be deleted. In case of failure: The NMS receives an exception as an indication of the failure of the request.

Use Case 7.6.4: (UC_23) NMS removes one or more Connections from a Call

Post-conditions	In case of success: 1) The corresponding Connections have been deleted. 2) SNCs that could not be deleted are in the Partial state. 3) The Connections in subordinate MLRAs have been deleted and the NMS has been notified. In case of failure: Nothing has changed in the System, i.e., the Call is unchanged.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 399}

7.6.5 NMS modifies a Call

Use Case 7.6.5: NMS modifies a Call

<table>
<tr><td>Use Case Name</td><td>NMS modifies a Call</td></tr>
<tr><td>Summary</td><td>The NMS sends a request to modify the following attributes of a Call:
• User label
• Owner
• Network Access Domain
• Additional information.

This use case only describes the modification of the attributes listed above. The number of connections in a Call can also be modified and this is described in Use Case 7.6.3: (UC_22) NMS adds one or more Connections to a Call and Use Case 7.6.4: (UC_23) NMS removes one or more Connections from a Call.

This request is best effort except for the case of the user label when user label uniqueness is requested.</td></tr>
<tr><td>Actors</td><td>NMS</td></tr>
<tr><td>Pre-conditions</td><td>The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.</td></tr>
<tr><td>Begins when</td><td>The NMS sends the modify the Call request to the EMS.</td></tr>
<tr><td>Description</td><td>1) The NMS sends the request to the EMS to modify a Call.
2) The EMS validates the request:
a) If the syntax is in error, an Invalid Input exception is raised.
b) If the Call to be modified does not exist, an Entity Not Found exception is raised.
c) If the Call has already the required information, the EMS returns success but no notification is generated.
d) If a new user label is provided and if the uniqueness of the user label is required, the EMS checks the user label for uniqueness; i.e., if a Call with the same value for the user label already exists in the EMS. If the user label is not unique a USer Label In Use exception is raised.
3) The EMS modifies the Call as requested.
4) The EMS replies with a success indication.
5) The appropriate attribute value change notifications are sent by the EMS to the notification service.</td></tr>
<tr><td>Ends when</td><td>In case of success:
The EMS returns the distinguishing information of the modified Call.
In case of failure:
The NMS receives an exception as an indication of the failure of the request.</td></tr>
</table>

Use Case 7.6.5: NMS modifies a Call

Post-conditions	In case of success: The Call has been modified. In case of failure: Nothing has changed in the System, i.e., the Call has not been modified.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 395}, {Requirement II. 396}

7.6.6 NMS retrieves all the Connections and SNCs for a specific list of Calls

Use Case 7.6.6: (UC_7a) NMS retrieves all the Connections and SNCs for a specific list of Calls

<table>
<tr><td>Use Case Name</td><td>NMS retrieves all the Connections and SNCs for a specific list of Calls</td></tr>
<tr><td>Summary</td><td>The NMS sends a request to retrieve the name and details of the Calls, associated Connections and SNCs in the scope of specified top-level MultiLayer Routing Area or MultiLayer Subnetwork.
The input of this operation are:
• A top-level MLRA name or MLSN name
• And optionally one of:
  • An ME name
  • A TP name
  • A Call name
The EMS is required to reply with:
• List of Call Objects and
• For each Call a list of:
  • Connection Objects that support the Call
  • SNC Objects that support theCall
Note:
• A Call that has at least one Connection which terminates on the specified TP itself or terminates on any TP contained in the specified TP will be returned.
• The reply can be an empty list, if there are no Calls established from any of the points controlled by the EMS.</td></tr>
<tr><td>Actors</td><td>NMS</td></tr>
<tr><td>Pre-conditions</td><td>The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.</td></tr>
<tr><td>Begins when</td><td>The NMS sends the retrieve all the Connections and SNCs for a specific list of Calls request to the EMS.</td></tr>
</table>

Use Case 7.6.6: (UC_7a) NMS retrieves all the Connections and SNCs for a specific list of Calls

Description	1) The NMS sends the retrieve all the Connections and SNCs for a specific list of Calls request to the EMS. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If a specified object is not of the right type (e.g., MLRA, MLSN, TP, ME, Call), an Invalid Input exception is raised. c) If the specified MLRA/MLSN/TP/ME/Call is not known to the EMS, an entity Not Found exception is raised. d) If the specified MLRA is not a top-level Routing Area, an Unable To Comply exception is raised. e) If an MLSN is specified and an MLRA exists, an Unable To Comply exception is raised. 3) If the request is valid the EMS returns to the NMS: a) A list of Calls. • If an ME or TP name is specified only the Calls that have at least one connection that terminates on the named ME or TP will be returned. b) For each Call, a list of Connections and/or SNCs. • All connections that support a Call will be returned regardless of the ME or TP from which they originate.
Ends when	In case of success: The NMS receives the list of Call, Connection and SNC objects, matching the request criteria. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: The NMS has received the requested information from the EMS. In case of failure: Nothing has changed in the System, i.e.,the NMS has not received the requested information.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 404}, {Requirement II. 405}, {Requirement II. 406}, {Requirement II. 407}, {Requirement II. 408}, {Requirement II. 409}, {Requirement II. 410}, {Requirement II. 411}.

7.6.7 NMS requests EMS to set/modify diversity and co-routing parameters of a Call

Use Case 7.6.7: (UC_33) NMS requests EMS to set/modify diversity and co-routing parameters of a Call

Use Case Name	NMS requests EMS to set/modify diversity and co-routing parameters of a Call

Use Case 7.6.7: (UC_33) NMS requests EMS to set/modify diversity and co-routing parameters of a Call

Summary	This operation allows the NMS to request the EMS to assign diversity and co-routing parameters to an existing Call and its supporting Connections. In the situation where a request from the NMS to the EMS will apply parameters that will enforce diversity between existing Connections of the Call (i.e. where different Route Group Labels are assigned to different Connections of the Call), different network preconditions need to be considered: • If the routes of the existing Connections already satisfy the diversity parameters that are to be applied to the Call then no rerouting will be necessary and the operation will be acceptable regardless of the state of the “Connection Route Rearrangement Allowed” parameter in the request. • If the routes of the existing Connections do not satisfy the diversity parameters that are to be applied to the Call and as a consequence rerouting will be necessary, the operation will only be acceptable if the value of the “Connection Route Rearrangement Allowed” parameter in the request is such that traffic impact is allowed. The operation will be rejected if the “Connection Route Rearrangement Allowed” parameter does not allow for rerouting. • If the EMS cannot determine whether the level of requested diversity is satisfied or not by the existing Call and “Connection Route Rearrangement Allowed” does not allow rerouting the operation will be rejected. In a Control Plane environment, if the EMS accepts the operation where rerouting is required, one or more Connections may enter the “Searching” state. There will be no rollback and the Connections may remain in the “Searching” state. In a non Control Plane environment, if the EMS accepts the operation where rerouting is required, one or more SNCs may enter the “Partial” state. If all diversity requirement parameters (i.e., “Link Diversity level of Effort”, “Node Diversity level of effort” and “Co-routing level of effort") are set to “None” and the “Number of Route Groups” parameter is either “NA” or “0”, the EMS will remove all diversity and co-routing requirements for the Call and the Route Group labels of the Connections will be set to an empty value. Note: • If all Connections of the Call are in different Route Groups and the “Link Diversity level of Effort” and “Node Diversity level of effort” are set to “Mandatory” then all Connections are required to be diverse. • If all Connections of a Call are in the same Route Group then all Connections are required to be co-routed (according to the level of effort specified in the Call). • If the diversity statements in the Call are “None” then the Control Plane has the choice to co-route or not. • As co-routing is always best effort there is not requirement for rerouting even when the Connection routes do not match the requested co-routing parameters. If the “Connection Route Rearrangement Allowed” parameter allows for rerouting the EMS/Control Plane may reroute to improve the co-routing situation. This is a local EMS decision.
Actors	NMS

Use Case 7.6.7: (UC_33) NMS requests EMS to set/modify diversity and co-routing parameters of a Call

Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends the set diversity parameters request to the EMS.
Description	1) The NMS sends the request to assign the diversity parameters to an existing Call to the EMS. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If a specified object is not of the right type (Call, Connection), an Invalid Input exception is raised. c) If a specified object (e.g., Call, Connection) is not known to the EMS, an Entity Not Found exception is raised. d) If the EMS does not control the originating end of the Call specified, an Unable To Comply exception is raised. e) If the EMS/Control Plane does not support specified diversity criteria, an Unsupported Routing Constraints exception is raised. f) If the EMS/Control Plane does not support specified Co-routing criteria, an Unsupported Routing Constraints exception is raised. g) If diversity or co-routing is requested and the “Number of Route Groups” is provided, then the EMS checks that all Connections of the Call do not have a “Route Group Label”. If this is not true, an Unable To Comply exception is raised. h) If diversity or co-routing is requested and no value is provided for the “Number of Route Groups”, then the EMS checks that all Connections of the Call already have a “Route Group Label” or a “Route Group Label” is provided by the NMS for the Connections that do not have a “Route Group Label”. If this is not true, an Unable To Comply exception is raised. i) If the “Connection Route Rearrangement Allowed” parameter is set to not allow rerouting and rerouting would be required in order to satisfy the request, an Unable To Comply exception is raised. j) If the EMS can not determine whether the level of requested diversity is satisfied or not by the existing Call and “Connection Route Rearrangement Allowed” does not allow rerouting, an Unable To Comply exception is raised. 3) If the request is valid and if no rerouting is required to satisfy the request: a) The EMS makes the necessary adjustments to satisfy the request. b) EMS sends an indication of completion of request to the NMS. c) EMS sends attribute value change notifications to reflect the changes. 4) If the request is valid and rerouting is required to satisfy the request: a) If the routes are not completely provided by the NMS, the EMS/Control Plane determines the routes of the Connections, according to the co-routing and diversity constraints associated with the request.

Use Case 7.6.7: (UC_33) NMS requests EMS to set/modify diversity and co-routing parameters of a Call

	b) If routes for all the Connections were found within timer T0, the Call state is set to "Established" (may not have been "Established" prior to request) and the Connection state is set to "Complete" (may not have been "Complete" prior to the request). The EMS sends state change and attribute value change notifications for the Call and Connections to reflect the changes (indicating the appropriate states of the Call/Connections and the diversity information). The EMS sends an indication of the completion of request to the NMS. c) If routes could not be found for all the Connections within timer T0, the Call state is set to "In Progress" and zero or more Connections have a state of "Complete" and one or more Connections have a state of "Searching" and the EMS sends an indication of the completion of request to the NMS. The EMS sends state change and sttribute value change notifications for the Call and Connection to reflect the changes (indicating the appropriate states of the Call/Connections). The EMS/Control Plane will keep trying to find routes within the constraints. For each Connection that is routed after timer T0 expires its state will change to "Complete" and the EMS sends an state change notification to reflect the change in the state of the Connection. When the final Connection of the Call changes state to "Complete" (i.e. all Connections are "Complete") the Call state changes to "Established" and the EMS sends attribute value change notifications to reflect the change in the state of the Call and the diversity information. Note: There is no roll-back. Note: • Each Connection created in the Control Plane is supported by Connection Segments (represented through the interface as Connections) in subordinate MLRAs throughout the network (between the source and destination of the Call). A Connection Segment creation, deletion or change is detected by the EMS managing the MLRA. The EMS sends the appropriate object creation, object deletion , state change or attribute value change notifications to reflect the change (refer to Use Case 8.2.7: (UC_8) EMS notifies of a new Call and its Connection(s)). Notifications are NOT reported for Connection Segments in Routing Nodes.
Ends when	In case of success: The NMS receives an indication of success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.

Use Case 7.6.7: (UC_33) NMS requests EMS to set/modify diversity and co-routing parameters of a Call

Post-conditions	In case of success: The requested diversity has been set and the NMS received the appropriate notifications and the diversity information. In case of failure: Nothing has changed in the System.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 402}, {Requirement II. 403}, {Requirement II. 400}, {Requirement II. 401}

7.7 Connection Management Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.7.1:	NMS creates a Subnetwork Connection (SNC)	{Requirement II. 082}, {Requirement II. 083}, {Requirement II. 084}
Use Case 7.7.2:	NMS activates a Subnetwork Connection (SNC)	{Requirement II. 086}, {Requirement II. 087}
Use Case 7.7.3:	NMS creates and activates a Subnetwork Connection (SNC)	{Requirement II. 088}, {Requirement II. 089}
Use Case 7.7.4:	NMS adds a route to a Subnetwork Connection (SNC)	{Requirement II. 241}, {Requirement II. 242}
Use Case 7.7.5:	NMS removes a route from a Subnetwork Connection (SNC)	{Requirement II. 243}, {Requirement II. 244}
Use Case 7.7.6:	NMS creates-modifies the route of a Subnetwork Connection (SNC)	{Requirement II. 245}, {Requirement II. 246}
Use Case 7.7.7:	NMS deactivates a Subnetwork Connection (SNC)	{Requirement II. 090}, {Requirement II. 091}
Use Case 7.7.8:	NMS deletes a Subnetwork Connection (SNC)	{Requirement II. 092}, {Requirement II. 093}
Use Case 7.7.9:	NMS deactivates and deletes a Subnetwork Connection (SNC)	{Requirement II. 094}, {Requirement II. 095}
Use Case 7.7.10:	EMS reroutes a Subnetwork Connection (SNC)	{Requirement II. 084}
Use Case 7.7.11:	NMS queries EMS Connection Management Mode	{Requirement II. 100}
Use Case 7.7.12:	NMS creates and activates a point-to-point Ethernet Service using fragmentation	
Use Case 7.7.13:	NMS modifies a point-to-point Ethernet Service with fragmentation	
Use Case 7.7.14:	NMS deletes a point-to-point Ethernet Service with fragmentation	
Use Case 7.7.15:	NMS creates a flexible IMA group	
Use Case 7.7.16:	NMS deletes a flexible IMA group	
Use Case 7.7.17:	NMS modifies the transport capacity or the routing targets of a flexible IMA group	

7.7.1 NMS creates a Subnetwork Connection (SNC)

Use Case 7.7.1: NMS creates a Subnetwork Connection (SNC)

Use Case Name	NMS creates a Subnetwork Connection (SNC)

Use Case 7.7.1: NMS creates a Subnetwork Connection (SNC)

<table>
<tr><td>Summary</td><td>
This use case describes how an NMS can create either a point-to-point SNC or the leg of a broadcast connection.
This use case requires that the EMS supports the Pending state for SNCs.
The EMS is required to create an SNC of the specified SNC type, layer rate, directionality and grade of impact between the specified A end and Z end termination points. If the EMS cannot meet any of these parameters, an appropriate exception is raised.
If non-network routed protocols are used for route determination and the NMS did not request network routing:
<blockquote>
The EMS creates an SNC of the specified static protection level that best matches the specified protection effort. If the protection effort is same, the EMS creates an SNC with the specified static protection level. If the protection effort is sameOrBetter or sameOrWorse, the EMS first attempts to provide the protection level requested. If it is not possible, the EMS attempts to provide better or worse static protection according to the protection effort parameter. However, if the NMS requests static protection level partiallyProtected with protection Effort sameOrBetter, the EMS may attempt to provide fullyProtected first.
If the NMS specifies routing constraints in the request and the EMS supports this feature, the EMS is required to either include or exclude specified resources during route selection based on the provided criteria even if there are pending, partial, or active SNCs using the required parts of the route.The NMS may specify the full route in the request. If the NMS does not specify routing constraints or does not provide a full route, then the EMS itself will select the full or partial route respectively. If the EMS does not support the routing constraints feature and routing constraints are specified in the request, or supports the feature but cannot satisfy the NMS routing criteria, an exception is raised.
If the EMS cannot use the routing constraints for a BLSR case, BLSRDirection and Timeslot parameters in additionalInformation may be used for route selection if specified by the NMS.
</blockquote>
If network routed protocols are used for route determination and the NMS requests network routing:
<blockquote>
The EMS requests the network determine the route of the SNC of the specified static protection level.
If the NMS specifies routing constraints in the request and the network supports this feature, the EMS passes these constraints to the network. If the network does not support the routing constraints feature or supports the feature but cannot satisfy the NMS routing criteria, an exception is raised.
If the EMS supports the capability to manage more than one route for the same SNC, then the route of the newly created SNC shall be the intended route.
If an exclusive SNC has been specified, then the EMS must find a route that does not conflict or share XCs or CTPs with any other existing SNC route, in any administrative state.
Once an exclusive (intended) route has been created by the EMS, any further creation operations in which conflicts are detected with the exclusive route shall raise an exception.
</blockquote>
</td></tr>
</table>

Use Case 7.7.1: NMS creates a Subnetwork Connection (SNC)

	For DWDM: A routing constraint for DWDM (frequency) can be specified similar to the use of timeslot for SONET/SDH. If an existing SNC respects all the conditions specified in the NMS request, the EMS is allowed to return the existing SNC. It is also allowed to attempt to create a different SNC. For rerouting behavior, please see Use Case 7.7.4: NMS adds a route to a Subnetwork Connection (SNC) and Use Case 7.7.5: NMS removes a route from a Subnetwork Connection (SNC). Where applicable, the exception contains a list of the failed cross-connections and the reason(s) for failure. Notes: From an NMS perspective, there is no limit on the number of SNCs in the pending state that use the same route. However, the EMS may limit the number of pending SNCs (depending on specific EMS implementation).
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends the SNC creation request to the EMS.
Description	1) The NMS sends the request to create an SNC to the EMS. 2) The EMS validates the request. If the request is not valid (invalid parameters, or the EMS does not support Pending SNC state), an exception is raised. 3) If an SNC with the same properties as specified in the NMS request already exists, the EMS may return that SNC. 4) If the EMS does not support routing constraints and they are specified in the request, or any of the mandatory parameters cannot be satisfied, or the NMS requested user label uniqueness and the specified user label already exists in the EMS, an exception is raised. 5) If non-network routed protocols are used for route determination and the NMS did not request network routing the EMS determines the route of the requested SNC using the specified static protection level and protection effort and optional routing constraints. If the routing constraints are not specified or a complete route is not defined by the NMS, then the EMS itself selects a full or partial route respectively. 6) If a server SNC is required and the EMS has freedom to create the server SNC for this request, the server SNC is created. If the EMS does not have this level of freedom, an exception is raised. Note that a server SNC is defined to be a SNC on a server layer needed to establish the traffic for the (client) SNC to be created. 7) If network routed protocols are used for route determination and the NMS requested network routing the EMS creates the SNC (this does not involve the entering of the cross-connection(s) at the ME(s)). The EMS initializes the SNC with the specified parameters of the request and the route. The SNC state is Pending. 8) The EMS replies with a success indication. 9) The EMS forwards a SNC object creation notification to the notification service.

Use Case 7.7.1: NMS creates a Subnetwork Connection (SNC)

Ends when	In case of success: 1) The NMS receives an indication of success of the action. 2) The EMS returns the created SNC name and its distinguishing information. In case of failure: The NMS receives an indication of failure of the action or exception.
Post-conditions	In case of success: 1) The SNC has been created and not activated. 2) The EMS has forwarded a SNC object creation notification to all NMSs. 3) If generic end point(s) were specified, then the defined end point(s) are replied. In case of failure: The SNC has not been created.
Exceptions	1) Processing failure: The requested operation could not be performed. 2) Not implemented: The EMS does not support this service. 3) Invalid input: At least 1 of the CTP references is invalid. 4) Invalid input: At least 1 of the CTP parameters is invalid. 5) Invalid input: In case a bundled SNC is requested, the GTP endpoints of the SNC do not match, i.e., the GTPs do not have the same number of CTPs and in a particular order with respect to their layer rates or are not of the same layer rate 6) Unable to comply: The EMS cannot create a SNC with the NMS-specified static protection level and protection effort. 7) Unsupported routing constraints: The SNC can not be created because of cross-connection or CTP conflicts with other SNCs. 8) Unsupported routing constraints: Timeslot conflicts with other SNCs. 9) The number of total pending SNCs has exceeded the maximum limit. The limit is dependent on EMS implementation. 10) Unsupported routing constraints: The EMS does not support routing constraints specified. 11) User label in use: The user label uniqueness constraint is not met. 12) Unsupported routing constraints: SNC cannot be created because of conflict with another active or partial SNC. 13) Object in use: The intended route is in conflict with an “exclusive” route of another SNC.
Traceability	{Requirement II. 082}, {Requirement II. 083}, {Requirement II. 084}

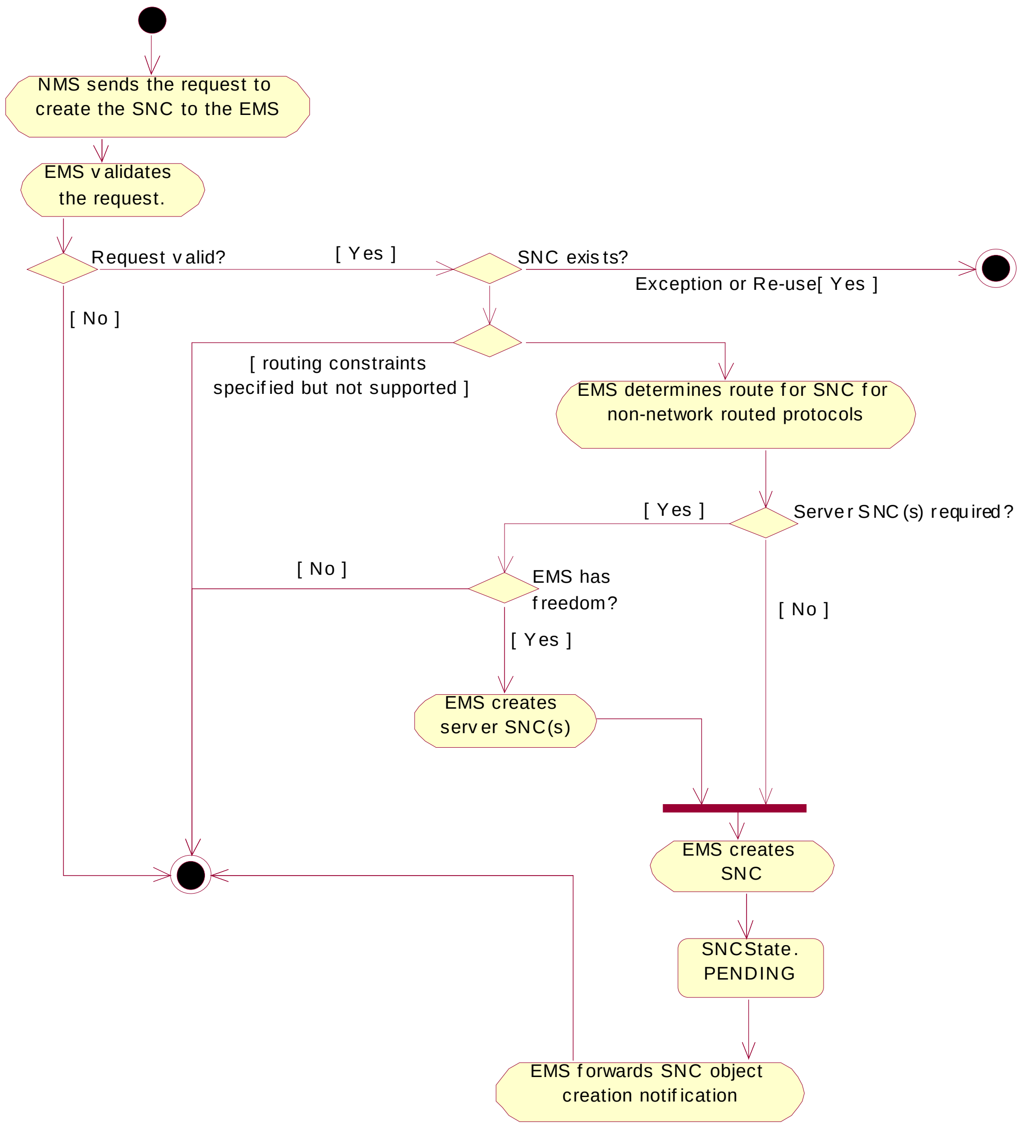


Figure 7.4: Activity Diagram for NMS creates a Subnetwork Connection (SNC)

7.7.2 NMS activates a Subnetwork Connection (SNC)

Use Case 7.7.2: NMS activates a Subnetwork Connection (SNC)

Use Case Name	NMS activates a Subnetwork Connection (SNC)
Summary	The NMS requests to activate a subnetwork connection (SNC). An SNC can be activated while in any state. If transmission parameters are specified for A end or Z end CTPs, the EMS will apply them either before or after the creation of the cross-connections, as appropriate. The alarm reporting on the CTPs and the containing TPs may be turned on by the EMS, unless otherwise specified via the alarm reporting transmission parameter. An already activated SNC can be activated again. Where applicable, the exception or error reason contains a list of the failed CTPs and the reason(s) for failure. If the addressed SNC has more routes, this operation unlocks all the routes, delegating the EMS and/or the network (e.g. restoration process) the actual activation of more appropriate route
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The SNC has been created.
Begins when	The NMS sends the SNC activation request to the EMS.

Use Case 7.7.2: NMS activates a Subnetwork Connection (SNC)

Description	1) The NMS sends the request to activate the specified SNC to the EMS. 2) The EMS validates the SNC reference (e.g. name). If the request is not valid, an exception is raised. 3) If the SNC is already activated, then the EMS replies with a success indication. The EMS may not send the commands to the NE a second time for the cross-connection establishment. However the commands may be sent for the transmission parameters. 4) If the SNC of the referenced name is pending and some contained cross-connections are on another (active) SNC, the EMS behavior depends on whether the SNC is a part of point-to-multipoint configuration or not. If the SNC is a part of point-to-multipoint configuration, the use case is carried out, otherwise an exception (resources occupied) is raised if cross-connection sharing is not supported. If no cross-connections to be established are on another active SNC, the use case is carried out. 5) If the specified aEnd and/or zEnd CTPs are not terminated and mapped at the appropriate connection rate, then the EMS behavior depends on the NMS specified EMS freedom. If the EMS does not have this level of freedom to terminate and map/un-terminate and un-map TPs, an exception is raised. If the NMS specified this level of freedom in the request, the EMS either terminates and maps containing or un-terminates and un-map the contained termination points of the aEnd and/or zEnd CTPs. Refer to Use Case 7.5.1: NMS provisions the mapping mode of a CTP and Use Case 7.5.2: NMS un-maps a server layer CTP. 6) If the TP parameters were specified for at least 1 of the A-end and/or Z-end CTPs then the EMS may provision the parameters on these CTPs before activation of the cross-connections. Refer to Use Case 7.5.7: NMS Provisions the TP Transmission Parameters, Use Case 7.5.8: NMS provisions alarm reporting on for a TP and Use Case 7.5.9: NMS provisions alarm reporting off for a TP. The EMS may put all the containing TPs in service. 7) If the NMS specified this level of freedom in the request and a server SNC is involved and is not active, the server SNC is activated. 8) The EMS initiates the activation of the SNC (which involves entering the cross-connection(s) at the ME(s)). 9) If TP parameters have not been applied before activation of cross-connections, they are applied now. See step #6. 10) If network routed protocols are used for route determination the EMS requests that the network determine the route of the requested SNC using the specified static protection level and protection effort and optional routing constraints. 11) If all of the cross-connections comprising the SNC have been established, then the EMS sets the SNC state of the SNC to active. The EMS updates the connection state of the affected CTPs to either sink, source, or bidirectionally connected. If there are more routes, then it is up to EMS the choice of better route to activate. Once a route has been successfully activated (its actual state is "active") then the SNC is enabled to transit to active state. 12) If network routed protocols are used for route determination and a complete route cannot be determined:

Use Case 7.7.2: NMS activates a Subnetwork Connection (SNC)

	• If the NMS requests that rerouting can be performed. The SNC state shall be set to partial and the network attempts to determine a new route for the SNC. Refer to Use Case 7.7.4: NMS adds a route to a Subnetwork Connection (SNC) and Use Case 7.7.5: NMS removes a route from a Subnetwork Connection (SNC) • If the NMS requests that rerouting cannot be performed. The SNC state shall be set to partial. 13) If not all cross-connections comprising the SNC have been established(*), then the EMS sets the SNC state of the SNC to partial. The EMS updates the connection state of those CTPs that were successfully established to either sink, source, or bidirectionally connected. The EMS replies with a failure indication and error reason. (*) in case of multi-route SNC, this means that the EMS (or the network) was not able to activate any (just unlocked) route of the SNC 14) If there has been a SNC state change, then the EMS forwards a SNC state change notification to the notification service. 15) If there has been a CTP connection state change, then the EMS forwards a CTP state change notification to the notification service. 16) If there are error conditions, (e.g. failure to provision TP transmission parameters) existing after establishing all the XCs, then the EMS will reply with a failure indication and error reason. In this case the SNC state will be active. Otherwise if there are no error conditions the EMS replies with a success indication.
Ends when	In case of success: The NMS receives an indication of success of the action. In case of failure: The NMS receives an indication of failure of the action.

Use Case 7.7.2: NMS activates a Subnetwork Connection (SNC)

Post-conditions	In case of success: 1) The SNC has been activated. 2) If the EMS has provisioned TP parameters, refer to the post-conditions as specified in Use Case 7.5.7: NMS Provisions the TP Transmission Parameters, Use Case 7.5.8: NMS provisions alarm reporting on for a TP and Use Case 7.5.9: NMS provisions alarm reporting off for a TP. 3) If the EMS has performed terminate and map or un-terminate and un-map, refer to the post-conditions (in case of success) as specified in to Use Case 7.5.1: NMS provisions the mapping mode of a CTP and Use Case 7.5.2: NMS un-maps a server layer CTP. 4) If there has been a change in the SNC state of the SNC, then the EMS will forward a state change notification for a SNC state to the NMS. 5) If there has been a CTP connection state change, then state change notification(s) are emitted to the NMS. In case of failure: 1) The SNC has not been completely activated, (i.e. zero or more cross-connections comprising of the SNC have been activated). 2) If the EMS has successfully provisioned TP parameters and failed to activate the SNC, refer to the post-conditions as specified in Use Case 7.5.7: NMS Provisions the TP Transmission Parameters, Use Case 7.5.8: NMS provisions alarm reporting on for a TP and Use Case 7.5.9: NMS provisions alarm reporting off for a TP. 3) If the EMS has successfully performed terminate and map or un-terminate and un-map and failed to activate the SNC, refer to the post-conditions as specified in to Use Case 7.5.1: NMS provisions the mapping mode of a CTP and Use Case 7.5.2: NMS un-maps a server layer CTP. 4) If there has been a change in the SNC state of the SNC as a result of a failure, then the EMS will forward a state change notification for a SNC state to the NMS. 5) If there has been a CTP connection state change as a result of the failure, then state change notification(s) are sent to the NMS.

Use Case 7.7.2: NMS activates a Subnetwork Connection (SNC)

Exceptions	1) Invalid input: Any input parameter is syntactical incorrect. 2) At least 1 of the CTP references failed to be provisioned. See exception list from Use Case 7.5.7: NMS Provisions the TP Transmission Parameters, Use Case 7.5.8: NMS provisions alarm reporting on for a TP and Use Case 7.5.9: NMS provisions alarm reporting off for a TP. 3) Not in valid state: At least 1 of the containing termination points of the CTP references failed to be terminated and mapped or contained CTP(s) failed to be un-terminated and unmapped. See exception list from Use Case 7.5.1: NMS provisions the mapping mode of a CTP and Use Case 7.5.2: NMS un-maps a server layer CTP. 4) NE communications loss. 5) Object in use: XC or CTP conflicts between the active route (with equal or higher priority) of this and other SNCs or when XC creation would involve a TP that has an existing fixed XC that does not match that required for the SNC. 6) Object in use: Timeslot conflicts with other SNCs. 7) Unable to comply: The SNC is in pending state and is in conflict with another active or partial SNC. 8) Entity not found: The SNC reference (e.g. name) or one or more TP references reference objects that do not exist. 9) Not implemented: The EMS does not support this service. 10) Processing failure: The requested operation could not be performed.
Traceability	{Requirement II. 086}, {Requirement II. 087}

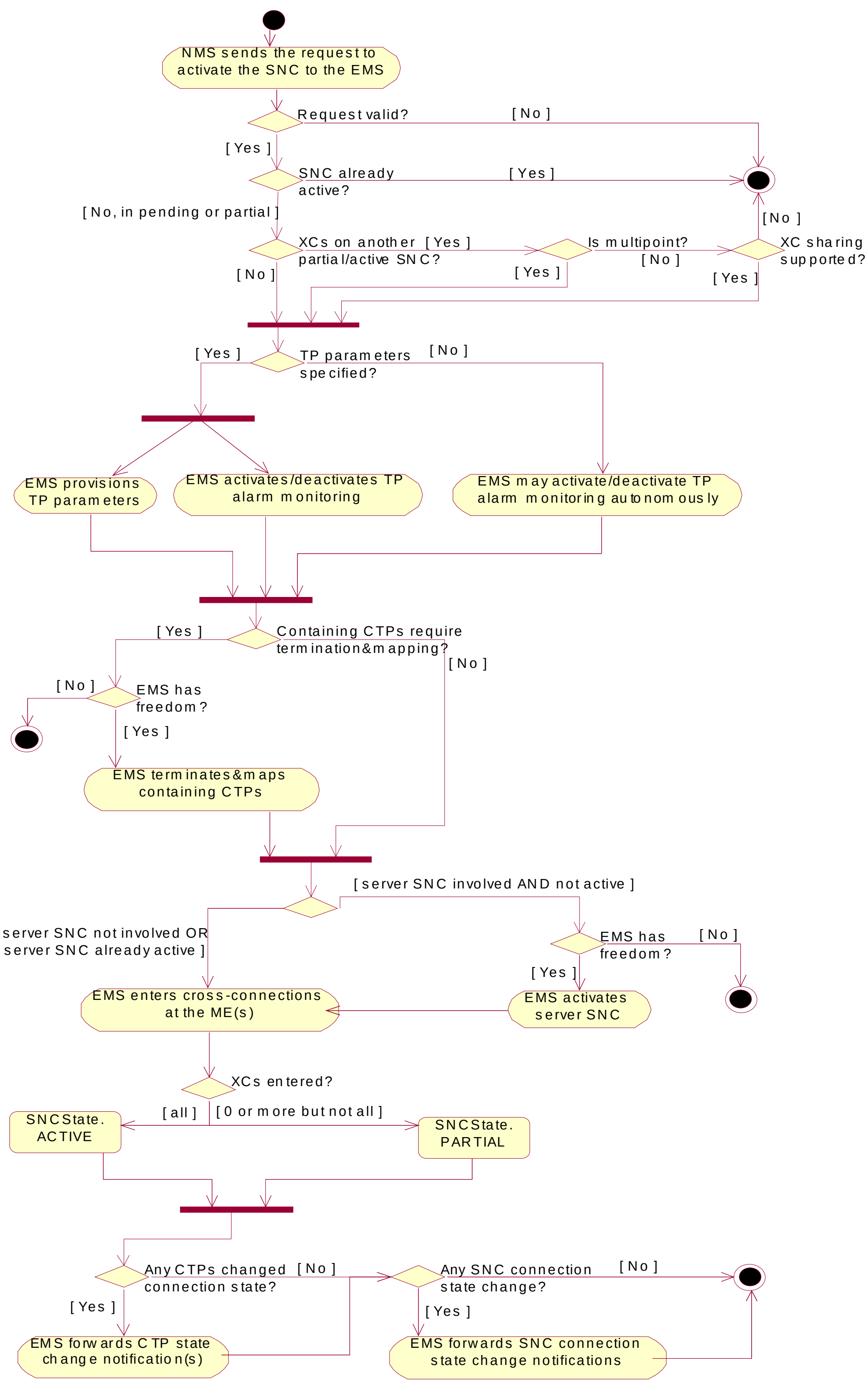


Figure 7.5: Activity Diagram for NMS activates a Subnetwork Connection (SNC)

7.7.3 NMS creates and activates a Subnetwork Connection (SNC)

Use Case 7.7.3: NMS creates and activates a Subnetwork Connection (SNC)

<table>
<tr><td>Use Case Name</td><td>NMS creates and activates a Subnetwork Connection (SNC)</td></tr>
<tr><td>Summary</td><td> This operation provides a way to create and activate a point-to-point subnetwork connection or a leg of a broadcast configuration in one request.
The EMS is required to create an SNC of the specified SNC type, layer rate, directionality and grade of impact between the specified A end and Z end termination points. If the EMS cannot meet any of these parameters, an appropriate exception is raised.
If non-network routed protocols are used for route determination and the NMS did not request network routing:
<blockquote> The EMS creates an SNC of the specified static protection level that best matches the specified protection effort. If the protection effort is same, the EMS creates an SNC with the specified static protection level. If the protection effort is sameOrBetter or sameOrWorse, the EMS first attempts to provide the protection level requested. If it is not possible, the EMS attempts to provide better or worse static protection according to the protection effort parameter. However, if the NMS requests static protection level partiallyProtected with protection Effort sameOrBetter, the EMS may attempt to provide fullyProtected first.
If the NMS specifies routing constraints in the request and the EMS supports this feature, the EMS is required to either include or exclude specified resources during route selection based on the provided criteria even if there are pending, partial, or active SNCs using the required parts of the route.The NMS may specify the full route in the request. If the NMS does not specify routing constraints or does not provide full route, then the EMS itself will select the full or partial route respectively. If the EMS does not support the routing constraints feature and routing constraints are specified in the request, or supports the feature but cannot satisfy the NMS routing criteria, an exception is raised.
If the EMS cannot use the routing constraints for a BLSR case, BLSRDirection and Timeslot parameters in additionalInformation may be used for route selection if specified by the NMS. </blockquote>
If network routed protocols are used for route determination and the NMS requests network routing:
<blockquote> The EMS requests the network determine the route of the SNC of the specified static protection level.
If the NMS specifies routing constraints in the request and the network supports this feature, the EMS passes these constraints to the network. If the network does not support the routing constraints feature or supports the feature but cannot satisfy the NMS routing criteria, an exception is raised. </blockquote>
If the EMS supports the capability to manage more routes for the same SNC, then the route of the newly activated SNC is the INTENDED route. The activation implies that the route is unlocked.
If EXCLUSIVE SNC has been specified, then the EMS must find a route that does not conflict or share XCs or CTPs with any other existing SNC route, in any administrative state. </td></tr>
</table>

Use Case 7.7.3: NMS creates and activates a Subnetwork Connection (SNC)

	Once an EXCLUSIVE (intended) route has been created by EMS, any further creation operation which conflicts with the exclusive route shall be refused. For DWDM: A routing constraint for DWDM (frequency) can be specified similar to the use of timeslot for SONET/SDH. If transmission parameters are specified for A end or Z end CTPs, the EMS will apply them either before or after the creation of the cross-connections, as appropriate. The alarm reporting on the CTPs and the containing TPs may be turned on by the EMS, unless otherwise specified via the alarm reporting transmission parameter. If the pending state is supported, it is possible for the SNC to be created but activation to be rejected if conflicting active or partial SNCs, the resulting SNC will be in pending state. If the pending state is not supported, then this is not possible and the SNC will not be created if activation is rejected. If an existing SNC respects all the conditions specified in the NMS request, the EMS is allowed to return the existing SNC. It is also allowed to attempt to create and activate a different SNC. For rerouting behavior, refer to Use Case 7.7.4: NMS adds a route to a Subnetwork Connection (SNC) and Use Case 7.7.5: NMS removes a route from a Subnetwork Connection (SNC). Where applicable, the exception contains a list of the failed cross-connections and the reason(s) for failure. Note: 1) From an NMS perspective, there is no limit on the number of SNCs in the pending state that use the same route. However, the EMS may limit the number of pending SNCs (depending on specific EMS implementation). 2) The EMS may not support pending SNCs at all. 3) For ATM specific behavior, refer to Use Case 7.15.1: NMS creates and activates an ATM Subnetwork Connection (SNC).
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends the SNC create and activate request to the EMS.

Use Case 7.7.3: NMS creates and activates a Subnetwork Connection (SNC)

Description	Case A: Pending SNC(s) not supported: 1) The NMS sends the request to create and activate a SNC to the EMS. 2) The EMS validates the request. If the request is not valid, an exception is raised. 3) If an SNC with the same properties as specified in the NMS request already exists, the EMS may reuse that SNC. 4) If the EMS (or the network for network routed) does not support routing constraints and they are specified in the request, or any of the mandatory parameters cannot be satisfied, or the NMS requested user label uniqueness and the specified user label already exists in the EMS, an exception is raised. 5) If some of cross-connections to be established for a referenced SNC are on another, active or partial SNC, the EMS behavior depends on whether the requested SNC is to be a part of broadcast configuration or not. If the requested SNC is to be a part of the broadcast configuration, the use case is carried out, otherwise an exception (resources occupied) is thrown if cross-connection sharing is not supported. If none of the cross-connections to be established are on another, active or partial SNC, the use case is carried out. 6) If non-network routed protocols are used for route determination and the NMS did not request network routing, the EMS determines the route of the requested SNC using the specified static protection level, protection effort and optional routing constraints. If routing constraints are not specified or a complete route is not defined by the NMS, then the EMS itself selects a full or partial route respectively. 7) If network routed protocols are used for route determination and the NMS requests network routing, the network determines the route of the requested SNC using the specified static protection level and optional routing constraints. If routing constraints are not specified or a complete route is not defined by the NMS, then the network itself selects a the route. 8) If a server SNC is required and the EMS has freedom to create the server SNC for this request, the server SNC is created if it does not already exist and then activated. If the NMS did not specify this level of freedom in the request, an exception is raised. 9) If the TP parameters were specified for at least 1 of the A-end and/or Z-end CTPs in the original create request, then the EMS may provision the parameters on these CTPs before activation of cross-connections. Refer to Use Case 7.5.7: NMS Provisions the TP Transmission Parameters, Use Case 7.5.8: NMS provisions alarm reporting on for a TP and Use Case 7.5.9: NMS provisions alarm reporting off for a TP. 10) If the specified aEnd and/or zEnd CTPs are not terminated and mapped at the appropriate connection rate, then the EMS behavior depends on the NMS specified EMS freedom. If the EMS does not have this level of freedom to terminate and map/un-terminate and un-map TPs, an exception is raised. If the NMS specified this level of freedom in the request, the EMS either terminates and maps containing or un-terminates and un-maps the contained termination points of the aEnd and/or sEnd CTPs. Refer to Use Case 7.5.1: NMS provisions the mapping mode of a CTP and Use Case 7.5.2: NMS un-maps a server layer CTP.

Use Case 7.7.3: NMS creates and activates a Subnetwork Connection (SNC)

	11) The EMS initiates the activation of the SNC (which involves entering the cross-connection(s) at the ME(s)). 12) If TP parameters have not been applied before activation of cross-connections, they are applied now. See step #9. 13) If all of the cross-connections comprising the SNC have been established, then the EMS sets the SNC state of the SNC to active. The EMS updates the SNC state of the affected CTPs to either sink, source, or bidirectionally connected. The EMS replies with a success indication. The EMS forwards a SNC object creation notification to the notification service.. The EMS forwards a CTP connection state change notification for all affected CTPs to the notification service. 14) If there are error conditions, (e.g. failure to provision TP transmission) existing after establishing all the cross-connections, then the EMS will reply with a failure indication and error reason. In this case the SNC state of the SNC will be active. 15) If one or more (possibly all) of the cross-connections comprising the SNC have been established (*) then the EMS sets the SNC state of the SNC to partial. The EMS updates the connection state of those CTPs that were successfully established to either sink, source, or bidirectionally connected. The EMS replies with a failure indication and error reason. (*) in case of multi-route SNC, this means that the EMS (or the network) was not able to activate any (just unlocked) route of the SNC. 16) The EMS forwards a SNC object creation notification to the notification service. 17) The EMS forwards a CTP connection state change notification for all CTPs that were successfully established to the notification service. 18) If none of the cross-connections comprising the SNC have been established, then the EMS replies with a failure indication and error reason. The SNC is not created. Case B: Pending SNC(s) supported: 1) The NMS sends the request to create and activate an SNC to the EMS. 2) The EMS creates the SNC. Refer to Use Case 7.7.1: NMS creates a Subnetwork Connection (SNC). 3) The EMS activates the SNC. Refer to Use Case 7.7.2: NMS activates a Subnetwork Connection (SNC).

Use Case 7.7.3: NMS creates and activates a Subnetwork Connection (SNC)

Ends when	Case A: Pending SNC(s) not supported: In case of success: 1) The SNC has been created and activated. 2) If the EMS has provisioned TP parameters, refer to the post-conditions as specified in Use Case 7.5.7: NMS Provisions the TP Transmission Parameters, Use Case 7.5.8: NMS provisions alarm reporting on for a TP and Use Case 7.5.9: NMS provisions alarm reporting off for a TP. 3) If the EMS has performed terminate and map or un-terminate and un-map, refer to the post-conditions (in case of success) as specified in Use Case 7.5.1: NMS provisions the mapping mode of a CTP and Use Case 7.5.2: NMS un-maps a server layer CTP. 4) The EMS has forwarded a SNC object creation notification to the notification service. 5) The EMS has forwarded CTP connection state change notification(s) to the notification service. In case of failure: 1) The SNC has not been created (i.e. invalid request or no cross-connections comprising the SNC were activated). 2) The SNC has been created but has not been completely activated, (i.e. one or more cross-connections comprising of the SNC have been activated). 3) If the EMS has successfully provisioned TP parameters and failed to activate the SNC, refer to the post-conditions as specified in Use Case 7.5.7: NMS Provisions the TP Transmission Parameters, Use Case 7.5.8: NMS provisions alarm reporting on for a TP and Use Case 7.5.9: NMS provisions alarm reporting off for a TP. 4) If the EMS has successfully performed terminate and map or un-terminate and un-map and failed to activate the SNC, refer to the post-conditions as specified in in Use Case 7.5.1: NMS provisions the mapping mode of a CTP and Use Case 7.5.2: NMS un-maps a server layer CTP. 5) If there has been a CTP connection state change as a result of the failure, then state change notification(s) are emitted to the notification service. Case B: Pending SNC(s) supported: Refer to the Refer to Use Case 7.7.1: NMS creates a Subnetwork Connection (SNC) and Use Case 7.7.2: NMS activates a Subnetwork Connection (SNC).

Use Case 7.7.3: NMS creates and activates a Subnetwork Connection (SNC)

Post-conditions	Case A: Pending SNC(s) not supported: 1) At least 1 of the CTP references is invalid. 2) At least 1 of the CTP parameters is invalid. 3) The EMS cannot meet specified static protection level and protection effort for the referenced SNC. 4) the EMS does not support the routing constraints specified 5) Timeslot conflicts with other SNCs. 6) The number of total pending SNCs has exceeded the maximum limit. The limit is dependent on EMS implementation. 7) At least 1 of the CTP references failed to be provisioned. See exception list from Use Case 7.5.7: NMS Provisions the TP Transmission Parameters, Use Case 7.5.8: NMS provisions alarm reporting on for a TP and Use Case 7.5.9: NMS provisions alarm reporting off for a TP. 8) At least 1 of the containing termination points of the CTP references failed to be channelized or contained CTP(s) failed to be de-channelized. See exception list from Use Case 7.5.1: NMS provisions the mapping mode of a CTP and Use Case 7.5.2: NMS un-maps a server layer CTP. 9) Communications failure between the EMS and the ME(s) and this prevents creation and activation of the SNC. 10) the SNC is in conflict with another active or partial SNC and can not be created 11) Cross-connection or CTP conflicts with other SNCs. 12) userLabel uniqueness constraint is not met 13) Non-specific EMS internal failure. Case B: Pending SNC(s) supported: 1) See exception list from Use Case 7.7.1: NMS creates a Subnetwork Connection (SNC). 2) See exception list from Use Case 7.7.2: NMS activates a Subnetwork Connection (SNC).
Exceptions	1) See exception list from Use Case 7.7.1: NMS creates a Subnetwork Connection (SNC). 2) See exception list from Use Case 7.7.2: NMS activates a Subnetwork Connection (SNC).
Traceability	{Requirement II. 088}, {Requirement II. 089}

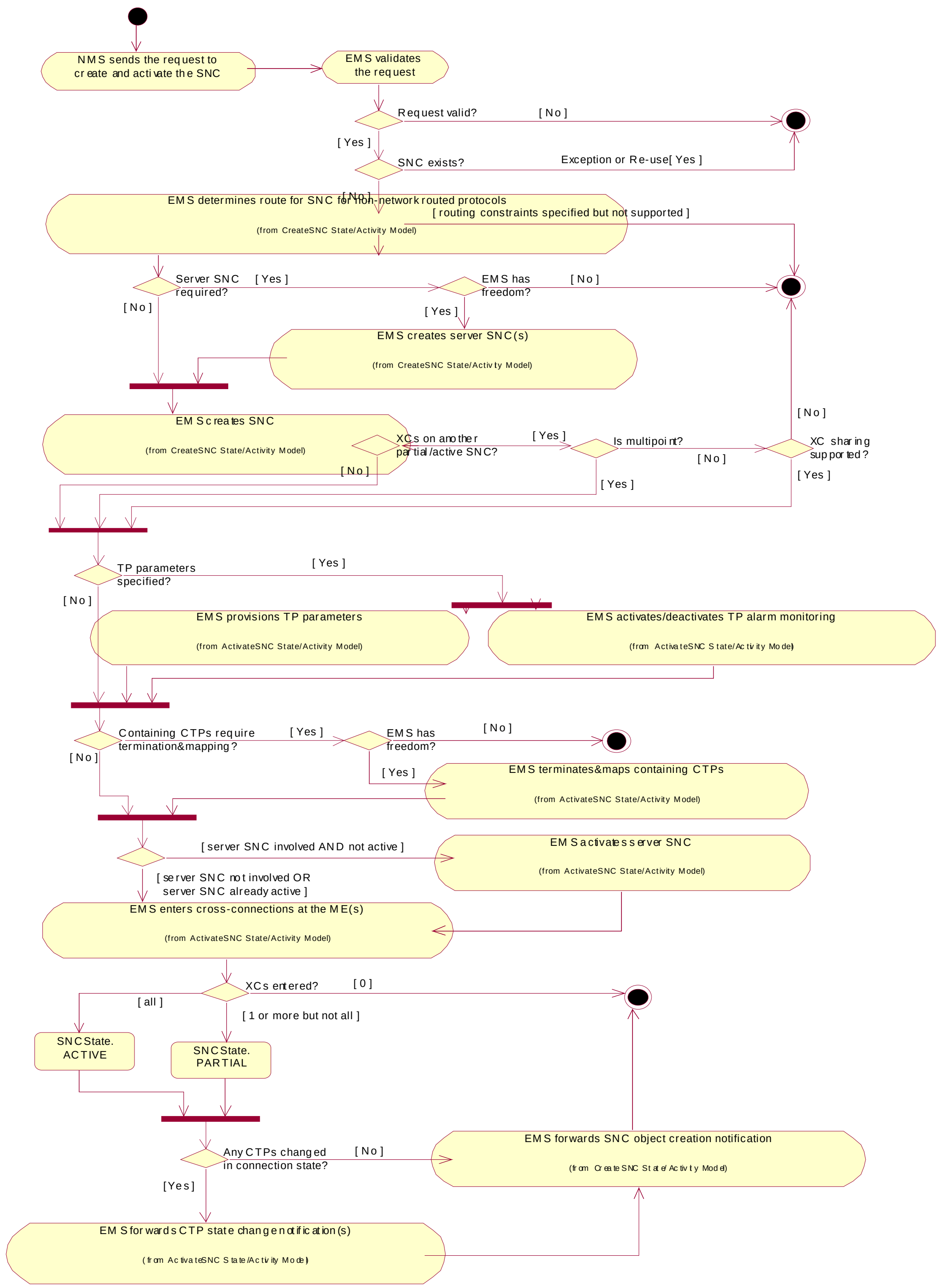


Figure 7.6: Activity Diagram for NMS creates and activates a Subnetwork Connection (SNC) (Pending not supported)

7.7.4 NMS adds a route to a Subnetwork Connection (SNC)

Use Case 7.7.4: NMS adds a route to a Subnetwork Connection (SNC)

Use Case Name	NMS adds a route to a Subnetwork Connection (SNC)
Summary	The NMS requests to add a protection route to a given Subnetwork Connection in an EMS. As a result of (successful) completion of this request, the EMS shall add the new route to the Subnetwork Connection, but shall not attempt to establish (on NEs) any cross connections as side effect of this operation, because the route is created in locked state.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS has a reference to the notification service. 3) An SNC exists
Begins when	he NMS sends an add route request to the EMS.
Description	1) It is possible to specify if the new added route becomes the intended one, and / or if it is exclusive. 2) It is possible to describe zero, more or all routing constraints, i.e. the whole route description.
Ends when	The EMS has completed the route creation.
Post-conditions	The newly added route is available at the interface, in locked state.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed. 3) Invalid input: Any input parameter is syntactical incorrect. 4) Entity not found: Fields of input parameters reference objects that do not exist. 5) Protection effort not met: The NMS requests a route with a static protection level (inherited from SNC) that cannot be met by the EMS. 6) Unable to comply: The EMS is unable to find an appropriate route. 7) Object in use: The route is in conflict with an “exclusive” (even locked) route of another SNC. 8) NE communication loss. 9) Unsupported routing constraints: The EMS does not support the routing constraints specified.
Traceability	{Requirement II. 241}, {Requirement II. 242}

7.7.5 NMS removes a route from a Subnetwork Connection (SNC)

Use Case 7.7.5: NMS removes a route from a Subnetwork Connection (SNC)

Use Case Name	NMS removes a route from a Subnetwork Connection (SNC)

Use Case 7.7.5: NMS removes a route from a Subnetwork Connection (SNC)

Summary	The NMS requests to remove a protection route from a subnetwork connection. As a result of (successful) completion of this request, the EMS shall delete the protection route of addressed Subnetwork Connection. Of course it is possible to delete a locked backup route which is "in use" by other SNC route, because this operation has no side effect on routes of any other SNCs, even if sharing XCs/TPs.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS has a reference to the notification service. 3) An SNC exists 4) The addressed route must not be in the unlocked state 5) The addressed route must not be the intended route
Begins when	The NMS sends a remove route request to the EMS.
Description	The route is removed, so at least the SNC remains with only the intended route.
Ends when	The EMS has completed the route removal.
Post-conditions	The route is no longer exists.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed.The EMS does not support the requested feature. 3) Invalid input: Any input parameter is syntactical incorrect. 4) Entity not found: Fields of input parameters reference objects that do not exist. 5) Not in valid state: The route is in the UNLOCKED state, or the route is the intended one. 6) NE communication loss.
Traceability	{Requirement II. 243}, {Requirement II. 244}

7.7.6 NMS creates-modifies the route of a Subnetwork Connection (SNC)

Use Case 7.7.6: NMS creates-modifies the route of a Subnetwork Connection (SNC)

Use Case Name	NMS creates-modifies the route of a Subnetwork Connection (SNC)
Summary	The NMS requests to modify a route of a Subnetwork Connection. As a result of (successful) completion of this request, the addressed SNC route is modified. If the SNC was in PENDING or PARTIAL state, then the state is unchanged. If the SNC was in ACTIVE state, then the output state is PARTIAL. In case the SNC has several routes, then the administrative state of the addressed route will always transit to LOCKED state.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS has a reference to the notification service.
Begins when	The NMS sends a modify route request to the EMS.
Description	Two classes of modification are available: • add protection leg, remove protection leg • reroute It is possible to describe zero, more or all routing constraints, i.e. the whole route or leg description. At least, it must be possible to • add/remove a protection leg to/from a simple SNC • change from simple to add drop type and vice versa
Ends when	The EMS has completed the route modification.
Post-conditions	The modified route is available at the interface, in locked state. It is not retrievable the current route in the network.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed. 3) Invalid input: Any input parameter is syntactical incorrect. 4) Object in use: The SNC can not be created because of XC or CTP conflicts with other SNCs. 5) Entity not found: Fields of input parameters reference objects that do not exist. 6) Protection effort not met: The NMS requests a new SNC with a static protection level and protection effort that cannot be met by the EMS. 7) Unable to comply: The SNC cannot be created because it cannot comply with any of the input parameter constraints for a reason different than the ones above. 8) Unsupported routing constraints: The EMS does not support the routing constraints specified. 9) User label in use: The user label supplied by the NMS is already in use.
Traceability	{Requirement II. 245}, {Requirement II. 246}

7.7.7 NMS deactivates a Subnetwork Connection (SNC)

Use Case 7.7.7: NMS deactivates a Subnetwork Connection (SNC)

Use Case Name	NMS deactivates a Subnetwork Connection (SNC)
Summary	The NMS requests that an SNC be deactivated from the EMS' managed subnetwork (i.e., de-provisioned from the EMS' managed subnetwork). However, as a result of this action, the EMS continues to maintain the SNC object within the EMS. The deactivate operation requires that the EMS support the Pending state for SNCs. Deactivating an SNC implies deletion in the ME of all the non-shared cross-connects that belong to this SNC. The PTPs are left in the same service state and are not put out-of-service. Some examples of why an NMS would use this use case include: • To free-up resources in the underlying managed subnetwork yet maintain a record of the SNC such that it could be quickly reactivated. For example, this is useful if 2 or more SNC share resources at different times of the day. • To maintain a record of the SNC in the EMS such that the EMS maintains knowledge of the network resources (e.g., aEnd CTP(s), zEnd CTP(s), route, etc.) which would be allocated for the SNC. If the addressed SNC has more routes, this operation locks all the routes, delegating the EMS and/or the network (e.g. restoration process) the actual deactivation of all XCs which are not shared with (routes of) other SNCs.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS must have knowledge of the identification of the SNC (e.g., the identifying name of the SNC).
Begins when	The NMS sends a request to the EMS to deactivate an SNC.

Use Case 7.7.7: NMS deactivates a Subnetwork Connection (SNC)

<table>
<tr><td>Description</td><td>
1) The EMS validates the specified SNC.
2) The EMS initiates the deactivation of the SNC. SNC deactivation involves EMS communication with Managed Elements. The EMS attempts to remove, from the applicable Managed Elements, all the non-shared cross-connections which comprise this SNC.
• An already deactivated SNC can be deactivated again with success (the EMS is not required to send the commands to the ME a second time, however).
• While in SNCS_PARTIAL state, it is possible to deactivate an SNC again, this corresponds to a retry.
3) If the EMS succeeds in deactivating the SNC (i.e., if all the non-shared cross-connections comprising the SNC on applicable Managed Elements have been removed), the state of the SNC changes to PENDING (or remains in PENDING if already in the PENDING state). The supporting CTPs are left in the same state and are not put out of service. The deactivation is successful even if some XCs representing fixed connectivity cannot be deleted. An SNC cannot be deactivated if it is composed solely of fixed cross-connects.
• The EMS provides a success indication to the NMS.
4) If the EMS fails to deactivate the SNC, a failure indication is sent to the NMS, and:
• If at least one, but not all of the non-shared cross-connections comprising the SNC on applicable Managed Elements have been removed, the SNC's state changes to PARTIAL (or remains in PARTIAL if already in the PARTIAL state);
• If none of the non-shared cross-connections comprising the SNC on applicable Managed Elements have been removed, the SNC's state remains ACTIVE.
5) If there has been an SNC state change, the EMS generates a state change notification when the SNC's state has been changed and sends it to the notification service.
6) For any cross-connection that has been successfully removed as a result of the deactivate SNC action, the EMS generates a state change notification for the associated connection termination points (CTPs), that have transitioned to Not Connected. Note that the CTP may still be Connected in the opposite direction as part of another SNC.
7) The NMS can request that the EMS set TP transmission parameters as a side-effect of the deactivate SNC action.
• The EMS should send AVC notifications for the successfully modified TPs.
• If a given entry in the list of transmission parameters specified by the NMS can not be successfully applied to the TP, for any reason, then an error reason is returned to the NMS.
• Existing TP transmission parameters for which no changes were requested will be left unchanged.
</td></tr>
</table>

Use Case 7.7.7: NMS deactivates a Subnetwork Connection (SNC)

	• The alarm reporting on the CTPs and the containing TPs may be turned off by the EMS as part of this request, unless otherwise specified by the NMS 8) If a Server SNC was involved in this SNC, the Server SNC does not support any other Client SNCs, and all cross-connections comprising the referenced SNC have been removed, then the EMS behavior depends on whether the EMS has freedom to deactivate the Server SNC. If the EMS does not have this level of freedom, only the Client SNC is deactivated.
Ends when	In case of success: The NMS receives an indication of success of the request. In case of failure: 1) The NMS receives an indication of failure of the request, or 2) The request times out.
Post-conditions	In case of success: 1) The network resources (CTPs) which had been used in the SNC are freed in the managed subnetwork (really, are no longer in active use by the SNC). 2) The EMS maintains the SNC object. 3) The SNC state is the PENDING state. 4) If a change of SNC state has occurred, the EMS has sent a state change notification to the Notification Service. 5) For any change of TP connection state, the EMS has sent a state change notification to the notification service. 6) For the successfully modified TPs, the EMS should send AVC notifications to the notification service In case of failure: 1) Some or all of the network resources (CTPs) associated with the SNC are still in active use by the SNC, as indicated in the TP Connection State. 2) The EMS maintains the SNC object. 3) The SNC state is either the ACTIVE (all network resources associated with the SNC are still in use by the SNC) or PARTIAL state (some network resources associated with the SNC are still in use by the SNC). 4) If a change of SNC state has occurred, the EMS sends a state change notification to the notification service. 5) For any change of TP connection state, the EMS has sent a state change notification to the notification service.

Use Case 7.7.7: NMS deactivates a Subnetwork Connection (SNC)

Exceptions	1) Service not implemented by the EMS 2) Non-specific internal EMS failure 3) The sncName provided by the NMS does not refer to an SNC object, or when any field in tpsToModify is invalid 4) The sncName or tpsToModify (provided by the NMS) references an object that does not exist 5) The SNC is fixed and can not be deactivated 6) Communications failure (or communications nonexistent) between the EMS and one or more of the underlying (applicable) Managed Elements. This exception is only used in the case where no change has been made to the SNC object or to any CTP. Whenever an exception is raised, it can be assumed that no network changes have been made to the SNC.
Traceability	{Requirement II. 090}, {Requirement II. 091}

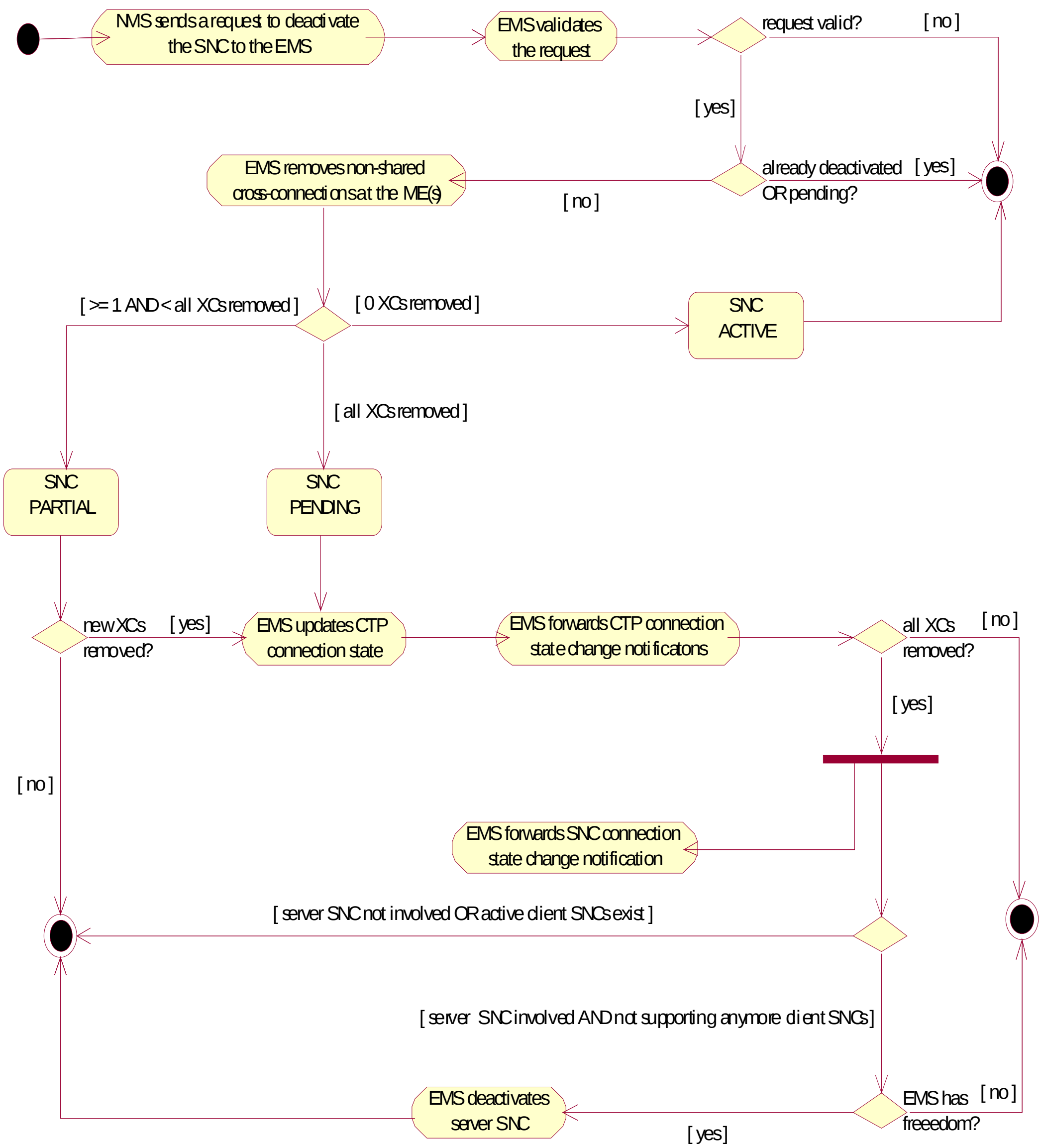


Figure 7.7: Activity Diagram for NMS deactivates a Subnetwork Connection (SNC)

7.7.8 NMS deletes a Subnetwork Connection (SNC)

Use Case 7.7.8: NMS deletes a Subnetwork Connection (SNC)

Use Case Name	NMS deletes a Subnetwork Connection (SNC)
Summary	The NMS requests that an SNC be deleted in the EMS (i.e., SNC object maintained by the EMS be deleted as a result of this action). This use case also includes the deletion a leg from a broadcast system. The delete operation requires that the EMS supports the Pending state for SNCs. Some examples of why an NMS would use this use case include: • To delete a record of the SNC from the EMS to free-up EMS resources. • To delete a record of the SNC from the EMS such that the EMS no longer has any network resources (e.g., aEnd CTP(s), zEnd CTP(s), route, etc.) allocated for the SNC. If the SNC has more routes, then the operation deletes the SNC, its intended and all back-up route(s).
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS must have knowledge of the identification of the SNC (e.g., the identifying name of the SNC). 3) The SNC must be in Pending state.
Begins when	The NMS sends a request to the EMS to delete an SNC.
Description	1) The EMS validates the SNC provided by the NMS as part of the request. 2) The EMS will only delete an SNC in the PENDING state. 3) The EMS initiates the deletion of the SNC in the EMS. SNC deletion is not expected to involve EMS communication with Managed Elements. • SNC deletion involves the EMS deleting the SNC object that the EMS maintains. If successful, the EMS provides a success indication to the NMS. • If the SNC is part of a broadcast system, the EMS only deletes the specified leg of the broadcast system and the other legs are left unchanged. 4) If the EMS fails to delete the SNC, a failure indication is sent to the NMS. The SNC state remains in PENDING. 5) The EMS generates an object deletion notification when the SNC object is deleted and sends it to the notification service. 6) If an Server SNC was involved in this SNC, and the Server SNC does not support any other Client SNCs, then the EMS behavior depends on whether the EMS has freedom to delete the Server SNC. If the EMS does not have this level of freedom, only the Client SNC is deleted.

Use Case 7.7.8: NMS deletes a Subnetwork Connection (SNC)

Ends when	In case of success: The NMS receives an indication of success of the request. In case of failure: 1) The NMS receives an indication of failure of the request, or 2) The request times out, or 3) The NMS receives an indication that the request was rejected (e.g., if the SNC that was requested to be deleted was in other than the PENDING state).
Post-conditions	In case of success: 1) The SNC object within the EMS has been deleted. 2) The EMS has sent an object deletion notification to the notification service. In case of failure: The SNC object still exists within the EMS.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed.Non-specific internal EMS failure. 3) Invalid input: Any input parameter is syntactically incorrect. 4) Entity not found: The sncName or tpsToModify (provided by the NMS) references an object that does not exist. 5) Unable to comply: The SNC is fixed and can not be deactivated 6) NE communication loss. Note: Whenever an exception is raised, it can be assumed that no network or EMS database changes have been made to the SNC.
Traceability	{Requirement II. 092}, {Requirement II. 093}

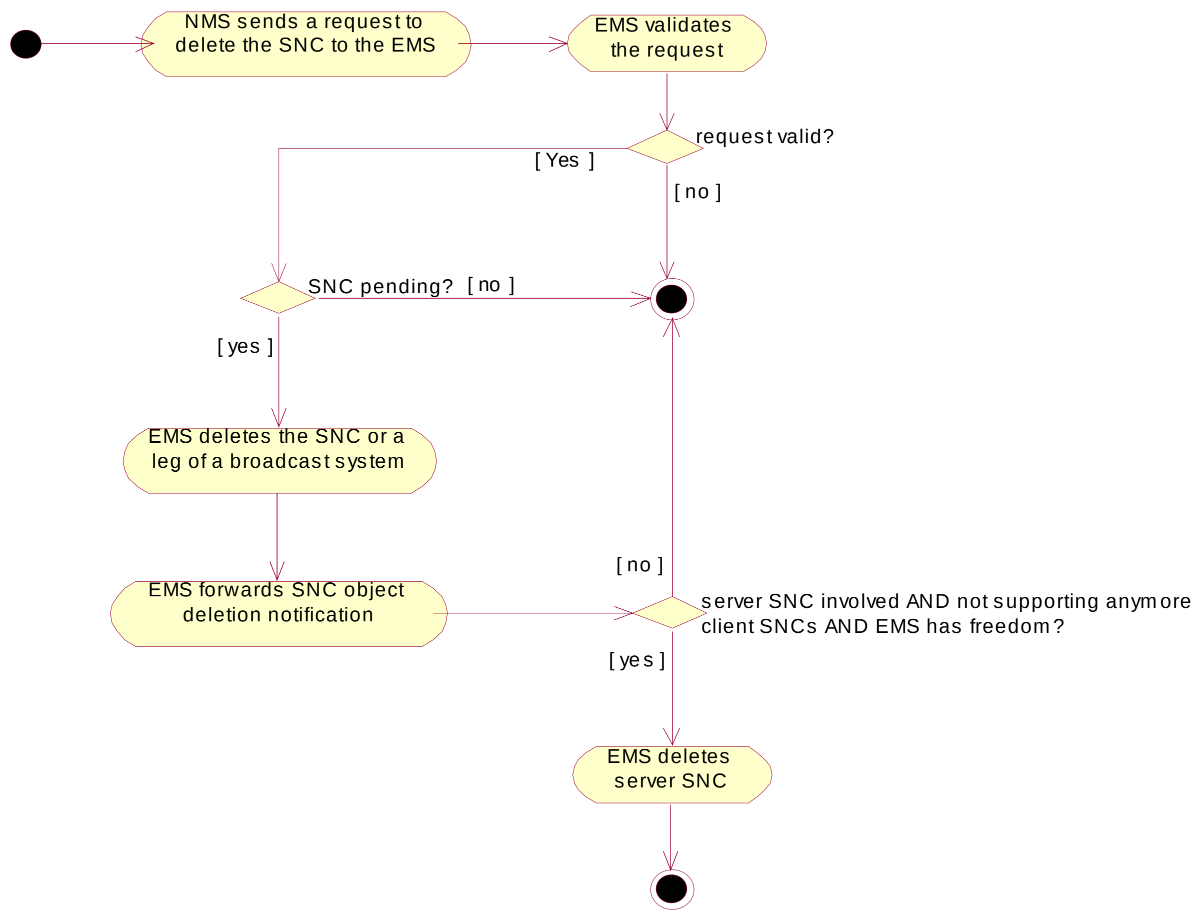


Figure 7.8: Activity Diagram for NMS deletes a Subnetwork Connection (SNC)

7.7.9 NMS deactivates and deletes a Subnetwork Connection (SNC)

Use Case 7.7.9: NMS deactivates and deletes a Subnetwork Connection (SNC)

Use Case Name	NMS deactivates and deletes a Subnetwork Connection (SNC)
Summary	The NMS requests that an SNC be deactivated and deleted. As a result of successfully completing this use case: • The SNC will be de-provisioned from the EMS' managed subnetwork, and • The SNC object will be removed from the EMS. If PENDING SNC(s) are supported by the EMS, this use case first uses the NMS Deactivates a Subnetwork Connection use case. If that use case is successfully completed, then the NMS Deletes a Subnetwork Connection use case is employed. This use case assumes that the deletion of the SNC **will never fail if pending SNC(s) are not supported by the EMS**. It is up to the EMS to enforce this assumption. Some examples of why an NMS would use this use case include: • To accomplish via a single request: the freeing-up of resources in the underlying managed subnetwork • The deletion of the record of the SNC from the EMS to free-up EMS resources • The removal of any EMS knowledge of the network resources allocated for the SNC. If the addressed SNC has more routes, this operation locks all the routes, which means that EMS and/or the network (e.g. restoration process) have no more control over these routes. All the currently active XCs for this SNC shall be removed, of any (active or partial) route. Then the operation deletes the SNC, its intended and all back-up route(s).
Actors	NMS
Pre-conditions	The pre-conditions of Use Case 7.7.7: NMS deactivates a Subnetwork Connection (SNC).
Begins when	The NMS sends a request to the EMS to deactivate and delete an SNC.

Use Case 7.7.9: NMS deactivates and deletes a Subnetwork Connection (SNC)

Description	Case A: Pending SNC(s) not supported 1) The EMS validates the specified SNC. If the request is invalid, an exception is raised. 2) The EMS initiates the deactivation of the SNC. SNC deactivation involves EMS communication with managed elements. The EMS attempts to remove, from the applicable managed elements, all the non-shared cross-connections which comprise this SNC. 3) If the EMS succeeds in deactivating the SNC (i.e., if all the cross-connections comprising the SNC on applicable managed elements have been removed), the EMS initiates the deletion of the SNC in the EMS. SNC deletion is not expected to involve EMS communication with managed elements. SNC deletion involves the EMS deleting the SNC object that the EMS maintains. The EMS provides a success indication to the NMS. The EMS generates an object deletion notification when the SNC object is deleted and sends it to the notification service. The deactivation is successful even if some XCs representing fixed connectivity cannot be deleted. An SNC cannot be deactivated if it is composed solely of fixed cross-connects. 4) If the EMS fails to deactivate the SNC, a failure indication is sent to the NMS, and: 5) If at least one, but not all cross-connections comprising the SNC on applicable managed elements have been removed, the SNC's state changes to PARTIAL (or remains in PARTIAL if already in the PARTIAL state); 6) If none of the cross-connections comprising the SNC on applicable managed elements have been removed, the SNC's state remains ACTIVE. 7) If there has been an SNC state change, the EMS generates a state change notification when the SNC's state has been changed and sends it to the notification service. 8) For any cross-connection that has been successfully removed as a result of the deactivate SNC action, the EMS generates a state change notification for the associated connection termination points (CTPs), indicating that the TP connection state has transitioned to Not Connected. 9) If an Server SNC was involved in this SNC, the Server SNC does not support any other Client SNCs and all cross-connections comprising the SNC have been removed, the EMS behavior depends on whether the EMS has freedom to deactivate and delete the Server SNC. If the EMS does not have this level of freedom, only the Client SNC is deactivated and deleted. Case B: Pending SNC(s) supported This use case first uses the "NMS Deactivates a Subnetwork Connection" use case.
	If the "NMS Deactivates a Subnetwork Connection" use case is successfully completed, this use case then uses the "NMS Deletes a Subnetwork Connection" use case.

Use Case 7.7.9: NMS deactivates and deletes a Subnetwork Connection (SNC)

Ends when	In case of success: The NMS receives an indication of success of the request. (Note that only a single success indication should be provided, not an indication for the successful deactivation of the SNC followed by another indication for the successful deletion of the SNC). In case of failure: 1) The NMS receives an indication of failure of the SNC deactivation action, or 2) The NMS receives an indication of failure of the SNC deletion action, or 3) The request times out.
Post-conditions	In case of success: This use case uses the Post Conditions of the Delete SNC use cases. In case of failure: 1) The SNC state is either ACTIVE (all network resources associated with the SNC are still in use by the SNC) or PARTIAL (some network resources associated with the SNC are still in use by the SNC). • Some or all of the network resources (CTPs) associated with the SNC are still in active use by the SNC, as indicated in the TP Connection State. • The EMS maintains the SNC object. 2) The network resources (CTPs) which had been used in the SNC are freed in the managed subnetwork and the SNC is in the PENDING state. However, (for whatever reason) the EMS maintains the SNC object. 3) The request times out. 4) If a change of SNC state has occurred, the EMS sends a state change notification to the notification service. For any change of TP connection state, the EMS has sent a state change notification to the notification service.
Exceptions	1) Processing failure: The requested operation could not be performed.Non-specific EMS internal failure. 2) Invalid input: Any input parameter is syntactical incorrect (e.g the sncName provided by the NMS does not refer to an SNC object, or when any field in tpsToModify is invalid). 3) Entity not found: The sncName or tpsToModify (provided by the NMS) reference an object that does not exist. 4) Unable to comply: The SNC is fixed and can not be deactivated. 5) NE communications loss. Note: Whenever an exception is raised, it can be assumed that no network changes have been made to the SNC

Use Case 7.7.9: NMS deactivates and deletes a Subnetwork Connection (SNC)

Traceability	{Requirement II. 094}, {Requirement II. 095}.

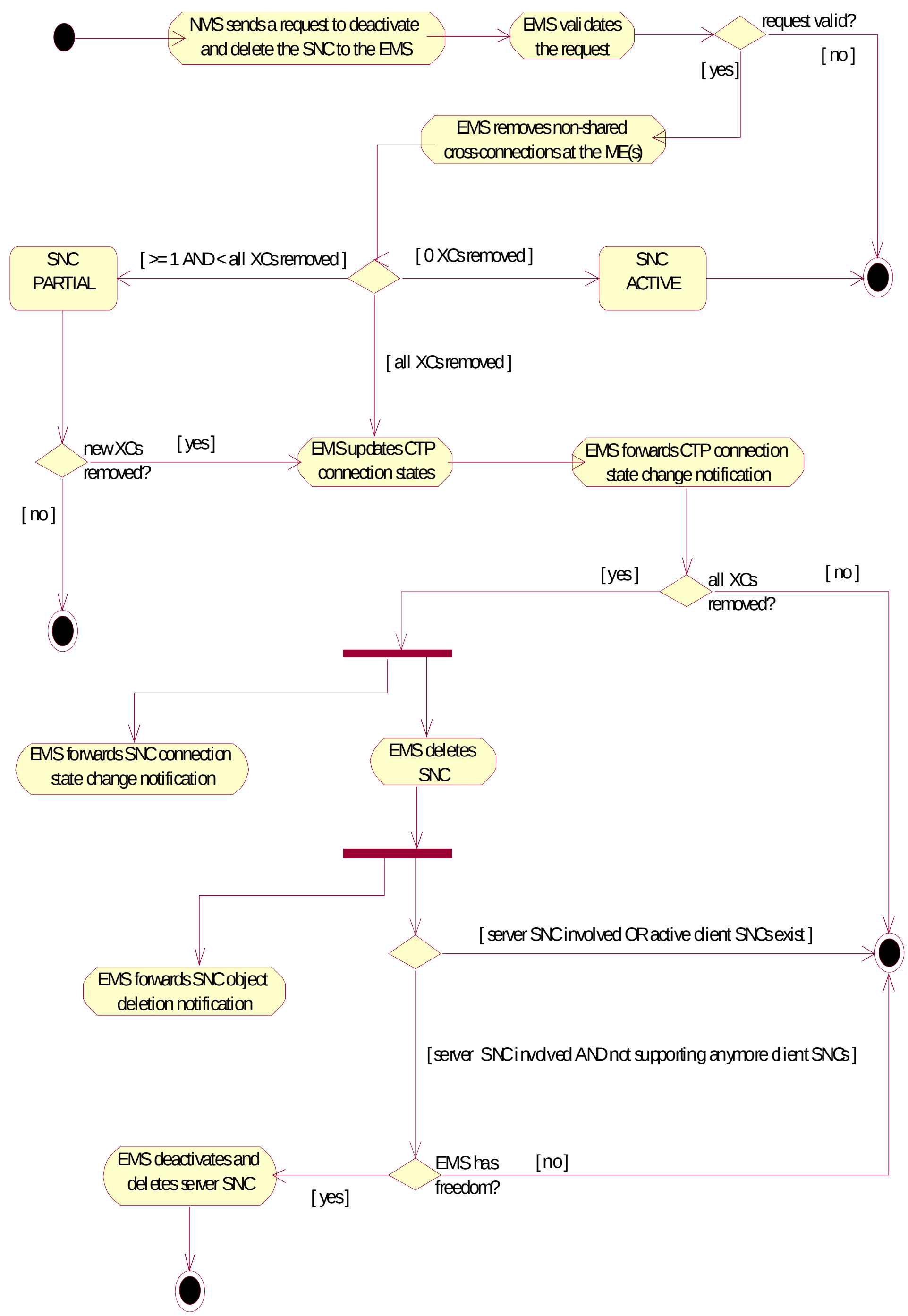


Figure 7.9: Activity Diagram for NMS deactivates and deletes a Subnetwork Connection (SNC)

7.7.10 EMS reroutes a Subnetwork Connection (SNC)

Use Case 7.7.10: EMS reroutes a Subnetwork Connection (SNC)

Use Case Name	EMS reroutes a Subnetwork Connection (SNC)
Summary	The EMS has the ability to reroute an SNC. There are a variety of reasons why this may happen. The NMS should know if it is possible for the EMS to reroute the SNC. The NMS will be notified when reroutes occur. The following are examples of when a reroute would occur in an EMS domain: a) Failure in the SNC or reversion (this is the main use) b) A lower cost route is available.
Actor(s)	NMS
Pre-Conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS has registered with the notification service. 3) The SNC is in an ACTIVE state before the route change is initiated. The SNC can also be in a PARTIAL state if the route change failed or in PENDING state if a route is available at that time and the route change is re-initiated.
Begins When	The NMS creates an SNC

Use Case 7.7.10: EMS reroutes a Subnetwork Connection (SNC)

Description	1) The NMS can specify at SNC create time if it wishes the EMS to perform automatic rerouting of SNCs. 2) The NMS can specify one of the following reroute behaviors (don't care", "yes" or "no") • If the NMS specifies don't care", the EMS can choose the behavior. • If the NMS specifies "yes" and the EMS does not support this feature, the EMS will raise an unable to comply exception with the error reason indicating re-route not supported. • If the NMS specifies "no" and the EMS only supports the specified connection using reroute, the EMS will raise an unable to comply exception with the error reason indicating only re-route supported. • The actual value for the SNC ("yes" or "no") instance is contained in the reply. 3) A condition occurs that results in the reroute of the SNC: • A route change notification is sent indicating that a re-route has started; the notification contains the original route that is no longer available. • The SNC is re-routed successfully a route change notification is sent indicating that the SNC has been re-routed successfully; the notification contains the new route. • The SNC cannot be re-routed a route change notification is sent indicating that the SNC re-routing has failed; the notification contains the original route. 4) If the EMS attempts to reroute an SNC it will send appropriate 'notifications' to the notification service. The EMS may send SNC state changes. The SNC state may not be impacted if the EMS can create a new connection in the network before deleting old.
Ends When	The EMS has completed it re-routing and the SNC has either been successfully re-routed or has failed.
Post-Conditions	The SNC will have particular reroute behavior as defined in the EMS reply.
Exceptions	1) Communication Failure with NE. 2) The EMS does not support the requested feature.
Traceability	{Requirement II. 084}

Note:

This is an optional feature which the EMS may not implement. The support of this feature by an EMS can be discovered by the NMS as a capability of the EMS.

This feature is of particular relevance to a network routed SNC but may also be applicable to NMS/EMS routed SNCs.

7.7.11 NMS queries EMS Connection Management Mode

Use Case 7.7.11: NMS queries EMS Connection Management Mode

Use Case Name	NMS queries EMS Connection Management Mode
Summary	The NMS requests the connection management mode (capabilities) from the EMS.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends the request to retrieve the connection management mode (capabilities) from the EMS.
Description	1) The NMS sends the request to retrieve the capabilities (which includes connection management mode) from the EMS. 2) The EMS replies with its supported capabilities (including the connection management mode indicating whether CC sharing and Pending SNCs are supported).
Ends when	In case of success: The NMS receives the EMS capabilities. In case of failure: None
Post-conditions	In case of success: The EMS has sent its capabilities to the NMS. In case of failure: None.
Exceptions	Processing failure: The requested operation could not be performed
Traceability	{Requirement II. 100}

7.7.12 NMS creates and activates a point-to-point Ethernet Service using fragmentation

The following use case is an example of provisioning an Ethernet Service using fragmentation, it should be noted that this is just one example of service provisioning using fragmentation and this use case can apply to other services using fragmentation such as digital video broadcast.

Use Case 7.7.12: NMS creates and activates a point-to-point Ethernet Service using fragmentation

Use Case Name	NMS creates and activates a point-to-point Ethernet Service using fragmentation

Use Case 7.7.12: NMS creates and activates a point-to-point Ethernet Service using fragmentation

Summary	The creation and activation of a point-to-point Ethernet service with fragmentation, such as SONET/SDH Virtual Concatenation or Inverse Multiplexing for ATM, involves the creation of multiple SNCs between two fragmentation capable A-end and Z-end termination points.The NMS must first discover the Ethernet ports and the capabilities of the termination points where fragmentation and reassembly occurs at the A-end and Z-end MEs. Then the NMS creates and activates as many SNCs as necessary to meet the requirements of the service order that is being implemented. This Use Case is an application of the following use case: • Use Case 7.7.3: NMS creates and activates a Subnetwork Connection (SNC)
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS asks the EMS(s) for the capabilities of the two Ethernet ports to be connected.
Description	1) The NMS sends requests to discover the capabilities of the two Ethernet ports to be connected (i.e., existing and potential CTPs with LR_Fragment layer rate). This provides the fragmentation options available for these Ethernet ports, especially the number of available fragments and their rate (VT1.5, STS1, STS3c, etc.). 2) The NMS verifies if it can meet the requirements of the service order using the available fragments (maximum bandwidth, etc.). 3) If the service order can be fulfilled and several fragmentation options are available (e.g., using STS1 or STS3c fragment from an OC12) then the NMS selects the scenario that best meet the requirement. The outcome is a series of SNCs that needs to be implemented for transporting the Ethernet service. 4) The NMS sends requests to create and activate each SNC needed to support the Ethernet service (to refer Use Case 7.7.3: NMS creates and activates a Subnetwork Connection (SNC). 5) The Ethernet service is provisioned successfully if the NMS receives a success indication from the EMS for each SNC creation and activation. 6) If one or several SNC could not be created and activated then the NMS may choose to leave the Ethernet service in a partial state or to delete the SNCs that were successfully created (i.e., rollback).
Ends when	In case of success: The NMS receives the indication that all SNCs are created and activated. In case of failure: The NMS receives an indication of creation or activation failure for at least one SNC

Use Case 7.7.12: NMS creates and activates a point-to-point Ethernet Service using fragmentation

Post-conditions	In case of success: 1) All SNCs required to fulfil the Ethernet service order have been activated. 2) The EMS has forwarded SNC creation notifications to all subscribing NMSs. In case of failure: Some or all SNCs have not been created.
Exceptions	1) Non-specific EMS internal failure. 2) Service not implemented by the EMS.
Traceability	

7.7.13 NMS modifies a point-to-point Ethernet Service with fragmentation

The following use case is an example of modifying an Ethernet Service using fragmentation, it should be noted that this is just one example of modifying a service using fragmentation and this use case can apply to other services using fragmentation such as digital video broadcast.

Use Case 7.7.13: NMS modifies a point-to-point Ethernet Service with fragmentation

Use Case Name	NMS modifies a point-to-point Ethernet Service with fragmentation
Summary	The modification described in this use case consists in changing the transport capacity by adding or deleting Subnetwork Connections to either increase or decrease the total bandwidth available to the point-to-point Ethernet service. This use case requires dynamic allocation (i.e.g., LCAS or LASR) to be supported on the A-end and Z-end MEs or the NMS would have to delete and re-create the bearer SNCs (service impacting operation). This Use Case is an application of the following existing use cases: • Use Case 7.7.3: NMS creates and activates a Subnetwork Connection (SNC) • Use Case 7.7.7: NMS deactivates a Subnetwork Connection (SNC) • Use Case 7.7.8: NMS deletes a Subnetwork Connection (SNC) Note that many other modification scenarios are possible including those involving changes to individual Subnetwork Connections (re-routing, protection level, etc.).
Actors	NMS
Pre-conditions	The NMS has executed: • Use Case 7.2.2: NMS creates a session with EMS • Use Case 7.7.12: NMS creates and activates a point-to-point Ethernet Service using fragmentation.
Begins when	The NMS has determined which SNCs need to be created or deleted

Use Case 7.7.13: NMS modifies a point-to-point Ethernet Service with fragmentation

Description	If the modification consists in increasing the transport capacity: 1) The NMS sends requests to discover the remaining capabilities of the two Ethernet ports. This provides for the number of fragments that are still available. 2) The NMS verifies if it can meet the capacity requirement of the new service order using the available fragments. 3) If the new service order can be fulfilled with the remaining fragments then the NMS sends requests to create and activate additional SNCs (to refer Use Case 7.7.3: NMS creates and activates a Subnetwork Connection (SNC)). 4) The Ethernet service is re-provisioned successfully if the NMS receives a success indication from the EMS for each SNC creation and activation. 5) If one or several SNC could not be created and activated then the NMS may choose to leave the Ethernet service in a partial state or to delete the SNCs that were successfully created (i.e., rollback). If the modification consists in decreasing the transport capacity: 1) Depending on the existing fragmentation and rate the NMS determines which SNCs need to be deleted to fulfil the new service order. 2) The NMS sends requests to deactivate the SNCs that are no longer needed (to refer Use Case 5.6.4 NMS Deactivates a Subnetwork Connection). 3) The NMS sends requests to delete the SNCs that are no longer needed (to refer Use Case 7.7.8: NMS deletes a Subnetwork Connection (SNC)). 4) The Ethernet service is re-provisioned successfully if the NMS receives a success indication from the EMS for each SNC deletion. 5) If one or several SNC could not be deleted (e.g., because they carry traffic and can't be put out of service) then the NMS needs to try again at a later time (Ethernet service is only partially modified).
Ends when	In case of success: The NMS receives the indication that all new SNCs are created and activated (capacity increase case) or it receives indication that all unneeded SNCs are deleted (capacity decrease case). In case of failure: The NMS receives an indication of creation or activation failure for at least one new SNC (capacity increase case) or it receives indication of deletion failure for at least one unneeded SNCs (capacity decrease case).

Use Case 7.7.13: NMS modifies a point-to-point Ethernet Service with fragmentation

Post-conditions	In case of success: 1) All new SNCs required to fulfil the Ethernet service order have been activated (capacity increase case) or all SNCs no longer required have been deleted (capacity decrease case). 2) The EMS has forwarded SNC creation notifications (capacity increase case) or deletion notifications (capacity decrease case) to all subscribing NMSs.. In case of failure: Some or all SNCs have not been created (capacity increase case) or deleted (capacity decrease case).
Exceptions	1) Non-specific EMS internal failure. 2) Service not implemented by the EMS.
Traceability	

7.7.14 NMS deletes a point-to-point Ethernet Service with fragmentation

The following use case is an example of deleting an Ethernet Service with fragmentation, it should be noted that this is just one example of deleting a service using fragmentation and this use case can apply to other services using fragmentation such as digital video broadcast.

Use Case 7.7.14: NMS deletes a point-to-point Ethernet Service with fragmentation

Use Case Name	NMS deletes a point-to-point Ethernet Service with fragmentation
Summary	The removal of a point-to-point Ethernet service with fragmentation, such as SONET/SDH Virtual Concatenation or Inverse Multiplexing for ATM, involves the deletion of all SNCs provisioned between the two fragmentation capable A-end and Z-end termination points.This Use Case is essentially an application of the following existing Use Cases: • Use Case 7.7.7: NMS deactivates a Subnetwork Connection (SNC) • Use Case 7.7.8: NMS deletes a Subnetwork Connection (SNC)
Actors	NMS
Pre-conditions	The NMS has executed: • Use Case 7.2.2: NMS creates a session with EMS • Use Case 7.7.12: NMS creates and activates a point-to-point Ethernet Service using fragmentation
Begins when	The NMS has determined which SNCs need to be deleted

Use Case 7.7.14: NMS deletes a point-to-point Ethernet Service with fragmentation

Description	1) The NMS sends requests to deactivate all SNCs involved in the Ethernet service (refer to Use Case 7.7.7: NMS deactivates a Subnetwork Connection (SNC)). 2) The NMS sends requests to delete all SNCs involved in the Ethernet service (refer to Use Case 7.7.8: NMS deletes a Subnetwork Connection (SNC)). 3) The Ethernet service is deleted successfully if the NMS receives a success indication from the EMS for all SNC deletions. 4) If one or several SNC could not be deleted (e.g., because they carry traffic and can't be put out of service) then the NMS needs to try again at a later time (Ethernet service is only partially deleted).
Ends when	In case of success: The NMS receives the indication that all SNCs are deleted. In case of failure: The NMS receives an indication of deletion failure for at least one SNC
Post-conditions	In case of success: 1) All SNCs required to delete the Ethernet service have been deleted. 2) The EMS has forwarded SNC deletion notifications to all subscribing NMSs In case of failure: Some or all SNCs have not been deleted
Exceptions	1) Non-specific EMS internal failure. 2) Service not implemented by the EMS.
Traceability	

7.7.15 NMS creates a flexible IMA group

Use Case 7.7.15: NMS creates a flexible IMA group

Use Case Name	NMS creates a flexible IMA group
Summary	This use case assumes that an IMA group FTP is offered by the EMS (i.e., it exists due to equipment capabilities) none of whose fragments is connected to a supporting CTP, i.e. the whole server capacity is “unloaded”. The NMS then “loads” the IMA group according to its transport capacity requirements by setting up appropriate SNCs.The NMS first discovers all server CTPs of the IMA group FTP, recognizes that all of them are unconnected, and then discovers the available unconnected supporting CTPs. Then the NMS creates and activates as many SNCs as necessary to meet the requirements of the bandwidth order that is being implemented. This Use Case is an application of the following use case: • Use Case 7.7.3: NMS creates and activates a Subnetwork Connection (SNC)
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS sends a request to the EMS for the capabilities of the IMA group TP and the available transport capabilities of the ME.
Description	1) The NMS sends requests to discover the server-side capabilities of the IMA group TP, i.e. the number of available fragments and their rate (e.g., E1 or DS1 or VC-12 or SHDSL), and the available transport capabilities of the ME, i.e. the number of unconnected supporting CTPs of the appropriate rate(s). 2) The NMS verifies if it can meet the requirements of the bandwidth demand using the available fragments (maximum bandwidth, etc.) and supporting TPs. 3) If the bandwidth demand can be fulfilled and several fragmentation options are available then the NMS selects the scenario that best meets the requirement. The outcome is a series of SNCs that needs to be implemented for transporting all of the ATM services of the IMA group. 4) The NMS sends requests to create and activate each SNC needed to support the capacity requirements (refer to Use Case 7.7.3: NMS creates and activates a Subnetwork Connection (SNC)). 5) The capacity requirements are met successfully if the NMS receives a success indication from the EMS for each SNC creation and activation. 6) If one or several SNCs could not be created and activated then the NMS may choose to meet the bandwidth demand only partially or to deactivate and delete the SNCs that were successfully created (i.e., rollback the successful work).
Ends When	In case of success: The NMS receives the indication that all SNCs are created and activated. In case of failure: The NMS receives an indication of creation or activation failure for at least one SNC.

Use Case 7.7.15: NMS creates a flexible IMA group

Post-Conditions	In case of success: 1) All SNCs required to fulfill the bandwidth demand have been activated. 2) The EMS has forwarded SNC creation notifications to all subscribing NMSs. In case of failure: Some or all SNCs have not been created..
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed.
Traceability	

7.7.16 NMS deletes a flexible IMA group

Use Case 7.7.16: NMS deletes a flexible IMA group

Use Case Name	NMS deletes a flexible IMA group
Summary	The unloading of a flexible IMA group involves the deletion of all SNCs that have been provisioned between the fragments of the IMA group and the supporting CTPs.This Use Case is an application of the following use case: • Use Case 7.7.9: NMS deactivates and deletes a Subnetwork Connection (SNC)
Actor(s)	NMS
Pre-Conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS and has executed Use Case 7.7.15: NMS creates a flexible IMA group (i.e., the NMS can only unload an IMA group that it has previously loaded itself).
Begins When	The NMS has determined which SNCs need to be deactivated and deleted
Description	1) The NMS determines which SNCs need to be deactivated and deleted and, optionally, locks the IMA group to initiate immediate or deferred shutdown. 2) The NMS sends requests to deactivate and delete all SNCs that are involved with server CTPs of the considered IMA group (refer to Use Case 7.7.9: NMS deactivates and deletes a Subnetwork Connection (SNC)) thereby disconnecting the involved supporting CTPs at the other sides of the SNCs. 3) The IMA group is unloaded successfully if the NMS receives a success indication from the EMS for all SNC deletions. 4) If one or several SNCs could not be deactivated and deleted (e.g., because they carry traffic and cannot be put out-of-service) then either the NMS needs to try again at a later time (IMA group is only partially unloaded) or the NMS may set the Administrative state of the involved CTPs to “shutting down”, when supported by the EMS, to indicate its intention of SNC deletion.
Ends When	In case of success: The NMS receives the indication that all SNCs are deleted. In case of failure: The NMS receives an indication of deletion failure for at least one SNC.
Post-Conditions	In case of success: 1) All SNCs required to unload the IMA group have been deleted. 2) The EMS has forwarded SNC deletion notifications to all subscribing NMSs. In case of failure: Some or all SNCs have not been deleted, and the involved CTPs could be in Administrative state “shutting down”.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed.
Traceability	

7.7.17 NMS modifies the transport capacity or the routing targets of a flexible IMA group

Use Case 7.7.17: NMS modifies the transport capacity or the routing targets of a flexible IMA group

Use Case Name	NMS modifies the transport capacity or the routing targets of a flexible IMA group.
Summary	The modification described in this use case consists in changing the transport capacity or the routing targets of an IMA group by adding or deleting Subnetwork Connections to either increase or decrease the total bandwidth available to the IMA group from available supporting CTPs or to redirect the transport to and from other supporting CTPs.This use case requires dynamic allocation by the LASR procedure to be supported on the considered IMA group since otherwise the NMS would have to delete and recreate SNCs which would unavoidably be a service impacting operation. This Use Case is an application of the following use cases: • Use Case 7.7.3: NMS creates and activates a Subnetwork Connection (SNC) • Use Case 7.7.9: NMS deactivates and deletes a Subnetwork Connection (SNC)
Actor(s)	NMS
Pre-Conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS and has executed Use Case 7.7.15: NMS creates a flexible IMA group (i.e., the NMS can only unload an IMA group that it has previously loaded itself).
Begins When	The NMS has determined which SNCs need to be created or deleted or recreated.
Description	If the modification consists in increasing the transport capacity: 1) The NMS sends requests to discover the remaining server-side capabilities of the IMA group TP. This provides for the number of still available fragments. 2) The NMS verifies if it can meet the requirements of the bandwidth increase demand using the still available fragments (maximum bandwidth, etc.). 3) If the bandwidth increase demand can be fulfilled with the remaining fragments then the NMS sends requests to create and activate additional SNCs (refer to Use Case 7.7.3: NMS creates and activates a Subnetwork Connection (SNC)). 4) The bandwidth increase demand is met successfully if the NMS receives a success indication from the EMS for each SNC creation and activation. 5) If one or several SNCs could not be created and activated then the NMS may choose to meet the bandwidth increase demand only partially or to delete the SNCs that were successfully created (i.e., rollback the successful work). If the modification consists in decreasing the transport capacity: 1) Depending on the existing fragmentation and layer rate the NMS determines which SNCs need to be deleted to fulfill the bandwidth decrease order. 2) The NMS sends requests to deactivate and delete the SNCs that are no longer needed (refer to Use Case 7.7.9: NMS deactivates and deletes a Subnetwork Connection (SNC)). 3) The IMA group is re-provisioned successfully if the NMS receives a success indication from the EMS for each SNC deletion.

Use Case 7.7.17: NMS modifies the transport capacity or the routing targets of a flexible IMA group

	4) If one or several SNCs could not be deactivated and deleted (e.g., because they carry traffic and cannot be put out-of-service) then either the NMS needs to try again at a later time (IMA group is only partially unloaded) or the NMS may set the Administrative state of the involved CTPs to "shutting down", when supported by the EMS, to indicate its intention of SNC deletion. If the modification consists in rerouting the physical transport: 1) The NMS executes the Use Case 7.7.16: NMS deletes a flexible IMA group for the old, no longer needed or usable, load of the IMA group. 2) The NMS executes the Use Case 7.7.15: NMS creates a flexible IMA group for the new load of the IMA group. This use case includes the determination of the supporting CTPs that constitute the new load.
Ends When	In case of success: The NMS receives the indication that all new SNCs are created and activated and/or all unneeded SNCs are deleted. In case of failure: The NMS receives an indication of creation or activation or deletion failure for at least one SNC.
Post-Conditions	In case of success: 1) All SNCs required to fulfill the capacity adjustment requirement have been activated or deleted. 2) The EMS has forwarded SNC creation and deletion notifications to all subscribing NMSs. In case of failure: Some or all SNCs have not been created or deleted, and the involved CTPs could be in Administrative state "shutting down".
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed.
Traceability	

7.8 Protection Management Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.8.1:	NMS retrieves all the Protection Groups of a Managed Element	{Requirement II. 059}, {Requirement II. 174}
Use Case 7.8.2:	Protection Switch Notification for Equipment, Trail and SNC Protection	{Requirement I. 046}, {Requirement II. 115}, {Requirement I. 074}, {Requirement II. 175}
Use Case 7.8.3:	NMS retrieves the protection switch information for Equipment, Trail and SNC Protection	{Requirement II. 114}, {Requirement II. 175}
Use Case 7.8.4:	NMS registers to receive protection switch notifications	{Requirement II. 115}
Use Case 7.8.5:	NMS invokes protection switch lockout to SNCP	{Requirement II. 116}

7.8.1 NMS retrieves all the Protection Groups of a Managed Element

Use Case 7.8.1: NMS retrieves all the Protection Groups of a Managed Element

Use Case Name	NMS retrieves all the Protection Groups of a Managed Element.
Summary	The NMS attempts to learn about the existence of all protection groups that exist in a network element. This use case applies to both TP and equipment protection groups.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The Managed Element exists within the control of the EMS.
Begins when	NMS requires the information about the existence of the protection groups in a Managed Element.
Description	1) The NMS requests the list of protection groups in a Managed Element. The NMS will send the name of the Managed Element as input. Note that the NMS can request all TP protection groups or all Equipment protection groups, but not both in the same request. 2) The EMS returns the list of all the protection groups contained in the Managed Element. 3) In the case of non-Equipment Protection Groups the EMS orders the protection group TPs in the list as follows: • The ProtectedTPs are always presented ahead of the protecting TP. • The TPs in the East direction are always presented contiguously ahead of the West directions. • In case of 4-fiber rings, there are three groups presented, two span groups and one 4-fiber ring group. • This ordering and scheme is applicable to all technologies. 4) If the EMS does not know the reversion Mode or the protection Scheme state, a value of UNKNOWN is returned. 5) For BLSR and 1:N MSP, non Pre-emptible traffic shall be ALLOWED, or NOT_ALLOWED. 6) The applicable parameters of each protection group type is returned. If not known, a value of UNKNOWN is returned. 7) The ProtectionScheme State is identified to be AUTOMATIC or FORCED_OR_LOCKED_OUT to switch. This indicates whether the protection scheme is free to switch or is constrained from switching. The protection scheme is constrained from switching when it is forced or locked. 8) The wtrTime is provided in seconds. If the EMS cannot obtain that value, a value of -1 is returned.
Ends when	The EMS completes the service.
Post-conditions	The NMS is aware of the protection groups in a Network Element.

Use Case 7.8.1: NMS retrieves all the Protection Groups of a Managed Element

Exceptions	1) Processing failure: The requested operation could not be performed. 2) Invalid input: The name of the Network Element in the request does not reference a managedElement object. 3) Entity not found: The name of the Network Element references object which does not exist. 4) NE communication loss.
Traceability	{Requirement II. 059}, {Requirement II. 174}

7.8.2 Protection Switch Notification for Equipment, Trail and SNC Protection

Use Case 7.8.2: Protection Switch Notification for Equipment, Trail and SNC Protection

Use Case Name	Protection Switch Notification for Equipment, Trail and SNC Protection.
Summary	This use case describes events that occur at the network level and how the NMS learns of them.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS 2) The NMS has executed Use Case 7.8.4: NMS registers to receive protection switch notifications. In the case of equipment protection, the NMS has registered to receive equipment protection switch notifications. 3) In the case of Trail and SNC Protection the Termination Points in question are in a protection configuration. 4) In the case of Equipment, the Equipment instances in question are in a protection configuration (only M:N equipment protection has been identified, so far)
Begins when	Either a network fault has occurred or a user triggers a switch from the EMS or the Craft creating a switch in the traffic source or the NMS triggers a switch.

Use Case 7.8.2: Protection Switch Notification for Equipment, Trail and SNC Protection

Description	1) In case of Trail protection switch (including the span switch in a 4-fiber ring configurations), the traffic source has switched from the protected to protecting or vice versa. 2) In case of a ring switch, the traffic has switched from the protected channels of a span to the protecting channels of the other span. 3) In case of a SNC protection switch, the traffic being received at the reliable TP (the output of the service selector), is switched from the worker TP to the protection TP or switched back. 4) The 1+1 and 1:N Trail protection (including the span switch in a 4-fiber ring) notification is raised against the Trail protection groups. 5) In the case of M:N equipment protection, the notification is raised against the equipment protection group. 6) In case of a ring switch the notification is raised against the Ring groups. 7) In case of a SNC protection switch, the notification is raised against the reliable TP. The EMS provides the following information to the NMS in the notification: 1) The protection type shall be provided to identify whether a protection switch is an Equipment protection, Trail protection or an SNC protection. 2) The switch reason shall be provided, which shall be Restored, Signal Fail, Signal Mismatch, Signal Degrade, Automatic Switch, Manual Switch, or Not Applicable. 3) In the case of Trail or SNC protection the layer rate shall be provided to which this switch is related. 4) The group name shall be provided, which identifies the protection group for which protection switch status is being reported. The group name shall be NULL if the protection type is SNC protection. 5) The protected TP shall be provided. For a SNC, this is always the reliable TP. For a 2F MSSP ring notification, this is the TP that is/was inactive during the switch. For a 4F MSSP ring switch notification, this is the worker TP that is/was inactive during the switch. For a 1:N MSP switch notification, this is the worker TP for which the protection switch occurred. For a revertive 1+1 MSP, this is always the worker TP. For a non-revertive 1+1 MSP switch notification, this is the TP that is inactive after the switch. In the case of equipment protection, the protected equipment instance shall be provided. For a M:N group, the protected equipment instance always identifies the worker equipment instance for which the switch occurred. 6) The switchAwayFromTP shall be provided. For a 2F MSSP ring switch, this is the TP that switched. For a 4F MSSP ring span switch, this is one of the TPs in the Trail1:N groups (worker or protection). In the case of equipment protection, the switchAwayFromEquipment is provided (this identifies the equipment instance being switched away from). 7) The switchToTP shall be provided, which identifies the TP that is the active source after the switch, or currently active if no protection switch is currently active. In the case of equipment protection, the switchToEquipment is provided (this identifies the equipment instance which is being switched to).

Use Case 7.8.2: Protection Switch Notification for Equipment, Trail and SNC Protection

Ends when	The NMS is notified of the switch.
Post-conditions	Subject to filter conditions, NMS knows of the present traffic source.
Exceptions	Not applicable.
Traceability	{Requirement I. 046}, {Requirement II. 115}, {Requirement I. 074}, {Requirement II. 175}

7.8.3 NMS retrieves the protection switch information for Equipment, Trail and SNC Protection

Use Case 7.8.3: NMS retrieves the protection switch information for Equipment, Trail and SNC Protection

Use Case Name	NMS retrieves the protection switch information for Equipment, Trail and SNC Protection.
Summary	This use case describes how the NMS learns of the traffic source of the protection groups and protected SNCs.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The Termination Points in question are in a protection configuration (Trail or SNC Protection). 3) In the case of Equipment, the equipment instances in question are in a protection configuration.
Begins when	NMS wishes to discover the present traffic source of a Trail or a SNC Protection configuration, or the active equipment instance in a Equipment protection group.

Use Case 7.8.3: NMS retrieves the protection switch information for Equipment, Trail and SNC Protection

Description	The EMS provides the following information to the NMS in the response to a query regarding the current protection switch status of a protection group or a SNC: 1) The protection type shall be provided to identify whether a protection switch is a Trail protection switch or a SNC protection switch. 2) The switch reason shall be provided, which shall be Restored, Signal Fail, Signal Mismatch, Signal Degrade, Automatic Switch, Manual Switch, or Not Applicable. 3) The layer rate shall be provided, which this switch is relevant to. not applicable for equipment protection). 4) The group name shall be provided, which identifies the protection group for which protection switch status is being reported. The group name shall be NULL if the protection type indicates SNC protection. 5) TP Protection: The protected TP shall be provided. For a SNC protection, this is always the reliable TP. For a retrieval of a 2Fiber MS SP ring, each TP is protected, and two SwitchData structures are returned. For a retrieval of a 4Fiber MS SP ring, each worker TP is protected, and two SwitchData structures are returned. For a retrieval of a 1:N Trail protection, each worker TP is protected, and N SwitchData structures are returned. For a revertive 1+1 Trail protection, this is always the worker TP. For a retrieval of a non-revertive 1+1 Trail protection switch, this is the active TP. 6) Equipment Protection: For a retrieval of an M:N group, the protected equipment always identifies a worker equipment instance. In this case, N ESwitchData structures are returned as a result of retrieve ESwitchData request (one for each worker equipment instance). 7) The switchToTP shall be provided, which identifies the TP that is the active source after the switch, or currently active if no protection switch is currently active. 8) In the case of equipment protection, the protected equipment instance shall be provided. For a M:N group, the protected equipment instance always identifies the worker equipment instance for which the switch occurred.
Ends when	The NMS is presented with all the information.
Post-conditions	NMS knows of the traffic source.
Exceptions	1) Not implemented: The EMS is unable to support this service. 2) Processing failure: The requested operation could not be performed. 3) Invalid input: The input object does not reference a protection group or a reliable CTP of a SNC object. 4) Entity not found: The input object does not exist. 5) NE communication loss..
Traceability	{Requirement II. 114}, {Requirement II. 175}

7.8.4 NMS registers to receive protection switch notifications

Use Case 7.8.4: NMS registers to receive protection switch notifications

Use Case Name	NMS registers to receive protection switch notifications
Summary	The NMS registers at the notification service related to the EMS, sets the appropriate filter to receive protection switch notifications, and connects to the notification service.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS has a reference to the notification service used by the EMS.
Begins when	The NMS sends a request to register itself at the notification service related to the EMS.
Description	1) The NMS registers at the notification service related to the EMS as a consumer of notifications (if this has not been done earlier). 2) The NMS sets the filter criteria needed to receive protection switch notifications from the EMS via the notification service. 3) The NMS connects to the notification service and thus is able to receive notifications matching the filter conditions specified (if this has not been done earlier). Note: The main filtering criteria is on the notification type (i.e., protection switch). In addition, the NMS can request other filtering criteria. Any of the parameters of the filterable body of the protection switch notification (Refer to {Requirement I. 046}) can be used.
Ends when	In case of success: The NMS receives a positive acknowledgement to its connection request to the notification service. In case of failure: The EMS returns an error indication.
Post-conditions	In case of success: The specified filter(s) are set up or modified. In case of failure: The NMS receives a negative acknowledgement to a request for registration, filter building or connection or a request times out.

Use Case 7.8.4: NMS registers to receive protection switch notifications

Exceptions	1) Filter creation: a) Invalid grammar 2) Filter building: a) Invalid constraint 3) Connection phase: a) Illegal consumer type b) Consumer already connected
Traceability	{Requirement II. 115}

7.8.5 NMS invokes protection switch lockout to SNCP

Use Case 7.8.5: NMS invokes protection switch lockout to SNCP

Use Case Name	NMS invokes protection switch lockout to SNCP
Summary	NMS applies protection switch lockout to a reliable CTP of a SNC that is protected by SNCP.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) A SNC currently exists that employs SNCP. 3) The NMS has determined which CTPs participate in the SNCP switch.
Begins when	A request to apply a protection command is applied.
Description	The command is applied to the reliable CTP that is defined as being able to perform a protection switch.
Ends when	The EMS responds that the command was applied or an exception is thrown.
Post conditions	The SNCP scheme is now locked and will not switch automatically such as when a transmission error would normally cause a protection switch.
Exceptions	1) Processing failure: The requested operation could not be performed. 2) Unable to comply: The CTP is not performing a protection switch in a SNCP. 3) Not implemented: The EMS does not support this service. 4) NE communication loss.
Traceability	{Requirement II. 116}

7.9 Fault Management Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.9.1:	NMS reconciles active alarms from an EMS	{Requirement II. 111}
Use Case 7.9.2:	NMS reconciles active alarms for a specified Managed Element	{Requirement II. 110}
Use Case 7.9.3:	NMS registers to receive alarms or threshold crossing alerts from an EMS	{Requirement II. 103}
Use Case 7.9.4:	NMS registers to receive RCAIs only, raw alarms only, or both RCAIs and raw alarms from an EMS	{Requirement II. 298}
Use Case 7.9.5:	EMS determines a more appropriate root cause than one previously reported	{Requirement II. 224}
Use Case 7.9.6:	EMS Notifies NMS of Alarms or Threshold Crossing Alert (TCA)s	{Requirement II. 126}
Use Case 7.9.7:	Alarm Acknowledgement in the NMS	{Requirement II. 155} {Requirement II. 157}

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.9.8:	Alarm Unacknowledgement in the NMS	{Requirement II. 156} {Requirement II. 158}
Use Case 7.9.9:	Alarm Acknowledgement in the EMS	{Requirement II. 157}
Use Case 7.9.10:	NMS reconciles Unacknowledged Active Alarms from an EMS	{Requirement II. 154}
Use Case 7.9.11:	NMS reconciles Unacknowledged Active Alarms for a specified Managed Element	{Requirement II. 154}
Use Case 7.9.12:	EMS discards an event to be sent to the NMS	{Requirement II. 177}
Use Case 7.9.13:	EMS succeeds in forwarding an event to the NMS again	{Requirement II. 177}
Use Case 7.9.14:	EMS sends a heartbeat notification to the NMS	{Requirement II. 178}

7.9.1 NMS reconciles active alarms from an EMS

Use Case 7.9.1: NMS reconciles active alarms from an EMS

Use Case Name	NMS reconciles active alarms from an EMS
Summary	The NMS requests the current list of active EMS-specific and non-EMS-specific alarms and TCAs that are under the control of an EMS (both those raised by the NEs and those raised by the EMS itself).. Some alarms may be filtered out (excluded) by specifying their probable causes or severities.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS for the current list of active EMS-specific and non-EMS-specific alarms and TCAs.
Description	1) The NMS sends a request to the EMS for its summary of active EMS-specific and non-EMS-specific alarms and TCAs, with parameters to filter out (exclude) certain values of probable cause or severity if the NMS does not wish to receive the whole list. These filtering criteria are independent of the filtering set up by the NMS for the notification service. 2) The EMS responds to the NMS request by returning a list of active alarms and TCAs that meet the specified filtering criteria or the request times out.
Ends when	In case of success: The NMS receives the current list of active EMS-specific and non-EMS-specific alarms and TCAs. Active alarms are those not inhibited neither by alarm reporting attribute, nor by assigned ASAP. In case of failure: The NMS receives a failure indication.
Post-conditions	In case of success: The NMS receives the current list of active EMS-specific and non-EMS-specific alarms and TCAs. In case of failure: The NMS does not receive a current list of active EMS-specific and non-EMS-specific alarms and TCAs from the EMS.
Exceptions	1) NMS request times out. 2) Processing failure: The requested operation could not be performed.
Traceability	{Requirement II. 111}

7.9.2 NMS reconciles active alarms for a specified Managed Element

Use Case 7.9.2: NMS reconciles active alarms for a specified Managed Element

Use Case Name	NMS reconciles active alarms for a specified Managed Element
Summary	NMS reconciles active alarms for a specified Managed Element.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS for the current list of active alarms and TCAs for a specified managed element
Description	1) The NMS sends a request to the EMS for its list of active alarms and TCAs for a specified managed element, with parameters to filter out (exclude) certain values of probable cause or severity if the NMS does not wish to receive the whole list. These filtering criteria are independent of the filtering set up by the NMS for the notification service. 2) The EMS responds to the NMS request by returning a list of active alarms and TCAs that meet the specified filtering criteria or the request times out.
Ends when	In case of success: The NMS receives the current list of active alarms and TCAs for the specified managed element from the EMS. Active alarms are those not inhibited neither by alarm reporting attribute, nor by assigned ASAP. In case of failure: The NMS receives a failure indication.
Post-conditions	In case of success: The NMS has the current list of active alarms and TCAs for the specified managed element from the EMS. In case of failure: The NMS does not receive the current list of active alarms and TCAs for the specified managed element from the EMS.
Exceptions	1) Processing failure: The requested operation could not be performed. 2) Invalid input: One of the input parameter contains undefined values. 3) NE communication loss.
Traceability	{Requirement II. 110}

7.9.3 NMS registers to receive alarms or threshold crossing alerts from an EMS

Use Case 7.9.3: NMS registers to receive alarms or threshold crossing alerts from an EMS

Use Case Name	NMS registers to receive alarms or threshold crossing alerts from an EMS

Use Case 7.9.3: NMS registers to receive alarms or threshold crossing alerts from an EMS

Summary	The NMS registers at the notification service related to the EMS, sets the appropriate filter to receive alarm notifications and/or threshold crossing alert notifications, and connects to the notification service.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS has a reference for the notification service used by the EMS.
Begins when	The NMS sends a request to register itself at the notification service related to the EMS.
Description	1) The NMS registers at the notification service related to the EMS as a consumer of notifications (if this has not been done earlier). 2) The NMS sets the filter criteria needed to receive alarm notifications and threshold crossing alert notifications from the EMS via the notification service. 3) The NMS connects to the notification service and thus is able to receive notifications matching the filter conditions specified (if this has not been done earlier). Note: The main filtering criteria is on the notification type (i.e., alarm and/or threshold crossing alert). In addition, the NMS can request other filtering criteria. Any of the parameters of the filterable body of the alarm or threshold crossing alert notification can be used. The commonest filtering criteria for alarms are the Probable Cause and Perceived Severity parameters. See Use Case 7.9.1: NMS reconciles active alarms from an EMS and Use Case 7.9.2: NMS reconciles active alarms for a specified Managed Element.
Ends when	In case of success: The NMS receives a positive acknowledgement to its connection request to the notification service. In case of failure: The EMS returns an error indication.
Post-conditions	In case of success: The specified filter(s) are set up or modified. In case of failure: The NMS receives a negative acknowledgement to a request for registration, filter building or connection or a request times out.

Use Case 7.9.3: NMS registers to receive alarms or threshold crossing alerts from an EMS

Exceptions	1) Filter creation: • Invalid grammar 2) Filter building • Invalid constraint 3) Connection phase: • Illegal consumer type • Consumer already connected
Traceability	{Requirement II. 103}

7.9.4 NMS registers to receive RCAIs only, raw alarms only, or both RCAIs and raw alarms from an EMS

When the NMS intends to receive RCAIs and raw alarms, it does not have to set any filter criterium (since according to SD1-26_OMGServicesUsage.pdf the filter criterium is either "rcaiIndicator" = FALSE for raw alarms or "rcaiIndicator" = TRUE for RCAIs).

Use Case 7.9.4: NMS registers to receive RCAIs only, raw alarms only, or both RCAIs and raw alarms from an EMS

Use Case Name	NMS registers to receive RCAIs only, raw alarms only, or both RCAIs and raw alarms from an EMS
Summary	The NMS registers at the notification service related to the EMS, sets the appropriate filter to receive RCAIs, raw alarms or both, and connects to the notification service.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS has a reference for the notification service used by the EMS.
Begins when	The NMS sends a request to register itself at the notification service related to the EMS.

Use Case 7.9.4: NMS registers to receive RCAIs only, raw alarms only, or both RCAIs and raw alarms from an EMS

Description	1) The NMS registers at the notification service related to the EMS as a consumer of notifications (if this has not been done earlier). 2) The NMS sets the filter criteria needed to receive RCAIs, raw alarms or both from the EMS via the notification service. 3) The NMS connects to the notification service and thus is able to receive notifications matching the filter conditions specified (if this has not been done earlier). Note: In addition to the notification type, the NMS can request filtering on any of the parameters of the filterable body of the notification. For this use case, it would be necessary to filter on the RCAI field.
Ends when	In case of success: The NMS receives a positive acknowledgement to its connection request to the notification service. In case of failure: The EMS returns an error indication.
Post-conditions	In case of success: The specified filter(s) are set up or modified. In case of failure: The NMS receives a negative acknowledgement to a request for registration, filter building or connection or a request times out.
Exceptions	1) Filter creation: • Invalid grammar 2) Filter building • Invalid constraint 3) Connection phase: • Illegal consumer type • Consumer already connected
Traceability	{Requirement II. 298}

7.9.5 EMS determines a more appropriate root cause than one previously reported

Use Case 7.9.5: EMS determines a more appropriate root cause than one previously reported

Use Case Name	EMS determines a more appropriate root cause than one previously reported
Summary	The EMS has reported a root cause to the NMS, and would like to revise the reported root cause.
Actors	EMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS has a reference for the notification service used by the EMS. 3) The EMS has sent a root cause alarm to the NMS and the alarm is still active
Begins when	The EMS determines a root cause that supersedes one or more previously reported (and still uncleared) root cause alarm indications.
Description	1) The EMS clears a root cause alarm that has been previously sent to the NMS. The EMS does not clear the raw alarms associated with the root cause. 2) The EMS sends a new root cause alarm to the NMS. (The idea is that this new root cause is more accurate than the initial root cause alarm.)
Ends when	In case of success: The NMS has received the new root cause alarm indication, and has received the clear(s) for the previous root cause alarm indication(s) that are superseded by the new root cause alarm indication. In case of failure: The NMS has not received the new root cause alarm and clear(s) for the previous root cause alarm indication(s) that are superseded by the new root cause alarm indication.
Post-conditions	In case of success: The NMS understands that the initial root cause alarm(s) is cleared and the new root cause alarm is active. In case of failure: The NMS is not aware that the initial root cause alarm has been upgraded/ corrected.
Exceptions	
Traceability	{Requirement II. 224}

7.9.6 EMS notifies NMS of Alarms or Threshold Crossing Alert (TCA)s

Use Case 7.9.6: EMS Notifies NMS of Alarms or Threshold Crossing Alert (TCA)s

Use Case Name	EMS notifies NMS of Alarm or Threshold Crossing Alert (TCA)s
Summary	The EMS detects an alarm or threshold crossing alert and notifies the NMS.

Use Case 7.9.6: EMS Notifies NMS of Alarms or Threshold Crossing Alert (TCA)s

Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.9.3: NMS registers to receive alarms or threshold crossing alerts from an EMS. 2) The notification service for these notifications is available.
Begins when	The EMS detects an alarm or threshold crossing alert.
Description	1) The EMS detects an alarm or threshold crossing alert and generates a notification to inform the NMS. The EMS does not generate the alarm notification if the object emitting the alarm has the alarm reporting attribute switched off, or the Alarm Severity Assignment Profile severity is assigned to NOTALARMED for that detected alarm probable cause. 2) The NMS receives the notification from the notification service.
Ends when	In case of success: The NMS receives the notification. In case of failure: The NMS does not receive the notification.
Post-conditions	In case of success: The NMS' database remains aligned with the EMS's database. In case of failure: The NMS' database is misaligned with the EMS's database.
Exceptions	
Traceability	{Requirement II. 126}

7.9.7 Alarm Acknowledgement in the NMS

Use Case 7.9.7: Alarm Acknowledgement in the NMS

Use Case Name	Alarm Acknowledgement in the NMS
Summary	The operator acknowledges one or more active alarms in the NMS. The NMS forwards the alarm acknowledgment through an interface operation to the EMS. The NMS will be notified in case that the alarm acknowledgment operation is successful.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The alarms to be acknowledged are active, unacknowledged, and the alarm bearing objects are managed by the EMS.
Begins when	The operator acknowledges one or more alarms in the NMS
Description	1) The NMS issues an interface request to acknowledge the alarm in the EMS. 2) The EMS acknowledges the alarm in the NE (if applicable). 3) The alarm is updated as “acknowledged” in the EMS. 4) The NMS is then notified that the alarm has been acknowledged in the EMS.
Ends when	In case of success: All NMS’ receive an alarm acknowledged notification after the operation completes successfully. In case of failure: No alarms are acknowledged in the EMS and the NMS does not receive any “alarm acknowledged” notifications.
Post-conditions	In case of success: All successfully acknowledged alarms are in an acknowledged state in the EMS. In case of failure: 1) Alarms that cannot be acknowledged are indicated to the NMS. 2) The NMS receives alarm notifications with an “alarm acknowledged” indication for all alarms successfully acknowledged in the EMS.
Exceptions	1) Processing failure: The requested operation could not be performed. 2) Invalid input: The input parameter are syntactical incorrect. 3) Not implemented: The EMS does not support this service.
Traceability	{Requirement II. 155}, {Requirement II. 157}

7.9.8 Alarm Unacknowledgement in the NMS

Use Case 7.9.8: Alarm Unacknowledgement in the NMS

Use Case Name	Alarm Unacknowledgement in the NMS

Use Case 7.9.8: Alarm Unacknowledgement in the NMS

Summary	The operator unacknowledges, in the NMS, one or more active and previously acknowledged alarms. The NMS forwards the alarm unacknowledgement through an interface operation to the EMS. The NMS will be notified in case that the alarm unacknowledgement operation is successful.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The alarms to be acknowledged are active, unacknowledged, and the alarm bearing objects are managed by the EMS.
Begins when	The operator unacknowledges, in the NMS, one or more active and previously acknowledged alarms. The NMS forwards the alarm unacknowledgement through an interface operation to the EMS. The NMS will be notified in case that the alarm unacknowledgement operation is successful.
Description	1) The NMS issues an interface request to unacknowledge the alarm in the EMS. 2) The EMS unacknowledges the alarm in the NE (if applicable). 3) The alarm is updated as “unacknowledged” in the EMS. 4) The NMS is then notified that the alarm has been unacknowledged in the EMS.
Ends when	In case of success: All NMS’ receive an alarm unacknowledged notification after the operation completes successfully. In case of failure: No alarms are unacknowledged in the EMS and the NMS does not receive any “alarm unacknowledged” notifications.
Post-conditions	In case of success: All successfully acknowledged alarms are in an unacknowledged state in the EMS. In case of failure: 1) Alarms that cannot be unacknowledged are indicated to the NMS. 2) The NMS receives alarm notifications with an “alarm unacknowledged” indication for all alarms successfully unacknowledged in the EMS.
Exceptions	1) Processing failure: The requested operation could not be performed 2) Invalid input: The input parameters are syntactical incorrect. 3) Not implemented: The EMS does not support this service.
Traceability	{Requirement II. 156}, {Requirement II. 158}

7.9.9 Alarm Acknowledgement in the EMS

Use Case 7.9.9: Alarm Acknowledgement in the EMS

Use Case Name	Alarm Acknowledgement in the EMS
Summary	The operator acknowledges alarms in the EMS through GUI Cut-Through or directly on the EMS GUI according to EMS alarm lifecycle acknowledgement steps.
Actors	None
Pre-conditions	The NMS has executed Use Case 7.9.3: NMS registers to receive alarms or threshold crossing alerts from an EMS.
Begins when	The operator acknowledges one or more alarms in the EMS.
Description	1) The EMS attempts to acknowledge the alarm in the NE (if applicable). 2) The alarm is updated as "acknowledged" in the EMS and NE if applicable. 3) The NMS is then notified that the alarm has been acknowledged in the EMS
Ends when	In case of success: The NMS receives an alarm acknowledged notification after the EMS operation completes successfully. In case of failure: The alarm is not acknowledged in the EMS and the NMS does not receive an "alarm acknowledged" notification.
Post-conditions	The alarms are acknowledged in the EMS. The NMS receives an alarm notification with an "alarm acknowledged" indication for those alarms that are acknowledged.
Exceptions	1) Internal Error 2) Loss of NE communication
Traceability	{Requirement II. 157}

7.9.10 NMS reconciles Unacknowledged Active Alarms from an EMS

Use Case 7.9.10: NMS reconciles Unacknowledged Active Alarms from an EMS

Use Case Name	NMS reconciles unacknowledged active alarms from an EMS
Summary	The NMS requests the current list of active unacknowledged EMS-specific and non-EMS-specific alarms and TCAs that are under the control of an EMS (both those raised by the NEs and those raised by the EMS itself). Some alarms may be filtered out (excluded) by specifying their probable causes or severities.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.

Use Case 7.9.10: NMS reconciles Unacknowledged Active Alarms from an EMS

Begins when	The NMS sends a request to the EMS for the current list of active unacknowledged EMS-specific and non-EMS-specific alarms and TCAs.
Description	1) The NMS sends a request to the EMS for its summary of unacknowledged active EMS-specific and non-EMS-specific alarms and TCAs, with parameters to filter out (exclude) certain values of probable cause or severity if the NMS does not wish to receive the whole list. These filtering criteria are independent of the filtering set up by the NMS for the notification service. 2) The EMS responds to the NMS request by returning a list of active unacknowledged alarms and TCAs that meet the specified filtering criteria or the request times out.
Ends when	In case of success: The NMS receives the list of active unacknowledged EMS-specific and non-EMS-specific alarms and TCAs. In case of failure: The NMS receives a failure indication.
Post-conditions	In case of success: The NMS receives the list of active unacknowledged EMS-specific and non-EMS-specific alarms and TCAs. In case of failure: The NMS receives a failure indication.
Exceptions	1) Processing failure: The requested operation could not be performed.
Traceability	{Requirement II. 154}

7.9.11 NMS reconciles Unacknowledged Active Alarms for a specified Managed Element

Use Case 7.9.11: NMS reconciles Unacknowledged Active Alarms for a specified Managed Element

Use Case Name	NMS reconciles unacknowledged active alarms for a specified Managed Element
Summary	NMS reconciles unacknowledged active alarms and TCAs for a specified Managed Element. Some alarms may be filtered out (excluded) by specifying their probable causes or severities.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS for the current list of unacknowledged active alarms and TCAs for a specified managed element.
Description	1) The NMS sends a request to the EMS for its list of active alarms and TCAs for a specified managed element, with parameters to filter out (exclude) certain values of probable cause or severity if the NMS does not wish to receive the whole list. These filtering criteria are independent of the filtering set up by the NMS for the notification service. 2) The EMS responds to the NMS request by returning a list of unacknowledged active alarms and TCAs that meet the specified filtering criteria or the request times out.
Ends when	In case of success: The NMS receives the list of unacknowledged active alarms and TCAs for the specified managed element from the EMS. In case of failure: The NMS receives a failure indication.
Post-conditions	In case of success: The NMS has the current list of unacknowledged active alarms and TCAs for the specified managed element from the EMS. In case of failure: The NMS does not receive the current list of unacknowledged active alarms and TCAs for the specified managed element from the EMS.
Exceptions	1) Processing failure: The requested operation could not be performed.
Traceability	{Requirement II. 154}

7.9.12 EMS discards an event to be sent to the NMS

Use Case 7.9.12: EMS discards an event to be sent to the NMS

Use Case Name	EMS discards an event to be sent to the NMS

Use Case 7.9.12: EMS discards an event to be sent to the NMS

Summary	The EMS discards an event, e.g. because the notification service became unavailable. The NMS is informed about this and must not assume to be synchronized any longer.
Actors	EMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) EMS discards an event.
Begins when	The EMS discards an event to be sent to the NMS.
Description	1) The EMS determines the type of notification that has been discarded. 2) If the notification is about object creation (OC), object deletion (OD), attribute value change (AVC), state change (SC) or route change (RC), the EMS informs the NMS that a lifecycle event was discarded. For all other notification types the NMS is informed that an alarm was discarded. The NMS may use operations provided by the NMS to do this or when the notification service is available may send alarms with an appropriate probable cause. 3) If a 2nd event has to be discarded, the EMS informs the NMS only if it didn't inform the NMS about this type of notification already. Thus at a time, the NMS will get informed about the loss of events at most twice, once about lifecycle events and once about alarms.
Ends when	In case of success: The NMS is aware that an event has been discarded. In case of failure: The NMS is unaware that an event has been discarded.
Post-conditions	In case of success: The NMS is aware that an event has been discarded. In case of failure: The NMS is unaware that an event has been discarded.
Exceptions	None
Traceability	{Requirement II. 177}

7.9.13 EMS succeeds in forwarding an event to the NMS again

Use Case 7.9.13: EMS succeeds in forwarding an event to the NMS again

Use Case Name	EMS succeeds in forwarding an event to the NMS again
Summary	The EMS is able to resume event forwarding and informs the NMS about this. The NMS may synchronize and can rely on upcoming events again
Actors	EMS
Pre-conditions	1) The NMS has successfully executed Use Case 7.2.2: NMS creates a session with EMS. 2) The EMS has informed the NMS that an event has been discarded before.
Begins when	The EMS succeeds to forward an event to the NMS.
Description	1) The EMS informs the NMS that event forwarding is resumed. There is one operation provided by the NMS for this. If the EMS has sent alarms about the event loss before it shall clear those alarms. 2) The NMS may start synchronization after that either for alarms, TCA and protection switch states alone or for the complete configuration dependent on what it was informed about before.
Ends when	In case of success: The NMS is informed that EMS has resumed event forwarding. In case of failure: The NMS is not informed that EMS has resumed event forwarding
Post-conditions	In case of success: The NMS is informed that the EMS has resumed event forwarding and may have started synchronization. In case of failure: The NMS is not informed that the EMS has resumed event forwarding and may have started synchronization.
Exceptions	
Traceability	{Requirement II. 177}

7.9.14 EMS sends a heartbeat notification to the NMS

Use Case 7.9.14: EMS sends a heartbeat notification to the NMS

Use Case Name	EMS sends a heartbeat notification to the NMS
Summary	The EMS sends a heartbeat notification to the NMS if there are no other notifications to be forwarded for some time. The NMS thus knows that the notification service is still available even if there are no other notification for a longer time.
Actors	EMS

Use Case 7.9.14: EMS sends a heartbeat notification to the NMS

Pre-conditions	The NMS has successfully executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS has successfully executed Use Case 7.2.2: NMS creates a session with EMS.
Description	1) The EMS sends out heartbeat notifications to the NMS on a regular basis. 2) The NMS can conclude that the notification service is unavailable if it doesn't receive any heartbeat notifications for some time.
Ends when	1) The NMS has successfully executed Use Case 7.2.4: NMS closes a session with an EMS. or 2) The EMS has successfully executed Use Case 7.2.5: EMS closes a session with an NMS
Post-conditions	The NMS has successfully executed Use Case 7.2.2: NMS creates a session with EMS.
Exceptions	None.
Traceability	{Requirement II. 178}

7.10 Equipment Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.10.1:	NMS unprovisions equipment	{Requirement II. 262}
Use Case 7.10.2:	NMS provisions equipment	{Requirement II. 136}
Use Case 7.10.3:	NMS provisions alarm reporting on/off for equipment	{Requirement II. 078}
Use Case 7.10.4:	NMS provisions alarm reporting on/off for an equipment holder	{Requirement II. 078}

7.10.1 NMS unprovisions equipment

Use Case 7.10.1: NMS unprovisions equipment

Use Case Name	NMS unprovisions equipment
Summary	An operator permanently unprovisions an equipment from the managed element. The successful result of this operation is the potential deletion of the equipment object and all of its related objects, such as Termination Points.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) Equipment to be unprovisioned is not in use.
Begins when	The NMS sends a request to unprovision an equipment or the NMS detects equipment has been unprovisioned.
Description	1) The NMS sends the request to the EMS 2) The EMS validates the equipment name. If the name cannot be found an exception is thrown. 3) The EMS may check if a service is assigned to a port of the equipment (e.g. a supported PTP is in service) in which case an exception will be thrown. 4) The EMS deletes all of the equipment related objects, such as Termination Points. 5) The EMS should attempt to set the equipment state to out of service by maintenance prior to unprovisioning the equipment. 6) The EMS unprovisions the equipment. 7) Appropriate notifications shall be sent.
Ends when	The EMS sends a response.
Post-conditions	The Equipment is unprovisioned and all its related objects deleted

Use Case 7.10.1: NMS unprovisions equipment

Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed. 3) Invalid input: The equipment name does not reference an equipment object. 4) Object in use: The equipment resources are in use. 5) Entity not found: The equipment name references an object which does not exist. 6) Unable to comply: The equipment can not be unprovisioned at the managedElement. 7) NE communication loss.
Traceability	{Requirement II. 262}

7.10.2 NMS provisions equipment

Use Case 7.10.2: NMS provisions equipment

Use Case Name	NMS provisions equipment
Summary	An operator permanently provisions an equipment in an equipment holder in a ME. The result of this operation may be the creation of the equipment object and all of its related objects such as Termination Points.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) Equipment holder do not has an expected equipment.
Begins when	The NMS sends a request to provision an equipment or the NMS detects equipment has been provisioned.
Description	1) The NMS sends request to EMS to provision an equipment. 2) The EMS validates the equipment CreateData. 3) If the equipmentHolder does not exist, an exception is thrown. 4) If the equipmentHolder already has an expected equipment, an exception is thrown. 5) The EMS provisions the equipment object and all of its related objects. 6) Appropriate notifications shall be sent.
Ends when	The EMS sends a response.
Post-conditions	Equipment and all of its related objects exist.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed. 3) Object in use: The equipment holder already has an expected equipment. 4) Invalid input: The equipment holder does not reference an equipmentHolder object. 5) Entity not found: The equipmentHolder references object that does not exist. 6) Unable to comply: The equipment can not be created at the managedElement. 7) NE communication loss.
Traceability	{Requirement II. 136}

7.10.3 NMS provisions alarm reporting on/off for equipment

Use Case 7.10.3: NMS provisions alarm reporting on/off for equipment

Use Case Name	NMS provisions alarm reporting on/off for equipment
Summary	The NMS requests to activate/deactivate all alarm reporting on an equipment.
Actors	NMS

Use Case 7.10.3: NMS provisions alarm reporting on/off for equipment

Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends the request to activate/deactivate alarm reporting to the EMS.
Description	1) The NMS sends the request to activate/deactivate alarm reporting for a specified equipment. 2) The EMS validates the equipment reference (e.g. name). 3) The EMS enables/disables alarm reporting on the specified equipment. 4) The EMS replies with a success indication. 5) Attribute Value Change notification(s) for the specified equipment are forwarded to the notification service indicating that alarm reporting has been activated/ deactivated for the specified equipment.
Ends when	In case of success: The NMS receives an indication of success of the action. In case of failure: The NMS receives an indication of failure of the action.
Post-conditions	In case of success: 1) Alarm monitoring is enabled/disabled on the specified equipment. Note: If alarm monitoring is enabled, this does not mean that alarm is reported anyway, because Alarm Severity Assignment Profile may perform further filtering. If alarm monitoring is disabled, then alarm reporting is disabled even if Alarm Severity Assignment Profile would allow it. 2) The EMS has forwarded an attribute value change notification if there was an attribute value change with the enabling/disabling of alarm monitoring on the equipment. In case of failure: None.
Exceptions	1) Invalid input: The equipment reference is invalid. 2) NE communications loss. 3) Entity not found: The specified equipment object does not exist. 4) Processing failure: The requested operation could not be performed. 5) Unable to comply: The alarm reporting can not be enabled/disabled
Traceability	{Requirement II. 078}

7.10.4 NMS provisions alarm reporting on/off for an equipment holder

Use Case 7.10.4: NMS provisions alarm reporting on/off for an equipment holder

Use Case Name	NMS provisions alarm reporting on/off for an equipment holder
Summary	The NMS requests to activate/deactivate all alarm reporting on an equipment holder.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends the request to activate/deactivate alarm reporting to the EMS.
Description	1) The NMS sends the request to activate/deactivate alarm reporting on for a specified equipment holder. 2) The EMS validates the equipment holder reference (e.g. name). 3) The EMS enables/disables alarm reporting on the specified equipment holder. 4) The EMS replies with a success indication. 5) Attribute Value Change notification(s) for the specified equipment holder are forwarded to the notification service indicating that alarm reporting has been activated/deactivated for the specified equipment holder.
Ends when	In case of success: The NMS receives an indication of success of the action. In case of failure: The NMS receives an indication of failure of the action.
Post-conditions	In case of success: 1) Alarm monitoring is enabled/disabled on the specified equipment holder. Note: If alarm monitoring is enabled, this does not mean that alarm is reported anyway, because Alarm Severity Assignment Profile may perform further filtering. If alarm monitoring is disabled, then alarm reporting is disabled even if Alarm Severity Assignment Profile would allow it. 2) The EMS has forwarded an attribute value change notification if there was an attribute value change with the enabling/disabling of alarm monitoring on the equipment holder. In case of failure: None.
Exceptions	1) Invalid input: The equipment reference is invalid. 2) NE communications loss. 3) Processing failure: The requested operation could not be performed. 4) Unable to comply: The alarm reporting can not be enabled/disabled.

Use Case 7.10.4: NMS provisions alarm reporting on/off for an equipment holder

Traceability	{Requirement II. 078}

7.11 Craft Related Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.11.1:	Craft modifies TP Transmission Parameter(s) of a TP	{Requirement I. 044}, {Requirement II. 066}
Use Case 7.11.2:	Craft/EMS creates a cross-connect (XC) in a Network Element (NE)	{Requirement I. 042}, {Requirement I. 045}, {Requirement II. 064}, {Requirement II. 067}
Use Case 7.11.3:	Craft/EMS Deletes a cross-connect (XC) in a Network Element (NE)	{Requirement I. 042}, {Requirement I. 045}, {Requirement II. 065}, {Requirement II. 067}
Use Case 7.11.4:	Craft inserts a plug-in card	{Requirement I. 042}, {Requirement I. 045}, {Requirement II. 064}, {Requirement II. 067}, {Requirement II. 133}, {Requirement II. 134}
Use Case 7.11.5:	Craft removes a plug-in card	{Requirement I. 043}, {Requirement I. 045}, {Requirement II. 065}, {Requirement II. 067}
Use Case 7.11.6:	Craft/EMS creates a Protection Group	{Requirement I. 042}, {Requirement II. 064}, {Requirement II. 112}, {Requirement II. 059}

7.11.1 Craft modifies TP Transmission Parameter(s) of a TP

Use Case 7.11.1: Craft modifies TP Transmission Parameter(s) of a TP

Use Case Name	Craft modifies Transmission Parameter(s) of a TP
Summary	The craft operator modifies transmission parameters on a TP of an NE using a craft interface or an EMS interface.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) Termination Point exists on NE. 3) NMS is attached to a notification service.
Begins when	EMS detects that the TP Transmission Parameters have been modified on the NE.

Use Case 7.11.1: Craft modifies TP Transmission Parameter(s) of a TP

Description	1) The EMS will send an attribute value change notification for all attributes that have changed. 2) The EMS will not update the SNCs which use this termination point, regardless of the SNC state.
Ends when	EMS has sent notifications.
Post-conditions	1) Transmission parameters are changed on a TP in NE. 2) EMS has sent notifications.
Exceptions	No Interface exceptions as no NMS-initiated operations
Traceability	{Requirement I. 044}, {Requirement II. 066}

7.11.2 Craft/EMS creates a cross-connect (XC) in a Network Element (NE)

Use Case 7.11.2: Craft/EMS creates a cross-connect (XC) in a Network Element (NE)

Use Case Name	Craft/EMS creates a cross-connect (XC) on a Network Element (NE)
Summary	Craft person enters a new XC in an NE.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. NMS is attached to a notification service.
Begins when	EMS detects that XC has been established on NE managed by the EMS.
Description	1) The EMS sends tpConnectionState state change notifications (state= 'bi-connected' or 'sink-connected' or 'source-connected'). The notification is sent even when the EMS exposes only a mesh and the TP is not an edge TP. 2) If the XC participates in an existing partial SNC and the XC is the last XC needed to complete the SNC, the SNC state changes to 'active' state. The EMS forwards SNC state change notifications. 3) If the XC participates in an existing partial SNC and the XC is not the last XC, the SNC state will remain in partial. 4) If the XC does not match in any existing partial or active SNC, then a new SNC object is created to represent it. The EMS forwards a SNC object creation notification. The SNC state of the newly created SNC is active. 5) No SNCs in pending state are affected.
Ends when	EMS sends appropriate notifications.
Post-conditions	The SNC object may be created as described above, and appropriate notifications (as described above) are sent.
Exceptions	No Interface exceptions as no NMS-initiated operations
Traceability	{Requirement I. 042}, {Requirement I. 045}, {Requirement II. 064}, {Requirement II. 067}

7.11.3 Craft/EMS deletes a cross-connect (XC) in a Network Element (NE)

Use Case 7.11.3: Craft/EMS Deletes a cross-connect (XC) in a Network Element (NE)

Use Case Name	Craft/EMS deletes a cross-connect (XC) in a Network Element (NE)
Summary	When a craft deletes a XC which is a part of an existing SNC, the EMS determines the applicable state of the SNC and provides applicable notifications (if any) to NMSs.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. NMS is attached to a notification service. The state of SNC is either active or partial.
Begins when	EMS detects deletion of XC on NE managed by EMS.

Use Case 7.11.3: Craft/EMS Deletes a cross-connect (XC) in a Network Element (NE)

Description	1) EMS detects the deletion of the XC, and finds the affected SNC; 2) If the state of the SNC is active and the deleted XC is not the only XC associated with the SNC (and at least one of the other XCs associated with the SNC is still in place on an NE), EMS changes the state of SNC to partial. 3) If the state of the SNC is active and EMS detects that the deleted XC is the only XC associated with the SNC, the EMS: • deletes the SNC if Pending state is not supported. • changes the SNC state to partial if pending state is supported. 4) If the state of the SNC is partial and EMS detects that the deleted XC is not the only XC associated with the SNC (nor the last remaining XC associated with the SNC), there will be no change on the SNC state. 5) If the state of the SNC is partial and EMS detects that the deleted XC is the last remaining XC associated with the SNC, the EMS: • leaves the state of the SNC in the partial state if pending is supported. • deletes the SNC if pending is not supported. 6) If the XC deletion caused a state change to the SNC, a State Change Notification is sent to the notification service. 7) If the XC deletion caused the deletion of the SNC, an Object Deletion Notification is sent to the notification service. 8) The EMS sends state change notifications for the TPs that have changed state.
Ends when	EMS sends applicable notifications (if any) to the notification service.
Post-conditions	EMS determines the new SNC state, sends state change notifications for the TPs, and sends applicable SNC Object Deletion and/or State Change Notifications (if any) to the notification service.
Exceptions	No Interface exceptions as no NMS-initiated operations
Traceability	{Requirement I. 042}, {Requirement I. 045}, {Requirement II. 065}, {Requirement II. 067}

7.11.4 Craft inserts a plug-in card

Use Case 7.11.4: Craft inserts a plug-in card

Use Case Name	Craft inserts a plug-in card
Summary	The Craft inserts a plug-in card.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. NMS is attached to a notification service.
Begins when	EMS detects insertion of plug-in card.
Description	1) If new equipment and auto-provisioning is supported, the following notifications are sent to the notification service: • Equipment object creation notification • Equipment Holder State Change • If it is a equipment supporting TPs, PTP Object Creation Notifications may be sent to the notification service (Refer to the provision Equipment use case) 2) If auto-provisioning is not supported, the following notifications may be sent to the notification service: • Equipment object creation notification if the equipment has not been provisioned. • Equipment Holder State Change 3) If it is insertion of a provisioned plug-in, the following notifications may be sent to the notification service. • Equipment Holder State Change • Equipment State changes and AVCs.
Ends when	The EMS sends applicable notifications to the notification service.
Post-conditions	The NMS is aware of the potential or modification of the PTPs, the state changes of the contained CTPs and the creation of any topological links.
Exceptions	No Interface exceptions as no NMS-initiated operations
Traceability	{Requirement I. 042}, {Requirement I. 045}, {Requirement II. 064}, {Requirement II. 067}, {Requirement II. 133}, {Requirement II. 134}

7.11.5 Craft removes a plug-in card

Use Case 7.11.5: Craft removes a plug-in card

Use Case Name	Craft removes a Plug-In card

Use Case 7.11.5: Craft removes a plug-in card

Summary	The Craft removes a plug-in that has no relevance to connection management. Removal of the plug-in implies a deletion of the equipment on the network element or via the EMS (TPs not associated to SNCs)
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. NMS is attached to a notification service.
Begins when	EMS detects removal of the plug-in card.
Description	1) If an equipment is removed, and the equipment has not already been de-provisioned, the following notifications may be sent to the notification service: • Equipment State change and AVCs • Equipment Holder State Change 2) If the equipment was in a de-provisioned state the following notifications may be sent to the notification service: • Equipment object deletion • Equipment Holder State Change
Ends when	The EMS sends applicable notifications to the notification service.
Post-conditions	The NMS is aware of the potential deletion or modification of the equipment/equipment holder.
Exceptions	No Interface exceptions as no NMS-initiated operations
Traceability	{Requirement I. 043}, {Requirement I. 045}, {Requirement II. 065}, {Requirement II. 067}

7.11.6 Craft/EMS creates a Protection Group

Use Case 7.11.6: Craft/EMS creates a Protection Group

Use Case Name	Craft/EMS creates a Protection Group
Summary	The Craft creates a Protection Group on a network element via the EMS or on the network element or the EMS detects a new protection group on communication association
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. NMS is attached to the notification service
Begins when	EMS detects that a Protection Group was created on a Managed Element
Description	1) The EMS identifies the protection group type. If the protection group identified pertains to a 4 fiber MS SP ring (BLSR) protection, the EMS sends three separate Object creation notifications (one each for the span groups and one for the combined groups). In all other cases, a single group is identified to be sent to the NMS. 2) The Object creation notification identifies the steady state switch status of the protection group. 3) Edge point boolean is set for this notification if any of the TPs forming the protection group is an edge point.
Ends when	The EMS sends applicable notifications to the notification service
Post-conditions	The NMS is aware of the existence of the line level protection.
Exceptions	No Interface exceptions as no NMS-initiated operations
Traceability	{Requirement I. 042}, {Requirement II. 064}, {Requirement II. 112}, {Requirement II. 059}

7.12 Performance Management Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.12.1:	NMS activates collection of Performance Monitoring Data (PMD) for a specified set of TPs	{Requirement II. 121}, {Requirement II. 132}
Use Case 7.12.2:	NMS deactivates collection of Performance Monitoring Data (PMD) for a specified set of TPs	{Requirement II. 121}
Use Case 7.12.3:	NMS retrieves current Performance Monitoring Data (PMD) for a specified set of TPs	{Requirement II. 131}
Use Case 7.12.4:	NMS retrieves the storage time of 24hr and 15min Performance Monitoring Data (PMD) records	{Requirement II. 124}
Use Case 7.12.5:	NMS retrieves PM capabilities of a Managed Element (ME)	{Requirement II. 054}, {Requirement II. 221}, {Requirement II. 222}
Use Case 7.12.6:	NMS retrieves historical Performance Monitoring Data (PMD) for a specified set of TPs	{Requirement II. 128}, {Requirement II. 129}, {Requirement II. 130}

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.12.7:	NMS sets PM thresholds on a TP	{Requirement I. 054}, {Requirement II. 125}
Use Case 7.12.8:	NMS retrieves PM threshold settings from a TP	{Requirement I. 054}, {Requirement II. 055}
Use Case 7.12.9:	NMS enables Threshold Crossing Alerts (TCA) for a specified set of TPs	{Requirement II. 122}, {Requirement II. 126}
Use Case 7.12.10:	NMS disables Threshold Crossing Alerts (TCA) for a specified set of TPs	{Requirement II. 122}
Use Case 7.12.11:	On demand retrieval of historical Performance Monitoring Data (PMD) for a specified set of TPs	{Requirement II. 163}
Use Case 7.12.12:	NMS retrieves Performance Monitoring Points (PMP) contained in an ME or a TP	{Requirement II. 221}
Use Case 7.12.13:	NMS modifies TCA Parameter Profile	{Requirement II. 238}, {Requirement II. 239}
Use Case 7.12.14:	NMS configures TCA Parameter Profile Pointer	{Requirement II. 238}, {Requirement II. 239}

Note:

{Requirement I. 047}, {Requirement II. 050} and {Requirement II. 056} specify the information that has to be provided by an EMS in the PM related notifications. These requirements are **not covered** by the following use cases.

7.12.1 NMS activates collection of Performance Monitoring Data (PMD) for a specified set of TPs

Use Case 7.12.1: NMS activates collection of Performance Monitoring Data (PMD) for a specified set of TPs

<table>
<tr><td>Use Case Name</td><td>NMS activates collection of Performance Monitoring Data (PMD) for a specified set of TPs</td></tr>
<tr><td>Summary</td><td>The NMS requests the EMS to activate measurement of PMD for a list of TP and layer rate measurement points. For each measurement point, location category (nearEnd (unidirectional) and/or farEnd (unidirectional) and/or bidirectional) and granularity (15min register and/or, 24hr register and/or NA) can be specified.</td></tr>
<tr><td>Actors</td><td>NMS</td></tr>
<tr><td>Pre-conditions</td><td>1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
2) The NMS must have knowledge of the ME or the TPs specified.
3) The EMS must have active communications established with the MEs, to which the TPs belong, which are specified.</td></tr>
<tr><td>Begins when</td><td>The NMS sends a request to the EMS to activate measurement of PMD.</td></tr>
<tr><td>Description</td><td>1) The NMS sends an activation request command to the EMS:
The NMS specifies the measurement points to which the request shall be applied. It is possible to identify all TPs of a ME or a set of individual TPs (PTPs and/or CTPs). For each measurement point the NMS has to define:
• the layer rate
• the granularity (15min-register and/or 24hr-register and/or NA) and
• the location (nearEnd (unidirectional) and/or farEnd (unidirectional) and/or bidirectional)
An empty list for layer rate, granularity or location means that the ME has to use every supported item.
2) The EMS validates the provided parameters.
3) If necessary the EMS sends the appropriate command(s) to the ME to activate the reporting of PMD from the MEs to the EMS.
4) The NMS may request the PM registers be cleared.
5) If PMD collection is already active for one or more of the specified TPs, the operation shall be considered a success.
6) The operation is best-effort. If Performance Monitoring could not be enabled for a subset of the TPs specified, a list identifying the non-enabled subset is returned.
7) If nearEnd/farEnd/bidirectional PMD collection cannot be separately activated for an ME, the EMS should hide this from the NMS.
8) For every item activated the EMS sends a state change notification on the monitoring state of the affected Performance Monitoring Point to the notification service</td></tr>
</table>

Use Case 7.12.1: NMS activates collection of Performance Monitoring Data (PMD) for a specified set of TPs

Ends when	In case of success: The NMS receives a confirmation In case of failure: The NMS receives an exception or gets an indication of the subset of TPs that could not be enabled..
Post-conditions	In case of success: The requested functionality of the TPs specified are enabled for collection of PMD. In case of failure: When an exception is received, nothing has changed in the EMS domain otherwise, the requested functionality of the TPs specified are enabled for collection of PMD **except** the ones identified as not enabled.
Exceptions	1) Processing failure: The requested operation could not be performed. 2) Invalid input: No TP/ME is specified in the input data or it contains at least one invalid data item 3) NE communication loss 4) Capacity exceeded: The maximum number of simultaneous enabled measurement points is exceeded
Traceability	{Requirement II. 121}, {Requirement II. 132}

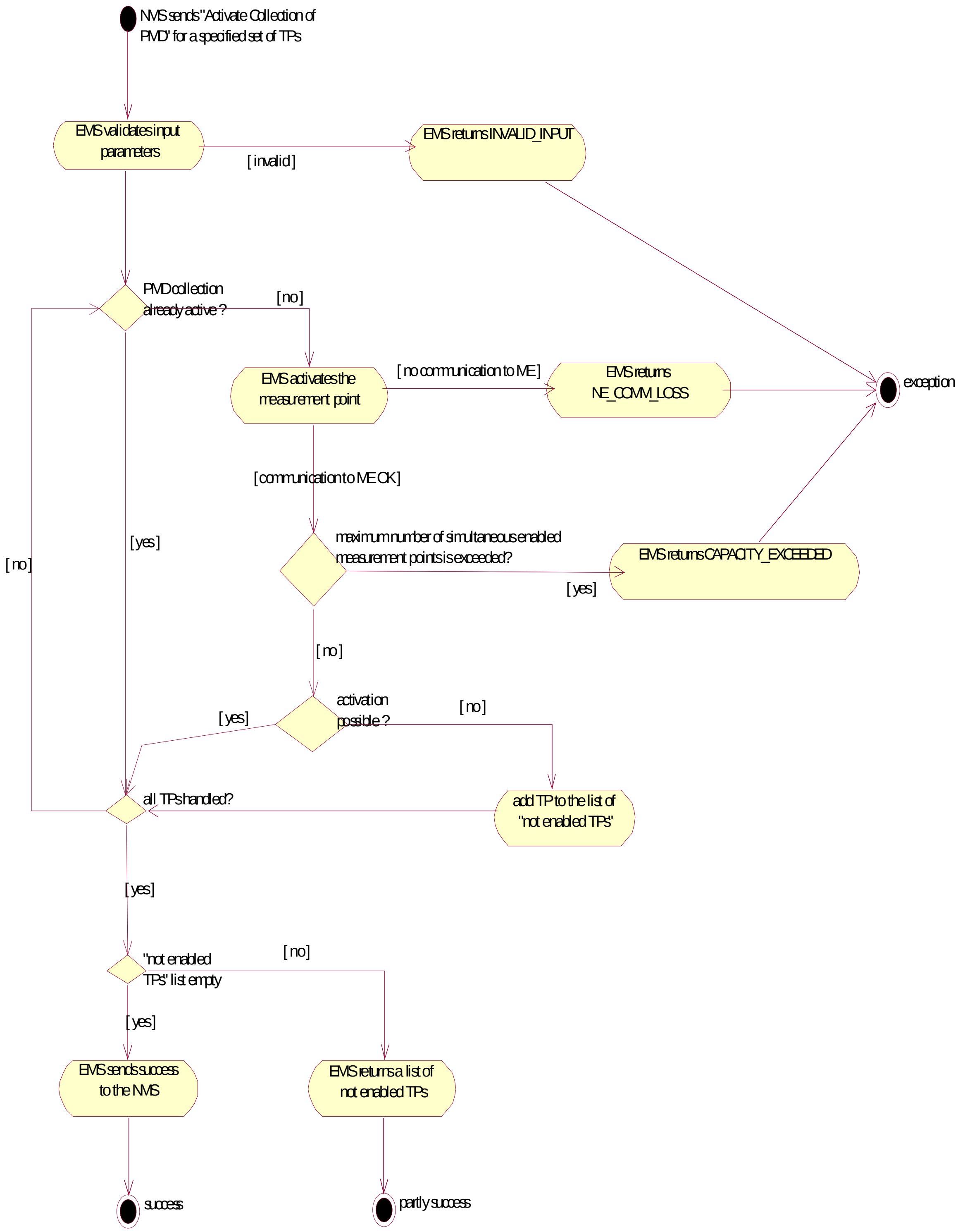


Figure 7.10: Activity Diagram for NMS activates collection of Performance Monitoring Data (PMD) for a specified set of TPs

7.12.2 NMS deactivates collection of Performance Monitoring Data (PMD) for a specified set of TPs

Use Case 7.12.2: NMS deactivates collection of Performance Monitoring Data (PMD) for a specified set of TPs

Use Case Name	NMS deactivates collection of Performance Monitoring Data (PMD) for a specified set of TPs
Summary	The NMS requests the EMS to de-activate measurement of PMD for a list of TP and layer rate measurement points. For each measurement point, location category (nearEnd (unidirectional) and/or farEnd (unidirectional) and/or bidirectional) and granularity (15min-register and/or 24hr-register and/or NA) has to be specified.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS must have knowledge of the ME or the TPs specified. 3) The EMS must have active communications established with the MEs, to which the specified TP belongs.
Begins when	The NMS sends a request to the EMS to stop measurement of PMD.
Description	1) The NMS sends a de-activation request command to the EMS. The NMS specifies the measurement points to which the request shall be applied. It is possible to identify all TPs of a ME or a set of individual TPs (PTPs and/or CTPs). For each measurement point the NMS has to define: • the layer rate • the granularity (15min-register and/or 24hr-register and/or NA) and • the location (nearEnd (unidirectional) and/or farEnd (unidirectional) and/or bidirectional) An empty list for layer rate, granularity or location means that the ME has to stop the collection of every supported item. Stop of data collection does not imply deleting data for the particular measurement point. 2) The EMS validates the provided parameters. 3) The EMS on its turn sends the appropriate command(s) to the ME to stop collecting of PMD for the specified set of TPs. 4) The operation is best-effort. If Performance Monitoring could not be disabled for a subset of the TPs specified, a list identifying the non-disabled subset is returned. 5) Following successful processing of the appropriate command(s), data collection stops immediately. This leads to incomplete collection period(s).These incomplete periods have to be marked by an appropriate status when reported. 6) If PMD collection is not active for one or more of the specified TPs, the operation shall be considered a success. 7) If nearEnd/farEnd/bidirectional PMD collection cannot be separately de-activated for an ME, the EMS should hide this from the NMS.

Use Case 7.12.2: NMS deactivates collection of Performance Monitoring Data (PMD) for a specified set of TPs

	8) For every item deactivated the EMS sends a state change notification on the monitoring state of the affected Performance Monitoring Point to the notification service.
Ends when	In case of success: The NMS receives a confirmation In case of failure: The NMS receives an exception or gets an indication of the subset of TPs that could not be disabled..
Post-conditions	In case of success The requested functionality of the TPs specified are disabled for collection of PMD. In case of failure: When an exception is received, nothing has changed in the EMS domain otherwise, the requested functionality of the TPs specified are disabled for collection of PMD **except** the ones identified as not disabled.
Exceptions	1) Processing failure: The requested operation could not be performed. 2) Invalid input: No TP/ME is specified in the input data or it contains at least one invalid data item. 3) NE communication loss.
Traceability	{Requirement II. 121}

7.12.3 NMS retrieves current Performance Monitoring Data (PMD) for a specified set of TPs

Use Case 7.12.3: NMS retrieves current Performance Monitoring Data (PMD) for a specified set of TPs

Use Case Name	NMS retrieves current Performance Monitoring Data (PMD) for a specified set of TPs
Summary	The NMS requests the EMS to send current PMD data to the NMS. A list of TP and layer rate measurement points have to be specified. It is possible to filter the amount of requested PMD based on a list of PM parameters. For each measurement point, location category (nearEnd (unidirectional) and/or farEnd (unidirectional) and/or bidirectional) and granularity (15min register and/or, 24hr register and/or NA) has to be specified.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS must have knowledge of the MEs and TPs specified. 3) PMD collection must have been enabled for the TPs specified. 4) The EMS must have active communications established with the ME, to which the TPs belong.
Begins when	The NMS requests the EMS to send current PMD.
Description	1) The NMS sends a request command to retrieve the current PMD to the EMS: The NMS specifies the measurement points and a list of PM parameters (e.g. BBE, ES) from which the current PMD are requested. It is possible to identify all TPs of a ME or a set of individual TPs (PTPs and/or CTPs). For each measurement point the NMS has to define: • the layer rate • the granularity (15min-register and/or 24hr-register and/or NA) and • the location (nearEnd (unidirectional) and/or farEnd (unidirectional) and/or bidirectional) An empty list for PM parameter, layer rate, granularity or location means that the ME has to use every supported item. 2) The EMS validates the provided parameters. 3) If the determination of a PM Parameter is Zero for a specified TP, the EMS returns also these records 4) If no current performance data are available for (some of) the PM parameters for (some of) the TPs specified in the request, the EMS returns UNAVAILABLE for those records. 5) In general PMD will not be complete for current 15min or 24hr registers. Incomplete records will be marked as INCOMPLETE. 6) The operation mode will be best-effort. Only the supported PM parameters have to be send by the EMS. 7) The EMS sends current PMD as specified in the request to the NMS.

Use Case 7.12.3: NMS retrieves current Performance Monitoring Data (PMD) for a specified set of TPs

Ends when	In case of success: The NMS receives a response with the requested current PMD. In case of failure: The NMS receives an exception.
Post-conditions	In case of success: Nothing has changed in the EMS domain. In case of failure: Nothing has changed in the EMS domain.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed. 3) Invalid input: No TP/ME is specified in the input data or it contains at least one invalid data item. 4) NE communication loss.
Traceability	{Requirement II. 131}

7.12.4 NMS retrieves the storage time of 24hr and 15min Performance Monitoring Data (PMD) records

Use Case 7.12.4: NMS retrieves the storage time of 24hr and 15min Performance Monitoring Data (PMD) records

Use Case Name	NMS retrieves the storage time of 24hr and 15min Performance Monitoring Data (PMD) records
Summary	The NMS requests the EMS to return the storage capacity or holding time in hours it keeps 24hr and 15min PMD records after the corresponding 24hr or 15min collecting registers has been closed. This is supposed to be the worst case time. There is no requirement on the EMS to support any predefined holding times.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. If the EMS has to contact the MEs to get the capacity, the EMS must also have an active communications established with every ME.
Begins when	The NMS sends a request to the EMS to be informed about PMD storage times.
Description	1) The NMS sends a request command to the EMS. 2) The EMS validates the request. 3) If the EMS does not support the storage of PMD, the minimum time across all MEs managed by the EMS has to be calculated by the EMS (if necessary by consulting each ME in its domain). 4) The EMS returns one PMD storage time for all 24hr collection registers and one PMD storage time for all 15min collection registers.
Ends when	In case of success: The NMS receives a response with the required information. In case of failure: The NMS receives an exception.
Post-conditions	In case of success: Nothing has changed in the EMS domain. In case of failure: Nothing has changed in the EMS domain.
Exceptions	1) Not implemented: The EMS does not support this service 2) Processing failure: The requested operation could not be performed. 3) NE communication loss.
Traceability	{Requirement II. 124}

7.12.5 NMS retrieves PM capabilities of a Managed Element (ME)

Use Case 7.12.5: NMS retrieves PM capabilities of a Managed Element (ME)

Use Case Name	NMS retrieves PM capabilities of a Managed Element (ME)
Summary	The NMS requests the EMS to return the PM capabilities (list of supported PM Parameters and location) of a ME for a specified layer rate
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) A communication session must be active between the EMS and the relevant ME.
Begins when	The NMS sends a request to the EMS to be informed about PM capabilities.
Description	1) The NMS sends a request command to the EMS: Within the request the NMS specifies the ME and the layer rate for which PM capabilities are to be returned. 2) The EMS validates the provided parameters. 3) The EMS returns a list of PM parameters that are supported for the ME and layer rate specified.
Ends when	In case of success: The NMS receives a response with the required information. In case of failure: The NMS receives an exception.
Post-conditions	In case of success: Nothing has changed in the EMS domain. In case of failure: Nothing has changed in the EMS domain.
Exceptions	1) Processing failure: The requested operation could not be performed. 2) Invalid input: The provided name does not reference a ME or layer rate contains undefined rates 3) Entity not found: The provided name references an object which does not exist 4) NE communication loss.
Traceability	{Requirement II. 054}, {Requirement II. 221}, {Requirement II. 222}

7.12.6 NMS retrieves historical Performance Monitoring Data (PMD) for a specified set of TPs

Use Case 7.12.6: NMS retrieves historical Performance Monitoring Data (PMD) for a specified set of TPs

Use Case Name	NMS retrieve historical Performance Monitoring Data (PMD) for a specified set of TPs

Use Case 7.12.6: NMS retrieves historical Performance Monitoring Data (PMD) for a specified set of TPs

Summary	The NMS requests the EMS to store history PMD within a specified time interval into a data file, and to send that data file to a specified destination using FTP mechanisms. A list of TP and layer rate, location categories (nearEnd (unidirectional) and/or farEnd (unidirectional) and/or bidirectional) and a granularity (15min, 24hr) for each measurement point has to be specified. It is possible to filter the amount of requested PMD based on a list of PM parameters. Username and password for file transfer to the given destination are also needed as input for this operation.
Actors	NMS, remote target
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS must have knowledge of the MEs and the TPs specified. 3) If the EMS has to retrieve the requested PMD from the ME, the EMS must also have an active communications established with the relevant MEs.
Begins when	The NMS sends a requests to the EMS to send PMD to a specific destination for a set of specified TPs.

Use Case 7.12.6: NMS retrieves historical Performance Monitoring Data (PMD) for a specified set of TPs

Description	1) The NMS sends the request command to the EMS: The NMS specifies the measurement points and a list of PM parameters (e.g. BBE, ES) from which the historical PMD are requested. It is possible to identify all TPs of a ME or a set of individual TPs (PTPs and/or CTPs). For each measurement point the NMS has to define: • the layer rate • the granularity (15min-register and/or 24hr-register and/or NA) and • the location (nearEnd (unidirectional) and/or farEnd (unidirectional) and/or bidirectional) An empty list for PM parameters, layer rate, granularity or location means that every supported item is requested. 2) Regarding the FTP mechanism the NMS has to specify: • the destination (hostname of the destination machine and full pathName of the target directory within the destination machine) to which the output file has to be sent • a user name for the file • a password for the file 3) In order to limit the amount of requested historical PMD the NMS may also provide a time frame (start time, end time). 4) The 'start' and 'end' time are aligned to UTC. 24hr time periods are considered to start at and include 0:00 am. 15min time periods are considered to start at and include 00/15/30/45 minute. 5) The NMS may force the EMS to contact also the MEs in order to get also the history data that has not been uploaded to the EMS yet. 6) The EMS validates the provided parameters. 7) The returned PM parameters are best-effort. Only the supported PM parameters have to be send in the file. 8) The EMS stores PMD as specified in the request into a data file, and sends that data file by FTP to the destination, using username and password as specified.hostname of the destination machine and the full pathName of the target directory within the destination machine 9) The EMS shall notify the NMS when the file transfer is completed or when the file transfer is failed. Notes concerning output file format: Considering a TP and a PM parameter the following holds true: If the NMS specified PM parameter is not supported for a TP there shall be no entry in the output file for this parameter and TP. Note: Use Case 7.12.5: NMS retrieves PM capabilities of a Managed Element (ME) can be used to check if PM parameter are supported for a TP in advance.

Use Case 7.12.6: NMS retrieves historical Performance Monitoring Data (PMD) for a specified set of TPs

	Otherwise, i.e. if the PM parameter is supported for a TP, if there are no measurement intervals at all, within the specified time interval, there is no entry in the output file for this parameter and TP. if there are some UNAVAILABLE measurement intervals, within the specified time interval, the description field has the value UNAVAILABLE. If there are a number of successive UNAVAILABLE periods the EMS has to combine these periods in the output file. if a measurement interval is only partly covered by the given time interval, a status field shall identify this and the related PM data record shall contain the number of monitored seconds. to indicate zero suppression, (i.e. if PM parameter has value zero over some successive measurement intervals), the related PMD record shall contain the number of full measurement periods with zero values.
Ends when	In case of success: The NMS receives an indication that the file transfer is completed. In case of failure: The NMS receives an exception or gets an indication that the file transfer has been failed
Post-conditions	In case of success: PMD for the set of TPs specified are sent to the requested destination. Nothing has changed in the EMS domain. In case of failure: Nothing has changed in the EMS domain.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed. 3) Invalid input: Any input parameter is invalid 4) NE communication loss.
Traceability	{Requirement II. 128}, {Requirement II. 129}, {Requirement II. 130}

7.12.7 NMS sets PM thresholds on a TP

Use Case 7.12.7: NMS sets PM thresholds on a TP

Use Case Name	NMS sets PM thresholds on a TP
Summary	The purpose of this use case is to modify the values of 15min- and/or 24hr- and/or instantaneous measurement PM Thresholds on a TP.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS must have knowledge of the TP specified. 3) A communication between the EMS and the relevant ME has to be active.
Begins when	The NMS sends the request to the EMS.
Description	1) The NMS send the request to the EMS: The NMS specifies the TP to which the request shall be applied to. It can be applied to a PTP or a CTP and a set of Performance Monitoring Points (PMPs). 2) The EMS checks if the operation is supported. 3) The EMS validates the input parameters. 4) The EMS checks the existence of the TP. 5) Thresholds within the TP that are not defined in the request remain as they are (i.e. thresholds currently applied at the specified TP and layer rate measurement points are not modified if the NMS does not explicitly specify them in the request). 6) The EMS set the thresholds of the TP in the NE as requested. This part is best effort. 7) The EMS returns a complete list of the current PM Thresholds for the specified TP. 8) For every set of PM thresholds set on a TP the EMS sends AVCs notifications for the affected PMPs to the notification service
Ends when	In case of success: The NMS receives a complete list of the current PM Thresholds for the specified TP response. In case of failure: The NMS receives an exception.
Post-conditions	In case of success: The requested PM Thresholds are modified on the TP. In case of failure: Nothing has changed in the EMS domain.

Use Case 7.12.7: NMS sets PM thresholds on a TP

Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed. 3) Invalid input: The name of the TP does not reference a Termination Point, or at least one of the provided parameters are not valid 4) Entity not found: The TP name references a TP which does not exist 5) NE communication loss.
Traceability	{Requirement I. 054}, {Requirement II. 125}

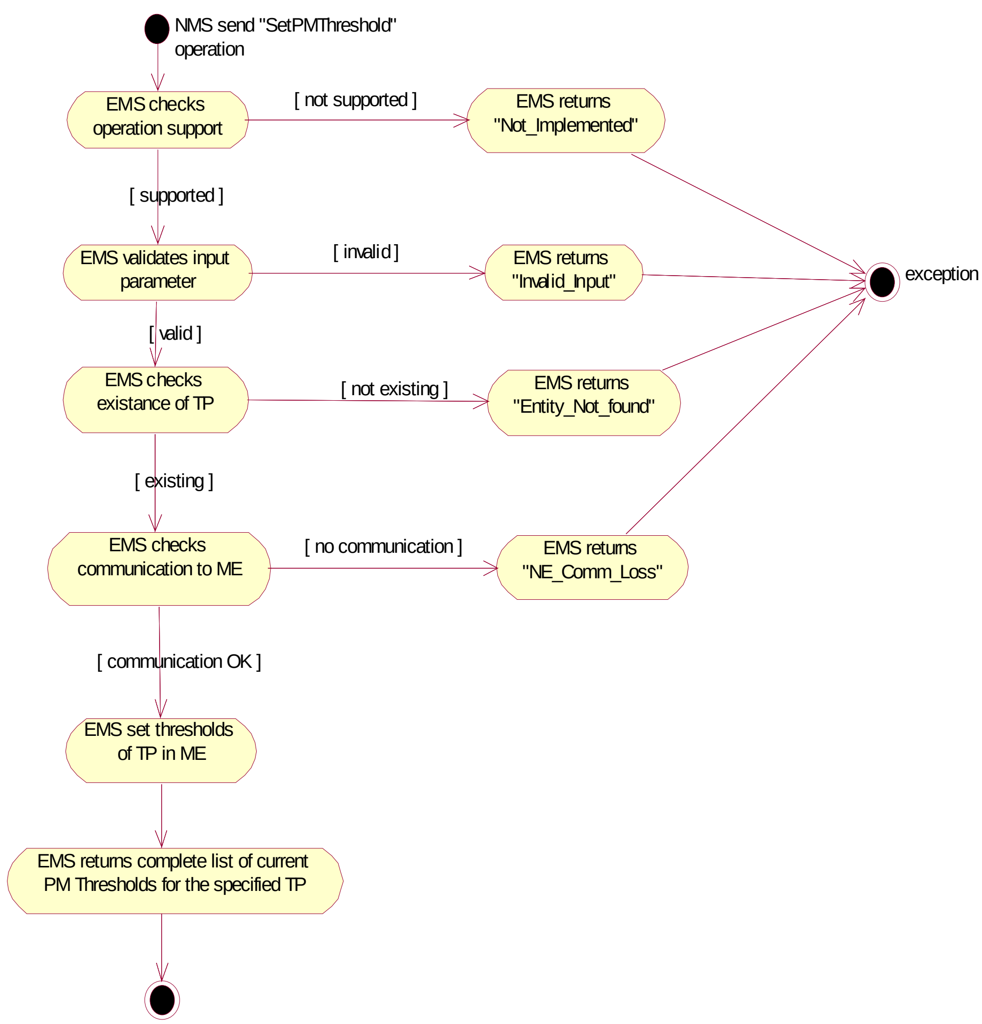


Figure 7.11: Activity Diagram for NMS sets PM thresholds on a TP

7.12.8 NMS retrieves PM threshold settings from a TP

Use Case 7.12.8: NMS retrieves PM threshold settings from a TP

Use Case Name	NMS retrieves PM threshold settings from a TP
Summary	The purpose of this use case is to retrieve the current values of 15min- and/or 24hr and/or instantaneous measurement PM Thresholds from a Termination Point.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS must have knowledge of the TP specified. 3) If the EMS has to contact the ME to get this information, a communication between the EMS and the relevant ME has to be active.
Begins when	The NMS sends the request to the EMS.
Description	1) The NMS send the request to the EMS. The NMS specifies the TP to which the request shall be applied to. It can be applied to a PTP or a CTP. A specific PM parameter and a Performance Monitoring Point (PMP). 2) The NMS specifies the specific TP measurement point to which the request shall be applied to. It can be applied to either a PTP or a CTP. 3) The EMS checks if the operation is supported. 4) The EMS validates the input parameters. 5) The EMS checks the existence of the TP. 6) The NMS has to define the PM parameter name. 7) The requested PM Threshold values are returned by the EMS.
Ends when	In case of success: The NMS receives a response with the current threshold settings for the TP. In case of failure: The NMS receives an exception.
Post-conditions	In case of success: Nothing has changed in the EMS domain. In case of failure: Nothing has changed in the EMS domain.
Exceptions	1) Processing failure: The requested operation could not be performed. 2) Invalid input: The name of the TP does not reference a Termination Point, the layer rate is undefined or the granularity is undefined. 3) Entity not found: The referenced TP does not exist. 4) NE communication loss.

Use Case 7.12.8: NMS retrieves PM threshold settings from a TP

Traceability	{Requirement I. 054}, {Requirement II. 055}

7.12.9 NMS enables Threshold Crossing Alerts (TCA) for a specified set of TPs

Use Case 7.12.9: NMS enables Threshold Crossing Alerts (TCA) for a specified set of TPs

Use Case Name	NMS enables Threshold Crossing Alerts (TCA) for a specified set of TPs
Summary	The NMS requests the EMS to enable the spontaneous (immediately after getting the event form the ME) generation of a TCA for a list of TP measurement points, in case when a PM Threshold value is reached or crossed.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS must have knowledge of the MEs and the TPs specified. 3) If the EMS needs to enable the alert generation in the ME, the EMS must also have an active communications established with the ME, to which the TP belongs.
Begins when	The NMS sends a request to the EMS to enable the generation of TCAs for a specified set of TPs.
Description	1) The NMS sends an enabling request command to the EMS: The NMS specifies the measurement points to which the request shall be applied. It is possible to identify all TPs of a ME or a set of individual TPs (PTPs and/or CTPs). For each measurement point the NMS has to define: • the layer rate • the granularity (15min-register and/or 24hr-register and/or NA • the location (nearEnd (unidirectional) and/or farEnd (unidirectional) and/or bidirectional) An empty list for layer rate, granularity or location means that the ME has to use every supported item. 2) The EMS validates the provided parameters. 3) If necessary the EMS sends the appropriate command(s) to the ME to enable the reporting of TCAs from the MEs to the EMS. 4) Following successful processing of the appropriate command(s), the generation of TCAs is enabled for the specified TPs. 5) If generation of TCAs is already enabled for one or more of the specified TPs, the operation shall be considered a success. 6) The operation is best-effort. If alert generation could not be enabled for a subset of the TPs specified, a list identifying the non-enabled subset is returned. 7) For every item enabled the EMS sends a state change notification on the supervision state of the affected Performance Monitoring Point to the notification service.
Ends when	In case of success: The NMS receives a confirmation In case of failure: The NMS receives an exception or gets an indication of the subset of TPs that could not be enabled.

Use Case 7.12.9: NMS enables Threshold Crossing Alerts (TCA) for a specified set of TPs

Post-conditions	In case of success: The requested threshold supervision functionality of the TPs specified are enabled for generating TCAs. In case of failure: When the NMS receives an exception, nothing has changed in the EMS domain otherwise, the requested threshold supervision functionality of the TPs specified are enabled for generating TCA **except** the ones identified as not enabled.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed. 3) Invalid input: Some input data contains invalid data. 4) Unable to comply: The EMS is unable to enable TCA for the specified TPs, i.e. no TPs were enabled. 5) NE communication loss.
Traceability	{Requirement II. 122}, {Requirement II. 126}

7.12.10 NMS disables Threshold Crossing Alerts (TCA) for a specified set of TPs

Use Case 7.12.10: NMS disables Threshold Crossing Alerts (TCA) for a specified set of TPs

Use Case Name	NMS disables Threshold Crossing Alerts (TCA) for a specified set of TPs
Summary	The NMS requests the EMS to disable the generation of a TCA for a list of TP measurement points.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS must have knowledge of the MEs and the TPs specified.
Begins when	The NMS sends a request to the EMS to disable the generation of TCAs for a specified set of TPs.
Description	1) The NMS sends a disabling request command to the EMS: The NMS specifies the measurement points to which the request shall be applied. It is possible to identify all TPs of a ME or a set of individual TPs (PTPs and/or CTPs). For each measurement point the NMS has to define: • the layer rate • the granularity (15min-register and/or 24hr-register and/or NA) • the location (nearEnd (unidirectional) and/or farEnd (unidirectional) and/or bidirectional) An empty list for layer rate, granularity or location means that the ME has to use every supported item. 2) The EMS validates the provided parameters. 3) Following successful processing of the command, the generation of TCAs is disabled for the specified TPs. 4) If generation of TCAs is already disabled for one or more of the specified TPs, the operation shall be considered a success. 5) The operation is best-effort. If alert generation could not be disabled for a subset of the TPs specified, a list identifying the non-disabled subset is returned. 6) For every item disabled the EMS sends a state change notification on the supervision state of the affected Performance Monitoring Point to the notification service.
Ends when	In case of success: The NMS receives a confirmation In case of failure: The NMS receives an exception or gets an indication of the subset of TPs that could not be disabled.

Use Case 7.12.10: NMS disables Threshold Crossing Alerts (TCA) for a specified set of TPs

Post-conditions	In case of success: The requested threshold supervision functionality of the TPs specified do no longer generate TCAs. In case of failure: When the NMS receives an exception, nothing has changed in the EMS domain otherwise, the requested threshold supervision functionality of the TPs specified do no longer generate TCAs **except** the ones identified as not disabled.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed. 3) Invalid input: Some input data contains invalid data. 4) Unable to comply: The EMS is unable to disable TCA for the specified TPs, i.e. no TPs were disabled. 5) NE communication loss
Traceability	{Requirement II. 122}

7.12.11 On demand retrieval of historical Performance Monitoring Data (PMD) for a specified set of TPs

Use Case 7.12.11: On demand retrieval of historical Performance Monitoring Data (PMD) for a specified set of TPs

Use Case Name	On demand retrieval of historical Performance Monitoring Data (PMD) for a specified set of TPs
Summary	The NMS requests the EMS to send history PMD data to the NMS. A list of TP and layer rate measurement points have to be specified. It is possible to filter the amount of requested PMD based on a list of PM parameters. For each measurement point, location category (nearEnd (unidirectional) and/or farEnd (unidirectional) and/or bidirectional) and granularity (15min register and/or, 24hr register and/or NA) has to be specified.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The NMS must have knowledge of the MEs and TPs specified. 3) PMD collection must have been enabled for the TPs specified. 4) The EMS must have active communications established with the ME, to which the TPs belong.
Begins when	The NMS sends a request to the EMS to get the historical PMD for the specified TP.
Description	1) The NMS sends a request command to retrieve the history PMD to the EMS: The NMS specifies the measurement points and a list of PM parameters (e.g. BBE, ES) from which the historical PMD are requested. It is possible to identify all TPs of a ME or a set of individual TPs (PTPs and/or CTPs). For each measurement point the NMS has to define: • the layer rate • the granularity (15min-register and/or 24hr-register and/or NA) and • the location (nearEnd (unidirectional) and/or farEnd (unidirectional) and/or bidirectional) An empty list for PM parameter, layer rate, granularity or location means that the ME has to use every supported item. 2) The EMS validates the provided parameters. 3) If the determination of a PM Parameter is Zero for a specified TP, the EMS returns also these records. 4) If no current performance data are available for (some of) the PM parameters for (some of) the TPs specified in the request, the EMS returns UNAVAILABLE for those records. 5) The operation mode will be best-effort. Only the supported PM parameters have to be send by the EMS. 6) The EMS sends history PMD as specified in the request to the NMS

Use Case 7.12.11: On demand retrieval of historical Performance Monitoring Data (PMD) for a specified set of TPs

Ends when	In case of success: The NMS receives a response with the requested historical PMD. In case of failure: The NMS receives an exception.
Post-conditions	In case of success: Nothing has changed in the EMS domain. In case of failure: Nothing has changed in the EMS domain.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed. 3) Invalid input: No TP/ME is specified in the input data or it contains at least one invalid data item 4) NE communication loss. 5) Unable to comply: The EMS supports retrieval of PM data only for a few buckets in past and NMS request is exceeding that time period. 6) Entity not found: One of the TPs does not exist.
Traceability	{Requirement II. 163}

7.12.12 NMS retrieves Performance Monitoring Points (PMP) contained in an ME or a TP

Use Case 7.12.12: NMS retrieves Performance Monitoring Points (PMP) contained in an ME or a TP

Use Case Name	NMS retrieves Performance Monitoring Points (PMP) contained in an ME or a TP
Summary	The NMS requests the EMS to return all Performance Monitoring Points (PMPs) contained in an ME or a TP. Note that the result provides not only the monitoring and supervision states but also the PM capabilities on a per TP basis and the PM thresholds.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The EMS must have active communications established with the ME, to which the TPs belong.
Begins when	The NMS sends a request to the EMS to get all PMPs contained in an ME or a TP.
Description	1) The NMS sends a request command to the EMS. Within the request the NMS specifies the ME or the TP for which the contained PMPs are to be returned. 2) The EMS validates the provided parameters. 3) The EMS returns a list of PMPs that are contained in the ME or TP specified.
Ends when	In case of success: The NMS receives a response with the requested PMP. In case of failure: The NMS receives an exception.
Post-conditions	In case of success: Nothing has changed in the EMS domain. In case of failure: Nothing has changed in the EMS domain.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed. 3) Invalid input: The provided name does not reference a ME or TP. 4) Entity not found: The provided name references an object which does not exist. 5) NE communication loss.
Traceability	{Requirement II. 221}

7.12.13 NMS modifies TCA Parameter Profile

Use Case 7.12.13: NMS modifies TCA Parameter Profile

Use Case Name	NMS modifies TCA Parameter Profile Pointer

Use Case 7.12.13: NMS modifies TCA Parameter Profile

Summary	The purpose of this use case is to configure the TCA thresholds contained in the Performance Monitoring Points of a TP by modifying the assignment of the TCA Parameter Profiles to this TP.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) TP specified must exist. 3) TCA Parameter Profile specified must exist. 4) A communication between the EMS and the relevant ME has to be active.
Begins when	The NMS sends the request to the EMS.

Use Case 7.12.13: NMS modifies TCA Parameter Profile

Description	1) The NMS send the request to the EMS: The NMS specifies the TP to which the request shall be applied to. It can be applied to a PTP, a FTP or a CTP. 2) The NMS has to define: • the TCA Parameter Profile to be removed; could be empty • the TCA Parameter Profile to be added; could be empty the granularity (15min-register or 24hr-register or NA) and for every threshold to be set 3) The EMS checks if the operation is supported. 4) The EMS validates the input parameters. 5) The EMS checks the existence of the TP. 6) If a TCA Parameter Profile has to be removed. Only the attribute TCA Parameter Profile Pointer of the TP has to be updated and an AVC notification has to be sent. 7) If a TCA Parameter Profile has to be added. The EMS checks the existence of the TCA Parameter Profile to be added. 8) If the TCA Parameter Profile to be added is already assigned to the TP the EMS shall overwrite the threshold values in the TP if necessary to make the values in the PMPs consistent with the values in the profile. (Note: Individual values may have been changed in the PMPs before via the Use Case 7.12.7: NMS sets PM thresholds on a TP). 9) When this is not successful, the EMS shall send an exception to the NMS explaining the reason. 10) If the TCA Parameter Profile is not already assigned to the TP but defines thresholds of the same layer as an already assigned TCA Parameter Profile the EMS shall send an exception to the NMS explaining the reason. 11) If the TCA Parameter Profile is not already assigned to the TP and defines thresholds of a different layer than the already assigned TCA Parameter Profiles, then the EMS shall set the threshold values as defined in the TCA Parameter Profile. When the thresholds could not be set the EMS shall send an exception to the NMS explaining the reason. **Note:** The attribute TCA Parameter Profile of the TP shall only be updated once within this operation and only one AVC notification shall be send.
Ends when	In case of success: The threshold values of the TP are consistent with the added/removed TCA Parameter Profiles.. In case of failure: The NMS receives an exception.

Use Case 7.12.13: NMS modifies TCA Parameter Profile

Post-conditions	In case of success: The threshold values of the TP are consistent with the added/removed TCA Parameter Profiles. In case of failure: Nothing has changed in the EMS domain.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed. 3) Invalid input: Any input parameter are syntactical incorrect. 4) Entity not found: TP or TCA Parameter Profile to be assigned does not exist. 5) Object in use: TCA Parameter Profile of the same layer is already assigned to the TP. 6) Unable to comply: The threshold values in the TP could not be configured. 7) NE communication loss.
Traceability	{Requirement II. 238}, {Requirement II. 239}

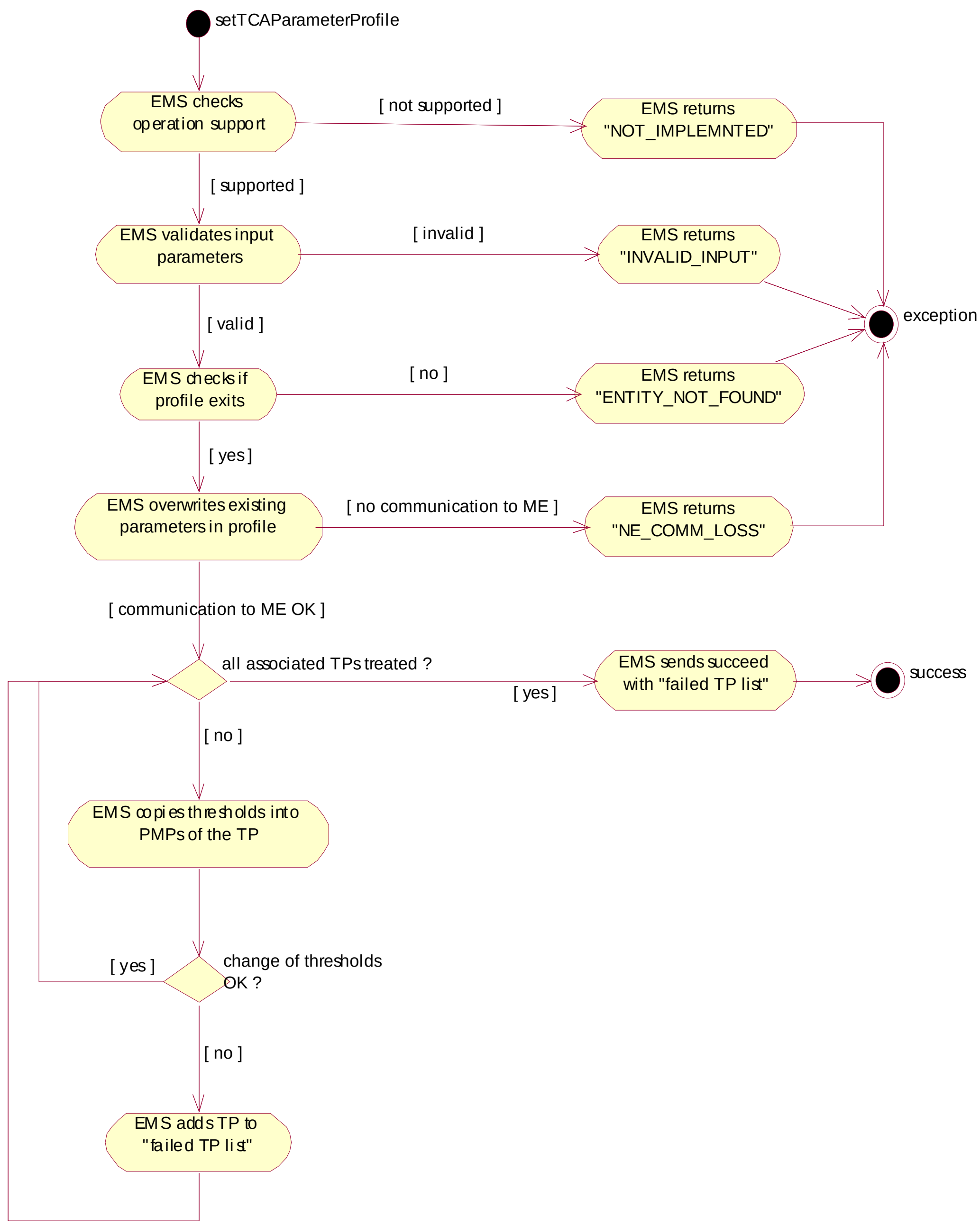


Figure 7.12: Activity Diagram for [NMS modifies TCA Parameter Profile]

7.12.14 NMS configures TCA Parameter Profile Pointer

Use Case 7.12.14: NMS configures TCA Parameter Profile Pointer

Use Case Name	NMS configures TCA Parameter Profile Pointer.
Summary	The purpose of this use case is to modify all the TCA thresholds contained in an already created TCA Parameter Profile. This operation overwrites all the existing threshold values of the profile with the new provided threshold values; i.e. it changes the profile completely.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) TCA Parameter Profile specified must exist. 3) A communication between the EMS and the relevant ME has to be active.
Begins when	The NMS sends the request to the EMS.
Description	1) The NMS send the request to the EMS: The NMS specifies the TCA Parameter Profile to which the request shall be applied to. 2) The NMS has to define the list of TCA parameters which contains the following information for each threshold value: • layer rate - already defined via the instance of the TCA Parameter Profile • parameter name - e.g. BBE, ES • granularity - i.e. 15 minute, 24 hours, N/A • location of the measurement • type of threshold - e.g. high, low • trigger flag - indicates if the threshold is for the trigger or the clear· • value of the threshold • unit of the threshold 3) The EMS checks if the operation is supported. 4) The EMS validates the input parameters. 5) The EMS checks the existence of the TCA Parameter Profile. 6) The EMS overwrites all parameters with the provided ones. **Note:** This includes also deletion of thresholds if the threshold is no longer contained in the provided list of TCA Parameters. 7) Then EMS changes all threshold values of all the TPs associated to this profile according to the new values. **Note:** Thresholds which have been deleted in the Profile will not be changed in the PMPs of the associated TPs. 8) The EMS has to return all TPs that could not be changed to the new threshold values due to some error reasons.

Use Case 7.12.14: NMS configures TCA Parameter Profile Pointer

Ends when	In case of success: The TCA Parameter Profile contains the new provided threshold values and every TP (except the TPs returned as failed TPs) assigned to this profile has been set to the new threshold values. In case of failure: The NMS receives an exception.
Post-conditions	In case of success: The TCA Parameter Profile contains the new provided threshold values. Every TP (except the TPs returned as failed TPs) assigned to this profile has been set to the new threshold values and if supervision is switched on, starts to use the new values for supervision. In case of failure: Nothing has changed in the EMS domain.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed. 3) Invalid input: One or more input parameters are syntactically incorrect. 4) Entity not found: The TCA Parameter Profile to be changed does not exist. 5) NE communication loss.
Traceability	{Requirement II. 238}, {Requirement II. 239}

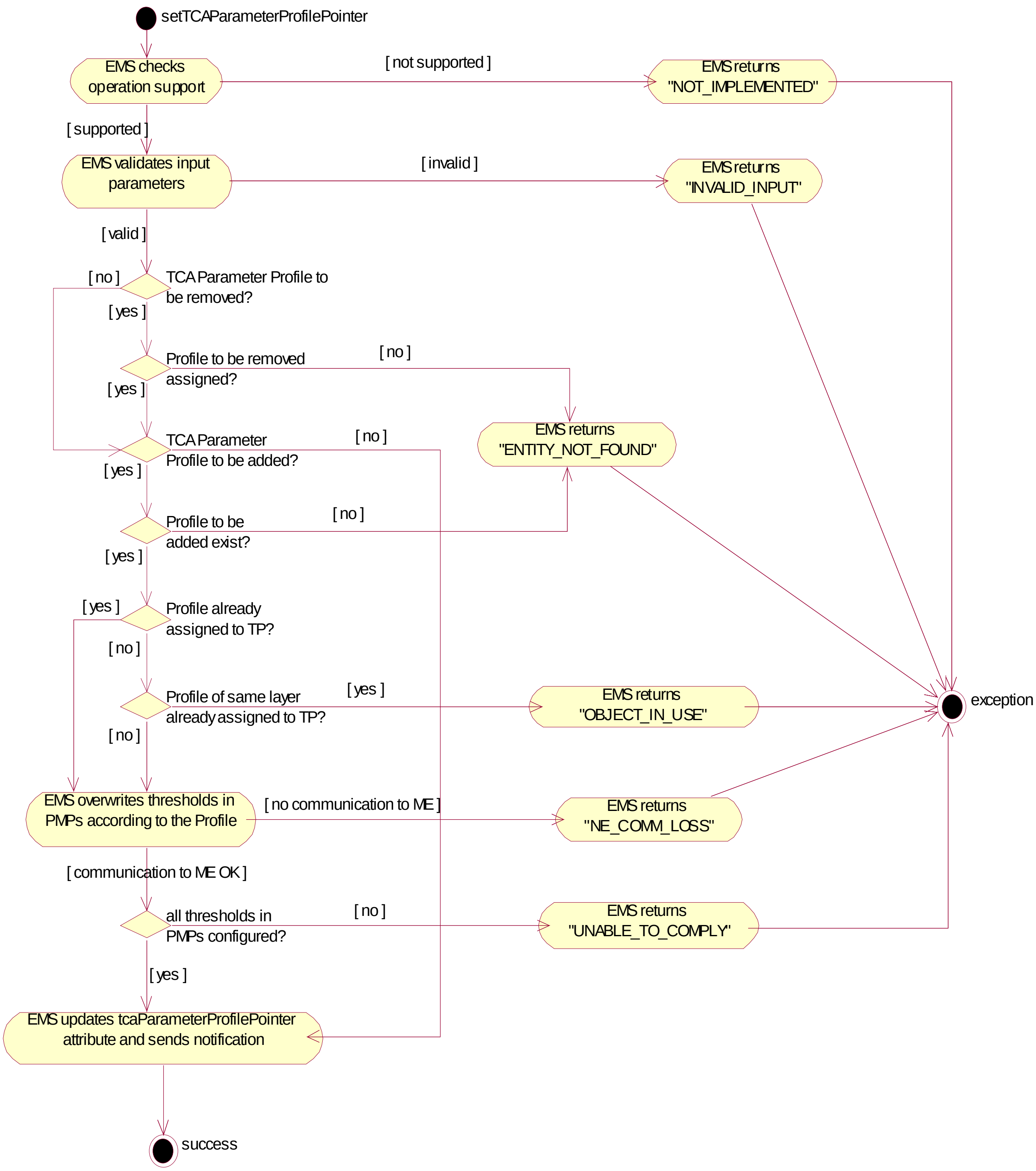


Figure 7.13: Activity Diagram for NMS configures TCA Parameter Profile Pointer

7.13 GUI Cut-Through Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.13.1:	NMS retrieves GUI Cut-Through window data	{Requirement II. 150}
Use Case 7.13.2:	Client based GCT launch	{Requirement II. 151}
Use Case 7.13.3:	Server based GCT launch	{Requirement II. 151}, {Requirement II. 152}

7.13.1 NMS retrieves GUI Cut-Through window data

Use Case 7.13.1: NMS retrieves GUI Cut-Through window data

Use Case Name	NMS retrieves GUI Cut-Through window data
Summary	The NMS is required to launch GCT. It retrieves the relevant GCT commands required to launch the GCT for each supported window type, or scope and context. Additionally, it receives indication if the EMS supports server-launch. The retrieval of the data is done through the NMS-EMS interface. Use Case 7.13.2: Client based GCT launch describes the client-based launch and Use Case 7.13.3: Server based GCT launch describes the server-based launch.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	NMS requires the relevant GCT window data
Description	NMS gets the list of GCT parameters for all windows supported by EMS through the NMS-EMS interface. EMS indicates if it also supports server-based launch.
Ends when	In case of success: The NMS has all required GCT information. In case of failure: The NMS does not have the required GCT window data; NMS cannot attempt to launch a GCT outside the interface.
Post conditions	The NMS has retrieved the GCT information from the EMS.
Exceptions	Processing failure: The requested operation could not be performed.
Traceability	{Requirement II. 150}

Note:

For Use Case 7.13.2: Client based GCT launch and Use Case 7.13.3: Server based GCT launch, NMS has to ensure that it is possible for GCT to be displayed on the GCT Display location. For example, the GCT server has permission to access a given display address; access control is turned off at the GCT client (e.g. the X-server of the NMS station) for the EMS GCT-server address (e.g. by executing xhost + [GCT host]).

7.13.2 Client based GCT launch

Use Case 7.13.2: Client based GCT launch

Use Case Name	Client based GCT launch
Summary	The EMS GUI is launched via a client-server protocol. The NMS initiates the GCT launch once it has retrieved the relevant application name and application arguments through the NMS-EMS interface. This USE CASE addresses the client-based GCT launch.
Actors	NMS and GCT server
Pre-conditions	1) The NMS has executed Use Case 7.13.1: NMS retrieves GUI Cut-Through window data. 2) NMS has ensured that it is possible for GCT to be displayed on the GCT Display location. 3) NMS has access to any other required server (e.g. ICA server for Citrix). For Citrix configuration The EMS applications can be accessed by a Citrix server running on the same OS platform (maybe different workstations). The EMS GCT application is registered with Citrix server under a certain application name. The relevant application name is known to the EMS.
Begins when	User request to display a GUI component that is managed by the EMS.
Description	1) If required, the NMS requests the GCT parameters (USE CASE 1) 2) The NMS determines which EMS context and scope has to be launched. 3) The NMS runs the GCT command using the parameters retrieved
Ends when	The EMS GCT application is displayed on GCT Display.
Post-conditions	The GCT application opens the display successfully at the NMS.
Exceptions	1) Processing Failure: The requested operation could not be performed. 2) Invalid input: One of the input parameter is invalid. 3) Entity not found: The object referenced does not exist. 4) Capacity exceeded: The maximum number of active GCTs has been reached.
Traceability	{Requirement II. 151}

7.13.3 Server based GCT launch

Use Case 7.13.3: Server based GCT launch

Use Case Name	Server based GCT launch (e.g. using an X-protocol)

Use Case 7.13.3: Server based GCT launch

Summary	The GUI can be launched via a pure X-protocol, initiated by the EMS / server side once it is requested to do so through the MTNM interface. This USE CASE addresses the optional server-based GCT launch.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.13.1: NMS retrieves GUI Cut-Through window data. 2) EMS has indicated that it supports server launch. 3) NMS has ensured that it is possible for GCT to be displayed on the GCT Display location.
Begins when	User request to display a GUI component that is managed by the EMS.
Description	1) The NMS request a launch of the GCT via the MTNM interface. 2) The EMS runs the GCT application on the X-server and attempts to redirect the display to the NMS display address. This step is outside of the MTNM interface and is shown as a dashed line in the figure below. 3) Once the GCT is displayed on NMS and the system call returns, the server launch operation returns and indicates if it supports closing of GCT window.
Ends when	The display of a given GUI application becomes available at the NMS X-terminal.
Post-conditions	1) The NMS is aware to whether the GCT has launched successfully (otherwise an exception would have been thrown). 2) If the EMS supports closing of GCT windows then it must maintain the aliases of the open GCT windows. 3) The EMS must notify the NMS system whenever the closing of the GCT being opened is not available.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Internal error. 3) Invalid Input. 4) Unable to comply. For example the EMS cannot support the (valid) context and scope combination requested. 5) Maximum GCT displays has been reached.
Traceability	{Requirement II. 151}, {Requirement II. 152}

7.14 ATM Provisioning Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.14.1:	NMS creates a Traffic Descriptor (TD)	{Requirement II. 097}, {Requirement II. 098}
Use Case 7.14.2:	NMS modifies a Traffic Descriptor (TD) on a VPCTP or VCCTP	{Requirement II. 074}
Use Case 7.14.3:	NMS deletes a Traffic Descriptor (TD)	{Requirement II. 099}

7.14.1 NMS creates a Traffic Descriptor (TD)

Use Case 7.14.1: NMS creates a Traffic Descriptor (TD)

Use Case Name	NMS creates a Traffic Descriptor (TD)
Summary	The use case describes how the NMS requests that an EMS create a new Traffic Descriptor. The NMS passes the traffic descriptor information to the EMS. The resulting Traffic Descriptor is returned as a result to the NMS.
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS sends the Add TD request to EMS.
Description	1) The NMS sends a create TD request to EMS with the TD distinguishing information. The NMS may request that the EMS enforce a unique user label (the user label is selected by the NMS and the EMS makes sure the user label is not already in use). 2) The EMS validates the request. The EMS creates the TD, assigns a unique name to the TD and stores it persistently. 3) The EMS replies with a success indication. 4) A Create TD Notification is sent by EMS to the notification service. Note: It is up to the internal implementation of EMS which data are stored in EMS persistently and which ones will be queried from ME as required.
Ends When	The EMS sends a reply to the NMS.
Post-Conditions	The TD has been established in the EMS
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed. 3) Invalid input: The NMS has supplied invalid input data. 4) User label in use: The user label supplied by the NMS is already in use. 5) Capacity exceeded: The Input data is valid but the EMS cannot create the TD because the EMS has already exceeded its maximum number of TDs.
Traceability	{Requirement II. 097}, {Requirement II. 098}

7.14.2 NMS modifies a Traffic Descriptor (TD) on a VPCTP or VCCTP

Use Case 7.14.2: NMS modifies a Traffic Descriptor (TD) on a VPCTP or VCCTP

Use Case Name	NMS modifies a Traffic Descriptor (TD) on a VPCTP or VCCTP
Summary	The NMS changes the Traffic Descriptors (ingress and/or egress) for a VPCTP or VCCTP.
Actors	NMS
Pre-conditions	1) The NMS has executed Use Case 7.2.2: NMS creates a session with EMS. 2) The TD to be associated with the VPCTP or VCCTP has to exist.
Begins when	The NMS sends the modify VPCTP or VCCTP request to the EMS
Description	1) The NMS sends the modify VPCTP or VCCTP request to the EMS. 2) The EMS validates the VPCTP or VCCTP identifier. 3) The EMS validates the TD Name(s). 4) The EMS communicates with the Network Elements. 5) The EMS changes the TD(s) if possible. This will fail under the following conditions: • The VPCTP is terminating, but was not explicitly created. • The input data is valid but the EMS cannot change the TD because resources are not available on the ME. • The input data is valid but the EMS cannot change the TD because the new TD does not provide enough resources for the existing VCs running over the VPCTP. 6) The EMS replies with a success indication. 7) The EMS will generate an AVC for the TP with the new TD name.
Ends when	The EMS sends a reply to the NMS.
Post-conditions	The TD(s) on the VPCTP or VCCTP has been updated and uses the traffic values contained in the associated TD.
Exceptions	1) Processing failure: The requested operation could not be performed. 2) Invalid input: VPCTP, VCCTP or TD identifier is invalid. 3) Entity not found: The VPCTP, VCCTP or TD does not exist. 4) Unable to comply: The traffic values could not be configured in the VPCTP or VCCTP. 5) NE communication loss.
Traceability	{Requirement II. 074}

7.14.3 NMS deletes a Traffic Descriptor (TD)

Use Case 7.14.3: NMS deletes a Traffic Descriptor (TD)

Use Case Name	NMS deletes a Traffic Descriptor (TD)
Summary	The NMS deletes a TD. Remark: • The NMS is responsible to maintain the consistency of TD across multiple administrative sub-networks. Routine integrity checking may be required. NMS must also check that the TD is not being used in any EMS under its network domain, before the TD can be deleted from the persistent storage. • This operation is idempotent. If the service is called with the name of a non-existent Traffic Descriptor, it will succeed.
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS sends the Delete TD request to the EMS.
Description	1) The NMS sends Delete TD request to the EMS. 2) The EMS validates the TD identifier and checks that this TD is not being used on by the EMS. 3) If no SNCs or TPs are using this TD, the EMS deletes the TD entry. 4) The EMS replies with a success indication. 5) A Delete TD Notification is sent by EMS to the notification service.
Ends When	The EMS sends a reply to the NMS.
Post-Conditions	The TD has been deleted in the EMS (Note. The TD can only be deleted in the NMS if and only if no EMS uses this TD.
Exceptions	1) Not implemented: The EMS does not support this service. 2) Processing failure: The requested operation could not be performed. 3) Object in use: The TD is being used. 4) Entity not found: The TD does not exist in the EMS. 5) Invalid input: The TD Name is invalid.
Traceability	{Requirement II. 099}

7.15 ATM Connection Management Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 7.15.1:	NMS creates and activates an ATM Subnetwork Connection (SNC)	{Requirement II. 088}, {Requirement II. 089}

Figure 7.14: ATM Connection Management Use Cases

7.15.1 NMS creates and activates an ATM Subnetwork Connection (SNC)

Use Case 7.15.1: NMS creates and activates an ATM Subnetwork Connection (SNC)

Use Case Name	NMS creates and activates an ATM Subnetwork Connection (SNC)
Summary	The NMS sets up a Subnetwork Connection on an EMS Subnetwork. The NMS supplies the CTP names or NSAP addresses. This use case extends the Use Case 7.7.3: NMS creates and activates a Subnetwork Connection (SNC). The only differences are with respect to that use case are specified here.
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS sends the Create and Activate SNC request to the EMS.

Use Case 7.15.1: NMS creates and activates an ATM Subnetwork Connection (SNC)

Description	1) The EMS configures the end point CTPs. The EMS should try to determine if the operation will succeed before making any configuration changes. This would be done as part of validation. It is preferable not to change the configuration of one CTP if the EMS can determine that the next CTP cannot be changed. For each CTP in the SNC the EMS will do the following: • If the CTP is already used by a PARTIAL or ACTIVE SNC, the request is rejected. • For a connection at the VP layer, i) If the VP CTP was previously explicitly configured to contain VC CTPs the request will be rejected. (i.e. if it is TERMINATED_AND_AVAILABLE_FOR_MAPPING with an assigned Traffic Descriptor (assigned bandwidth)) ii) If there are VC CTPs using the VP CTP, then the request will be rejected. iii) If the VP CTP is already NEITHER_TERMINATED_NOR_AVAILABLE_FOR_MAPPING it will be left in this mode. • For a connection at the VC layer, i) If the VP CTP was previously either explicitly or implicitly configured to contain VC CTPs the mapping mode of the VP CTP will be left unchanged (TERMINATED_AND_AVAILABLE_FOR_MAPPING). ii) If the VP CTP is configured as NEITHER_TERMINATED_NOR_AVAILABLE_FOR_MAPPING, the VP CTP will be implicitly configured to contain VC CTPs. Note that there will be no bandwidth assigned to the terminating VP CTP (The Traffic Descriptors will be empty). In the above cases, if the mapping mode is changed, an AVC will be generated. The TP Traffic Descriptors specified on the end points are used. 2) For the intermediate points along the route of the SNC • If the SNC is created using network routing protocols, there will be no notifications regarding the connection state or mapping mode of the intermediate points along the route • For all other cases, the EMS may have to configure the mapping mode. Also appropriate notifications will be sent for the connection state and the mapping mode changes. 3) The EMS activates the required SNC in the network. • A network routed SNC will be an atomic operation from the cross-connect perspective. However the SNC can still use the PARTIAL state. If an SPVX source switch cannot reach the destination, it is considered to be PARTIAL because one or more resources have been allocated. The behavior is that the network routing protocol will try to establish the connection when the required links are in place.

Use Case 7.15.1: NMS creates and activates an ATM Subnetwork Connection (SNC)

Description	• An EMS routed SNC may use the PARTIAL SNC state • In a network routed SNC, the configuration of the endpoints (if they are in the EMS domain) will be included as part of the activate operation. If the EMS fails to create and activate the SNC it will be in the PARTIAL state as some resources are allocated.
Post-Conditions	Refer to Use Case 7.7.3: NMS creates and activates a Subnetwork Connection (SNC).
Exceptions	Refer to Use Case 7.7.3: NMS creates and activates a Subnetwork Connection (SNC).
Traceability	{Requirement II. 088}, {Requirement II. 089}

8 Control Plane Management Use Cases

8.1 Actor-System Context Diagram

Refer to Figure 7.1.

8.1.1 Exceptions

Refer to Table 7.1.

8.2 Discovery and Inventory Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 8.2.1:	(UC_1) EMS notifies of changes in network capacity (increases/ decreases in capacity)	{Requirement II. 064}, {Requirement II. 065}, {Requirement II. 066}, {Requirement II. 359}, {Requirement II. 361}
Use Case 8.2.2:	(UC_4) NMS discovers MultiLayer Routing Area (MLRA)s	{Requirement II. 354}, {Requirement II. 355}, {Requirement II. 357}, {Requirement II. 358}
Use Case 8.2.3:	(UC_28) NMS discovers subordinate MultiLayer Routing Area (MLRA)s	{Requirement II. 356}
Use Case 8.2.4:	(UC_5) NMS discovers MLSNPP Links	{Requirement II. 359}, {Requirement II. 360}, {Requirement II. 361}, {Requirement II. 362}, {Requirement II. 363}, {Requirement II. 364}, {Requirement II. 365}
Use Case 8.2.5:	(UC_6) NMS requests available MLSNPP Link capacity	{Requirement II. 359}, {Requirement II. 360}, {Requirement II. 361}, {Requirement II. 366}
Use Case 8.2.6:	(UC_7) NMS discovers Calls and top level Connections	{Requirement II. 381}, {Requirement II. 405}, {Requirement II. 406}, {Requirement II. 407}, {Requirement II. 408}, {Requirement II. 409}, {Requirement II. 410}, {Requirement II. 411}
Use Case 8.2.7:	(UC_8) EMS notifies of a new Call and its Connection(s)	{Requirement II. 064}
Use Case 8.2.8:	(UC_9) EMS notifies of deleted Call and its top-level Connection(s)	{Requirement II. 065}

8.2.1 EMS notifies of changes in network capacity (increases/decreases in capacity)

Use Case 8.2.1: (UC_1) EMS notifies of changes in network capacity (increases/decreases in capacity)

Name	EMS notifies of changes in network capacity (increases/decreases in capacity)

Use Case 8.2.1: (UC_1) EMS notifies of changes in network capacity (increases/decreases in capacity)

<table>
<tr><td>Summary</td><td>
This use case describes how the NMS is informed by the EMS of an increase/decrease in network resources available to the Control Plane as a result of them being allocated/ deallocated to Control Plane by some action (either automatic or by a network operator).
This case covers notifications related to MLSNPPs and MLSNPPLinks including:

MLSNPPLinks internal to the MultiLayer Routing Area managed by the EMS. Such MLSNPPLinks may be E-NNI or I-NNI.
MLSNPPLinks partially managed by the EMS (i.e. where only one end of the link is managed by the EMS). Such MLSNPPLinks may:

be wholely within the operator network (i.e. the remote end is managed by another EMS owned by the operator) in which case the MLSNPPLink may be E-NNI or I-NNI
exit the operator network in which case the MLSNPPLinks may be E-NNI, UNI or “dumb”.

For an E-NNI only limited information, e.g., reachability, is available.
For a UNI no information is available for representing the other end of the MLSNPPLink other than an external address (access group/access point (TNA Name)) of the remote end.
For a dumb MLSNPPLink no information is available from the remote end.

Note:

This use case does not deal with notifications from failure or recovery of transport resources. The indication related to this case is constant regardless of the status of the MLSNPPLink and conveys potential capacity.
The MLSNPPLink and MLSNPP exist only when resources are allocated. Although an SNPPLink with no allocated resources may be visible to the Control Plane this will not be reported over the NML-EML interface.
An object creation notification is raised when an MSLNPPLink or MLSNPP is created as a result of the allocation of the initial resources (refer to Use Case 7.4.5: EMS notifies NMS of inventory change).
An attribute change notification is raised when the capacity of an MLSNPPLink or MLSNPP is increased or decreased (whilst still greater than zero capacity) (refer to Use Case 7.4.5: EMS notifies NMS of inventory change).
The MLSNPP/MLSNPPLink objects are created by the EMS as a result of the presence of the approriate SNPPs/SNPPLinks in the network.
The notification of MLSNPPLink and MLSNPP capacity change and creation/ deletion will be associated with a change in corresponding allocation state attributes of TPs.
The MLSNPP/MLSNPPLink creation and MLSNPP/MLSNPPLink capacity increase are the result of association establishment between the MLSNPP/ MLSNPPLink and the one or more CTPs (potential CTPs) that gives rise to the creation of SNPs. The MLSNPP/MLSNPPLink will carry information on the association between the CTP and SNP.

</td></tr>
</table>

Use Case 8.2.1: (UC_1) EMS notifies of changes in network capacity (increases/decreases in capacity)

Actor(s)	EMS
Pre-Conditions	The NMS has executed Use Case 7.4.1: NMS registers to receive network updates information from the EMS.
Begins When	EMS detects a change in network capacity available to the Control Plane as a result of some action (either automatic or network operator initiated).
Description	1) The EMS detects an increase/decrease in SNPP/SNPPLink capacity in the monitored network. The allocation results in either MLSNPP/MLSNPPLink creation/deletion or capacity increase/decrease of an existing MLSNPP/ MLSNPPLink. 2) The EMS generates a notification to inform the NMS. The notification will be an a) Object creation notification for an MLSNPPLink Creation b) Object creation notification for an MLSNPP Creation c) Object deletion notification for an MLSNPPLink Deletion d) Object deletion notification for an MLSNPP Deletion e) Attribute change notification for an MLSNPPLink capacity increase/ decrease f) Attribute change notification for an MLSNPP capacity increase/decrease 3) The NMS receives notifications from the notification service.
Ends When	In case of success: The NMS receives the notification. In case of failure: The NMS does not receive the notification.
Post-Conditions	In case of success: The NMS knows the increase in link capacity. The NMS database remains aligned with the EMS's database. In case of failure: The NMS does not know the increase in link capacity. The NMS database is misaligned with the EMS's database..
Exceptions	None.
Traceability	{Requirement II. 064}, {Requirement II. 065}, {Requirement II. 066}, {Requirement II. 359}, {Requirement II. 361}.

8.2.2 NMS discovers MultiLayer Routing Area (MLRA)s

Use Case 8.2.2: (UC_4) NMS discovers MultiLayer Routing Area (MLRA)s

Name	NMS discovers MultiLayer Routing Area (MLRA)s
Summary	This use case describes how the NMS retrieves the name and details of all the MultiLayer Routing Areas (MLRA) and Routing Nodes (refer to Use Case 7.4.3: NMS discovers the EMS network inventory). This use case also describes how the NMS can retrieve the name and details of only the top-level MLRA. Each MultiLayer Routing Area returned will at least include: • The name of the MLRA • For the lowest MLRA (i.e. where the MLRA represents a Routing Node), the name of the ME associated with the MLRA. The EMS shall return an empty ME list when it does not have the information, for example: • When the MLRA represents a Routing Node known to the EMS but where the EMS does not have visibility of the supporting ME. • When the MLRA does not represent a Routing Node. **Note:** A Routing Node may represent more than on ME under certain circumstances, however this release of the interface has not been proven for the cases where the MLRA represents more than one ME.
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS sends a request to the EMS to retrieve all Multi-Layer Routing Areas (MLRAs) including routing nodes.
Description	1) The NMS sends a request to retrieve the MLRAs or specifically the top-level MLRA. 2) The EMS validates the request. a) If the syntax is in error, then an Invalid Input exception is raised. b) If the specified top level MLRA object is not known to the EMS, an Entity Not Found exception is raised. 3) If the request is valid the EMS returns the list of MLRAs that match the specified criteria to the NMS.
Ends When	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.

Use Case 8.2.2: (UC_4) NMS discovers MultiLayer Routing Area (MLRA)s

Post-Conditions	In case of success: The NMS has been provided with all MLRA information. In case of failure: No information has been retrieved.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 354}, {Requirement II. 355}, {Requirement II. 357}, {Requirement II. 358}

8.2.3 NMS discovers subordinate MultiLayer Routing Area (MLRA)s

Use Case 8.2.3: (UC_28) NMS discovers subordinate MultiLayer Routing Area (MLRA)s

Name	NMS discovers subordinate MultiLayer Routing Area (MLRA)s
Summary	This use case describes how the NMS retrieves the name and details of all the MultiLayer Routing Areas (MLRA)s (and Routing Nodes) that are one level subordinate to a specific MultiLayer Routing Area (MLRA). The NMS provides the name of a MultiLayer Routing Area. The EMS returns a list of MLRAs as identified in (refer to Use Case 8.2.2: (UC_4) NMS discovers MultiLayer Routing Area (MLRA)s).
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS sends a request to the EMS to retrieve all MultiLayer Routing Areas (MLRAs) including Routing Nodes.
Description	1) The NMS sends a request to retrieve available MLRAs matching specified criteria. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified object is not of the right type (i.e, an MLRA), an Invalid Input exception is raised. c) If the specified MLRAobject is not known to the EMS, an Entity Not Found exception is raised 3) If the request is valid the EMS returns the list of MLRAs that match the specified criteria to the NMS.
Ends When	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The NMS has been provided with all MLRA information. In case of failure: No information has been retrieved.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 356}

8.2.4 NMS discovers MLSNPP Links

Use Case 8.2.4: (UC_5) NMS discovers MLSNPP Links

Name	NMS discovers MLSNPP Links
Summary	This use case describes how the NMS requests the MLSNPP Links allocated to the Control Plane. The EMS provides the information for the MLSNPP links matching the scope of the request. The NMS may specify one and only one of the following criteria: • **All**: the EMS should supply the properties of all MLSNPP Links. • **MLSNPP Link name**: the EMS should supply the properties of the MLSNPP Link that was specified by the NMS. • **MLRA name, internal MLSNPP Links**: the EMS should supply the properties of all MLSNPP Links that are contained in the specified MLRA (internal to the MLRA, i.e. I-NNIs of the specified MLRA) • **MLRA name, edge MLSNPP Links**: the EMS should supply the properties of all MLSNPP Links that are at the edge of the specified MLRA (an edge SNPP Link may be E-NNI, I-NNI or UNI). • **A pair of MLRA names**: the EMS should supply the properties of all MLSNPP Links that interconnect the specified MLRAs. • **TNA Name**: the EMS should supply the properties of the MLSNPP Links that match anyone of the specified TNA Names. • **PTP/CTP/FTP Name**: the EMS should supply the properties of the MLSNPP Links that are totally or partially supported by the referenced TPs. For example an MLSNPP Link shall be returned if that link terminates on an SNPP that is a pool of SNPs of which at least one is supported by one of the CTPs contained in a referenced PTP. The request can be qualified to indicate whether SNP details should be included in the response. The EMS provides properties of the retrieved MLSNPP links. Note: • For UNI links, no information is visible to the EMS beyond the local port other than the TNA Name of the link. • The objects referenced in the request (e.g., MLSNPP names, MLRA name) could belong to a different level of the routing hierarchy. • Since a Multi-Layer approach is used in the interface modeling, the referenced objects could belong to difference Transport Layers.
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS sends the MLSNPP Link retrieve request to the EMS.

Use Case 8.2.4: (UC_5) NMS discovers MLSNPP Links

Description	1) The NMS sends a request to retrieve the MLSNPP Link information that match the specified criteria. 2) The EMS validates the request. a) If the syntax is in error, an Invalid Input exception is raised b) If a specified object is not of the right type (valid types depending upon specific request are: MLRA, MLSNPP Link, PTP, CTP, FTP), an Invalid Input exception is raised c) If a specified object (MLRA, MLSNPP Link, PTP, CTP, FTP) is not known to the EMS, an Entity Not Found exception is raised d) If a specified TNA Name is not known to the EMS, an Entity Not Found exception is raised. e) If a specified object is not under the control of the Control Plane (e.g. in case the Solution Set represents both MLRA and MLSN through the same managed object), an Unable To Comply exception is raised. 3) If the request is valid the EMS returns a list of MLSNPP Links matching the criteria of the request.
Ends When	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The NMS knows all the MLSNPP Links, including their properties, within the scope of the request. In case of failure: No information has been retrieved.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 359}, {Requirement II. 360}, {Requirement II. 361}, {Requirement II. 362}, {Requirement II. 363},{Requirement II. 364}, {Requirement II. 365}

8.2.5 NMS requests available MLSNPP Link capacity

Use Case 8.2.5: (UC_6) NMS requests available MLSNPP Link capacity

Name	NMS requests available MLSNPP Link capacity
Summary	This use case describes how the NMS retrieves the available MLSNPP Link capacity. The EMS retrieves available capacity for the specified: • MLSNPP Link Name • Layer Rate list For each specified rate the EMS returns a count of possible connections availalable over the MLSNPP Link. If a layer rate is not specified then the EMS returns available capacity for each rate that is supported by the entity.
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS sends a request to the EMS to retrieve available MLSNPP Link Capacity.
Description	1) The NMS sends a request to retrieve available MLSNPP Link capacity. 2) The EMS validates the request. a) If the syntax is in error, an Invalid Input exception is raised. b) If a specified object is not of the right type (i.e., an MLSNPP Link), then an Invalid Input exception is raised. c) If a specified object (i.e., the MLSNPP Link) is not known to the EMS, an Entity Not Found exception is raised. d) If the specified layer rate is not supported by the specified MLSNPP Link, an Unable To Comply exception is raised. 3) If the request is valid the EMS returns the MLSNPP Link capacity information to the NMS.
Ends When	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The NMS has been provided with the requested capacity data. In case of failure: No information has been retrieved.
Exceptions	Refer to Table 7.1.

Use Case 8.2.5: (UC_6) NMS requests available MLSNPP Link capacity

Traceability	{Requirement II. 359}, {Requirement II. 360}, {Requirement II. 361}, {Requirement II. 366}

8.2.6 NMS discovers Calls and top level Connections

Use Case 8.2.6: (UC_7) NMS discovers Calls and top level Connections

Name	NMS discovers Calls and top level Connections

Summary	This use case describes how the NMS retrieves the name and details of the Calls and associated top level Connections in the scope of specified top level MultiLayer Routing Area.
	
	An EMS will only return the Calls and top level Connections for which it controls the originating end point. By implication this informs the NMS which EMS controls each Call (this information will be used in subsequent cases).
	
	The input of this operation are:
	• A top level MLRA name
	• And optionally one of:
	    • An ME name
	    • A Call name
	    • A Call ID
	
	The EMS is required to reply with:
	• List of Calls
	• For each Call a list of top level Connections
	
	Note:
	• Each Connection has the Call ID attribute and also contains information about the originating end point and the destination end point.
	• Connections will be reported regardless of their current state (i.e. regardless of whether they are "Searching" or "Complete" (refer to {Requirement I. 132}).
	• The reply can be an empty list, if there are no Calls established from any of the points controlled by the EMS.

Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS sends a request to the EMS to retrieve all the Calls and their supporting top level Connections.

Use Case 8.2.6: (UC_7) NMS discovers Calls and top level Connections

Description	1) The NMS sends the request to retrieve all the Calls andtheir top level Connections to the EMS. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If a specified object is not of the right type (i.e., MLRA, ME, Call), an Invalid Input exception is raised. c) If the specified MLRA name, ME name, Call name or Call ID is not known to the EMS, an Entity Not Found exception is raised. d) If the specified MLRA is not a top level routing area, an Unable To Comply exception is raised. 3) If the request is valid the EMS returns to the NMS the list (one item in the case of Call Name and Call ID) of Calls, and for each Call a list of the supporting top-level Connections. • If an ME name is specified only the Calls that have at least one Connection that originates from the named ME will be returned. • All connections that support a Call will be returned regardless of the ME from which they originate.
Ends When	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: The NMS has the received the requested information from the EMS. In case of failure: No information has been retrieved.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 381}, {Requirement II. 405}, {Requirement II. 406}, {Requirement II. 407}, {Requirement II. 408}, {Requirement II. 409}, {Requirement II. 410}, {Requirement II. 411}

8.2.7 EMS notifies of a new Call and its Connection(s)

Use Case 8.2.7: (UC_8) EMS notifies of a new Call and its Connection(s)

<table>
<tr><td>Name</td><td>EMS notifies of new Call and its Connection(s)</td></tr>
<tr><td>Summary</td><td>This use case describes how the NMS is informed by the EMS of Calls and Connections that have been established in the network (refer to Use Case 7.4.5: EMS notifies NMS of inventory change).

Notification(s) will be raised regardless of which system created the Call/Connection (e.g., UNI, Local Craft on Control Plane, EMS, NMS etc.). Notifications will be raised for both SCs and SPCs.

An EMS will notify the NMS of the creation of:
• each Call for which it controls the originating end point.
• each top level Connection for which it controls the originating end point.
• each Connection (representing a Connection Segment) existing in the scope of an MLRA subordinate to the top-level MLRA where the EMS manages the originating end point of the Connection in that MLRA.

In the case of a Call creation the EMS shall provide all of the following information:
• one object creation notification for the Call.
• zero, one or more object creation notification(s) for the top-level Connection(s) supporting the Call.
• zero, one or more object creation notification(s) for the Connection(s) in the scope of local MLRA, subordinate of top level Connection(s).
    • Other EMSs in the network may notify of the creation of Connections in the scope of MLRAs that are not that of the Call originating end point

Note:

• Connections are only available from MLRAs that are not representing a Routing Node . The connectivity within a Routing Node is considered as part of the route of the Connection in the MLRA one level superior to the Routing Node
• Connections will be reported regardless of their current state (i.e. regardless of whether they are “Searching” or “Complete” (refer to {Requirement I. 132}).
• A Connection may represent the home route or the actual route or a combination of both. Notifications are raised for Connections representing the home route and Connections representing the actual route as appropriate.
• A number of assumptions are made about the Control Plane to enable the Connection model identified to function, these are documented in the restrictions and limitations of the release.</td></tr>
<tr><td>Actor(s)</td><td>EMS</td></tr>
<tr><td>Pre-Conditions</td><td>The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.</td></tr>
<tr><td>Begins When</td><td>EMS detects a new Call established or Connection created.</td></tr>
</table>

Use Case 8.2.7: (UC_8) EMS notifies of a new Call and its Connection(s)

Description	1) The EMS detects one of a new: • Call established by network operator or end client in the monitored MLRA (i.e. the EMS directly manages the Call Originating Point), • Top Level Connection created by the network operator or end client • Subordinate Connection created as a result of a Top Level Connection creation 2) The EMS sends an appropriate object creation notification to the notification service.
Ends When	In case of success: The NMS receives a notification. In case of failure: The NMS does not receive a notification.
Post-Conditions	In case of success: The NMS knows the newly created Call/Connection. The NMS database remains aligned with the EMS's database. . In case of failure: The NMS does not know the newly created Call/Connection. The NMS database is misaligned with the EMS's database..
Exceptions	None
Traceability	{Requirement II. 064}

8.2.8 EMS notifies of deleted Call and its top-level Connection(s)

Use Case 8.2.8: (UC_9) EMS notifies of deleted Call and its top-level Connection(s)

Name	EMS notifies of deleted Call and its top-level Connection(s)
Summary	This use case describes how the NMS is informed by the EMS of Calls and Connections that have been deleted from the network. Notification(s) will be raised regardless of which system deleted the SC/SPC (UNI, Local Craft on Control Plane, EMS, NMS). An EMS will notify the NMS of the deletion of: • each Call for which it controls the originating end point • each top-level Connection for which it controls the Originating Point. • each Connection (representing a connection segment) existing in the scope of an MLRA subordinate to the top level MLRA where the EMS manages the originating point of the connection in that MLRA. Considering the deletion of a call the information related to this would be provided to the NMS by the EMS as • one object deletion notification for the Call, plus • zero, one or more object deletion notification(s) for the top level Connection(s) supporting the Call, plus • zero, one or more object deletion notification(s) for the Connection(s) in the scope of local MLRA, subordinate of top level Connection(s). ◦ Other EMSs in the network may notify of the deletion of Connections in the scope of MLRAs that are not that of the Call Originating Point Note: A connection deletion notification will include the call id.
Actor(s)	EMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	EMS detects a new Call established or Connection deleted by a network operator or end client.

Description	1) The EMS detects one of a:
	  • Call deleted by a network operator or end client in the monitored MLRA (i.e. the EMS directly manages the Call Originating Point),
	  • Top Level Connection deleted by a network operator or end client
	  • Subordinate Connection deleted as a result of a Top Level Connection deletion.
	2) The EMS generates an appropriate object deletion notification to inform the NMS.
	3) The NMS receives the notification from the Notification Service.

Use Case 8.2.8: (UC_9) EMS notifies of deleted Call and its top-level Connection(s)

Ends When	In case of success: The NMS receives a notification. In case of failure: The NMS does not receive a notification.
Post-Conditions	In case of success: The NMS knows the Call/Connection deleted. The NMS database remains aligned with the EMS's database. . In case of failure: The NMS does not know the Call/Connection deleted. The NMS database is misaligned with the EMS's database..
Exceptions	None
Traceability	{Requirement II. 065}

8.3 Provisioning Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 8.3.1:	(UC_10) NMS assigns a UNI MLSNPP Link to a Signaling Control-ler.	{Requirement II. 373}, {Requirement II. 375}
Use Case 8.3.2:	(UC_11) NMS requests to set the UNI signaling protocol and parameters	{Requirement II. 375}
Use Case 8.3.3:	(UC_11b) NMS modifies signaling parameters	{Requirement II. 375}
Use Case 8.3.4:	(UC_12) NMS requests EMS to enable the UNI signaling on an MLSNPP Link.	{Requirement II. 376}
Use Case 8.3.5:	(UC_13) NMS requests EMS to disable the UNI signaling on an MLSNPP Link	{Requirement II. 377}
Use Case 8.3.6:	(UC_14) NMS requests EMS to deassign a UNI MLSNPP Link from a Signalling Controller	{Requirement II. 374}
Use Case 8.3.7:	(UC_15) NMS requests EMS to assign TNA Names to components of an MLSNPP	{Requirement II. 379}
Use Case 8.3.8:	(UC_32) NMS requests EMS to assign TNA Names to components of an MLSNPP Link	{Requirement II. 378}
Use Case 8.3.9:	(UC_24) NMS requests EMS for subordinate MultiLayer Routing Area IDs involved in the route of a Connection	{Requirement II. 391}, {Requirement II. 405}, {Requirement II. 408}, {Requirement II. 410}
Use Case 8.3.10:	(UC_25) NMS requests EMS for the route details of a specified Call within a specified routing area	{Requirement II. 390}, {Requirement II. 405}, {Requirement II. 407}, {Requirement II. 408}, {Requirement II. 409}, {Requirement II. 410}
Use Case 8.3.11:	(UC_26) NMS requests EMS for the ID of each Call supported by a specified TP/SNPP/TNA	{Requirement II. 385}
Use Case 8.3.12:	(UC_27) NMS request EMS for the list of MLSNPPs	{Requirement II. 367}, {Requirement II. 368}, {Requirement II. 369}, {Requirement II. 370}, {Requirement II. 371}

8.3.1 NMS assigns a UNI MLSNPP Link to a Signaling Controller

Use Case 8.3.1: (UC_10) NMS assigns a UNI MLSNPP Link to a Signaling Controller.

Use Case Name	NMS assigns a UNI MLSNPP Link to a Signaling Controller

Use Case 8.3.1: (UC_10) NMS assigns a UNI MLSNPP Link to a Signaling Controller.

Summary	This use case describes how the NMS requests the EMS to assign the UNI MLSNPP Link to a Signalling Controller. An NMS shall provide an EMS with the following: • The name of an MLSNPPLink. • The Signalling Controller identifier. Note: For this Use Case to be successful the link must not be associated with a Signalling Controller (the Signalling Controller identifier must be an empty string).
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to assign the UNI MLSNPP Link to a Signalling Controller.
Description	1) The NMS sends the request to assign the specified MLSNPP Link to a Signalling Controller. 2) The EMS validates the request: a) If the syntax is in error, then an Invalid Input exception is raised b) If a specified object is not of the right type (i.e., an MLSNPP Link), then an Invalid Input exception is raised c) If the specified MLSNPP Link is not known to the EMS, then an Entity Not Found exception is raised. d) If the specified MLSNPP Link is not a UNI, then an Unable To Comply exception is raised. e) If the MLSNPP Link is already assigned to another Signaling Controller, then an Unable To Comply exception is raised. f) If the specified Signaling Controller is not know the the EMS, then an Unable To Comply exception is raised. 3) If the request is valid the EMS sets the Signaling Controller name for the MLSNPP Link and sends a response to the NMS. • If the MLSNPP Link is already assigned as requested the request is accepted. 4) If the request is valid the EMS sends appropriate AVC notifications to reflect the changes • If the MLSNPP Link is already assigned as requested no notification is sent.
Ends when	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.

Use Case 8.3.1: (UC_10) NMS assigns a UNI MLSNPP Link to a Signaling Controller.

Post-conditions	In case of success: • The MLSNPP Link has been associated with a Signalling Controller and an appropriate attribute is updated (i.e. from no association to association state). • The signalling protocol type attribute of MLSNPP Link either assumes the default signalling protocol value of the just associated interface or it is protocol not set, according to implementation. • The EMS has forwarded attribute value change notification(s) In case of failure: • Some or all of the above information is missing. • The original Signalling Controller configuration has not changed.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 373}, {Requirement II. 375}

8.3.2 NMS requests to set the UNI signaling protocol and parameters

Use Case 8.3.2: (UC_11) NMS requests to set the UNI signaling protocol and parameters

Use Case Name	NMS requests to set the UNI signaling protocol and parameters.
Summary	This use case describes how the NMS requests the EMS to select the signaling protocol and parameters to be used on the Signalling Controller associated with a specific MLSNPP Link. An NMS shall provide an EMS with the following: • The name of an MLSNPPLink • The Signalling Protocol identifier • A List of Signalling Parameters Note: For this use case to be successful the following conditions must be satisfied: • The signalling is disabled • The MLSNPP Link is a UNI
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to set the Signalling Controller.

Use Case 8.3.2: (UC_11) NMS requests to set the UNI signaling protocol and parameters

Description	1) The NMS sends the request to set the signaling protocol and parameters for the MLSNPP Link. 2) The EMS validates the request: a) If the syntax is in error, then an Invalid Input exception is raised. b) If a specified object is not of the right type (i.e., an MLSNPP Link), then an Invalid Input exception is raised. c) If the specified MLSNPP Link is not known to the EMS, then an Entity Not Found exception is raised. d) If the specified MLSNPP Link is not a UNI, then an Unable To Comply exception is raised. e) If a Signaling Controller is not assigned to the MLSNPP Link, then an Unable To Comply exception is raised. f) If the MLSNPP Link signaling is enabled, then an Unable To Comply exception is raised. g) If the Signalling Controller does not accept the provisioning of the specific protocol or parameters, then an Unable To Comply exception is raised. 3) If the request is valid the EMS sets the specified protocol and associated parameters (e.g. timers) depending upon the protocol selected and sends a response to the NMS. The following are protocol(s) that may be set: • RSVP-TE • CR-LDP • None 4) If the request is valid the EMS sends appropriate AVC notifications to reflect the changes.
Ends when	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: • The UNI signaling protocol has been selected for the referenced MLSNPP Link. The signalling protocol type attribute of MLSNPP Link is updated accordingly. • The requested parameters have been set. • The EMS has forwarded attribute value change notification(s). In case of failure: • Some or all of the above information is missing. • The original Signalling Controller configuration has not changed.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 375}

8.3.3 NMS modifies signaling parameters

Use Case 8.3.3: (UC_11b) NMS modifies signaling parameters

Use Case Name	NMS modifies signaling parameters.
Summary	This use case describes how the NMS requests the EMS to modify signalling parameters, but not the UNI signaling protocol. This modification can be executed even when signalling is enabled (refer to Use Case 8.3.4: (UC_12) NMS requests EMS to enable the UNI signaling on an MLSNPP Link.). An NMS shall provide an EMS with the following: • The name of an MLSNPPLink • A List of Signaling Parameters Note: • For this Use Case to be successful the following conditions must be satisfied: • the MLSNPP Link is a UNI
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to modify the signalling parameters.
Description	1) The NMS sends the request to modify the signalling parameters for the MLSNPP Link. 2) The EMS validates the request: a) If the syntax is in error, then an Invalid Input exception is raised. b) If a specified object is not of the right type (i.e., an MLSNPP Link), then an Invalid Input exception is raised. c) If the specified MLSNPP Link is not known to the EMS, then an Entity Not Found exception is raised. d) If the specified MLSNPP Link is not a UNI, then an Unable To Comply exception is raised. e) If a Signaling Controller is not assigned to the MLSNPP Link, then an Unable To Comply exception is raised. f) If the Signalling Controller does not accept the provisioning of the specific protocol or parameters, then an Unable To Comply exception is raised. 3) If the request is valid the EMS sets the specified parameters and sends a response to the NMS. 4) If the request is valid the EMS sends appropriate AVC notifications to reflect the changes.
Ends when	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request. There is no change in the signaling parameters.

Use Case 8.3.3: (UC_11b) NMS modifies signaling parameters

Post-conditions	In case of success: • The requested parameters have been set. • The EMS has forwarded attribute value change notification(s). In case of failure: • Some or all of the above information is missing. • The original Signalling Controller configuration has not changed.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 375}

8.3.4 NMS requests EMS to enable the UNI signaling on an MLSNPP Link.

Use Case 8.3.4: (UC_12) NMS requests EMS to enable the UNI signaling on an MLSNPP Link.

Use Case Name	NMS requests EMS to enable the UNI signaling on an MLSNPP Link.
Summary	This use case describes how the NMS requests the EMS to enable the UNI signalling for the Signalling Controller associated with a specific MLSNPP Link. An NMS shall provide an EMS with the following: • The name of an MLSNPPLink Note: • For this Use Case to be successful the following conditions must be satisfied: • the MLSNPP Link is a UNI
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to enable the Signalling Controller.
Description	1) The NMS sends the request to enable the signaling for the MLSNPP Link. 2) The EMS validates the request: a) If the syntax is in error, then an Invalid Input exception is raised. b) If a specified object is not of the right type (i.e., an MLSNPP Link), then an Invalid Input exception is raised c) If the specified MLSNPP Link is not known to the EMS, then an Entity Not Found exception is raised. d) If the specified MLSNPP Link is not a UNI, then an Unable To Comply exception is raised. e) If a Signaling Controller is not assigned to the MLSNPP Link, then an Unable To Comply exception is raised. f) If a protocol has not been assigned to the Signaling Controller for the MLSNPP Link, then an Unable To Comply exception is raised. 3) [Implementation Specific] If the Control Plane does not offer the capability to enable signaling on an MLSNPP Link that is supporting a Call, then an Unable To Comply exception is raised. Note: It may be possible to support SPC to SC migration in some implementations 4) If the request is valid the EMS enables signaling and sends a response to the NMS. 5) If the request is valid the EMS sends appropriate attribute value change notifications to reflect the changes.

Use Case 8.3.4: (UC_12) NMS requests EMS to enable the UNI signaling on an MLSNPP Link.

Ends when	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: • The referenced MLSNPP Link is now enabled for the UNI signaling. • The EMS has forwarded attribute value change notification(s). In case of failure: • Some or all of the above information is missing. • The original Signalling Controller configuration has not changed.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 376}

8.3.5 NMS requests EMS to disable the UNI signaling on an MLSNPP Link

Use Case 8.3.5: (UC_13) NMS requests EMS to disable the UNI signaling on an MLSNPP Link

Use Case Name	NMS requests EMS to disable the UNI signaling on an MLSNPP Link.
Summary	This use case describes how the NMS requests the EMS to disable the UNI signalling for the Signalling Controller associated with a specific MLSNPP Link. An NMS shall provide an EMS with the following: • The name of an MLSNPPLink
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to disable the Signalling Controller.
Description	1) The NMS sends the request to disable the signaling for the MLSNPP Link. 2) The EMS validates the request: a) If the syntax is in error, then an Invalid Input exception is raised. b) If a specified object is not of the right type (i.e., an MLSNPP Link), then an Invalid Input exception is raised. c) If the specified MLSNPP Link is not known to the EMS, then an Entity Not Found exception is raised. d) If the specified MLSNPP Link is not a UNI, then an Unable To Comply exception is raised. e) If a Signaling Controller is not assigned to the MLSNPP Link, then an Unable To Comply exception is raised. f) If Signaling Controller is not enabled for MLSNPP Link, then an Unable To Comply exception is raised. g) [Implementation Specific] If the Control Plane does not offer the capability to disable signaling if Calls are supported by the MLSNPPLink and one or more Calls are supported by the MLSNPPLink, then an Unable To Comply exception is raised. Note: It may be possible to support SPC to SC migration in some implementations 3) If the request is valid the EMS requests the Signaling Controller to disable signaling and then the EMS sends a response to the NMS. 4) If the request is valid the EMS sends appropriate attribute value change notifications to reflect the changes.
Ends when	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.

Use Case 8.3.5: (UC_13) NMS requests EMS to disable the UNI signaling on an MLSNPP Link

Post-conditions	In case of success: • The UNI signaling is disabled for the referenced MLSNPP Link. • The EMS has forwarded attribute value change notification(s) if the MLSNPP Link state has changed. In case of failure: • Some or all of the above information is missing. • The original Signalling Controller configuration has not changed.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 377}

8.3.6 NMS requests EMS to deassign a UNI MLSNPP Link from a Signalling Controller

Use Case 8.3.6: (UC_14) NMS requests EMS to deassign a UNI MLSNPP Link from a Signalling Controller

Use Case Name	NMS requests EMS to deassign a UNI MLSNPP Link from a Signalling Controller.
Summary	This use case describes how the NMS requests the EMS to deassign the UNI MLSNPP Link from a Signalling Controller. An NMS shall provide an EMS with the following: • The name of an MLSNPPLink
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to deassign the UNI MLSNPP Link from a Signaling Controller
Description	1) The NMS sends the request to deassign a MLSNPP Link from a Signaling Controller. 2) The EMS validates the request: a) If the syntax is in error, then an Invalid Input exception is raised. b) If a specified object is not of the right type (MLSNPP Link), then an Invalid Input exception is raised. c) If the specified MLSNPP Link is not known to the EMS, then an Entity Not Found exception is raised. d) If the specified MLSNPP Link is not a UNI, then an Unable To Comply exception is raised. e) If a Signaling Controller is not assigned to the MLSNPP Link, then an Unable To Comply exception is raised. f) If Signaling Controller is enabled for MLSNPP Link, then Unable To Comply exception is raised. 3) If the request is valid the EMS removes the relationship to the Signaling Controller and then the EMS sends a response to the NMS. Note: The Signaling Controller identifier is set to empty string. 4) If the request is valid the EMS sends appropriate attribute value change notifications to reflect the changes.
Ends when	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.

Use Case 8.3.6: (UC_14) NMS requests EMS to deassign a UNI MLSNPP Link from a Signalling Controller

Post-conditions	In case of success: • The MLSNPP Link is not associated with a Signalling Controller. • The EMS has forwarded attribute value change notification(s). In case of failure: Some or all of the above have not been completed.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 374}

8.3.7 NMS requests EMS to assign TNA Names to components of an MLSNPP

Use Case 8.3.7: (UC_15) NMS requests EMS to assign TNA Names to components of an MLSNPP

<table>
<tr><td>Use Case Name</td><td>NMS requests EMS to assign TNA Names to components of an MLSNPP.</td></tr>
<tr><td>Summary</td><td>This use case describes how the NMS requests the EMS to assign a TNA Name value to one or more components of a specified MLSNPP.

A TNA Name value may be assigned to the following components (each component has one TNA Name):
• An SNPP
• An SNP

A Group TNA Name value may be assigned to and SNPP in addition to the TNA Name.

An NMS shall provide an EMS with the following:
• An MLSNPP name
• A list of SNPP IDs each with:
    • a TNA Name
    • a Group TNA Name
    • both a TNA Name and a Group TNA Name
• A list of SNP ids each with a TNA Name

Note:

• If the NMS assignes a TNA Name this will overwrite an existing TNA Name.
• An empty string will indicate that no TNA Name has been assigned.
• If the NMS assigns an empty string this will clear the TNA Name (i.e., no TNA Name is assigned).
• Specification of the behaviour for connections already made to a component of the MLSNPP when its TNA Name is changed is outside the scope of this case and will depend upon the particular implementation. This use case allows for exceptions to be raised for implementations where the TNA Name change will be rejected.</td></tr>
<tr><td>Actors</td><td>NMS</td></tr>
<tr><td>Pre-conditions</td><td>The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.</td></tr>
<tr><td>Begins when</td><td>The NMS sends a request to the EMS to set the TNA Names on a specified MLSNPP.</td></tr>
</table>

Use Case 8.3.7: (UC_15) NMS requests EMS to assign TNA Names to components of an MLSNPP

Description	1) The NMS sends the request to set a TNA Name for each member of a list of components (SNPP and SNP) of a specified MLSNPP. 2) The EMS validates the request: a) If the syntax is in error, then an Invalid Input exception is raised. b) If a specified object is not of the right type (i.e., an MLSNPP Link), then an Invalid Input exception is raised. c) If the specified MLSNPP is not known to the EMS, then an Entity Not Found exception is raised. d) If any of the specified components (i.e., an SNPP and/or an SNP) of the MLSNPP does not exist, then an Entity Not Found exception is raised. e) [Implementation Specific] If the EMS determines that the TNA Name structure is not valid for any TNA Name, then an Invalid Input exception is raised. 3) If the request is valid, the EMS assigns the specified TNA Name (TNA Name and Group TNA Name as appropriate) to each specified component (SNPP and SNP) of the MLSNPP and then the EMS sends a response to the NMS identifying each component for which the TNA Names could not be assigned. Note: • The TNA Name assignment is best effort. • An empty TNA Name indicates that there is currently no assigned TNA Name. To clear the TNA Name for a specific SNPP/SNP the NMS sends an empty string referenced to the SNPP/SNP. 4) If the request is valid the EMS sends appropriate attribute value change notifications to reflect the changes.
Ends when	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: For each listed component the specified TNA Name is applied. In case of failure: The MLSNPP potentially does not have the requested TNA Names assigned.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 379}

8.3.8 NMS requests EMS to assign TNA Names to components of an MLSNPP Link

Use Case 8.3.8: (UC_32) NMS requests EMS to assign TNA Names to components of an MLSNPP Link

Use Case Name	NMS requests EMS to assign TNA Names to components of an MLSNPP Link.

Summary	This use case describes how the NMS requests the EMS to assign a TNA Name value to one or more components of a specified MLSNPP Link.
	A TNA Name value may be assigned to the following components (each component has one TNA Name):
	• An SNPP.
	• An SNP.
	A Group TNA Name value may be assigned to an SNPP in addition to the TNA Name.
	An NMS shall provide an EMS with the following:
	• An MLSNPP Link name.
	• A list of SNPP IDs each with:
	    • a TNA Name.
	    • a Group TNA Name.
	    • both a TNA Name and a Group TNA Name.
	• A list of SNP ids each with a TNA Name.
	Note:
	• If the NMS assignes a TNA Name this will overwrite an existing TNA Name.
	• An empty string will indicate that no TNA Name has been assigned.
	• If the NMS assigns an empty string this will clear the TNA Name (i.e., no TNA Name is assigned).
	• Specification of the behaviour for connections already made to a component of the MLSNPP Link when its TNA Name is changed is outside the scope of this case and will depend upon the particular implementation. This case allows for exceptions to be raised for implementations where the TNA Name change will be rejected.

Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to set the TNA Names on a specified MLSNPP Link.

Use Case 8.3.8: (UC_32) NMS requests EMS to assign TNA Names to components of an MLSNPP Link

Description	1) The NMS sends the request to set a TNA Name for each member of a list of components (SNPP and SNP) of a specified MLSNPP Link. 2) The EMS validates the request: a) If the syntax is in error, then an Invalid Input exception is raised. b) If a specified object is not of the right type (i.e., an MLSNPP Link), then an Invalid Input exception is raised. c) If the specified MLSNPP Link is not known to the EMS, then an Entity Not Found exception is raised. d) If any of the specified components (SNPP and SNP) of the MLSNPP Link does not exist, then an Entity Not Found exception is raised. e) [Implementation Specific] If the EMS determines that the TNA Name structure is not valid for any TNA Name, then an Invalid Input exception is raised. 3) If the request is valid, the EMS assigns the specified TNA Name (TNA Name and Group TNA Name as appropriate) to each specified component (SNPP and SNP) of the MLSNPP and then the EMS sends a response to the NMS identifying each component for which the TNA Names could not be assigned. Note: • The TNA Name assignment is best effort. • An empty TNA Name indicates that there is currently no assigned TNA Name. To clear the TNA Name for a specific SNPP/SNP the NMS sends an empty string referenced to the SNPP/SNP. 4) If the request is valid the EMS sends appropriate attribute value change notifications to reflect the changes.
Ends when	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: For each listed component the specified TNA Name is applied. In case of failure: The MLSNPP Link potentially does not have the requested TNA Names assigned
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 378}

8.3.9 NMS requests EMS for subordinate MultiLayer Routing Area IDs involved in the route of a Connection

Use Case 8.3.9: (UC_24) NMS requests EMS for subordinate MultiLayer Routing Area IDs involved in the route of a Connection

<table>
<tr><td>Use Case Name</td><td>NMS requests EMS for subordinate MultiLayer Routing Area IDs involved in the route of a Connection</td></tr>
<tr><td>Summary</td><td>This use case has been designed to operate in the following:
• A 2-level hierarchy, i.e. where the top-level MLRA only contains Routing Nodes.
• A 3-level hierarchy, i.e. where the top-level MLRA contains subordinate MLRAs and the subordinate MLRAs contain Routing Nodes.

This use case describes how the NMS retrieves the identifiers of the subordinate MLRAs at the lower partitioning level, which are known to be involved in the route of the Connection.

An NMS shall provide an EMS with the following:
• Name of a top-level MLRA.
• Name of the Connection in the MLRA.
• A filter parameter to indicate whether the response should include:
    • Home/intended route.
    • Actual route.
    • Both home/intended and actual routes.

This use case describes how the EMS returns to the NMS a list of MLRA identifiers which are 1-level subordinate to the input top-level MLRA and are involved in the route of the specified Connection. An empty list in the response indicates that the network has a 2-level hierarchy.

The response depends upon the signalling used in the network. The Control Plane component viewing the Connection from the originating point (or indeed any other point) may see all or only some of the routing areas involved in the Connection. Two key cases have been identified for the 3-level hierachy:
• The Control Plane component at the originating point of the Connection has a full view of the identifier of every routing area one level subordinate to the top-level routing area involved in the Connection. [“Full” case]
• The Control Plane component at the originating point of the Connection can only provide the identifiers of the local routing area, the neighbouring routing area and the destination routing area (and can not provide the identifiers of any other routing areas along the route). [“Sparce” case]

The EMS will indicate whether the response is a “Full” description of the route or a “Sparce” description.</td></tr>
</table>

Use Case 8.3.9: (UC_24) NMS requests EMS for subordinate MultiLayer Routing Area IDs involved in the route of a Connection	
	For the "Sparce" case the NMS will need to follow the route (i.e., starting from the originating routing area and progressing along the route to the subsequent routing areas) and use Use Case 8.3.10: (UC_25) NMS requests EMS for the route details of a specified Call within a specified routing area to find out the detail routes of each routing area. For the "Full" case the NMS can simply interrogate the EMSs for each routing area returned in any order by using Use Case 8.3.10: (UC_25) NMS requests EMS for the route details of a specified Call within a specified routing area to find out the detail routes of each routing area. For a 2-level hierarchy Use Case 8.3.10: (UC_25) NMS requests EMS for the route details of a specified Call within a specified routing area will only need to be used once to determine the full route. Note: • There may be no home/intended route in the case where none was provisioned. • There may be no actual route where the connection is in the Searching state or the actual route traverses a different MLRA to the home/intended route. • There will be no route information where the input MLRA is a Routing Node. • There will be more than one list of MLRA identifiers if more than one Connection of the same Call passes through the routing area referenced. • For an SNCP Connection, the home/intended route in the response will always include both main and spare legs of the SNCP protection. The actual route may specify only one leg, if only one leg is currently active in the network. • The route is described in full across the MLRA regardless of sharing of same Routing Node Connections among distinct routes of the same Call. • The output of Use Case 8.2.6: (UC_7) NMS discovers Calls and top level Connections is the intended input for this use case, i.e. which EMS controls each call. • The response shall contain the MLRAs at level N-1 only, N being the level of the input MLRA and Connection. • The concept of "recursive partitioning" means that any MLRA can be further partitioned into MLRAs at a lower partitioning level (the subordinate MLRAs), down to the MultiLayer Routing Node, which defines the bottom of the partitioning hierarchy. This use case has only been considered for the highest level MLRA as an input parameter.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to get all the subordinate MLRAs with Connection identifier.

Use Case 8.3.9: (UC_24) NMS requests EMS for subordinate MultiLayer Routing Area IDs involved in the route of a Connection

Description	1) The NMS sends the request get all subordinate MLRA identifiers with a Connection Id to the EMS. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If a specified object is not of the right type (ie., an MLRA or a Connection), an Invalid Input exception is raised. c) If a specified object (i.e., an MLRA or a Connection) is not known to the EMS, an Entity Not Found exception is raised. d) If the EMS does not control the node associated with the Connection originating point, an Unable To Comply exception is raised. e) If the EMS is unable to comply because the Connection State is in a Searching state, a Not In Valid State exception is raised. 3) If the request is valid the EMS returns the MLRA identifiers to the NMS.
Ends when	In case of success: The NMS receives the list of MLRA identifiers. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: The NMS has received the requested information from the EMS. In case of failure: Nothing has changed in the System, i.e., the NMS has not received the requested information.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 391}, {Requirement II. 405}, {Requirement II. 408}, {Requirement II. 410}

8.3.10 NMS requests EMS for the route details of a specified Call within a specified routing area

Use Case 8.3.10: (UC_25) NMS requests EMS for the route details of a specified Call within a specified routing area

Use Case Name	NMS requests EMS for the route details of a specified Call within a specified routing area.

Summary	This use case has only been validated in a 3-level hierarchy, i.e. where the top level MLRA contains subordinate MLRAs, and such subordinate MLRAs contain Routing Nodes.
	An NMS shall provide an EMS with the following:
	• ID of a Call.
	• Name of the Call. (optional)
	• MLRA Name.
	• SNPP/SNP name of one end of the Connection (optional).
	When a Call is supported by more than one Connection, the SNPP/SNP name is necessary to identify a specific Connection.
	• A filter parameter to indicate whether the response should include SNPPLink information.
	• A filter parameter to indicate whether the response should include:
	    • Home/intended route
	    • Actual route
	    • Both home/intended and actual routes
	This use case describes how the EMS returns to the NMS:
	• The Connection(s) of the referenced Call that spans the referenced MLRA.
	• Route details for the Connection(s) where they are available (i.e. where the referenced MLRA is one level above the Routing Nodes).
	    • The route details in terms of Routing Node Connections
	• Optionally:
	    • The edge MLSNPP Links to the MLRA through which the Connection(s) entered/exited the MLRA. The MLSNPP Links reference one local and one remote MLSNPP where the name of the remote MLSNPP includes the name of the neighbor MLRA.
	    • The internal MLSNPP Links of the MLRA for the case where route details are provided (i.e. where the referenced MLRA is one level above the Routing Nodes).
	The Routing Node Connections have the following information about their end points:
	• Actual route: Minimally the SNPP and SNP name and the PTP and CTP name and also the TNA name where specified on the endpoint.
	• Home/Intended route: Minimally the SNPP name or the TNA name.

Use Case 8.3.10: (UC_25) NMS requests EMS for the route details of a specified Call within a specified routing area

	Note: • There may be no home/intended route in the case where none was provisioned. • There may be no actual route where the Connection is in the Searching state or the actual route traverses a different MLRA to the home/intended route. • There will be no route information where the input MLRA is a Routing Node. • There will be more than one Connection and route descriptor if more than one Connection of the same Call passes through the routing area referenced and is within the scope of the SNPP/SNP (if an SNPP/SNP is specified). • If the request specifies "both home/intended and actual" and both the home/intended and the actual routes pass through the same point (e.g. if the actual route is following the home route - the normal case) there will be two Connections reported, i.e. one to represent the home/intended route and one the actual route. In the case where the actual route follows the home/intended route, the two Connections will be identical with respect to the points identified. • In MTNM 3.5 the Routing Node boundary coincides with the boundary of the NE fabric (the Routing Node may represent a subset of the capacity of the fabric but will be bounded by resources that are represented by MTNM TP and NOT abstract points). As a consequence the Routing Node Connection coincides with the cross connection. • For an SNCP Connection, the home/intended route in the response will always include both main and spare legs of the SNCP protection. The actual route may specify only one leg, if only one leg is currently active in the network. • The route is described in full across the MLRA regardless of sharing of same Routing Node Connections among distinct routes of the same Call. • The route details may not be available if the specified MLRA is not local to (managed directly by) the EMS to which the request is sent.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to retrieve all the route details for a specified Call.

Use Case 8.3.10: (UC_25) NMS requests EMS for the route details of a specified Call within a specified routing area

Description	1) The NMS sends the request to get the route details of a specified Call to the EMS. 2) The EMS validates the request: a) If the syntax is in error, then an Invalid Input exception is raised. b) If a specified object is not of the right type (i.e., an MLRA, a Call, an SNPP/SNP), then an Invalid Input exception is raised. c) If a specified object (i.e., an MLRA, a Call, an SNPP/SNP) is not known to the EMS, then an Entity Not Found exception is raised. d) If the EMS does not control the node associated with the Connection originating point, then an Unable To Comply exception is raised. 3) If the request is valid the EMS returns the Connection information to the NMS. a) If there are no connections that match the input criteria for the specified Call an empty list is returned b) If there are one or more Connections then the list of Connections is returned. c) For each Connection, if the Connection is in the Searching state then the Connection is returned with no route details and with the state set to Searching. Otherwise the connection is returned with route details.
Ends when	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: The NMS has received the requested information from the EMS. In case of failure: Nothing has changed in the System, i.e., the NMS has not received the requested information.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 390}, {Requirement II. 405}, {Requirement II. 407}, {Requirement II. 408}, {Requirement II. 409}, {Requirement II. 410}

8.3.11 NMS requests EMS for the ID of each Call supported by a specified TP/SNPP/TNA

Use Case 8.3.11: (UC_26) NMS requests EMS for the ID of each Call supported by a specified TP/SNPP/TNA

<table>
<tr><td>Use Case Name</td><td>NMS requests EMS for the ID or Name of each Call supported by a specified TP/SNPP/TNA</td></tr>
<tr><td>Summary</td><td>This use case describes how the NMS can retrieve the ID of all Calls supported by any specified TP, SNPP or TNA Name.

An NMS shall provide an EMS with the following:
• A TP name
• An SNPP ID
• A TNA Name
• A Group TNA Name

The EMS shall respond with:
• The ID of each Call that has its actual route supported by the specified input.
• The ID of each Call that has its home/intended route supported by the specified input.
• For each ID an parameter identifying whether it refers to a “home” or “actual” route. This parameter may be null if the information is not available.

Note:

• This case only returns Call IDs and not other details. It is only the Call ID that will be available to the Control Plane at an intermediate points along a Connection that supports the Call. An EMS that manages an intermediate point but not the originating point of the Call will only have access to the Call ID. The Control Plane will not be aware of any other details for Calls that originate outside the routing area to which the resources referenced belongs.
• This use case could be used during a planned engineering works impact analysis. This operation can be used in conjunction with information previously retrieved using (UC_7) NMS discovers Calls and top level Connections (and retained by the NMS), and kept up to date using (UC_8) EMS notifies of a new Call and its Connection(s), to provide the operator with full Call details and allow further enquiries. This could also be used in conjunction with other operations to move traffic off a PTP or SNPP.
• The key input here is a TP (primarily PTP - consider the maintenance scenario), however, SNPP and TNA Name have also been provided as there may be unforseen Control Plane operational scenarios that benefit.
• The only information guaranteed to be present relates to Calls whose actual routes pass through the specified input. In some cases where the Call is established it may have a currently inactive home route may also be known and therefore if a point that this home route passes through is referenced the Call name could be returned.
• For this operation to be successful the request must be sent to the EMS that controls the Control Plane and has access to the named TP/SNPP/TNA Name. For the TP case, this is the EMS that was able to respond with the SNPP-TP relationship in (UC_1) EMS notifies of changes in network capacity (increases/decreases in capacity).</td></tr>
</table>

Use Case 8.3.11: (UC_26) NMS requests EMS for the ID of each Call supported by a specified TP/SNPP/TNA

Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to retrieve the ID of each Call supported by a specified TP/SNPP/TNA name
Description	1) The NMS sends the request to discover the ID of each Call supported by a specified TP/SNPP/TNA Name to the EMS. 2) The EMS validates the request: a) If the syntax is in error, then an Invalid Input exception is raised. b) If a specified object is not of the right type (i.e., a TP or an SNPP), then an Invalid Input exception is raised c) If the specified TP/SNPP/TNA is not known to the EMS, then an Entity Not Found exception is raised. 3) If the request is valid the EMS returns the list of Call IDs to the NMS.
Ends when	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: The NMS has received the requested information from the EMS. In case of failure: The EMS has not received the requested information.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 385}

8.3.12 NMS request EMS for the list of MLSNPPs

Use Case 8.3.12: (UC_27) NMS request EMS for the list of MLSNPPs

Use Case Name	NMS request EMS for the list of MLSNPPs
Summary	This use case describes how the NMS requests from the EMS the MLSNPPs. The EMS provides the information for the MLSNPP matching the scope of the request The NMS may specify one and only one of the following criteria: • **All**: the EMS shall respond with the attributes of all MLSNPPs. • **MLSNPP name**: the EMS shall respond with the attributes of the MLSNPP that was specified by the NMS. • **MultiLayer Routing Area (MLRA) name**: the EMS shall respond with the attributes of all MLSNPPs that are named by the specified MLRA. • **TNA name**: the EMS shall respond with the attributes of the MLSNPP that match the specified TNA. • **PTP or CTP name**: the EMS shall respond with the attributes of all the MLSNPPs that are totally or even partially supported by the specified TP. Note: The interface is specified to only expose SNPPs that are pools of SNPs that represent CTPs of type TCP through the MLSNPP, the SNPPs that group SNPs that represent G.805 CPs are retrieved as a component of the MLSNPPLink (covered by Use Case 8.2.4: (UC_5) NMS discovers MLSNPP Links)
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to retrieve the MLSNPPs.

Use Case 8.3.12: (UC_27) NMS request EMS for the list of MLSNPPs

Description	1) The NMS sends a request to retrieve the MLSNPP information for the MLSNPPs that match the specified criteria. 2) The EMS validates the request: a) If the syntax is in error, then an Invalid Input exception is raised. b) If a specified object is not of the right type (i.e., an MLRA, an MLSNPP, a PTP, an FTP, or a CTP), then an Invalid Input exception is raised. c) If a specified object (i.e., an MLRA, an MLSNPP, a PTP, an FTP, or a CTP) does not exist, then an Entity Not Found exception is raised. d) If a specified TNA name does not exist, then an Entity Not Found exception is raised. e) If a specified object is not under the control of the Control Plane then an Unable To Comply exception is raised. 3) If the request is valid the EMS returns a list of MLSNPPs matching the specified input of the request. a) If no MLSNPPs match the criteria then the EMS will return an empty list. Note: The specified reference objects (e.g., MLSNPP names, MLRA name) could belong to different levels of the routing hierarchy. Further more, since multi-layer approach is used in modeling, the specified reference objects could belong to difference transport layers.
Ends when	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: The NMS knows all the MLSNPP, including their properties, within the scope of the retrieve request. In case of failure: The NMS does not know all the MLSNPPs, within the scope of the retrieve request.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 367}, {Requirement II. 368}, {Requirement II. 369}, {Requirement II. 370}, {Requirement II. 371}

9 Connectionless Management Use Cases

9.1 Actor-System Context Diagram

Refer to Section 7.1.

9.1.1 Exceptions

Refer to Section 7.1.1.

9.2 Termination Point Management Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 9.2.1:	NMS creates a Floating Termination Point	{Requirement II. 293}, {Requirement II. 299}
Use Case 9.2.2:	NMS deletes a Floating Termination Point	{Requirement II. 294}

9.2.1 NMS creates a FloatingTermination Point

Use Case 9.2.1: NMS creates a Floating Termination Point

Name	NMS creates a Floating Termination Point
Summary	This operation allows an NMS to create a Floating Termination Point (FTP). The EMS creates the requested FTP consistently with its implementation and the NMS specifications. If the EMS cannot create the FTP as specified, an appropriate exception is raised. The NMS specifies the FTP through choosing a location within the network element (e.g., shelf and card), a list of layer rates and related transmission parameters. After the EMS validates the input data, and if it can satisfy the input constraints (e.g., specific bandwidth as a virtually concatenated layer rate), it proceeds with implementing the FTP (and its contained CTPs, if applicable) and returns the name of the new FTP and all its attributes to the NMS. If the EMS fails to validate the input, it raises an appropriate exception. If transmission parameters are specified for the contained CTPs which are created (where applicable), the EMS will apply them after the creation of the FTP. The alarm reporting on the FTPs and the contained CTPs may be turned on by the EMS, unless otherwise specified via the alarm reporting transmission parameter. An object creation notification is sent to notify the NMS(s) about the existence of the new FTP (note that no notification is sent related to the contained CTPs in case virtual concatenation applies).
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS sends a request to the EMS to create an FTP.

Use Case 9.2.1: NMS creates a Floating Termination Point

Description	1) The NMS sends the request to the EMS to create an FTP with the provided parameters. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If explicit creation of the requested FTP not supported, an Unable To Comply exception is raised. c) If the specifed equipment object is not known to the EMS, an Entity Not Found exception is raised. d) If the EMS supports name specification by the NMS, and if a name was specified, and if there is already an FTP with the specified name, an Object In Use exception is raised. e) If uniqueness of the user label is required, the EMS checks the user label for uniqueness; i.e., if an FTP object with the same user label exists already then a User Label In Use exception is raised. f) If not all required resources are available (e.g., enough overall back-plane bandwidth, enough usable timeslots, etc.), a Capacity Exceeded exception is raised. g) If the input parameters are incompatible, an Unable To Comply exception is raised. 3) If the request is valid: a) The EMS creates the FTP and its contained CTPs if applicable (e.g., in case the FTP layer rate implies inverse multiplexing). CTP instances are either chosen by the EMS, or created according to NMS specification. b) If contained CTPs were created, and if transmission parameters were specified for them, the EMS applies the parameters. 4) The EMS replies with a success indication. a) The EMS returns the name of the new FTP. If the NMS specified a name, and if the EMS supports naming by NMS, the NMS specified name is returned, else an EMS created name is returned. 5) The EMS sends the appropriate object creation notification to the notification service. No notification is sent for the contained CTPs. (The CTPs can be retrieved by explicit request.)
Ends When	In case of success: The EMS returns the created FTP name and all its attributes to the NMS. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-Conditions	In case of success: 1) The FTP has been created. 2) The contained fragment CTPs have been created (if applicable). In case of failure: No change occurred to the system, i.e., the FTP has not been created.
Exceptions	Refer to Table 7.1

Use Case 9.2.1: NMS creates a Floating Termination Point

Traceability	{Requirement II. 293}, {Requirement II. 299}

9.2.2 NMS deletes a FloatingTermination Point

Use Case 9.2.2: NMS deletes a Floating Termination Point

Name	NMS deletes a Floating Termination Point
Summary	This operation allows the NMS to delete a Floating Termination Point (FTP) that already exists. The EMS is required to verify that the FTP or the contained fragment CTPs are not cross-connected. The deletion request will fail • if the FTP is a Call endpoint, or • if the FTP or any of its contained server CTPs is an SNC endpoint, or • if the CPTP cannot be explicitly deleted (e.g., was automatically created by the EMS). • If the FTP is a CPTP which is assigned or has the fdEdge role.
Actor(s)	NMS
Pre-Conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins When	The NMS sends the request to the EMS to delete an FTP.
Description	1) The NMS sends the request to the EMS to delete an FTP. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified FTP object is not known to the EMS, an Entity Not Found exception is raised. c) If the specified FTP object or any of its contained CTP objects are cross-connected, an Unable To Comply exception is raised. d) If the specified FTP object terminates a Call, an Unable To Comply exception is raised. e) If the specified FTP is a CPTP, and it is not in the unassigned role, an Object In Use exception is raised. f) If the specified FTP cannot be deleted (e.g., was not automatically created by the EMS), an Unable To Comply exception is raised. g) If there is any incompatibility among the input parameters (e.g., layered transmission parameters consistent with the list of layer rates), an Unable To Comply exception is raised. 3) If the request is valid the EMS deletes the FTP and any contained CTPs. 4) The EMS replies with a success indication. 5) The EMS sends an FTP object deletion notification to the notification service.
Ends When	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.

Use Case 9.2.2: NMS deletes a Floating Termination Point

Post-Conditions	In case of success: The FTP and its contained CTPs have been deleted.. In case of failure: No change occurred to the system, (i.e., the FTP or any of its contained CTPs have not been deleted or modified).
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 294}

9.3 Flow Domain Management Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 9.3.1:	NMS creates a Flow Domain	{Requirement II. 321}, {Requirement II. 322}
Use Case 9.3.2:	NMS deletes a Flow Domain	{Requirement II. 325}
Use Case 9.3.3:	NMS modifies a Flow Domain	{Requirement II. 323}, {Requirement II. 324}
Use Case 9.3.4:	NMS associates Matrix Flow Domain(s) to a Flow Domain	{Requirement II. 328}
Use Case 9.3.5:	NMS dissociates Matrix Flow Domain(s) to a Flow Domain	{Requirement II. 329}
Use Case 9.3.6:	NMS associates Connectionless Port Termination Point(s) to a Flow Domain	{Requirement II. 326}
Use Case 9.3.7:	NMS dissociates FD Edge Connectionless Port Termination Point(s) from a Flow Domain	{Requirement II. 327}

Figure 9.1, taken from ITU-T Y.1730, shows an example of a flow domain (provider network) which is made up of several MFDs. In this example all Edge CPTPs belong to some MFD.

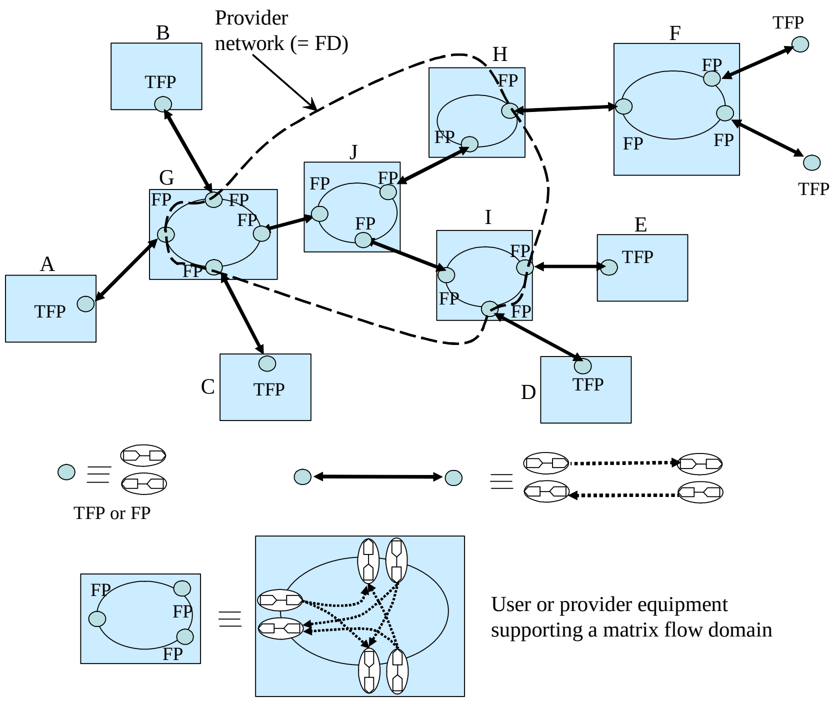


Figure 9.1: Multipoint-to-multipoint ETH reference model (Figure 8/Y.1730)

9.3.1 NMS creates a Flow Domain

Use Case 9.3.1: NMS creates a Flow Domain

Use Case Name	NMS creates a Flow Domain
Summary	This operation allows an NMS to create a Flow Domain (FD). The NMS can also associate Matrix Flow Domains (MFDs) and Connectionless Port Termination Points (CPTPs) to this new FD. The association of the CPTPs is done on a best effort basis.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to create a Flow Domain.
Description	1) The NMS sends the request to the EMS to create an FD. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If a FD object with the name specified already exists, then an Object In Use exception is raised. c) If uniqueness of the user label is required, the EMS checks the user label for uniqueness; i.e., if a FD object with the same user label exists already, then a User Label In Use exception is raised. d) If one of the specified resources (i.e., MFDs and/or CPTPs) does not exist an Entity Not Found exception is raised. e) If any of the MFDs to be associated is already associated to another FD, an Object In Use exception is raised. f) If any of the MFDs to be associated could not be associated to the FD, no MFD is associated and an Unable To Comply exception is raised. g) If a CPTP is not already assigned to one of the provided MFDs, an Unable To Comply exception is raised. 3) If the request is valid the EMS creates the FD as requested. 4) The EMS associates the requested CPTPs to the new FD (i.e., the “Port TP role state” attribute of the CPTPs is set to “fdEdge”). All names of the CPTPs that could not be associated shall be returned in the reply. 5) The EMS replies with a success indication 6) The EMS shall send an FD object creation notification to the notification service.
Ends when	In case of success: 1) The NMS receives an indication of the success of the operation. 2) The EMS returns the distinguishing information of the created FD. 3) In case not all requested CPTPs could be associated, the EMS returns the list of these CPTPs. In case of failure: The NMS receives an exception as an indication of the failure of the request.

Use Case 9.3.1: NMS creates a Flow Domain

Post-conditions	In case of success: 1) The FD has been created. 2) The MFDs have been associated to the new FD. 3) The CPTPs have been associated to the FD (i.e., the Port TP role state attribute of the CPTPs has been set to "fdEdge") or have been returned in the reply. In case of failure: Nothing has changed in the System, i.e., the FD has not been created and the appropriate exception is reported to the NMS.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 321}, {Requirement II. 322}

9.3.2 NMS deletes a Flow Domain

Use Case 9.3.2: NMS deletes a Flow Domain

Use Case Name	NMS deletes a Flow Domain
Summary	This operation allows an NMS to delete an existing Flow Domain (FD). The EMS is required to verify that no Flow Domain Fragments (FDFrs) exist within the FD. The operation dissociates the FD Edge Connectionless Port Termination Points (CPTPs), dissociates the Matrix Flow Domains (MFDs) and deletes the FD.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to delete a Flow Domain.
Description	1) The NMS sends the request to the EMS to delete an FD. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the FD object is not known to the EMS, an Entity Not Found exception is raised. c) If any FDFr is contained in the FD, an Object In Use exception is raised. d) If any of the MFD can not be dissociated from the specified FD, no MFD is dissociated and an Unable To Comply exception is raised. e) If any FD Edge CPTPs can not be dissociated from the FD (i.e., the Port TP role state attribute of the CPTPs was set to “assigned”) no FD Edge CPTP are dissociated and an Unable To Comply exception is raised. 3) If the request is valid the EMS deletes the Flow Domain. 4) The EMS replies with a success indication. 5) The EMS shall sends an FD object deletion notification to the notification service.
Ends when	In case of success: The NMS receives an indication of success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: 1) The FD Edge CPTPs associated to the FD have been de-associated (i.e., the Port TP role state attribute of the CPTPs has been set to "assigned"). 2) The MFDs associated to the FD have been de-associated. 3) The FD has been deleted. In case of failure: Nothing has changed in the System, i.e., the FD has not been deleted and the appropriate exception is reported to the NMS.
Exceptions	Refer to Table 7.1.

Use Case 9.3.2: NMS deletes a Flow Domain

Traceability	{Requirement II. 325}

9.3.3 NMS modifies a Flow Domain

Use Case 9.3.3: NMS modifies a Flow Domain

Use Case Name	NMS modifies a Flow Domain
Summary	This operation allows an NMS to modify a Flow Domain (FD) that already exists. The NMS can modify the user label, the owner, the network access domain, the connectionless layered parameters or the additional information of an existing FD.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to modify the attributes of a Flow Domain.
Description	1) NMS sends the request to the EMS to modify a FD. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified FD is not known to the EMS, an Entity Not Found exception is raised. c) If uniqueness of the user label is required, the EMS checks the user label for uniqueness; i.e., if a FD object with the same user label exists already, then a User Label In Use exception is raised. d) If the EMS can not satisfy any attribute that needs to be modified, an an Unable To Comply exception is raised. 3) If the request is valid: a) If the FD already has all the required information (i.e., no changes are required), the EMS shall not send any notifications to the notifcation service. b) Otherwise the EMS modifies the attributes of the FD as requested. 4) The EMS replies with a success indication 5) If the EMS made a change then appropriate notifications shall be sent to the notification service.
Ends when	In case of success: 1) The NMS receives an indication of the success of the operation. 2) The EMS returns the distinguishing information of the modified FD. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: The FD has been modified as requested. In case of failure: Nothing has changed in the System, i.e., the FD has not been modified.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 323}, {Requirement II. 324}

9.3.4 NMS associates Matrix Flow Domain(s) to a Flow Domain

Use Case 9.3.4: NMS associates Matrix Flow Domain(s) to a Flow Domain

Use Case Name	NMS associates Matrix Flow Domain(s) to a Flow Domain
Summary	This operation allows an NMS to associate one or more Matrix Flow Domain(s) (MFD(s)) to an existing Flow Domain (FD). The EMS is required to validate the data provided by the NMS, and associates the requested MFD(s) to the specified FD. If the EMS cannot associate the MFD as specified, an appropriate exception is raised. Note that best effort is not supported. The EMS verifies the server layer connectivity between the associated MFD(s), the “FD Connectivity State” attribute of the FD is modified accordingly; fully connected, not fully connected, unknown.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to associate one or more MFDs to a FD.
Description	1) The NMS sends the request to the EMS to associate additional MFD(s) to an existing FD. 2) The EMS validates the request: a) If the syntax is in errror, an Invalid Input exception is raised. b) If the specified FD is not known to the EMS, an Entity Not Found exception is raised. c) If one or more of the specified MFDs is not known to the EMS, an Entity Not Found exception is raised. d) If one or more of the specified MFDs are already associated to another FD, an Object In Use exception is raised. e) If any of the MFDs could not be associated, no MFD is associated and an Unable To Comply exception is raised. 3) If the request is valid the EMS associates the specified MFD(s) to the FD. 4) The EMS verifies the server layer connectivity between the associated MFD(s), the "FD Connectivity State" attribute is modified accordingly; fully connected, not fully connected, unknown. 5) The EMS replies with a success indication. 6) The EMS sends an attribute value change notification to the notification service; i.e., via a notification of the FD's "Matrix Flow Domains" attribute. Note that the notification includes the complete list of MFD names that are associated to the FD. 7) If the "FD Connectivity State" attribute is modified, the EMS sends a state change notification to the notification service.
Ends when	In case of success: The EMS returns the distinguishing information of the modified FD. In case of failure: The NMS receives an exception as an indication of the failure of the request.

Use Case 9.3.4: NMS associates Matrix Flow Domain(s) to a Flow Domain

Post-conditions	In case of success: The FD has been modified as requested. In case of failure: Nothing has changed in the System, i.e., the FD has not been modified.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 328}

9.3.5 NMS dissociates Matrix Flow Domain(s) to a Flow Domain

Use Case 9.3.5: NMS dissociates Matrix Flow Domain(s) to a Flow Domain

Use Case Name	NMS dissociates Matrix Flow Domain(s) to a Flow Domain
Summary	This operation allows an NMS to dissociate one or more Matrix Flow Domain(s) (MFD(s)) from an existing Flow Domain (FD). The EMS is required to validate the data provided by the NMS, and dissociates the requested MFD(s) from the specified FD. If the EMS cannot dissociate the MFD(s) as specified, an appropriate exception is raised. Note that best effort is not supported. The operation also dissociates the FD Edge Connectionless Port Termination Points (CPTPs) which are associated to the MFDs to be dissociated. Following the dissociation, the EMS verifies the server layer connectivity between the associated MFD(s), the “FD Connectivity State” attribute of the FD is modified accordingly; “fully connected”, “not fully connected”, or “unknown”.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to dissociate one or more MFD(s) to a FD.
Description	1) The NMS sends the request to the EMS to dissociate MFD(s) from an existing FD. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified FD is not known to the EMS, an Entity Not Found exception is raised. c) If one or more of the specified MFDs was not previously associated to the FD, an Unable To Comply exception is raised. d) If one or more of the specified MFDs has traffic (i.e., FDFr uses this MFD), an Object In Use exception is raised. e) If any FD Edge CPTPs can not be dissociated from the MFDs that are being dissociated (i.e., the Port TP role state attribute of the CPTPs was set to “assigned”) no FD Edge CPTP are dissociated and an Unable To Comply exception is raised. f) If any of the MFDs could not be dissociated, no MFD is dissociated and an Unable To Comply exception is raised. 3) If the request is valid the EMS dissociates the specified MFD(s) from the FD. 4) The EMS replies with a success indication. 5) The EMS sends an attribute value change notification to the notification service; i.e., via a notification of the FD's "Matrix Flow Domains" attribute. Note that the notification includes the complete list of MFD names that are associated to the FD. 6) The EMS verifies the server layer connectivity between the associated MFD(s), the “FD Connectivity State” attribute is modified accordingly; fully connected, not fully connected, unknown. 7) If the "FD Connectivity State" attribute is modified, the EMS sends a state change notification to the notification service.

Use Case 9.3.5: NMS dissociates Matrix Flow Domain(s) to a Flow Domain

Ends when	In case of success: The NMS receives an indication of success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: 1) The requested MFD(s) have been dissociated from the FD. 2) The corresponding FD Edge CPTPs have been dissociated from the FD. In case of failure: Nothing has changed in the System, i.e., no change in the MFD association to the FD.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 329}

9.3.6 NMS associates Connectionless Port Termination Point(s) to a Flow Domain

Use Case 9.3.6: NMS associates Connectionless Port Termination Point(s) to a Flow Domain

Use Case Name	NMS associates Connectionless Port Termination Point(s) to a Flow Domain
Summary	This operation allows an NMS to associate one or more Connectionless Port Termination Points (CPTPs) to an existing Flow Domain (FD). The EMS is required to validate the data provided by the NMS, and associate the requested CPTP(s) to the specified FD. The operation is best effort, i.e., the list of CPTPs that could not be associated will be returned in the reply.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to associate CPTP(s) to a FD.
Description	1) The NMS sends the request to the EMS to associate additional CPTP(s) to an existing FD. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified FD is not known to the EMS, an Entity Not Found exception is raised. c) If one or more of the specified CPTPs is not known to the EMS, an Entity Not Found exception is raised. d) If one or more of the specified CPTPs is not in the “assigned” CPTP “Port TP role state”, an Unable To Comply exception is raised. e) If one or more of the specified CPTPs is assigned to an MFD that is not associated to the FD, an Unable To Comply exception is raised. 3) If the request is valid the EMS associates the specified CPTP(s) to the FD (i.e., the “Port TP role state” attribute of the CPTPs is set to “FD Edge”). All CPTPs that could not be associated have to be returned in the reply. 4) The EMS replies with a success indication 5) The EMS sends an attribute value change notification to the notification service; i.e., a notification of the FD's “FD Edge CPTPs” attribute. Note that the notification includes the complete list of FD Edge CPTP names that are associated to the FD.
Ends when	In case of success: The NMS receives an indication of success of the request and a list of all CPTPs that could not be associated. In case of failure: The NMS receives an exception as an indication of the failure of the request.

Use Case 9.3.6: NMS associates Connectionless Port Termination Point(s) to a Flow Domain

Post-conditions	In case of success: The requested CPTPs (except the ones returned in the list of not associated CPTPs) have been associated as FD Edge CPTPs to the FD. In case of failure: Nothing has changed in the System, i.e., no change in the CPTP association to the FD.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 326}

9.3.7 NMS dissociates FD Edge Connectionless Port Termination Point(s) from a Flow Domain

Use Case 9.3.7: NMS dissociates FD Edge Connectionless Port Termination Point(s) from a Flow Domain

Use Case Name	NMS dissociates FD Edge Connectionless Port Termination Point(s) from a Flow Domain
Summary	This operation allows an NMS to dissociate one or more FD Edge Connectionless Port Termination Point(s) (CPTP(s)) from an existing Flow Domain (FD). The EMS is required to validate the data provided by the NMS, and dissociates the requested FD Edge CPTP(s) from the specified FD. The operation will be rejected if one or more of the specified FD Edge CPTPs carries traffic. If the EMS cannot dissociate the FD Edge CPTP(s) as specified, an appropriate exception is raised. Note that best effort is not supported.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to dissociate FD Edge CPTP(s) from an FD.
Description	1) The NMS sends the request to the EMS to disassociate FD Edge CPTPs from an existing FD. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified FD is not known to the EMS, an Entity Not Found exception is raised. c) If one or more of the specified FD Edge CPTPs is not known to the EMS, an Entity Not Found exception is raised. d) If one or more of the specified FD Edge CPTPs is not in the "FD Edge" CPTP "Port TP role state", an Unable To Comply exception is raised. e) If one or more of the specified FD Edge CPTPs is assigned to an MFD that is not associated to the FD, an Unable To Comply exception is raised. f) If one or more of the specified FD Edge CPTPs carries traffic (i.e., an FDFr is provisioned on this CPTP), an Object In Use exception is raised. g) If any of the CPTPs could not be dissociated, no CPTP is dissociated and an Unable To Comply exception is raised. 3) If the request is valid the EMS dissociates the specified FD Edge CPTP(s) from the FD (i.e., the "Port TP role state" attribute of the CPTPs is set to "assigned"). 4) The EMS replies with a success indication. 5) The EMS send an attribute value change notification to the notification service; i.e., via a notification of the FD's "FD Edge CPTPs" attribute. Note that the notification includes the complete list of FD Edge CPTP names that are associated to the FD.
Ends when	In case of success: The NMS receives an indication of success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.

Use Case 9.3.7: NMS dissociates FD Edge Connectionless Port Termination Point(s) from a Flow Domain

Post-conditions	In case of success: The requested FD Edge CPTPs have been disassociated from the FD. In case of failure: Nothing has changed in the System, i.e., no change in the FD Edge CPTP association to the FD.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 327}

9.4 Matrix Flow Domain Management Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 9.4.1:	NMS creates a Matrix Flow Domain (MFD)	{Requirement II. 305}, {Requirement II. 306}
Use Case 9.4.2:	NMS deletes a Matrix Flow Domain (MFD)	{Requirement II. 309}
Use Case 9.4.3:	NMS modifies a Matrix Flow Domain (MFD)	{Requirement II. 307}, {Requirement II. 308}
Use Case 9.4.4:	NMS assigns Connectionless Port Termination Point(s) to a Matrix Flow Domain	{Requirement II. 310}
Use Case 9.4.5:	NMS un-assigns Connectionless Port Termination Point(s) from a Matrix Flow Domain	{Requirement II. 311}

9.4.1 NMS creates a Matrix Flow Domain (MFD)

Use Case 9.4.1: NMS creates a Matrix Flow Domain (MFD)

Use Case Name	NMS creates a Matrix Flow Domain (MFD)
Summary	This operation allows an NMS to create a Matrix Flow Domain (MFD). The EMS creates the requested MFD consistent with its implementation and the NMS specifications. If the EMS cannot create the MFD as specified, an appropriate exception is raised. The NMS specifies the MFD through a list of FD Edge CPTPs and/or internal TPs that the MFD must contain, and a set of attributes that the MFD must satisfy. After the EMS validates the input data, and if it can satisfy the input constraints, it proceeds with implementing the MFD. If the EMS fails to validate the input, it raises an appropriate exception. The EMS implements the Matrix Flow Domain as specified by the NMS. It also assigns all the specified TPs to the MFD.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to create a MFD.

Use Case 9.4.1: NMS creates a Matrix Flow Domain (MFD)

Description	1) The NMS sends the request to the EMS to create an MFD with the provided parameters. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If uniqueness of the user label is required, the EMS checks the user label for uniqueness; i.e., if an MFD object with the same user label exists already then a User Label In Use exception is raised. c) If one of the specified TPs is not known to the EMS, an Entity Not Found exception is raised. d) If at least one of the MFD parameters could not be set, an Unable To Comply exception is raised. e) If any of the specified TPs is already in use by another MFD, an Object In Use exception is raised. 3) If the request is valid: a) the EMS creates the MFD. b) The EMS assigns the requested CPTPs to the MFD. 4) The EMS replies with a success indication 5) The EMS sends object creation notifications to the notification service.
Ends when	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: 1) The MFD has been created. 2) The requested TPs have been associated with the MFD. 3) The provided parameters have been set for the MFD. 4) The requested FD Edge CPTPs have been associated with the FD. In case of failure: Nothing has changed in the System, i.e., the MFD has not been created.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 305}, {Requirement II. 306}

9.4.2 NMS deletes a Matrix Flow Domain (MFD)

Use Case 9.4.2: NMS deletes a Matrix Flow Domain (MFD)

Use Case Name	NMS deletes a Matrix Flow Domain (MFD)
Summary	This operation allows the NMS to delete a Matrix Flow Domain (MFD) that already exists. The EMS is required to validate the MFD, verify that the MFD is not associated with a Flow Domain.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to delete an MFD.
Description	1) The NMS sends the request to the EMS to delete an MFD. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified MFD object is not known to the EMS, an Entity Not Found exception is raised. c) The MFD to be deleted must not be associated with a Flow Domain. If it is still associated, an Object In Use exception is raised. d) The MFD to be deleted must not be “fixed”. If it is fixed, an Unable To Comply exception is raised. 3) If the request is valid: a) The EMS deletes the MFD. b) The EMS releases all assigned TPs from the MFD. 4) The EMS replies with a success indication. 5) The EMS sends an object deletion notification to the notification service.
Ends when	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: The assigned TPs have become unassigned and the MFD has been deleted. In case of failure: Nothing has changed in the System, i.e., the MFD has not been deleted and no TPs have been modified.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 309}

9.4.3 NMS modifies a Matrix Flow Domain (MFD)

Use Case 9.4.3: NMS modifies a Matrix Flow Domain (MFD)

Use Case Name	NMS modifies a Matrix Flow Domain (MFD)
Summary	This operation allows the NMS to modify a Matrix Flow Domain (MFD) that already exists. The NMS can modify the user label, the owner, the attributes (e.g., Spanning Tree Protocol (STP) parameters) or the additional information of an existing Matrix Flow Domain.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to modify an MFD.
Description	1) The NMS sends a request to the EMS to modify an MFD. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified MFD is not known to the EMS, an Entity Not Found exception is raised. c) If uniqueness of the user label is required, the EMS checks the user label for uniqueness; i.e., if a MFD object with the same user label exists already, then a User Label In Use exception is raised. d) If the EMS can not satisfy any attribute that needs to be modified, an an Unable To Comply exception is raised. e) If the MFD already has the required information, the EMS replies with a success indication but no notification is generated. 3) If the request is valid: a) If the MFD already has all the required information (i.e., no changes are required), the EMS shall not send any notifications to the notifcation service. b) Otherwise the EMS modifies the attributes of the MFD as requested. 4) The EMS replies with a success indication. 5) If the EMS made a change then appropriate notifications shall be sent to the notification service.
Ends when	In case of success: 1) The NMS receives an indication of the success of the operation. 2) The EMS returns the distinguishing information of the modified MFD together with the list of attributes which could not be modified. In case of failure: The NMS receives an exception as an indication of the failure of the request.

Use Case 9.4.3: NMS modifies a Matrix Flow Domain (MFD)

Post-conditions	In case of success: The MFD has been modified as requested. In case of partial success: The MFD has been modified but not all requested modifications have been completed. In case of failure: Nothing has changed in the System, i.e., the MFD has not been modified.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 307}, {Requirement II. 308}

9.4.4 NMS assigns Connectionless Port Termination Point(s) to a Matrix Flow Domain

Use Case 9.4.4: NMS assigns Connectionless Port Termination Point(s) to a Matrix Flow Domain

Use Case Name	NMS assigns Connectionless Port Termination Point(s) to a Matrix Flow Domain
Summary	This operation allows an NMS to assign one or more Connectionless Port Termination Points (CPTPs) to an existing Matrix Flow Domain (MFD). The EMS is required to validate the data provided by the NMS, and assign the requested CPTP(s) to the specified MFD. Note that best effort is not supported.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to assign a list of CPTP(s) to an MFD
Description	1) The NMS sends a request to the EMS to assign a list of CPTP(s) to an existing MFD. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified MFD is not known to the EMS, an Entity Not Found exception is raised. c) If the specified MFD is fixed, an Unable To Comply exception is raised. d) If one or more of the specified CPTPs is not known to the EMS, an Entity Not Found exception is raised. e) If one or more of the specified CPTPs is not a potential CPTP for this MFD (i.e., is not in the "unassigned" CPTP "Port TP role state" or is not on the same equipment or same rack with backplane connectivity), an Unable To Comply exception is raised. 3) If the request is valid the EMS assigns the specified CPTP(s) to the MFD (i.e., the “Port TP role state” attribute of the CPTPs is set to “assigned”). 4) The EMS replies with a success indication. 5) The EMS send an attributev value change notification to the notification service; i.e., via a notification of the MFD's “Assigned CPTPs” attribute. Note that the notification includes the complete list of CPTP names that are assigned to the MFD.
Ends when	In case of success: The NMS receives an indication of success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: The requested CPTPs have been assigned to the MFD. In case of failure: Nothing has changed in the System, i.e., no change in the CPTP assignment to the MFD.
Exceptions	Refer to Table 7.1.

Use Case 9.4.4: NMS assigns Connectionless Port Termination Point(s) to a Matrix Flow Domain

Traceability	{Requirement II. 310}

9.4.5 NMS un-assigns Connectionless Port Termination Point(s) from a Matrix Flow Domain

Use Case 9.4.5: NMS un-assigns Connectionless Port Termination Point(s) from a Matrix Flow Domain

Use Case Name	NMS un-assigns Connectionless Port Termination Point(s) from a Matrix Flow Domain
Summary	This operation allows an NMS to un-assign one or more Connectionless Port Termination Points (CPTPs) to an existing Matrix Flow Domain (MFD). The EMS is required to validate the data provided by the NMS, and to un-assign the requested CPTP(s) from the specified MFD. Note that best effort is not supported.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	TThe NMS sends a request to the EMS to un-assign a list of CPTP(s) from an MFD
Description	1) The NMS sends a request to the EMS to un-assign a list of CPTP(s) from an existing MFD. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified MFD is not known to the EMS an Entity Not Found exception is raised. c) If the specified MFD is fixed, an Unable To Comply exception is raised. d) If one or more of the specified CPTPs is not known to the EMS, an Entity Not Found exception is raised. e) If one or more of the specified CPTPs is in the “unassigned” “Port TP role state”, a Not In Valid State exception is raised. f) If one or more of the specified CPTPs is used by a Flow Domain Fragment, an Object In Use exception is raised. g) If one or more of the specified CPTPs is not assigned to the specified MFD, an Unable To Comply exception is raised. 3) If the request is valid the EMS un-assigns the specified CPTP(s) from the MFD (i.e., the "Port TP role state" attribute of the CPTPs is set to “unassigned”). If any of the CPTPs could not be un-assigned, no CPTP is un-assigned and an Unable To Comply exception is raised. 4) The EMS replies with a success indication. 5) The EMS sends the appropriate notifications to the notification service.
Ends when	In case of success: The NMS receives an indication of success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.

Use Case 9.4.5: NMS un-assigns Connectionless Port Termination Point(s) from a Matrix Flow Domain

Post-conditions	In case of success: The requested CPTPs have been assigned to the MFD. In case of failure: Nothing has changed in the System, i.e., no change in the CPTP assignment to the MFD.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 311}

9.5 Traffic Conditioning Profile Management Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 9.5.1:	NMS creates a Traffic Conditioning Profile	{Requirement II. 333}, {Requirement II. 334}
Use Case 9.5.2:	NMS deletes a Traffic Conditioning Profile	{Requirement II. 336}
Use Case 9.5.3:	NMS modifies a Traffic Conditioning Profile	{Requirement II. 334}, {Requirement II. 335}
Use Case 9.5.4:	NMS configures Traffic Mapping Table	{Requirement II. 337}

9.5.1 NMS creates a Traffic Conditioning Profile

Use Case 9.5.1: NMS creates a Traffic Conditioning Profile

Use Case Name	NMS creates a Traffic Conditioning Profile
Summary	This operation allows an NMS to create aTraffic Conditioning (TC) Profile. EMS validates the data provided by NMS, and creates the requested TC Profile in accordance with the NMS parameter list. If the EMS cannot create the TC Profile as specified, an appropriate exception is raised.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to EMS to create a TC Profile.
Description	1) The NMS sends a request to EMS to create a TC Profile. The NMS provides the TC Profile name and a list of bandwidth parameters. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If uniqueness of the user label is required, the EMS checks the user label for uniqueness; i.e., if a TC Profile object with the same user label exists already, then a User Label In Use exception is raised. c) If the maximum number of TC Profiles in the EMS has already reached and the TC Profile cannot be created, a Capacity Exceeded exception is raised. 3) If the request is valid the EMS creates the TC Profile. 4) The EMS replies with success indication. 5) The EMS sends a TC Profile object creation notification to the notification service.
Ends when	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.

Use Case 9.5.1: NMS creates a Traffic Conditioning Profile

Post-conditions	In case of success: The TC Profile has been created by the EMS. In case of failure: Nothing has changed in the System, i.e., the TC Profile has not been created.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 333}, {Requirement II. 334}

9.5.2 NMS deletes a Traffic Conditioning Profile

Use Case 9.5.2: NMS deletes a Traffic Conditioning Profile

Use Case Name	NMS deletes a Traffic Conditioning Profile
Summary	This operation allows an NMS to delete a Traffic Conditioning (TC) Profile that already exists. The EMS is required to validate the request, verify that the TC Profile is not associated to a Flow Point (FP) or FD Edge Connectionless Port Termination Point (FD Edge CPTP), and deletes it.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to EMS to delete a TC Profile.
Description	1) The NMS sends a request to EMS to delete a TC Profile. The NMS provides the TC Profile name. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specifed TC Profile object is not known to the EMS, an Entity Not Found exception is raised. c) If any FP or CPTP within the EMS has an association to the TC Profile, an Object In Use exception is raised. 3) If the request is valid the EMS deletes the TC Profile. 4) The EMS replies with success indication. 5) The EMS sends a TC Profile object deletion notification to the notification service.
Ends when	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: The TC Profile has been deleted by the EMS. In case of failure: Nothing has changed in the System, i.e., the TC Profile has not been deleted.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 336}

9.5.3 NMS modifies a Traffic Conditioning Profile

Use Case 9.5.3: NMS modifies a Traffic Conditioning Profile

Use Case Name	NMS modifies a Traffic Conditioning Profile

Use Case 9.5.3: NMS modifies a Traffic Conditioning Profile

Summary	This operation allows an NMS to modify aTraffic Conditioning (TC) Profile. EMS validates the data provided by NMS, and modifies the requested TC Profile in accordance with the NMS parameter list. If the EMS cannot modify the TC Profile as specified, an appropriate exception is raised.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to EMS to modify a TC Profile
Description	1) The NMS sends a request to EMS to modify a TC Profile. The NMS provides the TC Profile name and a list of bandwidth parameters. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specirfied TC Profile object does not exist, an Entity Not Found exception is raised. 3) If the request is valid the EMS modifies the TC Profile with the new parameter list. Note: This will automatically modify the traffic conditioning in all associated TPs. 4) The EMS replies with success indication. 5) The EMS sends a TC Profile attribute value change notification to the notification service.
Ends when	In case of success: The NMS receives an indication of the success of the request. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: The TC Profile has been modified by the EMS. In case of failure: Nothing has changed in the System, i.e., the TC Profile has not been modified.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 334}, {Requirement II. 335}

9.5.4 NMS configures Traffic Mapping Table

Use Case 9.5.4: NMS configures Traffic Mapping Table

Use Case Name	NMS configures Traffic Mapping Table
Summary	This operation allows an NMS to configure the Traffic Mapping Table in an CPTP or FP. The NMS provides a complete (except default) new set of mappings which will overwrite all (except default) existing mappings.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to EMS to configure the Traffic Mapping Table of an FD Edge CPTP or FP.
Description	1) The NMS sends a request to EMS to configure the Traffic Mapping Table of an FD Edge CPTP or FP. 2) The NMS provides a complete (except default) new set of mappings. 3) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If at least one referenced TC Profile objects is not known to the EMS, an Entity Not Found exception is raised. 4) If the request is valid: a) If the provided set of mappings contains no mapping at all (i.e., is empty), all mappings, except the default one, are removed; i.e., the traffic units flowing through the TP are only conditioned by the default configuration. Note: The traffic units may still be conditioned specifically by another TP on this port. If the mappings cannot be removed, an Unable To Comply exception is raised. b) If the provided set of mappings contains mappings, all existing mappings (except default) are overwritten by the new set of mappings; i.e., the traffic units are conditioned by the new configuration. If the mappings cannot be overwritten, an Unable To Comply exception is raised. 5) The EMS replies with a success indication. 6) The EMS sends an attribute value change notification to the notification service.
Ends when	In case of success: The EMS has set the new mappings in the Traffic Mapping Table. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: The TP conditions the traffic units according to the modified Traffic Mapping Table. In case of failure: Nothing has changed in the System, i.e., the Traffic Mapping Tables have not been changed.

Use Case 9.5.4: NMS configures Traffic Mapping Table

Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 337}

9.6 Flow Domain Fragment Management Use Cases

Use Case	Use Case Name	Requirement(s) Fulfilled
Use Case 9.6.1:	NMS creates and activates a Flow Domain Fragment	{Requirement II. 345}, {Requirement II. 346}
Use Case 9.6.2:	NMS deactivates and deletes a Flow Domain Fragment	{Requirement II. 349}
Use Case 9.6.3:	NMS modifies a Flow Domain Fragment	{Requirement II. 347}, {Requirement II. 348}, {Requirement II. 350}, {Requirement II. 351}

9.6.1 NMS creates and activates a Flow Domain Fragment

Use Case 9.6.1: NMS creates and activates a Flow Domain Fragment

Use Case Name	NMS creates and activates a Flow Domain Fragment
Summary	This operation provides a way to create and activate a point-to-point or multipoint-to-multipoint Flow Domain Fragment (FDFr) in one request. The EMS is required to create an FDFr at the requested layer rate. If the EMS cannot meet any of the requested parameters, appropriate exceptions are raised. If the EMS works in the “connectivity-aware” mode, the NMS can request one of two creation results when not all Flow Points have potential connectivity to one another: • option 1) reject the creation request, or • option 2) add all FPs regardless of potential connectivity. If the FDFr is created, an accompanying FP is created for every edge CPTP specified. In case neither the EMS nor the network provides auto-routing, the NMS also has to provide the internal CPTPs that have to be used by the EMS when creating the FDFr. An accompanying FP is created for every internal CPTP specified. If transmission parameters are specified for the involved CPTPs and FPs, the EMS will apply them either before or after the creation of the Matrix Flow Domain Fragments (MFDFrs), as appropriate. The alarm reporting on the CPTPs and the containing FPs may be turned on by the EMS, unless otherwise specified via the alarm reporting transmission parameter. This operation is best effort except for option 1) when the EMS works in the “connectivity-aware” mode.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to create and activate a Flow Domain Fragment to the EMS.

Use Case 9.6.1: NMS creates and activates a Flow Domain Fragment

Description	1) The NMS sends a request to create and activate a Flow Domain Fragment to the EMS. 2) The EMS validates the request: a) TIf the syntax is in error, an Invalid Input exception is raised. b) If one of the referenced CPTP objects is not known to the EMS, an Entity Not Found exception is raised. c) If the specified CPTPs are not associated with the referenced FD, a TP Invalid Endpoint exception is raised. d) If an FDFr with the same properties as specified in the NMS request already exists, the EMS may reuse that FDFr. e) If any of the specified edge CPTPs do not have the “FD Edge” role, a Not In Valid State exception is raised. f) If any of the specified internal CPTPs do not have the “FD Internal” role, a Not In Valid State exception is raised. g) If the NMS has provided an FDFr name and the EMS does not support NMS supplied names, an Unable To Comply exception is raised. h) If uniqueness of the user label is required, the EMS checks the user label for uniqueness; i.e., if a FDFr object with the same user label exists already, then a User Label In Use exception is raised. i) If any of the mandatory input parameters cannot be satisfied, an Unable To Comply exception is raised. j) If the EMS is “connectivity-aware” and option 1 (see summary) was requested, and if not all of the FPs (to be created) have connectivity with one another, an Unable To Comply exception is raised. k) In traffic mapping tables specified for created FPs, if any map column headers are not recognized by the EMS, an Unable To Comply exception is raised. l) In traffic mapping tables specified for created FPs, if there are mismatches between the TrafficMapping_Table_Count and any column length, an Unable To Comply exception is raised. m) In traffic mapping tables specified for created FPs, if table contents would result in frames being mapped to more than one FDFr, an Unable To Comply exception is raised. n) If the FDFr being created would have less than two edge FPs, an Unable To Comply exception is raised. o) If the FDFr cannot use all of the specified internal CPTPs and/or all of the MFDFrs in the specified route, an Unable To Comply exception is raised.

Use Case 9.6.1: NMS creates and activates a Flow Domain Fragment	
	3) If the NMS provides transmission parameters for the involved CPTPs and FPs, it is up to the EMS to provision the parameters on these TPs before or after the activation of the Matrix Flow Domain Fragments, as appropriate. A CPTP or FP which cannot be provisioned, or for whom mandatory[a] transmission parameters cannot be set, must not be added to the new FDFr and has to be returned in the “failed TP list” list (for every FP which is not added to the FDFr, the corresponding server CPTP has to be added to the list). FPs for which only best-effort transmission parameters could not be set have to be added to the new FDFr and must be returned in the “parameter problems TP list” list. CPTPs for which only best-effort transmissions parameters could not be set have to be returned in the “parameter problems TP list”. If an internal CPTP cannot be provisioned, it is returned either in the “failed TP list” list, when the provisioning is mandatory, or to the “parameter problems TP list” list, when the provisioning is best-effort. The alarm reporting on the CPTPs and the contained FPs may be turned on by the EMS as part of this request, unless otherwise specified via the transmission parameter “AlarmReporting”. 4) If the specified value of the FDFr's IVID is not null, all FPs specified with IVID different from the the FDFr's IVID must not be associated to the new FDFr and their server CPTPs have to be returned in the “failed TP list” list. 5) The EMS initiates the activation of the FDFr (which involves establishing the Matrix Flow Domain Fragment(s) at the ME(s)). 6) If all of the Matrix Flow Domain Fragments comprising the FDFr have been established, then the EMS sets the FDFr state of the FDFr to active. The EMS updates the SNC state of the affected CTPs to either sink, source, or bidirectionally connected. 7) If there are error conditions, (e.g. failure to provision TP transmission) existing after establishing all the Matrix Flow Domain Fragments, then the EMS will reply with a failure indication and error reason. In this case the FDFr state of the FDFr will be active. 8) If one or more (possibly all) of the Matrix Flow Domain Fragment comprising the FDFr have not been established then the EMS sets the state of the FDFr to partial. The EMS updates the connection state of those CTPs that were successfully established to either sink, source, or bidirectionally connected. The EMS replies with a failure indication and error reason. 9) The EMS replies with a success indication. 10) The EMS sends FP and FDFr object creation notifications to the notification service.

Use Case 9.6.1: NMS creates and activates a Flow Domain Fragment

Ends when	In case of success: 1) The EMS returns the distinguishing information of the created FDFr in the success indication of the action. 2) The EMS sends object creation notifications for the created FDFr and FPs. The OC notification of “CTP” like objects like FPs is new to the interface. This because: • congestion prevention at the interface • FPs (like CTPs) have a naming convention, hence it is not strictly necessary even to read them. 3) The EMS sends the necessary AVC and SC notifications for the transmission parameters changed in the CPTPs. 4) In case of best-effort, and in case not all requested CPTPs could be associated, the EMS returns the CPTPs that could not be associated in the “failed TP list”. 5) In case not all provided best-effort2 transmissions parameters could be set on some CPTPs or FPs as requested, the EMS returns these TPs in the list of “parameter problems TP list”. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: 1) The FDFr and its FPs have been created. 2) The FDFr has been activated, or it has been partially activated, i.e., some but not all the necessary provisioning operations have been performed. 3) All requested CPTPs which are not contained in the "failed TP list" list have been associated to the new FDFr. 4) The transmission parameters have been set, as requested, to all CPTPs and FPs which are not contained in the “parameter problems TP list”. In case of failure: Nothing has changed in the System, i.e., the FDFr has not been created.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 345}, {Requirement II. 346}

a. Parameters are classified into mandatory and best-effort by bilateral agreement or in the Implementation Statement document.

9.6.2 NMS deactivates and deletes a Flow Domain Fragment

Use Case 9.6.2: NMS deactivates and deletes a Flow Domain Fragment

Use Case Name	NMS deactivates and deletes a Flow Domain Fragment
Summary	The NMS requests that a Flow Domain Fragment be deactivated and deleted. As a result of successfully completing this use case: • The FDFr will be de-provisioned from the EMS' managed Flow Domain, and the FDFr will be removed from the EMS. • The removal of any network resources allocated for the FDFr.
Actors	NMS
Pre-conditions	The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.
Begins when	The NMS sends a request to the EMS to deactivate and delete an FDFr
Description	1) The NMS sends the request to deactivate and delete a Flow Domain Fragment (FDFr) to the EMS. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified FDFr object is not known to the EMS, an Entity Not Found exception is raised. 3) If the request is valid: a) The EMS initiates the deactivation of the FDFr. FDFr deactivation involves EMS communication with managed elements. The EMS attempts to remove, from the applicable managed elements, all the Matrix FDFrs which comprise this FDFr. b) If any Matrix FDFr cannot be removed, its FPs must be returned in the "failed TP list" list. c) If the EMS succeeds in deleting the FDFr (i.e., if all the Matrix FDFrs comprising the FDFr on applicable managed elements have been removed), the EMS initiates the deletion of the FDFr in the EMS. FDFr deletion is not expected to involve EMS communication with managed elements. FDFr deletion involves the EMS deleting the FDFr object, and all associated FPs. d) If some MFDFrs were removed and some were not, the FDFr is not deleted from the EMS, and it is marked "Failed". 4) The EMS provides a success indication to the NMS. 5) The EMS sends FP and FDFr object deletion notifications to the notification service.

Use Case 9.6.2: NMS deactivates and deletes a Flow Domain Fragment

Ends when	In case of success: 1) The NMS receives an indication of success of the request. 2) The FPs of any MFDFrs which could not be removed are returned in the "failed TP list" list. 3) Object Deletion notifications have been sent for all FPs which were removed. 4) If the FDFr has been removed, an Object Deletion notification has been sent. In case of failure: The NMS receives an exception as an indication of the failure of the request.
Post-conditions	In case of success: 1) For all MFDFrs which are removed, the assocated FPs have been deleted. 2) If all MFDFrs were removed, the FDFr has been deleted. 3) If the FDFr was not deleted, it is marked as "Failed". In case of failure: Nothing has changed in the System, i.e., the FDFr has not been created.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 349}

9.6.3 NMS modifies a Flow Domain Fragment

Use Case 9.6.3: NMS modifies a Flow Domain Fragment

<table>
<tr><td>Use Case Name</td><td>NMS modifies a Flow Domain Fragment</td></tr>
<tr><td>Summary</td><td>This operation allows an NMS to add one or more Flow Points (FPs) to an existing Flow Domain Fragment (FDFr), and to remove one or more Flow Points (FPs) from an existing Flow Domain Fragment. Flow Points are added by designating the FD Edge CPTPs and FD Internal CPTPs to be added in two separate lists (one for edge and one for internal). Flow Points are removed by designating their server CPTPs (edge and internal) in a third list.

This operation also allows an NMS to modify the attributes and parameters of the same Flow Domain Fragment, and of other involved CPTPs and FPs, in one request together with FP addition/removal.

The NMS can modify the
• user label
• owner
• network access domain
• administrative state
• connectionless layered parameters of the FDFr
• layered parameters of the associated FPs and CPTPs
• additional information

of the FDFr.

For each added CPTP and associated FP, the NMS can provide the following parameters:
• Transmission parameters (incl. Traffic Mapping Table, Alarm Severity Assignment Profile names and TCA Parameter Profile names)
• ingress/egress Transmission descriptor names
• TCA parameter profile names.

If the EMS works in the “connectivity-aware” mode, the NMS can request one of two modification results when not all new FPs have potential connectivity to all FPs already in the FDFr:
• option 1) reject the modification request, or
• option 2) add all FPs regardless of potential connectivity.

The operation is best effort (except when the EMS is “connectivity-aware” and option 1 was requested).</td></tr>
<tr><td>Actors</td><td>NMS</td></tr>
<tr><td>Pre-conditions</td><td>The NMS has executed Use Case 7.2.2: NMS creates a session with EMS.</td></tr>
<tr><td>Begins when</td><td>The NMS sends a request to the EMS to modify a FDFr.</td></tr>
</table>

Use Case 9.6.3: NMS modifies a Flow Domain Fragment

Description	1) The NMS sends a request to modify a Flow Domain Fragment (FDFr) to the EMS. 2) The EMS validates the request: a) If the syntax is in error, an Invalid Input exception is raised. b) If the specified FDFr object is not known to the EMS, an Entity Not Found exception is raised. c) If uniqueness of the user label is required, the EMS checks the user label for uniqueness; i.e., if an FDFr object with the same user label exists already, then a User Label In Use exception is raised. d) If any of the TP objects specified for removal or modification is not known to the EMS, an Entity Not Found exception is raised. e) If any of the TP objects specified for removal or modification is not associated with the specified FDFr, an Entity Not Found exception is raised. f) If any of the CPTP objects specified for addition is not known to the EMS, an Entity Not Found exception is raised. g) If any of the CPTP objects specified for addition is not associated with the specified FDFr FD, an Entity Not Found exception is raised. Note that it is valid to include a CPTP which is already associated with the FDFr. No FP object creation notification is issued in this case, but there may be attribute value change notifications. h) If any of the CPTPs specified for addition as edge do not have the “FD Edge” role, a Not In Valid State exception is raised. i) If any of the CPTPs specified for addition as internal do not have the “FD Internal” role, a Not In Valid State exception is raised. j) If less than two end FPs would remain after successful completion of the operation, an Unable To Comply exception is raised. k) If any of the mandatory input parameters cannot be satisfied, an Unable To Comply exception is raised. l) In traffic mapping tables provided for specified FPs, if there is any unrecognized information or if there is internal inconsistency (e.g., inconsistent column lengths), an Unable To Comply exception is raised. m) In traffic mapping tables provided for specified FPs, if table contents would result in frames being mapped to more than one FDFr, Unable To Comply exception is raised. n) If fulfilling the command would create contradiction between the IVID value of the FDFr and the IVID value of any FP, an Unable To Comply exception is raised. o) If any of the mandatory transmission parameters of existing TPs cannot be set, an Unable To Comply exception is raised.

Use Case 9.6.3: NMS modifies a Flow Domain Fragment

	3) If the request is valid: a) If the existing value of the FDFr's IVID is not null (and is not being changed), any new FPs specified with IVID different from the the FDFr's IVID must not be associated to the FDFr and their server CPTPs have to be returned in the “failed TP list” list. b) Any new CPTP or FP which cannot be provisioned, or for whom mandatory transmission parameters cannot be set, must not be added to the FDFr and must be returned in the “failed TP list” list (for every FP which is not added to the FDFr, the corresponding server CPTP has to be added to the list). New CPTPs and FPs, for which only best-effort transmissions parameters could not be set, have to be added to the new FDFr and must be returned in the “parameter problems TP list” list. Existing CPTPs and FPs, for which mandatory transmissions parameters could not be modified, must be returned in the “failed TP list” list. Existing CPTPs and FPs, for which only best-effort transmissions parameters could not be modified, must be returned in the “parameter problems TP list” list. If the EMS works in the “connectivity-aware” mode, it checks if the FPs to be added have potential connectivity to the already existing FPs of this FDFr. If not all of the new FPs have connectivity, and if option 1 (see summary) was requested, an Unable To Comply exception is raised c) The EMS modifies FDFr attributes and parameters as requested. Flow Points are created as clients of CPTPs which are successfully added to the FDFr, and their parameters are set. Flow Points, which are clients of CPTPs successfully removed from the FDFr, are deleted. The parameters of other involved TPs are modified as requested. 4) The EMS provides with a success indication. 5) The EMS sends object create, object delete, attribute value and state change notifications to the notification service.
Ends when	In case of success: 1) The NMS receives an indication of success of the action. 2) The EMS returns TPs which could not be added, or whose “mandatory” parameters could not be set/modified, in the “failed TP list”. 3) The EMS returns TPs, whose “best effort” parameters could not be set/modified, in the “parameter problems TP list”. 4) The EMS returns the modified FDFr with its distinguishing information, and sends In case of failure: The NMS receives an exception as an indication of the failure of the request.

Use Case 9.6.3: NMS modifies a Flow Domain Fragment

Post-conditions	In case of success: 1) The Flow Domain Fragment has been modified as requested. 2) The transmission parameters have been set/modified, as requested, to all CPTPs and FPs which are not contained in the “parameter problems TP list” or “failed TP list”. 3) The TPs which could not be added to the FDFr are returned in the “failed TP list”. 4) The EMS has forwarded a Flow Domain Fragment object modification notification and notifications for all objects which were added, removed, or modified. In case of failure: Nothing has changed in the System, i.e., the FDFr has not been created.
Exceptions	Refer to Table 7.1.
Traceability	{Requirement II. 347}, {Requirement II. 348}, {Requirement II. 350}, {Requirement II. 351}

10 Business Requirement Model UML Diagrams

A requirement model is an implementation neutral (i.e. logical) description covering a problem statement (Section 2 and Section 3), a set of requirement statements (Section 4) and a set of use cases (Section 7).

A requirement model is written from a business perspective and is produced prior to the start of implementation of systems, applications and interfaces.

10.1 Static Model

The complete static model can be found in the MTNM Information Agreement, NML-EML Interface, (TMF 608). Section 10.1.1 contains the UML classes (including the names of the attributes and operations) and is provided to facilitate traceability between the requirements statements in Section 4 and the UML Information Model specified in TMF 608.

10.1.1 Classes

Table 10.1 contains a list of the UML Classes defined in TMF 608. The color coding that is used is the same as that used within TMF 608.

objects unchanged with respect to Version 3.0
objects changed with respect to Version 3.0
new objects in Version 3.5

Table 10.1: UML Model Classes

#	UML Classes
1	AID
2	AlarmInformation
3	AlarmSeverityAssignmentProfile
4	ASAPCreateModifyData
5	Attribute
6	Call
7	CallCreateData
8	CallModifyData
9	Capacity
10	CommonCreateModifyData
11	CommonResourceInfo

Table 10.1: UML Model Classes

#	UML Classes
12	CommonTransmissionParameters
13	Connection
14	ConnectionCreateData
15	ConnectionlessTechnologyManager
16	ConnectionParameterProfile
17	ConnectionTerminationPoint
18	Consumer
19	ControlPlaneManager
20	CrossConnect
21	DiversityData
22	DiversityInfo
23	DiversityViolations
24	EMS
25	EMSSession
26	EProtectionSwitch
27	Equipment
28	EquipmentCreateData
29	EquipmentHolder
30	EquipmentProtectionGroup
31	ESwitchData
32	Event
33	EventInformation
34	FDCreateData
35	FDFrCreateData
36	FDFrModifyData
37	FDModifyData
38	FileTransferStatus
39	Filter
40	FloatingTerminationPoint
41	Flow Domain
42	FlowDomainFragment

Table 10.1: UML Model Classes

#	UML Classes
43	FTPCreateData
44	GCTProfileInfo
45	GroupTerminationPoint
46	HeartbeatInformation
47	HistoricalPMDataFile
48	LayeredSNPP
49	LayeredSNPPLink
50	Log
51	LogAVC
52	LogCapacityThresholdAlarm
53	LogObjectCreation
54	LogObjectDeletion
55	LogProcessingErrorAlarm
56	LogRecord
57	LogStateChange
58	ManagedElement
59	MatrixFlowDomain
60	MatrixFlowDomainFragment
61	MFDCreateData
62	MFDModifyData
63	MultiLayerRoutingArea
64	MultiLayerSNPP
65	MultiLayerSNPPLink
66	MultiLayerSubnetwork
67	NMSSession
68	ObjectInformation
69	PerformanceManager
70	PerformanceMonitoringPoint
71	PhysicalTerminationPoint
72	PMCurrentData
73	PMData

Table 10.1: UML Model Classes

#	UML Classes
74	PMHistoricData
75	PMHistoricMeasurement
76	PMMeasurement
77	PMParameter
78	PMPStateChangeInformation
79	PMThreshold
80	ProtectionGroup
81	ProtectionSwitch
82	Route
83	RouteCreateData
84	Session
85	SessionFactory
86	SNCCreateData
87	SNCModifyData
88	SNCRouteChangeData
89	SNP
90	SNPP
91	SNPPLink
92	SoftwareBackupStatus
93	SoftwareManager
94	SubnetworkConnection
95	Supplier
96	SwitchData
97	TCAParameter
98	TCAParameterProfile
99	TCProfileCreateModifyData
100	TDCreateData
101	TerminationPoint
102	ThresholdCrossingAlertInformation
103	TLCreateData
104	TopologicalLink

Table 10.1: UML Model Classes

#	UML Classes
105	TPData
106	TPPool
107	TPPoolCreateData
108	TrafficConditioningProfile
109	TrafficDescriptor
110	TransmissionDescriptor
111	TransmissionDescriptorCreateData
112	TransmissionParameters
113	Version

10.1.1.1 AID

Refer to {Requirement I. 057}.

10.1.1.2 AlarmInformation

Table 10.2: UML Class AlarmInformation (derived from *EventInformation*) Attributes - *{Requirement I. 048}*

Attribute Name	***{Requirement I. 048}***
isClearable	1) Clearable
layer	2) Layer rate
probableCause	3) Probable Cause
perceivedSeverity	4) Perceived severity
serviceAffecting	5) Service affecting
probableCauseQualifier	6) Probable Cause Qualifier
affectedPTPs	7) Affected PTPs
additionalText	8) Additional text
nativeEMSName	9) Native EMS Name
nativeProbableCause	10) Native Probable Cause
acknowledgeIndication	11) Acknowledgement
rootCauseAlarmIndication	12) Root Cause Alarm indication
X.733::EventType	13) X.733 Event Type
X.733::SpecificProblems	14) X.733 Specific problems
X.733::BackedUpStatus	15) X.733 Backed-up status
X.733::BackUpObject	16) X.733 Back-up object
X.733::TrendIndication	17) X.733 Trend indication
X.733::CorrelatedNotifications	18) X.733 Correlated notifications
X.733::MonitoredAttributes	19) X.733 Monitored attributes
X.733::ProposedRepairActions	20) X.733 Proposed repair actions
X.733::AdditionalInformation	21) X.733 Additional Information

10.1.1.3 AlarmSeverityAssignmentProfile

Table 10.3: UML Class AlarmSeverityAssignmentProfile (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 080}*

Attribute Name	***{Requirement I. 081}***
fixed	1) Fixed
alarmSeverityAssignmentList	2) Alarm severity assignments

Table 10.4: UML Class AlarmSeverityAssignmentProfile Operations

Operation Name	
getASAPAssociatedResourceNames	{Requirement II. 209}

10.1.1.4 ASAPCreateModifyData

Table 10.5: UML Class ASAPCreateModifyData Attributes - *{Requirement II. 197}*

Attribute Name	*{Requirement II. 197}*
userLabel	1) User label
forceUniqueness	2) User label uniqueness
owner	3) Owner
alarmSeverityAssignmentList	4) Alarm severity assignments
additionalInfo	5) Additional information

10.1.1.5 Attribute

Table 10.6: UML Class Attribute Attributes - *{Requirement I. 044}* and *{Requirement I. 045}*

Attribute Name	*{Requirement I. 044}*	*{Requirement I. 045}*
id	1) Attribute name	1) State attribute name
value	2) Attribute value	2) State attribute value

10.1.1.6 Call

Table 10.7: UML Class Call (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 133}*

Attribute Name	*{Requirement I. 134}*
callId	1) Call ID
networkAccessDomain	2) Network Access Domain
aEnd	3) A-end
zEnd	4) Z-end
callParameters	5) Call parameter
callState	6) Call state
callDiversity (*association* to DiversityData)	7) Call diversity
diversityViolations (*association* to DiversityViolations)	8) Diversity violations

Table 10.8: UML Class Call (derived from *CommonResourceInfo*) Operations

Operation Name	Requirement
getConnection	{Requirement II. 389}
getAllConnectionsWithSNPP	
addConnections	{Requirement II. 397}
removeConnections	{Requirement II. 399}
modifyDiversityAndCorouting	{Requirement II. 402}

10.1.1.7 CallCreateData

Table 10.9: UML Class CallCreateData Attributes - *{Requirement II. 393}*

Attribute Name	*{Requirement II. 393}*
callName	1) Name
userLabel	2) User label
forceUniqeness	3) User label uniqueness
owner	4) Owner
networkAccessDomain	5) Network Access Domain
aEnd	7) A-end TP
zEnd	8) Z-end TP
callParameters	9) Call Parameter (best effort)
establishCall() parameter	10) Connection-related data - OPTIONAL
establishCall() parameter	11) TP-related data - OPTIONAL
callDiversity	12) Co-routing level of effort
	13) Link Diversity level of effort
	14) Node Diversity level of effort
	15) Link SRG Type
	16) Node SRG Type
establishCall() parameter	17) Number of Route Groups
additionalInfo	18) Additional information

10.1.1.8 CallModifyData

Table 10.10: UML Class CallCreateData Attributes - *{Requirement II. 396}*

Attribute Name	*{Requirement II. 396}*
userLabel	1) User label
forceUniqeness	2) User label uniqueness
owner	3) Owner
networkAccessDomain	4) Network access domain
additionalInfo	5) Additional information

10.1.1.9 Capacity

Table 10.11: UML Class Capacity Attributes -

Attribute Name	
layerRate	
capacity	

10.1.1.10 CommonCreateModifyData

Table 10.12: UML Class CommonCreateModifyData Attributes

Attribute Name	
userLabel	
forceUniqueness	
owner	
networkAccessDomain	
additionalInfo	
association to TransmissionParameters	

10.1.1.11 CommonResourceInfo

Table 10.13: UML Class CommonResourceInfo Attributes - *{Requirement I. 060}*

Attribute Name	***{Requirement I. 060}***
name	1) Name
userLabel	2) User label
nativeEmsName	3) Native EMS name
owner	4) Owner
additionalInfo	5) Additional information

Table 10.14: UML Class CommonResourceInfo Operations

Operation Name	
setAdditionalInfo	{Requirement II. 223}
setNativeEmsName	{Requirement II. 077}

Table 10.14: UML Class CommonResourceInfo Operations

Operation Name	
setOwner	{Requirement II. 076}
setUserLabel	{Requirement II. 075}
setNetworkAccessDomain	{Requirement II. 193}

10.1.1.12 CommonTransmissionParameters

Table 10.15: UML Class CommonTransmissionParameters Attributes

Attribute Name	
layerRate	{Requirement I. 022}
transmissionParameters	{Requirement I. 024}

10.1.1.13 Connection

Table 10.16: UML Class Connection (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 129}* and *{Requirement I. 130}*

Attribute Name	***{Requirement I. 131}***
connectionID	1) Connection ID
layerRate	2) Layer Rate
aEnd	3) A-end ID
zEnd	4) Z-end ID
aEnd	5) A-end TP
zEnd	6) Z-end TP
aEnd	7) A-end TNA Name
zEnd	8) Z-end TNA Name
direction	9) Directionality
rerouteAllowed	10) Reroute Allowed
staticProtectionLevel	11) Static Protection Level
maximumCost	12) Maximum Cost
protectionEffort	13) Protection Effort
routingConstraintEffort	14) Routing Constraint Effort
connectionType	15) Connection Type
connectionSetupType	16) Connection Setup Type
connectionState	17) Connection State
revertive	18) Revertive
networkAccessDomain	19) Network Access Domain
priority	20) Priority
alarmReportingIndication	21) Alarm Reporting Indication
asapPointer	22) Alarm Severity Assignment Profile Pointer

Table 10.16: UML Class Connection (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 129}* and *{Requirement I. 130}*

Attribute Name	***{Requirement I. 131}***
routeGroupLabel	23) Route Group Label
usingHomeRoute	24) Using Home Route
association to Call	25) Call ID
association to SubnetworkConnection	26) Supporting SNCs

Table 10.17: UML Class Connection (derived from *CommonResourceInfo*) Operations

Operation Name	**Requirement**
addRoute	{Requirement II. 241}
getBackupRoutes	{Requirement II. 256}
getIntendedRoute	{Requirement II. 261}
getRouteAndMLSNPPLinks	{Requirement II. 390}
setAlarmReportingOff	{Requirement II. 160}
setAlarmReportingOn	{Requirement II. 159}
setIntendedRoute	{Requirement II. 251}
setRoutesAdminState	{Requirement II. 249}
switchRoute	{Requirement II. 247}

10.1.1.14 ConnectionCreateData

Table 10.18: UML Class ConnectionCreateData Attributes - *{Requirement II. 398}*

Attribute Name	*{Requirement II. 398}*
connectionName	1) Name
userLabel	2) User Label
forceUniqueness	3) Force uniqueness
owner	4) Owner
layerRate	5) Layer Rate
aEnd	6) A-end ID
	8) A-end TP
	10) A-end TNA Name
zEnd	7) Z-end ID
	9) Z-end TP
	11) Z-end TNA Name
direction	12) Directionality
rerouteAllowed	13) Reroute Allowed
staticProtectionLevel	14) Static Protection Level
maximumCost	15) Maximum Cost
protectionEffort	16) Protection Effort
routingConstraintEffort	17) Routing Constraint Effort
connectionType	18) Connection Setup Type
revertive	19) Revertive
networkAccessDomain	20) Network Access Domain
priority	21) Priority
alarmReportingIndication	22) Alarm Reporting Indication
asapPointer	23) Alarm Severity Assignment Profile Pointer
routeGroupLabel	24) Route Group Label
additionalInfo	26) Additional information

Table 10.18: UML Class ConnectionCreateData Attributes - *{Requirement II. 398}*

Attribute Name	*{Requirement II. 398}*
associations to CrossConnect ConnectionTerminationPoint ManagedElement MultiLayerSNPPLink SNP SNPP TerminationPoint	27) Routing Constraints

10.1.1.15 ConnectionlessTechnologyManager

Table 10.19: UML Class ConnectionlessTechnologyManager Operations

Operation Name	Requirement
getAssigningMFD	{Requirement II. 292}
getAllFlowDomains	{Requirement II. 314}
getFlowDomainsByUserLabel	{Requirement II. 317}
getFlowDomain	{Requirement II. 318}
getAllTCProfiles	{Requirement II. 330}
getTCProfile	{Requirement II. 331}
createMFD	{Requirement II. 305}
deleteMFD	{Requirement II. 309}
modifyMFD	{Requirement II. 307}
createFlowDomain	{Requirement II. 321}
deleteFlowDomain	{Requirement II. 325}
modifyFlowDomain	{Requirement II. 323}
createTCProfile	{Requirement II. 333}
deleteTCProfile	{Requirement II. 336}
modifyTCProfile	{Requirement II. 335}
createFTP	{Requirement II. 293}
deleteFTP	{Requirement II. 294}

10.1.1.16 ConnectionParameterProfile

Table 10.20: UML Class ConnectionlParameterProfile Attributes

Attribute Name	Requirement
serviceClassParameters	

10.1.1.17 ConnectionTerminationPoint

Table 10.21: UML Class ConnectionTerminationPoint (derived from *TerminationPoint*) Attributes - *{Requirement I. 006}*

Attribute Name	*{Requirement I. 062}*
connectionState	1) Connection state
tpMappingMode	2) Mapping mode
association to TrafficDescriptor	3) Ingress TD
association to TrafficDescriptor	4) Egress TD

Table 10.22: UML Class ConnectionTerminationPoint (derived from *TerminationPoint*) Operations - *{Requirement I. 006}*

Operation Name	
associateWithTrafficDescriptor	{Requirement II. 074}
getAssociatedGroupTerminationPoint	{Requirement II. 184}
getContainingTPs	{Requirement II. 047}
getContainingTPsNames	{Requirement II. 048}
getTPMappingMode	{Requirement II. 071}
performProtectionCommand	{Requirement II. 116}
retrieveSwitchData	{Requirement II. 117}
terminateAndMap	{Requirement II. 068} (SONET/SDH)
terminateAndMap	{Requirement II. 073} (ATM)
unTerminateAndUnmap	{Requirement II. 069}

10.1.1.18 Consumer

Table 10.23: UML Class Consumer Operations

Operation Name	
pushEvent	

10.1.1.19 ControlPlaneManager

Table 10.24: UML Class ControlPlaneManager Operations

Operation Name	Requirements
getAllTopLevelMLRAs	{Requirement II. 357}
getAllTopLevelMLRANames	{Requirement II. 358}
getAllMLRAs	{Requirement II. 354}
getMLRA	{Requirement II. 355}
getAllMLSNPPLinks	{Requirement II. 360}
getAllMLSNPPLinksWithTP	{Requirement II. 362}
getAllMLSNPPLinksWithMLRAs	{Requirement II. 365}
getAllMLSNPPLinksWithTNA	{Requirement II. 364}
getMLSNPPLink	{Requirement II. 361}
getAllMLSNPPs	{Requirement II. 371}
getAllMLSNPPsWithTP	{Requirement II. 369}
getAllMLSNPPsWithTNA	{Requirement II. 370}

10.1.1.20 CrossConnect

Table 10.25: UML Class CrossConnect Attributes - *{Requirement I. 019}*

Attribute Name	*{Requirement I. 020}*
ccType	1) Type
direction	2) Directionality
active	3) Active
fixed	4) Fixed
additionalInfo	5) Additional information
associations to: ConnectionTerminationPoint GroupTerminationPoint	6) aEnd TP(s)
associations to: ConnectionTerminationPoint GroupTerminationPoint	7) zEnd TP(s)
connectionID	8) Connection ID

10.1.1.21 DiversityData

Table 10.26: UML Class DiversityData Attributes - *{Requirement I. 134}*

Attribute Name	*Call diversity*
coroutingLevelOfEffort	Co-routing Level of Effort
linkDiversityLevelOfEffort	Link Diversity Level of Effort
nodeDiversityLevelOfEffort	Node Diversity Level of Effort
linkSRGType	Link SRG Type
nodeSRGType	Node SRG Type

10.1.1.22 DiversityInfo

Table 10.27: UML Class DiversityInfo Attributes

Attribute Name	
srgType	
sharedResourceList	

10.1.1.23 DiversityViolations

Table 10.28: UML Class DiversityViolations Attributes - *{Requirement I. 134}*

Attribute Name	***Diversity violations***
diversitySynthesis	Diversity synthesis
linkDiversityViolationsList	Link diversity violations
nodeDiversityViolationsList	Node diversity violations
linkPartialDiversityViolationsList	Link partial diversity list
nodePartialDiversityViolationsList	Node partial diversity list

10.1.1.24 EMS

Table 10.29: UML Class EMS (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 001}*

Attribute Name	***{Requirement I. 061}***
emsVersion	1) Software version
type	2) Type
asapPointer	3) Alarm severity assignment profile

Table 10.30: UML Class EMS (derived from *CommonResourceInfo*) Operations

Operation Name	
acknowledgeAlarms	{Requirement II. 155}
assignASAP	{Requirement II. 201}
createASAP	{Requirement II. 196}
createTopologicalLink	{Requirement II. 168}
createTrafficDescriptor	{Requirement II. 097}
createTransmissionDescriptor	{Requirement II. 190}
deassignASAP	{Requirement II. 203}
deleteASAP	{Requirement II. 200}
deleteTopologicalLink	{Requirement II. 170}
deleteTrafficDescriptor	{Requirement II. 099}
deleteTransmissionDescriptor	{Requirement II. 192}
destroyGCT	{Requirement II. 152}

Table 10.30: UML Class EMS (derived from *CommonResourceInfo*) Operations

Operation Name	
getAllASAPNames	{Requirement II. 206}
getAllASAPs	{Requirement II. 205}
getAllEMSAndMEActiveAlarms	{Requirement II. 270}
getAllEMSAndMEUnacknowledgedActiveAlarms	{Requirement II. 154}
getAllEMSSystemActiveAlarms	{Requirement II. 111}
getAllEMSSystemUnacknowledgedActiveAlarms	{Requirement II. 288}
getAllManagedElementNames	{Requirement II. 003}
getAllManagedElements	{Requirement II. 002}
getAllTopLevelSubnetworkNames	{Requirement II. 010}
getAllTopLevelSubnetworks	{Requirement II. 011}
getAllTopLevelTopologicalLinkNames	{Requirement II. 013}
getAllTopLevelTopologicalLinks	{Requirement II. 012}
getAllTrafficDescriptorNames	{Requirement II. 061}
getAllTrafficDescriptors	{Requirement II. 060}
getAllTransmissionDescriptorNames	{Requirement II. 188}
getAllTransmissionDescriptors	{Requirement II. 187}
getASAP	{Requirement II. 207}
getASAPByResource	{Requirement II. 208}
getCapabilities	{Requirement II. 146}
getEMS	{Requirement II. 001}
getGCTProfileInfo	{Requirement II. 151}
getLog	{Requirement II. 271}
getManagedElement	{Requirement II. 006}
getMultiLayerSubnetwork	{Requirement II. 008}
getTopLevelTopologicalLink	{Requirement II. 014}
getTrafficDescriptor	{Requirement II. 062}
getTransmissionDescriptor	{Requirement II. 189}
launchGCT	{Requirement II. 147}
modifyASAP	{Requirement II. 198}
modifyTransmissionDescriptor	{Requirement II. 353}
unacknowledgeAlarms	{Requirement II. 156}

10.1.1.25 EMSSession

Table 10.31: UML Class EMSSession (derived from *Session*) Operations

Operation Name	
getSupplier	

10.1.1.26 EProtectionSwitch

Table 10.32: UML Class EProtectionSwitch (derived from *Event*) Attributes - *{Requirement I. 074}*

Attribute Name	*{Requirement I. 074}*
eProtectionGroupType	1) Type
eSwitchReason	2) Switch reason
groupName	3) EPG
protectedE	4) Protected Equipment
switchAwayFromE	5) Switch away from Equipment
switchToE	6) Switch to Equipment
emsTime	7) EMS timestamp
neTime	8) NE timestamp

10.1.1.27 Equipment

Table 10.33: UML Class Equipment (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 064}*

Attribute Name	*{Requirement I. 032}*
serviceState	1) Service state
alarmReportingIndicator	2) Alarm reporting
expectedEquipmentObjectType	3) Expected equipment type
installedEquipmentObjectType	4) Installed equipment type
installedPartNumber	5) Installed part number
installedSerialNumber	6) Installed serial number
installedVersion	7) Installed version
manufacturer	8) Manufacturer
protectionRole	9) Protection role
protectionSchemeState	10) Protection scheme state
asapPointer	11) Alarm severity assignment profile

Table 10.33: UML Class Equipment (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 064}*

Attribute Name	*{Requirement I. 032}*
manufacturerDate	12) Manufacturer date

Table 10.34: UML Class Equipment (derived from *CommonResourceInfo*) Operations

Operation Name	
getSupportedEquipment	{Requirement II. 227}
getSupportedEquipmentNames	{Requirement II. 228}
getSupportedMFDs	{Requirement II. 300}
getSupportedPTPNames	{Requirement II. 051}
getSupportedPTPs	{Requirement II. 050}
getSupportingEquipment	{Requirement II. 225}
getSupportingEquipmentNames	{Requirement II. 226}
setAlarmReportingOff	{Requirement II. 078}
setAlarmReportingOn	{Requirement II. 078}

10.1.1.28 EquipmentCreateData

Table 10.35: UML Class EquipmentCreateData Attributes - *{Requirement II. 263}*

Attribute Name	*{Requirement II. 263}*
userLabel	1) User label
forceUniqueness	2) User label uniqueness
owner	3) Owner
expectedEquipmentType	4) Expected equipment type
alarmReportingIndication	5) Alarm reporting
asapPointer	6) Alarm severity assignment profile
manufacturer	7) Manufacturer
protectionRole	8) Protection role
protectionSchemeState	9) Protection scheme state
additionalCreationInfo	10) Additional information.

10.1.1.29 EquipmentHolder

Table 10.36: UML Class EquipmentHolder (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 033}*

Attribute Name	*{Requirement I. 065}*
alarmReportingIndicator	1) Alarm reporting
holderType	2) Type
acceptableEquipmentList	4) Acceptable equipment types
expectedOrInstalledEquipment	3) Expected or installed equipment
holderState	5) State
asapPointer	6) Alarm severity assignment profile
location	7) Location
manufacturer	8) Manufacturer
manufacturerDate	9) Manufacturer date

Table 10.37: UML Class EquipmentHolder (derived from *CommonResourceInfo*) Operations

Operation Name	
getAllEquipment	{Requirement II. 279}

Table 10.37: UML Class EquipmentHolder (derived from *CommonResourceInfo*) Operations

Operation Name	
getAllEquipmentNames	{Requirement II. 280}
getContainedEquipment	{Requirement II. 058}
provisionEquipment	{Requirement II. 136}
setAlarmReportingOff	{Requirement II. 079}
setAlarmReportingOn	{Requirement II. 079}
unprovisionEquipment	{Requirement II. 262}

10.1.1.30 EquipmentProtectionGroup

Table 10.38: UML Class EquipmentProtectionGroup (derived from CommonResourceInfo) Attributes - *{Requirement I. 072}*

Attribute Name	*{Requirement I. 073}*
eProtectionGroupType	1) Type
protectionSchemeState	2) Protection scheme state
reversionMode	3) Reversion mode
association to Equipment	4) Protected Equipment
association to Equipment	5) Protecting Equipment
ePgpParameters	6) PG parameter list
asapPointer	7) Alarm severity assignment profile

Table 10.39: UML Class EquipmentProtectionGroup (derived from CommonResourceInfo) Operations

Operation Name	
retrieveESwitchData	{Requirement II. 175}

10.1.1.31 ESwitchData

Table 10.40: UML Class ESwitchData Attributes - *{Requirement II. 269}*

Attribute Name	*{Requirement II. 269}*
eProtectionGroupType	1) Type
eSwitchReason	2) Switch reason
association to EquipmentProtectionGroup	3) EPG
association to Equipment	4) Protected Equipment
association to Equipment	5) Switch to Equipment
additionalInfo	6) Additional information

10.1.1.32 Event

Table 10.41: UML Class Event Attributes - *{Requirement I. 068}*

Attribute Name	*{Requirement I. 068}*
notificationId	1) Identifier
eventType	2) Type

10.1.1.33 EventInformation

Table 10.42: UML Class EventInformation (derived from *Event*) Attributes - *{Requirement I. 093}*

Attribute Name	*{Requirement I. 093}*
objectName	1) Object Name
objectType	2) Object Type
neTime	4) NE timestamp
emsTime	3) EMS timestamp
edgePointRelated	5) Edge Point

10.1.1.34 FDCreateData

Table 10.43: UML Class FDCreateData (derived from *CommonCreateModifyData*) Attributes - *{Requirement II. 322}*

Attribute Name	*{Requirement II. 322}*
name	1) Name
inherited from CommonCreateModifyData	2) User label
inherited from CommonCreateModifyData	3) User label uniqueness
inherited from CommonCreateModifyData	4) Owner
inherited from CommonCreateModifyData	5) Network Access Domain
association to MatrixFlowDomainFragment	6) List of Matrix Flow Domains
inherited from CommonCreateModifyData	7) Connectionless layered parameters
inherited from CommonCreateModifyData	8) Additional information

10.1.1.35 FDFrCreateData

Table 10.44: UML Class FDFrCreateData (derived from *CommonCreateModifyData*) Attributes - *{Requirement II. 346}*

Attribute Name	*{Requirement II. 346}*
name	1) Name
inherited from CommonCreateModifyData	2) User label
inherited from CommonCreateModifyData	3) User label uniqueness
inherited from CommonCreateModifyData	4) Owner
inherited from CommonCreateModifyData	5) Network Access Domain
inherited from CommonCreateModifyData	6) Connectionless layered parameters
direction	7) Directionality
association to TerminationPoint	8) aEnd TPs
association to TerminationPoint	9) zEnd TPs
association to TerminationPoint	10) Internal TPs
association to MatrixFlowDomainFragment	11) MFDFrs
fullRoute	12) Full route
createAndActivateFDFr() parameter	13) Termination Point(s) to configure
fdfrType	14) FDFr type
connectivityRequirement	15) Connectivity requirement
administrativeState	16) Administrative state

Table 10.44: UML Class FDFrCreateData (derived from *CommonCreateModifyData*) Attributes - *{Requirement II. 346}*

Attribute Name	{Requirement II. 346}
inherited from CommonCreateModifyData	17) Additional information

10.1.1.36 FDFrModifyData

Table 10.45: UML Class FDFrModifyData (derived from *CommonCreateModifyData*) Attributes - *{Requirement II. 348}*

Attribute Name	{Requirement II. 348}
inherited from CommonCreateModifyData	1) User label
inherited from CommonCreateModifyData	2) User label uniqueness
inherited from CommonCreateModifyData	3) Owner
inherited from CommonCreateModifyData	4) Network Access Domain
inherited from CommonCreateModifyData	5) Connectionless layered parameters
association to TerminationPoint	6) TPs to remove
association to TerminationPoint	7) aEnd TPs
association to TerminationPoint	8) zEnd TPs
association to TerminationPoint	9) Internal TPs
modifyFDFr() parameter	10) Termination Point(s) to modify
administrativeState	11) Administrative state
inherited from CommonCreateModifyData	12) Additional information

10.1.1.37 FDModifyData

Derived from ***CommonCreateModifyData***.

10.1.1.38 FileTransferStatus

Table 10.46: UML Class FileTransferStatus (derived from *Event*) Attributes - *{Requirement I. 058}*

Attribute Name	*{Requirement I. 058}*
fileName	1) File name
transferStatus	2) Transfer status
percentComplete	3) Percentage complete
failureReason	4) Reason for failure

10.1.1.39 Filter

Table 10.47: UML Class Filter Attributes

Operation Name	
addConstraints	
match	

10.1.1.40 FloatingTerminationPoint

derived from PhysicalTerminationPoint and ConnectionTerminationPoint, no new attributes, no new operations - {Requirement I. 075}.

10.1.1.41 Flow Domain

Table 10.48: UML Class Flow Domain (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 105}*

Attribute Name	*{Requirement I. 106}*
	1) Layered transmission parameters
networkAccessDomain	2) Network Access Domain
fdConnectivityState	3) FD connectivity state
fdType	4) FD type

Table 10.49: UML Class Flow Domain (derived from *CommonResourceInfo*) Operations

Operation Name	Requirement
associateCPTPsWithFlowDomain	{Requirement II. 326}
associateMFDsWithFlowDomain	{Requirement II. 328}
createAndActivateFDFr	{Requirement II. 345}

Table 10.49: UML Class Flow Domain (derived from *CommonResourceInfo*) Operations

Operation Name	Requirement
deactivateAndDeleteFDFr	{Requirement II. 349}
deassociateCPTPsFromFlowDomain	{Requirement II. 327}
deassociateMFDsFromFlowDomain	{Requirement II. 329}
getAllAssociatedMFDs	{Requirement II. 316}
getAllCPTPs	{Requirement II. 315}
getAllFDFrs	{Requirement II. 338}
getAllFDFrsWithCPTP	{Requirement II. 339}
getAllTopologicalLinks	{Requirement II. 380}
getFDFr	{Requirement II. 340}
getFDFrByUserLabel	{Requirement II. 341}
getFDrWithFP	{Requirement II. 342}
getTransmissionParameters	{Requirement II. 319}
modifyFDFr	{Requirement II. 347}

10.1.1.42 FlowDomainFragment

Table 10.50: UML Class FlowDomainFragment (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 110}*

Attribute Name	*{Requirement I. 111}*
direction	1) Directionality
association to TransmissionParameters	2) Layered transmission parameters
association to TerminationPoint	3) aEnd TPs
association to TerminationPoint	4) zEnd TPs
networkAccessDomain	5) Network Access Domain
flexible	6) Flexible
administrativeState	7) Administrative state
fdfrState	8) FDFr state
fdfrType	9) FDFr type

Table 10.51: UML Class FlowDomainFragment (derived from *CommonResourceInfo*) Operations

Operation Name	Requirement
getFDFrRoute	{Requirement II. 343}
getTransmissionParameter	{Requirement II. 344}

10.1.1.43 FTPCreateData

Table 10.52: UML Class FDFrCreateData (derived from *CommonCreateModifyData*) Attributes - *{Requirement II. 299}*

Attribute Name	***{Requirement II. 299}***
inherited from CommonCreateModifyData	1) User label
inherited from CommonCreateModifyData	2) User label uniqueness
inherited from CommonCreateModifyData	3) Owner
inherited from CommonCreateModifyData	4) Network Access Domain
equipmentName	5) Equipment name
ingressTransmissionDescriptorName	6) Ingress TMD
egressTransmissionDescriptorName	7) Egress TMD
inherited from CommonCreateModifyData	8) Layered transmission parameters
inherited from CommonCreateModifyData	9) Additional information

10.1.1.44 GCTProfileInfo

Table 10.53: UML Class -GCTProfileInfo Attributes

Attribute Name	
serverLaunchCapability	
gctHostName	
emsGctPlatform	
guiCutThroughDataList	

10.1.1.45 GroupTerminationPoint

Table 10.54: UML Class - GroupTerminationPoint (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 069}*

Attribute Name	*{Requirement I. 070}*
listOfTPs	1) Contained TPs
connectionState	2) Connection State
networkAccessDomain	3) Network Access Domain
alarmReportingIndication	4) Alarm reporting
asapPointer	5) Alarm severity assignment profile

Table 10.55: UML Class GroupTerminationPoint (derived from *CommonResourceInfo*) Operations

Operation Name	
setAlarmReportingOn	{Requirement II. 219}
setAlarmReportingOff	{Requirement II. 220}
getTPGroupingRelationships	{Requirement II. 272}

10.1.1.46 HeartbeatInformation

Table 10.56: UML Class HeartbeatInformation (derived from *Event*) Attributes - *{Requirement I. 077}*

Attribute Name	*{Requirement I. 077}*
objectName	1) Object Name
objectType	2) Object Type
emsTime	3) EMS timestamp

10.1.1.47 HistoricalPMDataFile

Table 10.57: UML Class HistoricalPMDataFile Attributes - *{Requirement II. 284}*

Attribute Name	*{Requirement II. 284}*
startTime	1) Start time
endTime	2) End time

10.1.1.48 LayeredSNPP

Table 10.58: UML Class LayeredSNPP Attributes - *{Requirement I. 126}*

Attribute Name	*{Requirement I. 126}*
layerRate	1) Layer rate
association to SNPP	2) SNPP list

10.1.1.49 LayeredSNPPLink

Table 10.59: UML Class LayeredSNPPLink Attributes - *{Requirement I. 123}*

Attribute Name	*{Requirement I. 123}*
layerRate	1) Layer rate
association to SNPPLink	2) SNPP Link list

10.1.1.50 Log

Table 10.60: UML Class Log Attributes - *{Requirement I. 090}*

Attribute Name	*{Requirement I. 091}*
operationalState	1) Operational state
administrativeState	2) Administrative state
logSize	3) Size
logFullAction	4) Full action
logDuration	5) Duration
logScheduling	6) Scheduling
availabilityStatus	7) Availability status
logRecordCompaction	8) Record compaction
logCapacityAlarmThresholds	9) Capacity alarm thresholds
discriminatorConstruct	10) Discriminator construct

Table 10.61: UML Class Log Operations

Operation Name	*{Requirement II. 254}*
deleteLogRecords	1) delete Log Records
getLogRecords	2) get Log Records
retrieveLogRecords	3) retrieve Log Records
setAdministrativeState	4) set Administrative State
setCapacityAlarmThreshold	5) set Capacity Alarm Threshold
setDiscriminatorConstruct	6) set Discriminator Construct
setLogFullAction	7) set Log Full Action
setMaxLogSize	8) set Max Log Size
setMaxRecordLife	9) set Max Record Life
setWeekMask	10) set Week Mask

10.1.1.51 LogAVC

Table 10.62: UML Class LogAVC (derived from *Event*) Attributes - *{Requirement I. 095}*

Attribute Name	*{Requirement I. 095}*
attributeType	1) Attribute identifier
oldvalue	2) Old attribute value
newValue	3) New attribute value

10.1.1.52 LogCapacityThresholdAlarm

Table 10.63: UML Class LogCapacityThresholdAlarm (derived from *Event*) Attributes - *{Requirement I. 096}*

Attribute Name	*{Requirement I. 096}*
observedValue	1) Observed value
crossedValue	2) Crossed value
perceivedSeverity	3) Perceived severity

10.1.1.53 LogObjectCreation

derived from Event.

10.1.1.54 LogObjectDeletion

derived from Event.

10.1.1.55 LogProcessingErrorAlarm

Table 10.64: UML Class LogProcessingErrorAlarm (derived from *Event*) Attributes - *{Requirement I. 097}*

Attribute Name	***{Requirement I. 097}***
errorNumber	1) Error number
errorReason	2) Error reason

10.1.1.56 LogRecord

Table 10.65: UML Class LogRecord Attributes -

Attribute Name	
recordId	
time	
eventInfo	

10.1.1.57 LogStateChange

Table 10.66: UML Class LogStateChange (derived from *Event*) Attributes - *{Requirement I. 098}*

Attribute Name	***{Requirement I. 098}***
stateType	1) State identifier
newValue	2) New state value

10.1.1.58 ManagedElement

Table 10.67: UML Class ManagedElement (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 002}*

Attribute Name	***{Requirement I. 003}***
location	1) Location
version	2) Software version
productName	3) Product name
communicationState	4) Communication state
supportedRates	5) Supported connection layer rate(s)
emsInSyncState	6) Synchronization state
networkAccessDomain	7) Network Access Domain
manufacturer	8) Manufacturer

Table 10.67: UML Class ManagedElement (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 002}*

Attribute Name	*{Requirement I. 003}*
asapPointer	9) Alarm severity assignment profile
manufacturerDate	10) Manufacturer date

Table 10.68: UML Class ManagedElement (derived from *CommonResourceInfo*) Operations

Operation Name	
createGroupTerminationPoint	{Requirement II. 164}
deleteGroupTerminationPoint	{Requirement II. 166}
getActiveMaintenanceOperations	{Requirement II. 138}
getAllActiveAlarms	{Requirement II. 110}
getAllCrossConnections	{Requirement II. 063}
getAllEProtectionGroups	{Requirement II. 174}
getAllEquipment	{Requirement II. 056}
getAllEquipmentNames	{Requirement II. 057}
getAllFixedCrossConnections	{Requirement II. 183}
getAllFTPNames	{Requirement II. 211}
getAllFTPNames	{Requirement II. 213}
getAllFTPs	{Requirement II. 210}
getAllFTPs	{Requirement II. 212}
getAllGroupTerminationPointNames	{Requirement II. 172}
getAllGroupTerminationPoints	{Requirement II. 171}
getAllMFDs	{Requirement II. 295}
getAllNUTTPNames	{Requirement II. 274}
getAllPreemptibleTPNames	{Requirement II. 275}
getAllProtectedTPNames	{Requirement II. 276}
getAllProtectionGroups	{Requirement II. 059}
getAllPTPNames	{Requirement II. 215}
getAllPTPNames	{Requirement II. 217}
getAllPTPNamesWithoutFTPs	{Requirement II. 034}
getAllPTPNamesWithoutFTPs	{Requirement II. 036}

Table 10.68: UML Class ManagedElement (derived from *CommonResourceInfo*) Operations

Operation Name	
getAllPTPs	{Requirement II. 214}
getAllPTPs	{Requirement II. 216}
getAllPTPsWithoutFTPs	{Requirement II. 033}
getAllPTPsWithoutFTPs	{Requirement II. 035}
getAllUnacknowledgedActiveAlarms	{Requirement II. 287}
getContainingSubnetworkNames	{Requirement II. 009}
getEProtectionGroup	{Requirement II. 175}
getEquipment	{Requirement II. 056}
getGroupTerminationPoint	{Requirement II. 173}
getMFDs	{Requirement II. 296}
getProtectionGroup	{Requirement II. 112}
getTP	{Requirement II. 027}
modifyGroupTerminationPoint	{Requirement II. 167}

10.1.1.59 MatrixFlowDomain

Table 10.69: UML Class MatrixFlowDomain (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 103}*

Attribute Name	*{Requirement I. 104}*
association to TransmissionParameters	1) Layered transmission parameters
networkAccessDomain	2) Network access domain
flexible	3) Flexible
tmdState	4) TMD state

Table 10.70: UML Class MatrixFlowDomain Operations

Operation Name	Requirement
assignCPTPsToMFD	{Requirement II. 310}
getAllAssignableCPTPs	{Requirement II. 304}
getAllAssignedCPTPs	{Requirement II. 297}
getAllSupportingEquipmentNames	{Requirement II. 053}
getAllSupportingEquipments	{Requirement II. 052}

Table 10.70: UML Class MatrixFlowDomain Operations

Operation Name	Requirement
getAssociatingFlowDomain	{Requirement II. 303}
getTransmissionParameters	{Requirement II. 302}
setTransmissionDescriptorAssociation	{Requirement II. 312}
setTransmissionDescriptorAssociation	{Requirement II. 313}
unassignCPTPsFromMFD	{Requirement II. 311}
verifyTMDAssignmentToMFD	{Requirement II. 278}

10.1.1.60 MatrixFlowDomainFragment

Table 10.71: UML Class MatrixFlowDomainFragment (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 114}*

Attribute Name	*{Requirement I. 115}*
association to ConnectionTerminationPoint	1) aEnd TPs
association to ConnectionTerminationPoint	2) zEnd TPs
direction	3) Directionality
flexible	4) Flexible
active	5) Active
mfdfrType	6) MFDFr type
additionalInfo	7) Additional information

10.1.1.61 MFDCreateData

Derived from ***CommonCreateModifyData***.

10.1.1.62 MFDModifyData

Derived from ***CommonCreateModifyData***.

10.1.1.63 MultiLayerRoutingArea

Table 10.72: UML Class MultiLayerRoutingArea (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 116}*

Attribute Name	*{Requirement I. 117}*
lateredRoutingAreaList	1) Layered Routing Area List
routingAreaLevel	2) Routing Area Level
association to ManagedElement	3) Supporting ME Name
sharedRiskGroup	4) Shared Risk Group
association to MultiLayerRoutingArea	5) Superior MultiLayer Routing Area
association to MultiLayerSubnetwork	6) Supporting MultiLayer Subnetwork List

Table 10.73: UML Class MultiLayerRoutingArea (derived from *CommonResourceInfo*) Operations

Operation Name	Requirement
establishCall	{Requirement II. 392}
getAllCallIdsWithSNPPOrTNAName	{Requirement II. 385}
getAllCallIdsWithTP	{Requirement II. 385}
getAllCallsAndTopLevelConnections	{Requirement II. 381}
getAllCallsAndTopLevelConnectionsAndSNCs	{Requirement II. 415}
getAllCallsAndTopLevelConnectionsWithME	{Requirement II. 388}
getAllCallsAndTopLevelConnectionsAndSNCsWithME	{Requirement II. 416}
getAllCallsAndTopLevelConnectionsAndSNCsWithTP	{Requirement II. 417}
getAllEdgeMLSNPPLinks	{Requirement II. 363}
getAllInternalMLSNPPLinks	{Requirement II. 359}
getAllMLSNPPLinks	{Requirement II. 360}
getAllMLSNPPs	{Requirement II. 367}
getAllSubordinateMLRAs	{Requirement II. 356}
getAllSubordinateRAIdsWithConnection	{Requirement II. 391}
getCall	{Requirement II. 387}
getCallAndTopLevelConnections	{Requirement II. 383}
getCallAndTopLevelConnectionsAndSNCs	{Requirement II. 386}
getConnectionAndRouteDetails	
getMLSNPPLink	{Requirement II. 368}

Table 10.73: UML Class MultiLayerRoutingArea (derived from *CommonResourceInfo*) Operations

Operation Name	Requirement
modifyCall	{Requirement II. 395}
releaseCall	{Requirement II. 394}

10.1.1.64 MultiLayerSNPP

Table 10.74: UML Class MultiLayerSNPP (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 118}*

Attribute Name	*{Requirement I. 119}*
direction	1) Directionality
association to LayeredSNPP	2) Layered SNPP List

Table 10.75: UML Class MultiLayerSNPP (derived from *CommonResourceInfo*) Operations

Operation Name	Requirement
setTNAName	{Requirement II. 379}

10.1.1.65 MultiLayerSNPPLink

Table 10.76: UML Class MultiLayerSNPPLink (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 120}*

Attribute Name	*{Requirement I. 121}*
association to MultiLayerRoutingArea	1) aEnd MLRA Name
association to MultiLayerRoutingArea	2) zEnd MLRA Name
association to LayeredSNPPLink	3) Layered SNPP Link List
aTNAGroupName	4) aTNA Group Name
zTNAGroupName	5) zTNA Group Name
aTNAName	6) aTNA Name
zTNAName	7) zTNA Name
association to Capacity	8) Capacity
direction	9) Direction
interfaceType	10) Interface Type
signallingProtocol	11) Signaling protocol
cost	12) Cost
discovered	13) Discovered
availability	14) Availability
linkSRG	15) Link Shared Risk Group
signallingEnabled	16) Signaling enabled
signallingControllerIdentifier	17) Signaling controller Identifier
signallingParameters	18) Signaling parameters

Table 10.77: UML Class MultiLayerSNPPLink (derived from *CommonResourceInfo*) Operations

Operation Name	Requirement
assignSignallingController	{Requirement II. 373}
deassignSignallingController	{Requirement II. 374}
setSignallingProtocolAndParameters	{Requirement II. 375}
modifySignallingProtocolParameters	{Requirement II. 375}
enableSignalling	{Requirement II. 376}
disableSignalling	{Requirement II. 377}
getAvailableCapacity	{Requirement II. 366}

Table 10.77: UML Class MultiLayerSNPPLink (derived from *CommonResourceInfo*) Operations

Operation Name	Requirement
setTNAName	{Requirement II. 378}

10.1.1.66 MultiLayerSubnetwork

Table 10.78: UML Class MultiLayerSubnetwork (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 012}*

Attribute Name	*{Requirement I. 013}*
subnetworkType	1) Type
supportedRates	2) Supported SNC layer rate(s)
networkAccessDomain	3) Network Access Domain

Table 10.79: UML Class MultiLayerSubnetwork (derived from *CommonResourceInfo*) Operations

Operation Name	Requirement
activateSNC	{Requirement II. 086}
checkValidSNC	{Requirement II. 085}
createAndActivateSNC	{Requirement II. 088}
createModifiedSNC	{Requirement II. 245}
createSNC	{Requirement II. 082}
createSNC	{Requirement II. 083}
createSNC	{Requirement II. 153}
createTPPool	{Requirement II. 264}
deactivateAndDeleteSNC	{Requirement II. 094}
deactivateAndDeleteSNC	{Requirement II. 096}
deactivateSNC	{Requirement II. 090}
deleteSNC	{Requirement II. 092}
deleteTPPool	{Requirement II. 266}
establishCall	{Requirement II. 392}
getAllCallsAndSNCs	{Requirement II. 382}
getAllCallsAndSNCsWithME	{Requirement II. 412}
getAllCallsAndSNCsWithTP	{Requirement II. 413}
getAllEdgePointNames	{Requirement II. 038}

Table 10.79: UML Class MultiLayerSubnetwork (derived from *CommonResourceInfo*) Operations

Operation Name	Requirement
getAllEdgePointNames	{Requirement II. 040}
getAllEdgePoints	{Requirement II. 037}
getAllEdgePoints	{Requirement II. 039}
getAllFixedSubnetworkConnectionNames	{Requirement II. 180}
getAllFixedSubnetworkConnectionNamesWithTP	{Requirement II. 182}
getAllFixedSubnetworkConnections	{Requirement II. 179}
getAllFixedSubnetworkConnectionsWithTP	{Requirement II. 181}
getAllManagedElementNames	{Requirement II. 005}
getAllManagedElements	{Requirement II. 004}
getAllSubnetworkConnectionNames	{Requirement II. 019}
getAllSubnetworkConnectionNamesWithTP	{Requirement II. 023}
getAllSubnetworkConnections	{Requirement II. 018}
getAllSubnetworkConnections	{Requirement II. 020}
getAllSubnetworkConnectionsWithTP	{Requirement II. 022}
getAllTopologicalLinkNames	{Requirement II. 016}
getAllTopologicalLinks	{Requirement II. 015}
getAllTPPoolNames	{Requirement II. 032}
getAllTPPools	{Requirement II. 031}
getAssociatedTP	{Requirement II. 049}
getCall	{Requirement II. 414}
getCallAndSNCs	{Requirement II. 384}
getSNC	{Requirement II. 024}
getSNCsByUserLabel	{Requirement II. 025}
getTopologicalLink	{Requirement II. 017}
getTPPool	{Requirement II. 268}
modifyCall	{Requirement II. 395}
modifySNC	{Requirement II. 257}
modifyTPPool	{Requirement II. 267}
releaseCall	{Requirement II. 394}
swapSNC	{Requirement II. 258}

10.1.1.67 NMSSession

Table 10.80: UML Class NMSSession (derived from *Session*) Operations *{Requirement II. 177}*

Operation Name	Requirement
eventLossOccurred	{Requirement II. 177}
eventLossCleared	{Requirement II. 177}
alarmLossOccurred	{Requirement II. 177}

10.1.1.68 ObjectInformation

derived from EventInformation, contains the Attribute(s) of the object (Refer to {Requirement I. 042}, {Requirement I. 044} and {Requirement I. 045}).

10.1.1.69 PerformanceManager

Table 10.81: UML Class PerformanceManager Operations

Operation Name	Requirement
clearPMData	{Requirement II. 132}
createTCAParameterProfile	{Requirement II. 236}
deleteTCAParameterProfile	{Requirement II. 240}
disablePMData	{Requirement II. 121}
disableTCA	{Requirement II. 122}
enablePMData	{Requirement II. 121}
enableTCA	{Requirement II. 122}
getAllCurrentPMData	{Requirement II. 131}
getAllPMPNames	{Requirement II. 222}
getAllPMPs	{Requirement II. 221}
getAllTCAParameterProfileNames	{Requirement II. 233}
getAllTCAPrameterProfiles	{Requirement II. 273}
getHistoryPMData	{Requirement II. 128}
getHistoryPMData	{Requirement II. 129}
getHistoryPMData	{Requirement II. 130}
getHoldingTime	{Requirement II. 124}
getMEPMCapabilities	{Requirement II. 054}
getTCAParameterProfile	{Requirement II. 234}
getTPHistoryPMData	{Requirement II. 163}
setTCAParameterProfile	{Requirement II. 238}

10.1.1.70 PerformanceMonitoringPoint

Table 10.82: UML Class PerformanceMonitoringPoint (derived from *CommonResourceInfo*) Attributes *{Requirement I. 084}*

Attribute Name	*{Requirement I. 085}*
layerRate	1) Layer rate
pmLocation	2) Location
granularity	3) Granularity
supervisionState	4) Supervision state

Table 10.82: UML Class PerformanceMonitoringPoint (derived from *CommonResourceInfo*) Attributes *{Requirement I. 084}*

Attribute Name	*{Requirement I. 085}*
monitoringState	5) Monitoring state
association to PMParameter	6) PM parameters

10.1.1.71 PhysicalTerminationPoint

Table 10.83: UML Class PhysicalTerminationPoint (derived from *TerminationPoint*) Operations

Operation Name	Requirement
getAllSupportingEquipment	{Requirement II. 052}
getAllSupportingEquipmentNames	{Requirement II. 053}
getContainingPGNames	{Requirement II. 289}

10.1.1.72 PMCurrentData

Table 10.84: UML Class PMCurrentData (derived from *PMData*) Attributes

Attribute Name	*{Requirement II. 282}*
retrievalTime	3) Retrieval time

10.1.1.73 PMData

Table 10.85: UML Class PMData Attributes - *{Requirement II. 282}*

Attribute Name	*{Requirement II. 282}*
layerRate	1) Layer rate
granularity	2) Granularity
association to PMCurrentData	3) Retrieval time
association to PMMeasurement	4) PM parameter measurements

10.1.1.74 PMHistoricData

Table 10.86: UML Class PMHistoricData (derived from *PMData*) Attributes - *{Requirement II. 284}*

Attribute Name	*{Requirement II. 284}*
userLabel	5) User label

10.1.1.75 PMHistoricMeasurement

Table 10.87: UML Class PMHistoricMeasurement(derives from *PMMeasurement*) Attributes - *{Requirement II. 284}*

Attribute Name	*{Requirement II. 284}*
periodEndTime	6) Period end time
monitoredTime	7) Monitored time

Table 10.87: UML Class PMHistoricMeasurement(derives from *PMMeasurement*) Attributes - *{Requirement II. 284}*

Attribute Name	***{Requirement II. 284}***
numberOfPeriods	8) Number of periods

10.1.1.76 PMMeasurement

Table 10.88: UML Class PMMeasurement Attributes - *{Requirement II. 282}*

Attribute Name	*{Requirement II. 282}*
pmParName	1) PM parameter
pmLocation	2) Location
value	3) Value
unit	4) Measurement unit
status	5) Status

10.1.1.77 PMParameter

Table 10.89: UML Class PMParameter Attributes - *{Requirement I. 085}*

Attribute Name	*{Requirement I. 085}*
pmParameterName	6) PM parameters

10.1.1.78 PMPStateChangeInformation

Table 10.90: UML Class PMPStateChangeInformation derived from *Event*) Attributes *{Requirement I. 083}*

Attribute Name	*{Requirement I. 083}*
pmpNameList	1) PMP name(s)
attributeList	2) Attribute value(s)
emsTime	3) EMS timestamp
neTime	4) NE timestamp

10.1.1.79 PMThreshold

Table 10.91: UML Class PMThreshold Attributes - *{Requirement I. 099}*

Attribute Name	
thresholdType	1) Threshold type
triggerFlag	2) Trigger
value	3) Value
unit	4) Measurement units

10.1.1.80 ProtectionGroup

Table 10.92: UML Class ProtectionGroup (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 034}*

Attribute Name	*{Requirement I. 066}*
protectionGroupType	1) Type
protectionSchemeState	2) Protection scheme state
reversionMode	3) Reversion mode
rate	4) Layer Rate
association to a list of PhysicalTerminationPoint	5) Protection related PTPs
pgpParameters	6) PG parameters
apsProtocolType	7) APS protocol type
asapPointer	8) Alarm severity assignment profile

Table 10.93: UML Class ProtectionGroup (derived from *CommonResourceInfo*) Operations

Operation Name	
performProtectionCommand	{Requirement II. 116}
retrieveSwitchData	{Requirement II. 117}

10.1.1.81 ProtectionSwitch

Table 10.94: UML Class ProtectionSwitch (derived from *Event*) Attributes *{Requirement I. 046}*

Attribute Name	*{Requirement I. 046}*
protectionType	1) Type
switchReason	2) Switch reason
layer	3) Layer rate
groupName	4) PG
protectedTP	5) Protected TP
switchAwayFromTP	6) Switch away from TP
switchToTP	7) Switch to TP

10.1.1.82 Route

Table 10.95: UML Class Route Attributes - *{Requirement I. 021}*

Attribute Name	*{Requirement I. 089}*
routeId	1) Identifier
association to CrossConnect	2) Contained XCs
intended	3) Intended
actualState	4) Actual state
administrativeState	5) Administrative state
inUseBy	6) In use by
exclusive	7) Exclusive
additionalInfo	8) Additional information

10.1.1.83 RouteCreateData

Table 10.96: UML Class RouteCreateData Attributes - *{Requirement II. 242}*

Attribute Name	*{Requirement II. 242}*
operation parameter	1) SNC or Connection name
operation parameter	2) Grade of impact
operation parameter	3) EMS freedom level
intended	4) Intended
exclusive	5) Exclusive

Table 10.96: UML Class RouteCreateData Attributes - *{Requirement II. 242}*

Attribute Name	***{Requirement II. 242}***
association to CrossConnect	6) Routing constraint data
fullRoute	7) Complete route
additionalCreationInfo	8) Additional information

10.1.1.84 Session

Table 10.97: UML Class - Session Operations

Operation Name	
ping	
endSession	

10.1.1.85 SessionFactory

Table 10.98: UML Class - SessionFactory Operations - *{Requirement II. 141}*

Operation Name	
getSession	{Requirement II. 141}

10.1.1.86 SNCCreateData

Table 10.99: UML Class - SNCCreateData Attributes *{Requirement II. 084}*

Attribute Name	*{Requirement II. 084}*
userLabel	1) User label
forceUniqueness	2) User label uniqueness
owner	3) Owner
direction	4) Directionality
staticProtectionLevel	5) Static protection level
protectionEffort	6) Protection effort
sncType	7) SNC Type
layerRate	8) Layer Rate
associations to: CrossConnect, TerminationPoint, ManagedElement, SubnetworkConnection, TopologicalLink, GroupTerminationPoint	9) Routing constraint data
fullRoute	10) Complete route
networkRouted	11) Network routed
rerouteAllowed	12) Reroute allowed
networkReroute	13) Network reroute
revertive	14) Revertive

Table 10.99: UML Class - SNCCreateData Attributes *{Requirement II. 084}*

Attribute Name	*{Requirement II. 084}*
priority	15) Priority
intendedRouteExclusive	16) Exclusive intended route
associations to: ConnectionTerminationPoint, FloatingTerminationPoint	17) aEnd TP(s)
associations to: ConnectionTerminationPoint, FloatingTerminationPoint	18) zEnd TP(s)
additionalCreationInfo	19) Additional information
bundledSNCIndicator	20) Bundled SNC
mustRemoveGTPs	21) GTP deletion
alarmReportingIndication	22) Alarm reporting
asapPointer	23) Alarm severity assignment profile
aEndPointsRole	24) aEnd point role
zEndPointsRole	25) zEnd point role
networkAccessDomain	26) Network Access Domain
operation parameter	27) Grade of impact
operation parameter	28) EMS freedom level

10.1.1.87 SNCModifyData

Table 10.100: UML Class - SNCModifyData (derived from *SNCCreateData*) Attributes - *{Requirement II. 246}*

Attribute Name	***{Requirement II. 246}***
operation parameter	1) SNC name
operation parameter	2) Route identifier
modifyType	3) Modification type
retainOldSNC	4) Retain SNC
modifyServerLayersAllowed	5) Modify server layers
addedOrNewRoute	6) Added or new route
removedRoute	7) Removed route
operation parameter	8) Termination Point (TP)s to modify

10.1.1.88 SNCRouteChangeData

Table 10.101: UML Class - SNCRouteChangeData (derived from *EventInformation*) Attributes - *{Requirement I. 059}*

Attribute Name	***{Requirement I. 059}***
routeChangeState	1) Route change state
association to Route	2) Route

10.1.1.89 SNP

Table 10.102: UML Class - SNP Attributes - *{Requirement I. 128}*

Attribute Name	***{Requirement I. 128}***
snppId	1) SNP ID
association to ConnectionTerminationPoint	2) TP Name
tnaName	3) TNA Name

10.1.1.90 SNPP

Table 10.103: UML Class - SNPP Attributes - *{Requirement I. 127}*

Attribute Name	***{Requirement I. 127}***
snppId	1) SNPP ID
association to SNP	2) SNP list
raId	3) Routing Area ID
tnaName	4) TNA Name
tnaGroupName	5) TNAGroup Name

10.1.1.91 SNPPLink

Table 10.104: UML Class - SNPPLink Attributes - *{Requirement I. 124}*

Attribute Name	***{Requirement I. 125}***
snppLinkId	1) SNPP Link ID
association to SNPP	2) aEnd SNPP
association to SNPP	3) zEnd SNPP

10.1.1.92 SoftwareBackupStatus

Table 10.105: UML Class SoftwareBackupStatus (derived from *Event*) Attributes - *{Requirement I. 086}*

Attribute Name	*{Requirement I. 086}*
meName	1) NE Name
backupStatus	2) Backup status
emsTime	3) EMS timestamp
neTime	4) NE timestamp

10.1.1.93 SoftwareManager

Table 10.106: UML Class SoftwareManager Operations

Operation Name	Requirement
backupME	{Requirement II. 229}
getMEBackupStatus	{Requirement II. 230}
abortMEBackup	{Requirement II. 231}
getBackupList	{Requirement II. 232}

10.1.1.94 SubnetworkConnection

Table 10.107: UML Class SubnetworkConnection (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 014}*

Attribute Name	*{Requirement I. 015}*
sncState	1) State
direction	2) Directionality
rates	3) Layer rate
staticProtectionLevel	4) Static protection level
sncType	5) Type
associations to TerminationPoint, GroupTerminationPoint	6) aEnd TPs
associations to TerminationPoint, GroupTerminationPoint	7) zEnd TPs
networkRouted	8) Network routed
rerouteAllowed	9) Reroute allowed

Table 10.107: UML Class SubnetworkConnection (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 014}*

Attribute Name	*{Requirement I. 015}*
networkReroute	10) Network reroute
revertive	11) Revertive
networkAccessDomain	12) Network Access Domain
alarmReportingIndication	13) Alarm reporting
correlationId	14) Correlation identifier
bundledSNCIndicator	15) Bundled SNC
mustRemoveGTPs	16) GTP deletion
fixed	17) Fixed
asapPointer	18) Alarm severity assignment profile
retainOldSNC	19) Retain SNC
priority	20) Priority
aEndPointsRole	21) aEnd point role
zEndPointsRole	22) zEnd point role
routeGroupLabel	23) Route Group Label
association to Call	24) CallID

Table 10.108: UML Class SubnetworkConnection (derived from *CommonResourceInfo*) Operations

Operation Name	Requirement
addRoute	{Requirement II. 241}
getBackupRoutes	{Requirement II. 256}
getBackupRoutes	{Requirement II. 260}
getIntendedRoute	{Requirement II. 261}
getRoute	{Requirement II. 026}
getRouteAndTopologicalLinks	{Requirement II. 218}
removeRoute	{Requirement II. 243}
setAlarmReportingOff	{Requirement II. 160}
setAlarmReportingOn	{Requirement II. 159}
setIntendedRoute	{Requirement II. 251}
setRoutesAdminState	{Requirement II. 249}

Table 10.108: UML Class SubnetworkConnection (derived from *CommonResourceInfo*) Operations

Operation Name	Requirement
switchRoute	{Requirement II. 247}

10.1.1.95 Supplier

No attributes, no operations.

10.1.1.96 SwitchData

Table 10.109: UML Class SwitchData Attributes *{Requirement II. 195}*

Attribute Name	*{Requirement II. 195}*
protectionType	1) Type
switchReason	2) Switch reason
layerRate	3) Layer rate
additionalInfo	4) PG
association to ProtectionGroup	5) Protected TP
association to TerminationPoint	6) Switch away from TP
association to TerminationPoint	7) Switch to TP
additionalInfo	8) Additional information

10.1.1.97 TCAParameter

Table 10.110: UML Class TCAParameter Attributes - *{Requirement I. 088}*

Attribute Name	*{Requirement I. 035}*
pmParameterName	1) Name
granularity	2) Granularity
pmLocation	3) Location
thresholdType	4) Threshold type
triggerFlag	5) Trigger
value	6) Value
unit	7) Measurement units

10.1.1.98 TCAParameterProfile

Table 10.111: UML Class TCAParameterProfile (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 067}*

Attribute Name	*{Requirement I. 087}*
layerRate	1) Layer rate
associatedTPs	2) Associated TPs
association to TCAParameter	3) TCA Parameters

Table 10.112: UML Class TCAParameterProfile (derived from *CommonResourceInfo*) Operations

Operation Name	Requirement
getAssociatedTPs	{Requirement II. 285}

10.1.1.99 TCProfileCreateModifyData

Table 10.113: UML Class TCProfileCreateModifyData Attributes - *{Requirement II. 334}*

Attribute Name	*{Requirement II. 334}*
userLabel	1) User label
forceUniqueness	2) User label uniqueness
owner	3) Owner
association to TransmissionParameters	4) Layered traffic conditioning parameters
additionalInfo	5) Additional information

10.1.1.100 TDCreateData

Table 10.114: UML Class TDCreateData Attributes - *{Requirement II. 098}*

Attribute Name	*{Requirement II. 098}*
userLabel	1) User label
forceUniqueness	2) User label uniqueness
owner	3) Owner
serviceCategory	4) Service Category
conformanceDefinition	5) Conformance Profile.
trafficParameters	6) Traffic Parameters
additionalCreationInfo	7) Additional information

10.1.1.101 TerminationPoint

Table 10.115: UML Class TerminationPoint (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 004}*

Attribute Name	*{Requirement I. 005}*
direction	1) Directionality
tpProtectionAssociation	2) Protection association
edgePoint	3) Edge Point
networkAccessDomain	4) Network Access Domain
equipmentProtected	5) Equipment protected
egressTMDState	6) Ingress TMD state
ingressTMDState	7) Egress TMD state
association to: GroupTerminationPoint or TPPool	8) GTP or TPPool
association to TransmissionParameters	9) Layered transmission parameters
association to ingress TransmissionDescriptor	10) Ingress TMD
association to egress TransmissionDescriptor	11) Egress TMD
association to TCAParameterProfile	12) TCA parameter profile
association to AlarmSeverityAssignmentProfile	13) Alarm severity assignment profile
association to PerformanceMonitoringPoint	14) Performance monitoring point

Table 10.116: UML Class TerminationPoint (derived from *CommonResourceInfo*) Operations

Operation Name	Requirement
getActiveMaintenanceOperations	{Requirement II. 138}
getContainedCurrentCTPNames	{Requirement II. 044}
getContainedCurrentCTPs	{Requirement II. 043}
getContainedInUseCTPNames	{Requirement II. 046}
getContainedInUseCTPs	{Requirement II. 045}
getContainedPotentialCTPNames	{Requirement II. 042}
getContainedPotentialCTPs	{Requirement II. 041}
getPotentialFixedCCs	{Requirement II. 185}
getTCAParameter	{Requirement II. 055}
getTPGroupingRelationships	{Requirement II. 030}
getTransmissionParameters	{Requirement II. 291}
performMaintenanceOperation	{Requirement II. 137}
setTCAParameter	{Requirement II. 125}
setTCAParameterProfilePointer	{Requirement II. 235}
setTransmissionDescriptorAssociation	{Requirement II. 194}
verifyTMDAssignment	{Requirement II. 278}

10.1.1.102 ThresholdCrossingAlertInformation

Table 10.117: UML Class ThresholdCrossingAlertInformation (derived from *EventInformation*) Attributes - *{Requirement I. 047}*

Attribute Name	*{Requirement I. 047}*
nativeEMSName	1) Native EMS name
isClearable	2) Clearable
perceivedSeverity	3) Perceived severity
layerRate	4) Layer rate
granularity	5) Granularity
pmParameterName	6) Parameter name
pmLocation	7) Parameter location
thresholdType	8) Threshold type
value	9) Value
unit	10) Measurement units
acknowledgeIndication	11) Acknowledgement

10.1.1.103 TLCreateData

Table 10.118: UML Class TLCreateData Attributes - *{Requirement II. 169}*

Attribute Name	*{Requirement II. 169}*
userLabel	1) User label
forceUniqueness	2) User label uniqueness
owner	3) Owner
direction	4) Directionality
association to TerminationPoint	5) aEnd Termination Point (TP)
association to TerminationPoint	6) zEnd Termination Point (TP)
layerRate	7) Layer Rate
networkAccessDomain	8) Network Access Domain
alarmReportingIndication	9) Alarm reporting
asapPointer	10) Alarm severity assignment profile
additionalCreationInfo	11) Additional information.

10.1.1.104 TopologicalLink

Table 10.119: UML Class TopologicalLink (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 010}*

Attribute Name	*{Requirement I. 011}*
direction	1) Directionality
association to TerminationPoint	2) aEnd Termination Point (TP)
association to TerminationPoint	3) zEnd Termination Point (TP)
layerRate	4) Layer rate
networkAccessDomain	5) Network Access Domain
alarmReportingIndication	6) Alarm reporting

Table 10.120: UML Class TopologicalLink (derived from *CommonResourceInfo*) Operations

Operation Name	Requirement
setAlarmReportingOn	{Requirement II. 161}
setAlarmReportingOff	{Requirement II. 162}

10.1.1.105 TPData

Table 10.121: UML Class TPData Attributes

Attribute Name	
tpMappingMode	

10.1.1.106 TPPool

Table 10.122: UML Class TPPool Attributes - *{Requirement I. 009}*

Attribute Name	*{Requirement I. 094}*
numberOfMembers	1) Contained members
association to GroupTerminationPoint, TerminationPoint	2) Number of members
numberOfIdleMembers	3) Number of idle members
association to CommonTransmissionParameters	4) Layered transmission parameters
descriptionOfUse	5) Description of use

Table 10.123: UML Class TPPool (derived from *CommonResourceInfo*) Operations - *{Requirement I. 009}*

Operation Name	Requirement
getTPGroupingRelationships	{Requirement II. 028}

10.1.1.107 TPPoolCreateData

Table 10.124: UML Class TPPoolCreateData Attributes - *{Requirement II. 265}*

Attribute Name	***{Requirement II. 265}***
userLabel	1) User label
forceUniqueness	2) User label uniqueness
owner	3) Owner
operation parameter	4) Containing Subnetwork
containedMembers	5) Contained members
association to TransmissionParameters	6) Layered transmission parameters
descriptionOfUse	7) Description of use
additionalCreationInfo	8) Additional information.

10.1.1.108 TrafficConditioningProfile

Table 10.125: UML Class TrafficConditioningProfile (derived from *CommonResourceInfo*) Attributes - *{Requirement I. 107}*

Attribute Name	***{Requirement I. 108}***
networkAccessDomain	1) Network access domain
defaultProfile	2) Default profile
association to TransmissionParameters	3) Layered traffic conditioning parameters

Table 10.126: UML Class TrafficConditioningProfile (derived from *CommonResourceInfo*) Operations

Operation Name	**Requirement**
getTCProfileAssociatedTPs	{Requirement II. 332}

10.1.1.109 TrafficDescriptor

Table 10.127: UML Class TrafficDescriptor Attributes - *{Requirement I. 025}*

Attribute Name	***{Requirement I. 063}***
serviceCategory	1) Service category
conformanceDefn	2) Conformance profile
trafficParameters	3) Traffic parameters

Table 10.128: UML Class TrafficDescriptor Operations

Operation Name	**Requirement**
getAssociatedCTPs	{Requirement II. 029}

10.1.1.110 TransmissionDescriptor

Table 10.129: UML Class - TransmissionDescriptor (derived from *CommonResourceInfo*) - *{Requirement I. 078}*

Attribute Name	***{Requirement I. 079}***
association to TransmissionParameters	1) Layered transmission parameters
additionalTPInfo	2) Additional Object information
externalRepresentationReference	3) External representation

Table 10.130: UML Class - TransmissionDescriptor (derived from *CommonResourceInfo*)

Operation Name	**Requirement**
getAssociatedTPs	{Requirement II. 186}
getAllAssociatedMFDs	{Requirement II. 301}
getTransmissionParameters	{Requirement II. 352}

10.1.1.111 TransmissionDescriptorCreateData

Table 10.131: UML Class TransmissionDescriptorCreateData Attributes - *{Requirement II. 191}*

Attribute Name	*{Requirement II. 191}*
userLabel	1) User label
forceUniqueness	2) User label uniqueness
owner	3) Owner
association to TransmissionParameters	4) Layered transmission parameters
additionalTPInfo	5) Additional Object information
externalRepresentationReference	7) External representation
additionalCreationInfo	8) Additional information

10.1.1.112 TransmissionParameters

Table 10.132: UML Class - TransmissionParameters Attributes

Attribute Name	Requirement
layerRate	{Requirement I. 022}
transmissionParameters	{Requirement I. 024}

Table 10.133: UML Class - TransmissionParameters Operations

Operation Name	Requirement
setTransmissionParameters	{Requirement II. 072}

10.1.1.113 Version

Table 10.134: UML Class - Version Operations - *{Requirement I. 030}*

Operation Name	Requirement
getVersion	{Requirement II. 286}

10.2 Dynamic Model

The dynamic model can be found in the MTNM Information Agreement, NML-EML Interface, (TMF 608).

11 Traceability Matrices

These matrices show the traceability of requirement statements from Section 4, Section 5 and Section 6 to Use Cases from Section 7, Section 9 and Section 8.

11.1 Use Case versus Requirements

Table 11.1 provides a mapping between the Use Cases in Section 7, Section 9 and Section 8 and the requirements in Section 4, Section 5 and Section 6. Column 1 of Table 11.1 lists all the Use Cases and column 2 identifies the corresponding requirements.

Table 11.1: Use Case to Requirement Map

Use Case	Requirement
NMS-EMS Session Management Use Cases	
Use Case 7.2.1: EMS (Re)starts	{Requirement I. 041}
	{Requirements IV. 003}
Use Case 7.2.2: NMS creates a session with EMS	{Requirement I. 041}
	{Requirement II. 139}
	{Requirement II. 140}
	{Requirement II. 144}
	{Requirement II. 145}
Use Case 7.2.3: NMS retrieves the interface version used by the EMS	{Requirement I. 030}
	{Requirement I. 031}
Use Case 7.2.4: NMS closes a session with an EMS	{Requirement II. 139}
	{Requirement II. 140}
Use Case 7.2.5: EMS closes a session with an NMS	{Requirement II. 139}
	{Requirement II. 140}
Use Case 7.2.6: NMS detects that an EMS is unavailable	{Requirement II. 145}
	{Requirement III. 001}
	{Requirement III. 002}
Use Case 7.2.7: EMS detects that an NMS is unavailable	{Requirement I. 041}
	{Requirement II. 145}
	{Requirement III. 001}
	{Requirement III. 002}

Table 11.1: Use Case to Requirement Map

Use Case	Requirement
EMS-NE Session Management Use Cases	
Use Case 7.3.1: EMS loses communication to a Network Element	{Requirement I. 003}
	{Requirement II. 067}
Discovery and Inventory Use Cases	
Use Case 7.4.1: NMS registers to receive network updates information from the EMS	{Requirement II. 064}
	{Requirement II. 065}
	{Requirement II. 066}
	{Requirement II. 067}
Use Case 7.4.2: NMS resynchronizes its database with the EMS	{Requirement II. 001}
	{Requirement II. 002}
	{Requirement II. 004}
	{Requirement II. 006}
	{Requirement II. 011}
	{Requirement II. 008}
	{Requirement II. 012}
	{Requirement II. 015}
	{Requirement II. 018}
	{Requirement II. 020}
	{Requirement II. 022}
	{Requirement II. 024}
	{Requirement II. 026}
	{Requirement II. 031}
	{Requirement II. 033}
	{Requirement II. 040}
	{Requirement II. 041}
	{Requirement II. 043}
	{Requirement II. 047}
	{Requirement II. 050}
	{Requirement II. 056}
	{Requirement II. 058}

Table 11.1: Use Case to Requirement Map

Use Case	Requirement
	{Requirement II. 059}
	{Requirement II. 060}
Use Case 7.4.2: NMS resynchronizes its database with the EMS	{Requirement II. 001}
	{Requirement II. 002}
	{Requirement II. 004}
	{Requirement II. 006}
	{Requirement II. 011}
	{Requirement II. 008}
	{Requirement II. 012}
	{Requirement II. 015}
Use Case 7.4.3: NMS discovers the EMS network inventory	{Requirement II. 001}
	{Requirement II. 002}
	{Requirement II. 011}
	{Requirement II. 012}
	{Requirement II. 015}
	{Requirement II. 018}
	{Requirement II. 031}
	{Requirement II. 033}
	{Requirement II. 041}
	{Requirement II. 043}
	{Requirement II. 056}
	{Requirement II. 059}
	{Requirement II. 060}
	{Requirement II. 295}
	{Requirement II. 314}
	{Requirement II. 338}
	{Requirement II. 343}
Use Case 7.4.4: NMS queries EMS concerning inventory	{Requirement II. 002}
	{Requirement II. 004}
	{Requirement II. 006}
	{Requirement II. 011}

Table 11.1: Use Case to Requirement Map

Use Case	Requirement
	{Requirement II. 008}
	{Requirement II. 012}
	{Requirement II. 015}
	{Requirement II. 018}
	{Requirement II. 020}
	{Requirement II. 022}
	{Requirement II. 024}
	{Requirement II. 026}
	{Requirement II. 033}
	{Requirement II. 040}
Use Case 7.4.4: NMS queries EMS concerning inventory	{Requirement II. 047}
	{Requirement II. 050}
	{Requirement II. 052}
	{Requirement II. 056}
	{Requirement II. 058}
	{Requirement I. 100}
	{Requirement II. 292}
	{Requirement II. 295}
	{Requirement II. 296}
	{Requirement II. 297}
	{Requirement II. 300}
	{Requirement II. 301}
	{Requirement II. 303}
	{Requirement II. 304}
	{Requirement II. 314}
	{Requirement II. 315}
	{Requirement II. 316}
	{Requirement II. 317}
	{Requirement II. 318}
	{Requirement II. 330}
	{Requirement II. 332}

Table 11.1: Use Case to Requirement Map

Use Case	Requirement
	{Requirement II. 338}
	{Requirement II. 339}
	{Requirement II. 340}
	{Requirement II. 341}
	{Requirement II. 342}
	{Requirement II. 343}
Use Case 7.4.5: EMS notifies NMS of inventory change	{Requirement II. 064}
	{Requirement II. 065}
	{Requirement II. 066}
	{Requirement II. 067}
Provisioning Use Cases	
Use Case 7.5.1: NMS provisions the mapping mode of a CTP	{Requirement II. 068}
Use Case 7.5.2: NMS un-maps a server layer CTP	{Requirement II. 069}
Use Case 7.5.3: NMS provisions the User Label	{Requirement II. 075}
Use Case 7.5.4: NMS provisions the Owner	{Requirement II. 076}
Use Case 7.5.5: NMS provisions the Additional Information	{Requirement II. 223}
Use Case 7.5.6: NMS provisions the Native EMS Name	{Requirement II. 077}
Use Case 7.5.7: NMS Provisions the TP Transmission Parameters	{Requirement II. 072}
Use Case 7.5.8: NMS provisions alarm reporting on for a TP	{Requirement II. 108}
Use Case 7.5.9: NMS provisions alarm reporting off for a TP	{Requirement II. 109}
Use Case 7.5.10: NMS creates a Topological Link (TL)	{Requirement II. 168}
Use Case 7.5.11: NMS deletes a Topological Link (TL)	{Requirement II. 170}
Use Case 7.5.12: NMS creates a Transmission Descriptor (TMD)	{Requirement II. 190}
Use Case 7.5.13: NMS modifies a Transmission Descriptor (TMD) on a TP	{Requirement II. 194}
	{Requirement II. 277}
	{Requirement II. 312}
	{Requirement II. 313}
Use Case 7.5.14: NMS sets the Transmission Descriptor (TMD) Profile Pointer	{Requirement II. 194}
Use Case 7.5.15: NMS modifies a Transmission Descriptor (TMD)	{Requirement I. 102}
	{Requirement II. 302}

Table 11.1: Use Case to Requirement Map

Use Case	Requirement
	{Requirement II. 319}
	{Requirement II. 344}
	{Requirement II. 352}
Use Case 7.5.16: NMS retrieves scoped Transmission Parameters	{Requirement II. 192}
Use Case 7.5.17: NMS deletes a Transmission Descriptor (TMD)	{Requirement II. 192}
Use Case 7.5.18: NMS creates a Group Termination Point (GTP)	{Requirement II. 164}
	{Requirement II. 165}
Use Case 7.5.19: NMS modifies a Group Termination Point (GTP)	{Requirement II. 167}
Use Case 7.5.20: NMS deletes a Group Termination Point (GTP)	{Requirement II. 166}
Use Case 7.5.21: NMS creates a Termination Point Pool (TP Pool)	{Requirement II. 264}
	{Requirement II. 265}
Use Case 7.5.22: NMS modifies a Termination Point Pool (TP Pool)	{Requirement II. 267}
Use Case 7.5.23: NMS deletes a Termination Point Pool (TP Pool)	{Requirement II. 266}
Use Case 7.5.24: NMS assigns an Alarm Severity Assignment Profile (ASAP) to a CTP	{Requirement II. 201}
Use Case 7.5.25: NMS locks (in a forced response deferred/graceful fashion) or unlocks a number of IMA links to modify the transport capacity of the corresponding fixed IMA group	
Use Case 7.5.26: NMS requests dynamic provisioning of an IMA group by the EMS subject to a prescribed bandwidth that is communicated as number and connectable layer rate of the IMA links	
Use Case 7.5.27: NMS provisions the IMA virtual link between two peer IMA groups as a topological link	
Use Case 7.5.28: NMS unprovisions an IMA virtual link between IMA groups	
Use Case 7.5.29: NMS provisions or re-provisions a single DSL line	
Use Case 7.5.30: NMS provisions multiple DSL lines by using TMDs	
Call Management Use Cases	
Use Case 7.6.1: (UC_20) NMS establishes a Call	{Requirement II. 392}
	{Requirement II. 393}
Use Case 7.6.2: (UC_21) NMS releases a Call	{Requirement II. 394}
Use Case 7.6.3: (UC_22) NMS adds one or more Connections to a Call	{Requirement II. 397}
	{Requirement II. 398}
Use Case 7.6.4: (UC_23) NMS removes one or more Connections from a Call	{Requirement II. 399}
Use Case 7.6.5: NMS modifies a Call	{Requirement II. 395}

Table 11.1: Use Case to Requirement Map

Use Case	Requirement
	{Requirement II. 396}
Use Case 7.6.6: (UC_7a) NMS retrieves all the Connections and SNCs for a specific list of Calls	{Requirement II. 404}
	{Requirement II. 405}
	{Requirement II. 406}
	{Requirement II. 407}
	{Requirement II. 408}
	{Requirement II. 409}
	{Requirement II. 410}
	{Requirement II. 411}
Use Case 7.6.7: (UC_33) NMS requests EMS to set/modify diversity and co-routing parameters of a Call	{Requirement II. 400}
	{Requirement II. 401}
	{Requirement II. 402}
	{Requirement II. 403}
Connection Management Use Cases	
Use Case 7.7.1: NMS creates a Subnetwork Connection (SNC)	{Requirement II. 082}
	{Requirement II. 083}
	{Requirement II. 084}
Use Case 7.7.2: NMS activates a Subnetwork Connection (SNC)	{Requirement II. 086}
	{Requirement II. 087}
Use Case 7.7.3: NMS creates and activates a Subnetwork Connection (SNC)	{Requirement II. 088}
	{Requirement II. 089}
Use Case 7.7.4: NMS adds a route to a Subnetwork Connection (SNC)	{Requirement II. 241}
	{Requirement II. 242}
Use Case 7.7.5: NMS removes a route from a Subnetwork Connection (SNC)	{Requirement II. 243}
	{Requirement II. 244}
Use Case 7.7.6: NMS creates-modifies the route of a Subnetwork Connection (SNC)	{Requirement II. 245}
	{Requirement II. 246}
Use Case 7.7.7: NMS deactivates a Subnetwork Connection (SNC)	{Requirement II. 090}
	{Requirement II. 091}

Table 11.1: Use Case to Requirement Map

Use Case	Requirement
Use Case 7.7.8: NMS deletes a Subnetwork Connection (SNC)	{Requirement II. 092}
	{Requirement II. 093}
Use Case 7.7.9: NMS deactivates and deletes a Subnetwork Connection (SNC)	{Requirement II. 094}
	{Requirement II. 095}
Use Case 7.7.10: EMS reroutes a Subnetwork Connection (SNC)EMS reroutes a Subnetwork Connection (SNC)	{Requirement II. 084}
Use Case 7.7.11: NMS queries EMS Connection Management Mode	{Requirement II. 100}
Use Case 7.7.12: NMS creates and activates a point-to-point Ethernet Service using fragmentation	
Use Case 7.7.13: NMS modifies a point-to-point Ethernet Service with fragmentation	
Use Case 7.7.14: NMS deletes a point-to-point Ethernet Service with fragmentation	
Use Case 7.7.15: NMS creates a flexible IMA group	
Use Case 7.7.16: NMS deletes a flexible IMA group	
Use Case 7.7.17: NMS modifies the transport capacity or the routing targets of a flexible IMA group	
Protection Management Use Cases	
Use Case 7.8.1: NMS retrieves all the Protection Groups of a Managed Element	{Requirement II. 059}
	{Requirement II. 174}
Use Case 7.8.2: Protection Switch Notification for Equipment, Trail and SNC Protection	{Requirement I. 046}
	{Requirement I. 074}
	{Requirement II. 115}
	{Requirement II. 175}
Use Case 7.8.3: NMS retrieves the protection switch information for Equipment, Trail and SNC Protection	{Requirement II. 114}
	{Requirement II. 175}
Use Case 7.8.4: NMS registers to receive protection switch notifications	{Requirement II. 115}
Use Case 7.8.5: NMS invokes protection switch lockout to SNCP	{Requirement II. 116}
Fault Management Use Cases	
Use Case 7.9.1: NMS reconciles active alarms from an EMS	{Requirement II. 111}
Use Case 7.9.2: NMS reconciles active alarms for a specified Managed Element	{Requirement II. 110}
Use Case 7.9.3: NMS registers to receive alarms or threshold crossing alerts from an EMS	{Requirement II. 103}

Table 11.1: Use Case to Requirement Map

Use Case	Requirement
Use Case 7.9.4: NMS registers to receive RCAIs only, raw alarms only, or both RCAIs and raw alarms from an EMS	{Requirement II. 298}
Use Case 7.9.5: EMS determines a more appropriate root cause than one previously reported	{Requirement II. 224}
Use Case 7.9.6: EMS Notifies NMS of Alarms or Threshold Crossing Alert (TCA)s	{Requirement II. 126}
Use Case 7.9.7: Alarm Acknowledgement in the NMS	{Requirement II. 155}
	{Requirement II. 157}
Use Case 7.9.8: Alarm Unacknowledgement in the NMS	{Requirement II. 156}
	{Requirement II. 158}
Use Case 7.9.9: Alarm Acknowledgement in the EMS	{Requirement II. 157}
Use Case 7.9.10: NMS reconciles Unacknowledged Active Alarms from an EMS	{Requirement II. 154}
Use Case 7.9.11: NMS reconciles Unacknowledged Active Alarms for a specified Managed Element	{Requirement II. 154}
Use Case 7.9.12: EMS discards an event to be sent to the NMS	{Requirement II. 177}
Use Case 7.9.13: EMS succeeds in forwarding an event to the NMS again	{Requirement II. 177}
Use Case 7.9.14: EMS sends a heartbeat notification to the NMS	{Requirement II. 178}
Equipment Use Cases	
Use Case 7.10.1: NMS unprovisions equipment	{Requirement II. 262}
Use Case 7.10.2: NMS provisions equipment	{Requirement II. 136}
Use Case 7.10.3: NMS provisions alarm reporting on/off for equipment	{Requirement II. 078}
Use Case 7.10.4: NMS provisions alarm reporting on/off for an equipment holder	{Requirement II. 078}
Craft Related Use Cases	
Use Case 7.11.1: Craft modifies TP Transmission Parameter(s) of a TP	{Requirement I. 044}
	{Requirement II. 066}
Use Case 7.11.2: Craft/EMS creates a cross-connect (XC) in a Network Element (NE)	{Requirement I. 042}
	{Requirement I. 045}
	{Requirement II. 064}
	{Requirement II. 067}
Use Case 7.11.3: Craft/EMS Deletes a cross-connect (XC) in a Network Element (NE)	{Requirement I. 042}
	{Requirement I. 045}
	{Requirement II. 065}
	{Requirement II. 067}

Table 11.1: Use Case to Requirement Map

Use Case	Requirement
Use Case 7.11.4: Craft inserts a plug-in card	{Requirement I. 042}
	{Requirement I. 045}
	{Requirement II. 064}
	{Requirement II. 067}
	{Requirement II. 133}
	{Requirement II. 134}
Use Case 7.11.5: Craft removes a plug-in card	{Requirement I. 043}
	{Requirement I. 045}
	{Requirement II. 065}
	{Requirement II. 067}
Use Case 7.11.6: Craft/EMS creates a Protection Group	{Requirement I. 042}
	{Requirement II. 059}
	{Requirement II. 064}
	{Requirement II. 112}
Performance Management Use Cases	
Use Case 7.12.1: NMS activates collection of Performance Monitoring Data (PMD) for a specified set of TPs	{Requirement II. 121}
	{Requirement II. 132}
Use Case 7.12.2: NMS deactivates collection of Performance Monitoring Data (PMD) for a specified set of TPs	{Requirement II. 121}
Use Case 7.12.3: NMS retrieves current Performance Monitoring Data (PMD) for a specified set of TPs	{Requirement II. 131}
Use Case 7.12.4: NMS retrieves the storage time of 24hr and 15min Performance Monitoring Data (PMD) records	{Requirement II. 124}
Use Case 7.12.5: NMS retrieves PM capabilities of a Managed Element (ME)	{Requirement II. 054}
	{Requirement II. 221}
	{Requirement II. 222}
Use Case 7.12.6: NMS retrieves historical Performance Monitoring Data (PMD) for a specified set of TPs	{Requirement II. 128}
	{Requirement II. 129}
	{Requirement II. 130}
Use Case 7.12.7: NMS sets PM thresholds on a TP	{Requirement I. 054},
	{Requirement II. 125}

Table 11.1: Use Case to Requirement Map

Use Case	Requirement
Use Case 7.12.8: NMS retrieves PM threshold settings from a TP	{Requirement I. 054}
	{Requirement II. 055}
Use Case 7.12.9: NMS enables Threshold Crossing Alerts (TCA) for a specified set of TPs	{Requirement II. 122}
	{Requirement II. 126}
Use Case 7.12.10: NMS disables Threshold Crossing Alerts (TCA) for a specified set of TPs	{Requirement II. 122}
	{Requirement II. 122}
Use Case 7.12.11: On demand retrieval of historical Performance Monitoring Data (PMD) for a specified set of TPs	{Requirement II. 163}
	{Requirement II. 221}
Use Case 7.12.13: NMS modifies TCA Parameter Profile	{Requirement II. 238}
	{Requirement II. 239}
Use Case 7.12.14: NMS configures TCA Parameter Profile Pointer	{Requirement II. 238}
	{Requirement II. 239}
GUI Cut-Through Use Cases	
Use Case 7.13.1: NMS retrieves GUI Cut-Through window data	{Requirement II. 150}
Use Case 7.13.2: Client based GCT launch	{Requirement II. 151}
Use Case 7.13.3: Server based GCT launch	{Requirement II. 151}
	{Requirement II. 152}
ATM Provisioning Use Cases	
Use Case 7.14.1: NMS creates a Traffic Descriptor (TD)	{Requirement II. 097}
	{Requirement II. 098}
Use Case 7.14.2: NMS modifies a Traffic Descriptor (TD) on a VPCTP or VCCTP	{Requirement II. 074}
Use Case 7.14.3: NMS deletes a Traffic Descriptor (TD)	{Requirement II. 099}
ATM Connection Management Use Cases	
Use Case 7.15.1: NMS creates and activates an ATM Subnetwork Connection (SNC)	{Requirement II. 088}
	{Requirement II. 089}
Connectionless Management Use Cases	
Termination Point Management Use Cases	
Use Case 9.2.1: NMS creates a Floating Termination Point	{Requirement II. 293}
Use Case 9.2.2: NMS deletes a Floating Termination Point	{Requirement II. 294}

Table 11.1: Use Case to Requirement Map

Use Case	Requirement
Flow Domain Management Use Cases	
Use Case 9.3.1: NMS creates a Flow Domain	{Requirement II. 321}
	{Requirement II. 322}
Use Case 9.3.2: NMS deletes a Flow Domain	{Requirement II. 325}
Use Case 9.3.3: NMS modifies a Flow Domain	{Requirement II. 323}
	{Requirement II. 324}
Use Case 9.3.4: NMS associates Matrix Flow Domain(s) to a Flow Domain	{Requirement II. 328}
Use Case 9.3.5: NMS dissociates Matrix Flow Domain(s) to a Flow Domain	{Requirement II. 329}
Use Case 9.3.6: NMS associates Connectionless Port Termination Point(s) to a Flow Domain	{Requirement II. 326}
Use Case 9.3.7: NMS dissociates FD Edge Connectionless Port Termination Point(s) from a Flow Domain	{Requirement II. 327}
Matrix Flow Domain Management Use Cases	
Use Case 9.4.1: NMS creates a Matrix Flow Domain (MFD)	{Requirement II. 305}
	{Requirement II. 306}
Use Case 9.4.2: NMS deletes a Matrix Flow Domain (MFD)	{Requirement II. 309}
Use Case 9.4.3: NMS modifies a Matrix Flow Domain (MFD)	{Requirement II. 307}
	{Requirement II. 308}
Use Case 9.4.4: NMS assigns Connectionless Port Termination Point(s) to a Matrix Flow Domain	{Requirement II. 310}
Use Case 9.4.5: NMS un-assigns Connectionless Port Termination Point(s) from a Matrix Flow Domain	{Requirement II. 311}
Traffic Conditioning Profile Management Use Cases	
Use Case 9.5.1: NMS creates a Traffic Conditioning Profile	{Requirement II. 333}
	{Requirement II. 334}
Use Case 9.5.2: NMS deletes a Traffic Conditioning Profile	{Requirement II. 336}
Use Case 9.5.3: NMS modifies a Traffic Conditioning Profile	{Requirement II. 334}
	{Requirement II. 335}
Use Case 9.5.4: NMS configures Traffic Mapping Table	{Requirement II. 337}
Flow Domain Fragment Management Use Cases	
Use Case 9.6.1: NMS creates and activates a Flow Domain Fragment	{Requirement II. 345}
	{Requirement II. 346}

Table 11.1: Use Case to Requirement Map

Use Case	Requirement
Use Case 9.6.2: NMS deactivates and deletes a Flow Domain Fragment	{Requirement II. 349}
Use Case 9.6.3: NMS modifies a Flow Domain Fragment	{Requirement II. 347}
	{Requirement II. 348}
	{Requirement II. 350}
	{Requirement II. 351}
Control Plane Management Use Cases	
Discovery and Inventory Use Cases	
Use Case 8.2.1: (UC_1) EMS notifies of changes in network capacity (increases/decreases in capacity)	{Requirement II. 064}
	{Requirement II. 065}
	{Requirement II. 066}
	{Requirement II. 359}
	{Requirement II. 361}
Use Case 8.2.2: (UC_4) NMS discovers MultiLayer Routing Area (MLRA)s	{Requirement II. 354}
	{Requirement II. 355}
	{Requirement II. 357}
	{Requirement II. 358}
Use Case 8.2.3: (UC_28) NMS discovers subordinate MultiLayer Routing Area (MLRA)s	{Requirement II. 356}
Use Case 8.2.4: (UC_5) NMS discovers MLSNPP Links	{Requirement II. 359}
	{Requirement II. 360}
	{Requirement II. 361}
	{Requirement II. 362}
	{Requirement II. 363}
	{Requirement II. 364}
	{Requirement II. 365}
Use Case 8.2.5: (UC_6) NMS requests available MLSNPP Link capacity	{Requirement II. 359}
	{Requirement II. 360}
	{Requirement II. 361}
	{Requirement II. 366}
Use Case 8.2.6: (UC_7) NMS discovers Calls and top level Connections	{Requirement II. 381}
	{Requirement II. 405}

Table 11.1: Use Case to Requirement Map

Use Case	Requirement
	{Requirement II. 406}
	{Requirement II. 407}
	{Requirement II. 408}
	{Requirement II. 409}
	{Requirement II. 410}
	{Requirement II. 411}
Use Case 8.2.7: (UC_8) EMS notifies of a new Call and its Connection(s)	{Requirement II. 064}
Use Case 8.2.8: (UC_9) EMS notifies of deleted Call and its top-level Connection(s)	{Requirement II. 065}
Provisioning Use Cases	
Use Case 8.3.1: (UC_10) NMS assigns a UNI MLSNPP Link to a Signaling Controller.	{Requirement II. 373}, {Requirement II. 375}
Use Case 8.3.2: (UC_11) NMS requests to set the UNI signaling protocol and parameters	{Requirement II. 375}
Use Case 8.3.3: (UC_11b) NMS modifies signaling parameters	{Requirement II. 375}
Use Case 8.3.4: (UC_12) NMS requests EMS to enable the UNI signaling on an MLSNPP Link.	{Requirement II. 376}
Use Case 8.3.5: (UC_13) NMS requests EMS to disable the UNI signaling on an MLSNPP Link	{Requirement II. 377}
Use Case 8.3.6: (UC_14) NMS requests EMS to deassign a UNI MLSNPP Link from a Signalling Controller	{Requirement II. 374}
Use Case 8.3.7: (UC_15) NMS requests EMS to assign TNA Names to components of an MLSNPP	{Requirement II. 379}
Use Case 8.3.8: (UC_32) NMS requests EMS to assign TNA Names to components of an MLSNPP Link	{Requirement II. 378}
Use Case 8.3.9: (UC_24) NMS requests EMS for subordinate MultiLayer Routing Area IDs involved in the route of a Connection	{Requirement II. 391}
	{Requirement II. 405}
	{Requirement II. 408}
	{Requirement II. 410}
Use Case 8.3.10: (UC_25) NMS requests EMS for the route details of a specified Call within a specified routing area	{Requirement II. 390}
	{Requirement II. 405}
	{Requirement II. 407}
	{Requirement II. 408}
	{Requirement II. 409}

Table 11.1: Use Case to Requirement Map

Use Case	Requirement
	{Requirement II. 410}
Use Case 8.3.11: (UC_26) NMS requests EMS for the ID of each Call supported by a specified TP/SNPP/TNA	{Requirement II. 385}
Use Case 8.3.12: (UC_27) NMS request EMS for the list of MLSNPPs	{Requirement II. 367}
	{Requirement II. 368}
	{Requirement II. 369}
	{Requirement II. 370}
	{Requirement II. 371}

11.2 UML Object Class versus Requirements

Table 11.2 provides a mapping between the UML Classes in Section 10.1.1 and the requirements in Section 4, Section 5 and Section 6. Column 1 of Table 11.2 lists all UML Classes and column 2 identifies the corresponding requirement. Where there is no entry in column 2 this means that there is no corresponding requirement for that particular UML Class. When a specific UML Class has more than one requirement then there are multiple entries for that UML Class.

Table 11.2: UML Class to Requirement Map

	UML Class	Requirement
1	AID	{Requirement I. 057}
2	AlarmInformation	{Requirement I. 048}
3	AlarmSeverityAssignmentProfile	{Requirement I. 080}
	AlarmSeverityAssignmentProfile	{Requirement I. 081}
4	ASAPCreateModifyData	{Requirement II. 197}
5	Attribute	{Requirement I. 044}
	Attribute	{Requirement I. 045}
6	Call	{Requirement I. 133}
	Call	{Requirement I. 134}
7	CallCreateData	{Requirement II. 393}
8	CallModifyData	{Requirement II. 396}
9	Capacity	
10	CommonCreateModifyData	
11	CommonResourceInfo	{Requirement I. 060}
12	CommonTransmissionParameters	

Table 11.2: UML Class to Requirement Map

	UML Class	Requirement
13	Connection	{Requirement I. 129}
	Connection	{Requirement I. 130}
	Connection	{Requirement I. 131}
14	ConnectionCreateData	{Requirement II. 398}
15	ConnectionlessTechnologyManager	
16	ConnectionParameterProfile	
17	ConnectionTerminationPoint	{Requirement I. 006}
	ConnectionTerminationPoint	{Requirement I. 062}
18	Consumer	
19	ControlPlaneManager	
20	CrossConnect	{Requirement I. 019}
	CrossConnect	{Requirement I. 020}
21	DiversityData	{Requirement I. 134}
22	DiversityInfo	
23	DiversityViolations	{Requirement I. 134}
24	EMS	{Requirement I. 001}
	EMS	{Requirement I. 061}
25	EMSSession	
26	EProtectionSwitch	{Requirement I. 074}
27	Equipment	{Requirement I. 032}
	Equipment	{Requirement I. 064}
28	EquipmentCreateData	{Requirement II. 263}
29	EquipmentHolder	{Requirement I. 033}
	EquipmentHolder	{Requirement I. 065}
30	EquipmentProtectionGroup	{Requirement I. 072}
	EquipmentProtectionGroup	{Requirement I. 073}
31	ESwitchData	{Requirement II. 175}
32	Event	{Requirement I. 068}
33	EventInformation	{Requirement I. 093}
34	FDCreateData	{Requirement II. 322}
35	FDFrCreateData	{Requirement II. 346}

Table 11.2: UML Class to Requirement Map

	UML Class	Requirement
36	FDFrModifyData	{Requirement II. 348}
37	FDModifyData	
38	FileTransferStatus	{Requirement I. 058}
39	Filter	
40	FloatingTerminationPoint	{Requirement I. 075}
41	Flow Domain	{Requirement I. 105}
	Flow Domain	{Requirement I. 106}
42	FlowDomainFragment	{Requirement I. 110}
	FlowDomainFragment	{Requirement I. 111}
43	FTPCreateData	{Requirement II. 299}
44	GCTProfileInfo	
45	GroupTerminationPoint	{Requirement I. 069}
	GroupTerminationPoint	{Requirement I. 070}
46	HeartbeatInformation	{Requirement I. 077}
47	HistoricalPMDataFile	
48	LayeredSNPP	{Requirement I. 126}
49	LayeredSNPPLink	{Requirement I. 123}
50	Log	{Requirement I. 090}
	Log	{Requirement I. 091}
51	LogAVC	{Requirement I. 095}
52	LogCapacityThresholdAlarm	{Requirement I. 096}
53	LogObjectCreation	
54	LogObjectDeletion	
55	LogProcessingErrorAlarm	{Requirement I. 097}
56	LogRecord	
57	LogStateChange	{Requirement I. 098}
58	ManagedElement	{Requirement I. 002}
	ManagedElement	{Requirement I. 003}
59	MatrixFlowDomain	{Requirement I. 103}
	MatrixFlowDomain	{Requirement I. 104}

Table 11.2: UML Class to Requirement Map

	UML Class	Requirement
60	MatrixFlowDomainFragment	{Requirement I. 114}
	MatrixFlowDomainFragment	{Requirement I. 115}
61	MFDCreateData	
62	MFDModifyData	
63	MultiLayerRoutingArea	{Requirement I. 116}
	MultiLayerRoutingArea	{Requirement I. 117}
64	MultiLayerSNPP	{Requirement I. 118}
	MultiLayerSNPP	{Requirement I. 119}
65	MultiLayerSNPPLink	{Requirement I. 120}
	MultiLayerSNPPLink	{Requirement I. 121}
66	MultiLayerSubnetwork	{Requirement I. 012}
	MultiLayerSubnetwork	{Requirement I. 013}
67	NMSSession	
68	ObjectInformation	
69	PerformanceManager	
70	PerformanceMonitoringPoint	{Requirement I. 084}
	PerformanceMonitoringPoint	{Requirement I. 085}
71	PhysicalTerminationPoint	{Requirement I. 007}
72	PMCurrentData	{Requirement II. 282}
73	PMData	{Requirement II. 282}
74	PMHistoricData	{Requirement II. 284}
75	PMHistoricMeasurement	{Requirement II. 284}
76	PMMeasurement	{Requirement II. 282}
77	PMParameter	{Requirement I. 085}
78	PMPStateChangeInformation	{Requirement I. 083}
79	PMThreshold	{Requirement I. 099}
80	ProtectionSwitch	{Requirement I. 046}
81	ProtectionGroup	{Requirement I. 034}
	ProtectionGroup	{Requirement I. 066}
82	Route	{Requirement I. 021}
	Route	{Requirement I. 089}

Table 11.2: UML Class to Requirement Map

	UML Class	Requirement
83	RouteCreateData	{Requirement II. 242}
84	Session	
85	SessionFactory	{Requirement II. 141}
86	SNCCreateData	{Requirement II. 084}
87	SNCModifyData	{Requirement II. 246}
88	SNCRouteChangeData	{Requirement I. 059}
89	SNP	{Requirement I. 128}
90	SNPP	{Requirement I. 127}
91	SNPPLink	{Requirement I. 124}
92	SoftwareBackupStatus	{Requirement I. 086}
93	SoftwareManager	
94	SubnetworkConnection	{Requirement I. 014}
	SubnetworkConnection	{Requirement I. 015}
95	Supplier	
96	SwitchData	
97	TCAParameter	{Requirement I. 035}
	TCAParameter	{Requirement I. 088}
98	TCAParameterProfile	{Requirement I. 067}
	TCAParameterProfile	{Requirement I. 087}
99	TCProfileCreateModifyData	{Requirement II. 334}
100	TDCreateData	{Requirement II. 098}
101	TerminationPoint	{Requirement I. 004}
	TerminationPoint	{Requirement I. 005}
102	ThresholdCrossingAlertInformation	{Requirement I. 047}
103	TLCreateData	{Requirement II. 169}
104	TopologicalLink	{Requirement I. 010}
	TopologicalLink	{Requirement I. 011}
105	TPData	
106	TPPool	{Requirement I. 009}
	TPPool	{Requirement I. 094}
107	TPPoolCreateData	{Requirement II. 265}

Table 11.2: UML Class to Requirement Map

	UML Class	Requirement
108	TrafficConditioningProfile	{Requirement I. 107}
	TrafficConditioningProfile	{Requirement I. 108}
109	TrafficDescriptor	{Requirement I. 025}
	TrafficDescriptor	{Requirement I. 063}
110	TransmissionDescriptor	{Requirement I. 078}
	TransmissionDescriptor	{Requirement I. 079}
111	TransmissionDescriptorCreateData	{Requirement II. 191}
112	TransmissionParameters	
113	Version	{Requirement I. 030}

11.3 Requirements by Category

11.3.1 Category I

Table 11.3 provides a mapping between the Category I requirements in Section 4.1, Section 5.1 and Section 6.1 and the UML Classes in Section 10.1.1. There are some Category I requirements that do not have a corresponding UML Class, in these cases a comment has been added to summarize the requirement (these comments are in a **bold font**).

Table 11.3: Category I requirements

Requirement	UML Class or Comment
{Requirement I. 001}	EMS
{Requirement I. 002}	ManagedElement
{Requirement I. 003}	ManagedElement
{Requirement I. 004}	TerminationPoint
{Requirement I. 005}	TerminationPoint
{Requirement I. 006}	ConnectionTerminationPoint
{Requirement I. 007}	PhysicalTerminationPoint
{Requirement I. 008}	Edge Termination Point
{Requirement I. 009}	TPPool
{Requirement I. 010}	TopologicalLink
{Requirement I. 011}	TopologicalLink
{Requirement I. 012}	MultiLayerSubnetwork

Table 11.3: Category I requirements

Requirement	UML Class or Comment
{Requirement I. 013}	MultiLayerSubnetwork
{Requirement I. 014}	SubnetworkConnection
{Requirement I. 015}	SubnetworkConnection
{Requirement I. 016}	SNC names must not be re-used.
{Requirement I. 017}	SNC States
{Requirement I. 018}	SNC Configurations
{Requirement I. 019}	CrossConnect
{Requirement I. 020}	CrossConnect
{Requirement I. 021}	Route
{Requirement I. 022}	Layer rate
{Requirement I. 023}	Connection states
{Requirement I. 024}	TP Parameters
{Requirement I. 025}	TrafficDescriptor
{Requirement I. 026}	TD Combinations
{Requirement I. 027}	TD Service Category
{Requirement I. 028}	TD Conformance Profile
{Requirement I. 029}	TD Parameters
{Requirement I. 030}	Version
{Requirement I. 031}	Interface versioning
{Requirement I. 032}	Equipment
{Requirement I. 033}	EquipmentHolder
{Requirement I. 034}	ProtectionGroup
{Requirement I. 035}	TCAParameter
{Requirement I. 036}	PTP naming
{Requirement I. 037}	CTP naming
{Requirement I. 038}	Subnetwork type
{Requirement I. 039}	SNC configurations
{Requirement I. 040}	SNC types
{Requirement I. 041}	Event Notifications
{Requirement I. 042}	Object Creation Notification
{Requirement I. 043}	Object Deletion Notification

Table 11.3: Category I requirements

Requirement	UML Class or Comment
{Requirement I. 044}	Attribute Value Change Notification Attribute
{Requirement I. 045}	State Change Notification Attribute
{Requirement I. 046}	Protection Switch Notification ProtectionSwitch
{Requirement I. 047}	Threshold Crossing Alert Notification ThresholdCrossingAlertInformation
{Requirement I. 048}	Alarm Notification AlarmInformation
{Requirement I. 049}	Alarm - Probable Cause Qualifier
{Requirement I. 050}	Alarm - Clearable
{Requirement I. 051}	Alarm - Probable Cause
{Requirement I. 052}	Alarm - Perceived severity
{Requirement I. 053}	Alarm - Service affecting
{Requirement I. 054}	Alarm - Alarm indication
{Requirement I. 055}	Deleted - replaced with an update to {Requirement I. 048}.
{Requirement I. 056}	Alarm - Additional text
{Requirement I. 057}	AID
{Requirement I. 058}	File Transfer Status Notification FileTransferStatus
{Requirement I. 059}	Route Change Notification SNCRouteChangeData
{Requirement I. 060}	CommonResourceInfo
{Requirement I. 061}	EMS
{Requirement I. 062}	ConnectionTerminationPoint
{Requirement I. 063}	TrafficDescriptor
{Requirement I. 064}	Equipment
{Requirement I. 065}	EquipmentHolder
{Requirement I. 066}	ProtectionGroup
{Requirement I. 067}	TCAParameterProfile
{Requirement I. 068}	Event
{Requirement I. 069}	GroupTerminationPoint

Table 11.3: Category I requirements

Requirement	UML Class or Comment
{Requirement I. 070}	GroupTerminationPoint
{Requirement I. 071}	Alarm - Acknowledgement
{Requirement I. 072}	EquipmentProtectionGroup
{Requirement I. 073}	EquipmentProtectionGroup
{Requirement I. 074}	EProtectionSwitch
{Requirement I. 075}	FloatingTerminationPoint
{Requirement I. 076}	Bundled SNC
{Requirement I. 077}	HeartbeatInformation
{Requirement I. 078}	TransmissionDescriptor
{Requirement I. 079}	TransmissionDescriptor
{Requirement I. 080}	AlarmSeverityAssignmentProfile
{Requirement I. 081}	AlarmSeverityAssignmentProfile
{Requirement I. 082}	Alarm Severity Assignment
{Requirement I. 083}	PMPStateChangeInformation
{Requirement I. 084}	PerformanceMonitoringPoint
{Requirement I. 085}	PerformanceMonitoringPoint
{Requirement I. 086}	SoftwareBackupStatus
{Requirement I. 087}	TCAParameterProfile
{Requirement I. 088}	TCAParameter
{Requirement I. 089}	Route
{Requirement I. 090}	Log
{Requirement I. 091}	Log
{Requirement I. 092}	SNC end point role
{Requirement I. 093}	EventInformation
{Requirement I. 094}	TPPool
{Requirement I. 095}	LogAVC
{Requirement I. 096}	LogCapacityThresholdAlarm
{Requirement I. 097}	LogProcessingErrorAlarm
{Requirement I. 098}	LogStateChange
{Requirement I. 099}	Not used
{Requirement I. 100}	Connectionless Port Termination Point (CPTP)

Table 11.3: Category I requirements

Requirement	UML Class or Comment
{Requirement I. 101}	Flow Domain Edge CPTP (FD EdgeCPTP)
{Requirement I. 102}	Flow Point (FP)
{Requirement I. 103}	MultiLayerSNPP
{Requirement I. 104}	MultiLayerSNPP
{Requirement I. 105}	Flow Domain
{Requirement I. 106}	Flow Domain
{Requirement I. 107}	TrafficConditioningProfile
{Requirement I. 108}	TrafficConditioningProfile
{Requirement I. 109}	Traffic Mapping Table shall represent current TP configuration.
{Requirement I. 110}	FlowDomainFragment
{Requirement I. 111}	FlowDomainFragment
{Requirement I. 112}	Flow Domain Fragment Route
{Requirement I. 113}	Flow Domain Fragment Route
{Requirement I. 114}	MatrixFlowDomainFragment
{Requirement I. 115}	MatrixFlowDomainFragment
{Requirement I. 116}	MultiLayerRoutingArea
{Requirement I. 117}	MultiLayerRoutingArea
{Requirement I. 118}	MatrixFlowDomain
{Requirement I. 119}	MatrixFlowDomain
{Requirement I. 120}	MultiLayerSNPPLink
{Requirement I. 121}	MultiLayerSNPPLink
{Requirement I. 122}	Edge MLSNPPLink
{Requirement I. 123}	LayeredSNPPLink
{Requirement I. 124}	SNPPLink
{Requirement I. 125}	SNPPLink
{Requirement I. 126}	LayeredSNPP
{Requirement I. 127}	SNPP
{Requirement I. 128}	SNP
{Requirement I. 129}	Connection
{Requirement I. 130}	Connection
{Requirement I. 131}	Connection

Table 11.3: Category I requirements

Requirement	UML Class or Comment
{Requirement I. 132}	Connection States
{Requirement I. 133}	Call
{Requirement I. 134}	Call
{Requirement I. 135}	Call States

11.3.2 Category II

Table 11.4 provides a mapping between the Category II requirements in Section 4.2, Section 5.2 and Section 6.2, the Use Cases in Section 7, Section 9 and Section 8 and the UML Class operations in Section 10.1.1. There are some Category II requirements that do not have a corresponding UML Class operation, in these cases a comment has been added to summarize the requirement (these comments are in a **bold font**). In some cases the comments take the form of a suggested UML Class operation name that would be required to meet the requirement (these comments are in a **bold red font**).

Table 11.4: Category II requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirement II. 001}	Use Case 7.4.2: Use Case 7.4.3:	EMS::getEMS
{Requirement II. 002}	Use Case 7.4.2: Use Case 7.4.3: Use Case 7.4.4:	EMS::getAllManagedElements
{Requirement II. 003}		EMS::getAllManagedElementNames
{Requirement II. 004}	Use Case 7.4.2: Use Case 7.4.4:	MultiLayerSubnetwork::getAllManagedElements
{Requirement II. 005}		MultiLayerSubnetwork::getAllManagedElementNames
{Requirement II. 006}	Use Case 7.4.2: Use Case 7.4.4:	EMS::getManagedElementByName
{Requirement II. 007}		Deleted - requirement was already covered by {Requirement II. 011}
{Requirement II. 008}	Use Case 7.4.2: Use Case 7.4.4:	EMS::getMultiLayerSubnetwork
{Requirement II. 009}		ManagedElement::getContainingSubnetworkNames
{Requirement II. 010}		EMS::getAllTopLevelSubnetworkNames
{Requirement II. 011}	Use Case 7.4.2: Use Case 7.4.3: Use Case 7.4.4:	EMS::getAllTopLevelSubnetworks
{Requirement II. 012}	Use Case 7.4.2: Use Case 7.4.3: Use Case 7.4.4:	EMS::getAllTopLevelTopologicalLinks

Table 11.4: Category II requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirement II. 013}		EMS::getAllTopLevelTopologicalLinkNames
{Requirement II. 014}		EMS::getTopLevelTopologicalLink
{Requirement II. 015}	Use Case 7.4.2: Use Case 7.4.3: Use Case 7.4.4:	MultiLayerSubnetwork::getAllTopologicalLinks
{Requirement II. 016}		MultiLayerSubnetwork::getAllTopologicalLinkNames
{Requirement II. 017}		MultiLayerSubnetwork::getTopologicalLink
{Requirement II. 018}	Use Case 7.4.2: Use Case 7.4.3: Use Case 7.4.4:	MultiLayerSubnetwork::getAllSubnetworkConnections
{Requirement II. 019}		MultiLayerSubnetwork::getAllSubnetworkConnectionNames
{Requirement II. 020}	Use Case 7.4.2: Use Case 7.4.4:	MultiLayerSubnetwork::getAllSubnetworkConnections
{Requirement II. 021}		MultiLayerSubnetwork::getAllSubnetworkConnectionNames
{Requirement II. 022}	Use Case 7.4.2: Use Case 7.4.4:	MultiLayerSubnetwork::getAllSubnetworkConnectionsWithTP
{Requirement II. 023}		MultiLayerSubnetwork::getAllSubnetworkConnectionNamesWithTP
{Requirement II. 024}	Use Case 7.4.2: Use Case 7.4.4:	MultiLayerSubnetwork::getSNC
{Requirement II. 025}		MultiLayerSubnetwork::getSNCsByUserLabel
{Requirement II. 026}	Use Case 7.4.2: Use Case 7.4.4:	SubnetworkConnection::getRoute
{Requirement II. 027}		ManagedElement::getTP
{Requirement II. 028}		TPPool::getTPGroupingRelationships
{Requirement II. 029}		TrafficDescriptor::getAssociatedCTPs
{Requirement II. 030}		TerminationPoint::getTPGroupingRelationships
{Requirement II. 031}	Use Case 7.4.2: Use Case 7.4.3:	MultiLayerSubnetwork::getAllTPPools
{Requirement II. 032}		MultiLayerSubnetwork::getAllTPPoolNames
{Requirement II. 033}	Use Case 7.4.2: Use Case 7.4.3: Use Case 7.4.4:	ManagedElement::getAllPTPsWithoutFTPs
{Requirement II. 034}		ManagedElement::getAllPTPNamesWithoutFTPs
{Requirement II. 035}		ManagedElement::getAllPTPsWithoutFTPs
{Requirement II. 036}		ManagedElement::getAllPTPNamesWithoutFTPs

Table 11.4: Category II requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirement II. 037}		MultiLayerSubnetwork::getAllEdgePoints
{Requirement II. 038}		MultiLayerSubnetwork::getAllEdgePointNames
{Requirement II. 039}		MultiLayerSubnetwork::getAllEdgePoints
{Requirement II. 040}	Use Case 7.4.2: Use Case 7.4.4:	MultiLayerSubnetwork::getAllEdgePointNames
{Requirement II. 041}	Use Case 7.4.2: Use Case 7.4.3:	TerminationPoint::getContainedPotentialCTPs
{Requirement II. 042}		TerminationPoint::getContainedPotentialCTPNames
{Requirement II. 043}	Use Case 7.4.2: Use Case 7.4.3:	TerminationPoint::getContainedCurrentCTPs
{Requirement II. 044}		TerminationPoint::getContainedCurrentCTPNames
{Requirement II. 045}		TerminationPoint::getContainedInUseCTPs
{Requirement II. 046}		TerminationPoint::getContainedInUseCTPNames
{Requirement II. 047}	Use Case 7.4.2: Use Case 7.4.4:	ConnectionTerminationPoint::getContainingTPs
{Requirement II. 048}		ConnectionTerminationPoint::getContainingTPsNames
{Requirement II. 049}		MultiLayerSubnetwork::getAssociatedTP
{Requirement II. 050}	Use Case 7.4.2: Use Case 7.4.4:	Equipment::getSupportedPTPs
{Requirement II. 051}		Equipment::getSupportedPTPNames
{Requirement II. 052}	Use Case 7.4.4:	PhysicalTerminationPoint::getAllSupportingEquipment MatrixFlowDomain::getAllSupportingEquipments
{Requirement II. 053}		PhysicalTerminationPoint::getAllSupportingEquipmentNames MatrixFlowDomain::getAllSupportingEquipmentNames
{Requirement II. 054}	Use Case 7.12.5:	PerformanceManager::getMEPMCapabilities
{Requirement II. 055}	Use Case 7.12.8:	TerminationPoint::getTCAParameter
{Requirement II. 056}	Use Case 7.4.2: Use Case 7.4.3: Use Case 7.4.4:	ManagedElement::getAllEquipment ManagedElement::getEquipment
{Requirement II. 057}		ManagedElement::getAllEquipmentNamess
{Requirement II. 058}	Use Case 7.4.2: Use Case 7.4.4:	EquipmentHolder::getContainedEquipment

Table 11.4: Category II requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirement II. 059}	Use Case 7.11.6: Use Case 7.4.2: Use Case 7.4.3: Use Case 7.8.1:	ManagedElement::getAllProtectionGroups
{Requirement II. 060}	Use Case 7.4.2: Use Case 7.4.3:	EMS::getAllTrafficDescriptors
{Requirement II. 061}		EMS::getAllTrafficDescriptorNames
{Requirement II. 062}		EMS::getTrafficDescriptorByName
{Requirement II. 063}		ManagedElement::getAllCrossConnections
{Requirement II. 064}	Use Case 7.11.2: Use Case 7.11.4: Use Case 7.11.6: Use Case 7.4.1: Use Case 7.4.5: Use Case 8.2.1: Use Case 8.2.7:	Object Creation Notifications.
{Requirement II. 065}	Use Case 7.11.3: Use Case 7.11.5: Use Case 7.4.1: Use Case 7.4.5: Use Case 8.2.1: Use Case 8.2.8:	Object Deletion Notifications.
{Requirement II. 066}	Use Case 7.11.1: Use Case 7.4.1: Use Case 7.4.5: Use Case 8.2.1:	Attribute Value Change Notifications.
{Requirement II. 067}	Use Case 7.11.2: Use Case 7.11.3: Use Case 7.11.4: Use Case 7.11.5: Use Case 7.3.1: Use Case 7.4.1: Use Case 7.4.5:	State Change Notifications.
{Requirement II. 068}	Use Case 7.5.1:	ConnectionTerminationPoint::terminateAndMap
{Requirement II. 069}	Use Case 7.5.2:	ConnectionTerminationPoint::unTerminateAndUnmap
{Requirement II. 070}		Deleted - requirement was already covered by {Requirement II. 066}.
{Requirement II. 071}		ConnectionTerminationPoint::getTPMappingMode
{Requirement II. 072}	Use Case 7.5.7:	TransmissionParameters::setTransmissionParameters
{Requirement II. 073}		ConnectionTerminationPoint::terminateAndMap

Table 11.4: Category II requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirement II. 074}	Use Case 7.14.2:	ConnectionTerminationPoint::associateWithTrafficDescriptor
{Requirement II. 075}	Use Case 7.5.3:	CommonResourceInfo::setUserLabel
{Requirement II. 076}	Use Case 7.5.4:	CommonResourceInfo::setOwner
{Requirement II. 077}	Use Case 7.5.6:	CommonResourceInfo::setNativeEmsName
{Requirement II. 078}	Use Case 7.10.3: Use Case 7.10.4:	Equipment::setAlarmReportingOff Equipment::setAlarmReportingOn
{Requirement II. 079}		EquipmentHolder::setAlarmReportingOff EquipmentHolder::setAlarmReportingOn
{Requirement II. 080}		Create an SNC with a specific Route.
{Requirement II. 081}		Deleted - requirement was already covered by {Requirement II. 067}.
{Requirement II. 082}	Use Case 7.7.1:	MultiLayerSubnetwork::createSNC
{Requirement II. 083}	Use Case 7.7.1:	MultiLayerSubnetwork::createSNC
{Requirement II. 084}	Use Case 7.7.1: Use Case 7.7.10:	SNCCreateData
{Requirement II. 085}		MultiLayerSubnetwork::checkValidSNC
{Requirement II. 086}	Use Case 7.7.2:	MultiLayerSubnetwork::activateSNC
{Requirement II. 087}	Use Case 7.7.2:	SNC Activation Data.
{Requirement II. 088}	Use Case 7.7.3: Use Case 7.15.1:	MultiLayerSubnetwork::createAndActivateSNC
{Requirement II. 089}	Use Case 7.15.1: Use Case 7.7.3:	SNC Creation and Activation Data.
{Requirement II. 090}	Use Case 7.7.7:	MultiLayerSubnetwork::deactivateSNC
{Requirement II. 091}	Use Case 7.7.7:	SNC Deactivation Data.
{Requirement II. 092}	Use Case 7.7.8:	MultiLayerSubnetwork::deleteSNC
{Requirement II. 093}	Use Case 7.7.8:	SNC Deletion Data.
{Requirement II. 094}	Use Case 7.7.9:	MultiLayerSubnetwork::deactivateAndDeleteSNC
{Requirement II. 095}	Use Case 7.7.9:	SNC Deactivation and Deletion Data.
{Requirement II. 096}		MultiLayerSubnetwork::deactivateAndDeleteSNC
{Requirement II. 097}	Use Case 7.14.1:	EMS::createTrafficDescriptor
{Requirement II. 098}	Use Case 7.14.1:	TDCreateData
{Requirement II. 099}	Use Case 7.14.3:	EMS::deleteTrafficDescriptor
{Requirement II. 100}	Use Case 7.7.11:	EMS::getCapabilities

Table 11.4: Category II requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirement II. 101}		NMS shall be able to subscribe to non-EMS alarms.
{Requirement II. 102}		NMS shall be able to subscribe to EMS alarms.
{Requirement II. 103}	Use Case 7.9.3:	Alarm filtering - ability to create filters.
{Requirement II. 104}		Alarm filtering - ability to specify filter behavior.
{Requirement II. 105}		Multiple NMS shall be able to subscribe to alarms.
{Requirement II. 106}		NMS shall be able to un-subscribe to alarms.
{Requirement II. 107}		NMS shall be able to un-subscribe to alarms based on filter criteria.
{Requirement II. 108}	Use Case 7.5.8:	TransmissionParameters::setTransmissionParameters
{Requirement II. 109}	Use Case 7.5.9:	TransmissionParameters::setTransmissionParameters
{Requirement II. 110}	Use Case 7.9.2:	ManagedElement::getAllActiveAlarms
{Requirement II. 111}	Use Case 7.9.1:	EMS::getAllEMSSystemActiveAlarms
{Requirement II. 112}	Use Case 7.11.6:	ManagedElement::getProtectionGroup
{Requirement II. 113}		Deleted - requirement was already covered by {Requirement II. 059}
{Requirement II. 114}	Use Case 7.8.3:	ProtectionGroup::retrieveSwitchData
{Requirement II. 115}	Use Case 7.8.2: Use Case 7.8.4:	EMS shall send Protection Switch Notifications.
{Requirement II. 116}	Use Case 7.8.5:	ConnectionTerminationPoint::performProtectionCommand ProtectionGroup::performProtectionCommand
{Requirement II. 117}		ConnectionTerminationPoint::retrieveSwitchData ProtectionGroup::retrieveSwitchData
{Requirement II. 118}		Monitor and collect PM parameters
{Requirement II. 119}		Monitor and collect PM parameters on a TP basis.
{Requirement II. 120}		Enable and disable PM on SNC endpoints.
{Requirement II. 121}	Use Case 7.12.1: Use Case 7.12.2:	PerformanceManager::disablePMData PerformanceManager::enablePMData
{Requirement II. 122}	Use Case 7.12.10: Use Case 7.12.10: Use Case 7.12.9:	PerformanceManager::disableTCA PerformanceManager::enableTCA
{Requirement II. 123}		Deleted - replaced with {Requirement II. 221} and {Requirement II. 222}.
{Requirement II. 124}	Use Case 7.12.4:	PerformanceManager::getHoldingTime
{Requirement II. 125}	Use Case 7.12.7:	TerminationPoint::setTCAParameter
{Requirement II. 126}	Use Case 7.12.9: Use Case 7.9.6:	EMS shall send TCA Notifications.

Table 11.4: Category II requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirement II. 127}		Deleted - duplicate of {Requirement II. 055}
{Requirement II. 128}	Use Case 7.12.6:	PerformanceManager::getHistoryPMData
{Requirement II. 129}	Use Case 7.12.6:	PerformanceManager::getHistoryPMData
{Requirement II. 130}	Use Case 7.12.6:	PerformanceManager::getHistoryPMData
{Requirement II. 131}	Use Case 7.12.3:	PerformanceManager::getAllCurrentPMData
{Requirement II. 132}	Use Case 7.12.1:	PerformanceManager::clearPMData
{Requirement II. 133}	Use Case 7.11.4:	Get actual equipment configuration
{Requirement II. 134}	Use Case 7.11.4:	Identify equipment mismatch configurations.
{Requirement II. 135}		Get installed equipment information.
{Requirement II. 136}	Use Case 7.10.2:	EquipmentHolder::provisionEquipment
{Requirement II. 137}		TerminationPoint::performMaintenanceOperation
{Requirement II. 138}		ManagedElement::getActiveMaintenanceOperations TerminationPoint::getActiveMaintenanceOperations
{Requirement II. 139}	Use Case 7.2.2: Use Case 7.2.4: Use Case 7.2.5:	Identification of NMS.
{Requirement II. 140}	Use Case 7.2.2: Use Case 7.2.4: Use Case 7.2.5:	Authentication of NMS.
{Requirement II. 141}		SessionFactory::getSession
{Requirement II. 142}		Establish multiple sessions.
{Requirement II. 143}		Terminate a session.
{Requirement II. 144}	Use Case 7.2.2:	Detect security violations.
{Requirement II. 145}	Use Case 7.2.2: Use Case 7.2.6: Use Case 7.2.7:	Detect communication failures.
{Requirement II. 146}		EMS::getCapabilities
{Requirement II. 147}		EMS::launchGCT
{Requirement II. 148}		GCT
{Requirement II. 149}		GCT
{Requirement II. 150}	Use Case 7.13.1:	GCT
{Requirement II. 151}	Use Case 7.13.2: Use Case 7.13.3:	EMS::getGCTProfileInfo
{Requirement II. 152}	Use Case 7.13.3:	EMS::destroyGCT

Table 11.4: Category II requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirement II. 153}		MultiLayerSubnetwork::createSNC
{Requirement II. 154}	Use Case 7.9.10: Use Case 7.9.11:	EMS::getAllEMSSystemActiveAlarms
{Requirement II. 155}	Use Case 7.9.7:	EMS::acknowledgeAlarms
{Requirement II. 156}	Use Case 7.9.8:	EMS::unacknowledgeAlarms
{Requirement II. 157}	Use Case 7.9.7: Use Case 7.9.9:	Subscribe to an acknowledged alarm.
{Requirement II. 158}	Use Case 7.9.8:	Subscribe to an unacknowledged alarm.
{Requirement II. 159}		SubnetworkConnection::setAlarmReportingOn
{Requirement II. 160}		SubnetworkConnection::setAlarmReportingOff
{Requirement II. 161}		TopologicalLink::setAlarmReportingOn
{Requirement II. 162}		TopologicalLink::setAlarmReportingOff
{Requirement II. 163}	Use Case 7.12.11:	PerformanceManager::getTPHistoryPMData
{Requirement II. 164}		ManagedElement::createGroupTerminationPoint
{Requirement II. 165}		GTP Creation Data.
{Requirement II. 166}		ManagedElement::deleteGroupTerminationPoint
{Requirement II. 167}		ManagedElement::modifyGroupTerminationPoint
{Requirement II. 168}	Use Case 7.5.10:	EMS::createTopologicalLink
{Requirement II. 169}		TLCreateData
{Requirement II. 170}	Use Case 7.5.11:	EMS::deleteTopologicalLink
{Requirement II. 171}		ManagedElement::getAllGroupTerminationPoints
{Requirement II. 172}		ManagedElement::getAllGroupTerminationPointNames
{Requirement II. 173}		ManagedElement::getGroupTerminationPoint
{Requirement II. 174}	Use Case 7.8.1:	ManagedElement::getAllEProtectionGroups
{Requirement II. 175}	Use Case 7.8.2: Use Case 7.8.3:	ESwitchData EquipmentProtectionGroup::retrieveESwitchData ManagedElement::getEProtectionGroup
{Requirement II. 176}		Determine active Equipment in Protection Group.
{Requirement II. 177}	Use Case 7.9.12: Use Case 7.9.13:	NMSSession::eventLossOccurred NMSSession::eventLossCleared NMSSession::alarmLossOccurred
{Requirement II. 178}	Use Case 7.9.14:	Event Channel availability.
{Requirement II. 179}		MultiLayerSubnetwork::getAllFixedSubnetworkConnections

Table 11.4: Category II requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirement II. 180}		MultiLayerSubnetwork::getAllFixedSubnetworkConnectionNames
{Requirement II. 181}		MultiLayerSubnetwork::getAllFixedSubnetworkConnectionsWithTP
{Requirement II. 182}		MultiLayerSubnetwork::getAllFixedSubnetworkConnectionNamesWithTP
{Requirement II. 183}		ManagedElement::getAllFixedCrossConnections
{Requirement II. 184}		ConnectionTerminationPoint::getAssociatedGroupTerminationPoint
{Requirement II. 185}		TerminationPoint::getPotentialFixedCCs
{Requirement II. 186}		TransmissionDescriptor::getAssociatedTPs
{Requirement II. 187}		EMS::getAllTransmissionDescriptors
{Requirement II. 188}		EMS::getAllTransmissionDescriptorNames
{Requirement II. 189}		EMS::getTransmissionDescriptorByName
{Requirement II. 190}	Use Case 7.5.12:	EMS::createTransmissionDescriptor
{Requirement II. 191}		TransmissionDescriptorCreateData
{Requirement II. 192}	Use Case 7.5.17:	EMS::deleteTransmissionDescriptor
{Requirement II. 193}		CommonResourceInfo::setNetworkAccessDomain
{Requirement II. 194}	Use Case 7.5.14:	TerminationPoint::setTransmissionDescriptorAssociation
{Requirement II. 195}		SwitchData
{Requirement II. 196}		EMS::createASAP
{Requirement II. 197}		ASAPCreateModifyData
{Requirement II. 198}		EMS::modifyASAP
{Requirement II. 199}		ASAP Modification Data.
{Requirement II. 200}		EMS::deleteASAP
{Requirement II. 201}	Use Case 7.5.24:	EMS::assignASAP
{Requirement II. 202}		ASAP Assignment Data.
{Requirement II. 203}		EMS::deassignASAP
{Requirement II. 204}		ASAP Deassignment Data.
{Requirement II. 205}		EMS::getAllASAPs
{Requirement II. 206}		EMS::getAllASAPNames
{Requirement II. 207}		EMS::getASAP
{Requirement II. 208}		EMS::getASAPByResource
{Requirement II. 209}		AlarmSeverityAssignmentProfile::getASAPAssociatedResourceNames
{Requirement II. 210}		ManagedElement::getAllFTPs

Table 11.4: Category II requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirement II. 211}		ManagedElement::getAllFTPNames
{Requirement II. 212}		ManagedElement::getAllFTPs
{Requirement II. 213}		ManagedElement::getAllFTPNames
{Requirement II. 214}		ManagedElement::getAllPTPs
{Requirement II. 215}		ManagedElement::getAllPTPNames
{Requirement II. 216}		ManagedElement::getAllPTPs
{Requirement II. 217}		ManagedElement::getAllPTPNames
{Requirement II. 218}		SubnetworkConnection::getRouteAndTopologicalLinks
{Requirement II. 219}		GroupTerminationPoint::setAlarmReportingOn
{Requirement II. 220}		GroupTerminationPoint::setAlarmReportingOff
{Requirement II. 221}	Use Case 7.12.12:	PerformanceManager::getAllPMPs
{Requirement II. 222}		PerformanceManager::getAllPMPNames
{Requirement II. 223}	Use Case 7.5.5: Use Case 7.9.4:	CommonResourceInfo::setAdditionalInfo
{Requirement II. 224}	Use Case 7.9.5:	EMS shall be able to change the root cause for an alarm.
{Requirement II. 225}		Equipment::getSupportingEquipment
{Requirement II. 226}		Equipment::getSupportingEquipmentNames
{Requirement II. 227}		Equipment::getSupportedEquipment
{Requirement II. 228}		Equipment::getSupportedEquipmentNames
{Requirement II. 229}		SoftwareManager::backupME
{Requirement II. 230}		SoftwareManager::getMEBackupStatus
{Requirement II. 231}		SoftwareManager::abortMEBackup
{Requirement II. 232}		SoftwareManager::getBackupList
{Requirement II. 233}		PerformanceManager::getAllTCAParameterProfileNames
{Requirement II. 234}		PerformanceManager::getTCAParameterProfile
{Requirement II. 235}		TerminationPoint::setTCAParameterProfile
{Requirement II. 236}		PerformanceManager::createTCAParameterProfile
{Requirement II. 237}		TCAParameterProfile Creation Data.
{Requirement II. 238}	Use Case 7.12.13: Use Case 7.12.14:	PerformanceManager::setTCAParameterProfile
{Requirement II. 239}	Use Case 7.12.13: Use Case 7.12.14:	TCAParameterProfile Modification Data.

Table 11.4: Category II requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirement II. 240}		PerformanceManager::deleteTCAParameterProfile
{Requirement II. 241}	Use Case 7.7.4:	SubnetworkConnection::addRoute
{Requirement II. 242}	Use Case 7.7.4:	RouteCreateData
{Requirement II. 243}	Use Case 7.7.5:	SubnetworkConnection::removeRoute
{Requirement II. 244}	Use Case 7.7.5:	Route Removal Data.
{Requirement II. 245}	Use Case 7.7.6:	MultiLayerSubnetwork::createModifiedSNC
{Requirement II. 246}	Use Case 7.7.6:	SNCModifyData
{Requirement II. 247}		SubnetworkConnection::switchRoute
{Requirement II. 248}		Switch Route Data.
{Requirement II. 249}		SubnetworkConnection::setRoutesAdminState
{Requirement II. 250}		Set route administrative state data
{Requirement II. 251}		SubnetworkConnection::setIntendedRoute
{Requirement II. 252}		Set Intended Route Data.
{Requirement II. 253}		Retrieval of all Alarms and Events.
{Requirement II. 254}		Log::deleteLogRecords Log::getLogRecords Log::retrieveLogRecords Log::setAdministrativeState Log::setCapacityAlarmThreshold Log::setDiscriminatorConstruct Log::setLogFullAction Log::setMaxLogSize Log::setMaxRecordLife Log::setWeekMask
{Requirement II. 255}		One instance of the Log Service.
{Requirement II. 256}		SubnetworkConnection::getBackupRoutes
{Requirement II. 257}		MultiLayerSubnetwork::modifySNC
{Requirement II. 258}		MultiLayerSubnetwork::swapSNC
{Requirement II. 259}		Swap SNC Data.
{Requirement II. 260}		SubnetworkConnection::getBackupRoutes
{Requirement II. 261}		SubnetworkConnection::getIntendedRoute
{Requirement II. 262}	Use Case 7.10.1:	EquipmentHolder::unprovisionEquipment
{Requirement II. 263}		EquipmentCreateData
{Requirement II. 264}		MultiLayerSubnetwork::createTPPool

Table 11.4: Category II requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirement II. 265}		TPPoolCreateData
{Requirement II. 266}		MultiLayerSubnetwork::deleteTPPool
{Requirement II. 267}		MultiLayerSubnetwork::modifyTPPool
{Requirement II. 268}		MultiLayerSubnetwork::getTPPool
{Requirement II. 269}		ESwitchData
{Requirement II. 270}		EMS::getAllEMSAndMEActiveAlarms
{Requirement II. 271}		EMS::getLog
{Requirement II. 272}		GroupTerminationPoint::getTPGroupingRelationships
{Requirement II. 273}		PerformanceManager::getAllTCAParameterProfiles
{Requirement II. 274}		ManagedElement::getAllNUTTPNames
{Requirement II. 275}		ManagedElement::getAllPreemptibleTPNames
{Requirement II. 276}		ManagedElement::getAllProtectedTPNames
{Requirement II. 277}	Use Case 7.5.14:	TMD Assignment behavior.
{Requirement II. 278}		TerminationPoint::verifyTMDAssignment MatrixFlowDomain::verifyTMDAssignmentToMFD
{Requirement II. 279}		EquipmentHolder::getAllEquipment
{Requirement II. 280}		EquipmentHolder::getAllEquipmentNames
{Requirement II. 281}		Get current PM Data.
{Requirement II. 282}		PMData
{Requirement II. 283}		PMMeasurement
{Requirement II. 284}		Historical PM Data File Format.
{Requirement II. 285}		TCAParameterProfile::getAssociatedTPs
{Requirement II. 286}		Version::getVersion
{Requirement II. 287}		EMS::getAllEMSAndMEUnacknowledgedActiveAlarms
{Requirement II. 288}		EMS::getAllEMSSystemUnacknowledgedActiveAlarms
{Requirement II. 289}		PhysicalTerminationPoint::getContainingPGNames
{Requirement II. 290}		Connectivity awareness capability.
{Requirement II. 291}		TerminationPoint::getTransmissionParameters
{Requirement II. 292}	Use Case 7.4.4:	ConnectionlessTechnologyManager::getAssigningMFD
{Requirement II. 293}	Use Case 9.2.1:	ConnectionlessTechnologyManager::createFTP
{Requirement II. 294}	Use Case 9.2.2:	ConnectionlessTechnologyManager::deleteFTP

Table 11.4: Category II requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirement II. 295}	Use Case 7.4.4:	ManagedElement::getAllMFDs
{Requirement II. 296}	Use Case 7.4.4:	ManagedElement::getMFD()
{Requirement II. 297}	Use Case 7.4.4:	MatrixFlowDomain::getAllAssignedCPTPs
{Requirement II. 298}	Use Case 7.9.4:	EMS shall be able to indicate a raw alarm.
{Requirement II. 299}	Use Case 9.2.1:	FTP Creation Data.
{Requirement II. 300}	Use Case 7.4.4:	Equipment::getSupportedMFDs
{Requirement II. 301}	Use Case 7.4.4:	TransmissionDescriptor::getAllAssociatedMFDs
{Requirement II. 302}	Use Case 7.5.16:	MatrixFlowDomain::getTransmissionParameters
{Requirement II. 303}	Use Case 7.4.4:	MatrixFlowDomain::getAssociatingFlowDomain
{Requirement II. 304}	Use Case 7.4.4:	MatrixFlowDomain::getAllAssignableCPTPs
{Requirement II. 305}	Use Case 9.4.1:	ConnectionlessTechnologyManager::createMFD
{Requirement II. 306}	Use Case 9.4.1:	MFD Creation Data.
{Requirement II. 307}	Use Case 9.4.3:	ConnectionlessTechnologyManager::modifyMFD
{Requirement II. 308}	Use Case 9.4.3:	MFD Modification Data.
{Requirement II. 309}	Use Case 9.4.2:	ConnectionlessTechnologyManager::deleteMFD
{Requirement II. 310}	Use Case 9.4.4:	MatrixFlowDomain::assignCPTPsToMFD
{Requirement II. 311}	Use Case 9.4.5:	MatrixFlowDomain::unassignCPTPsToMFD
{Requirement II. 312}	Use Case 7.5.14:	MatrixFlowDomain::setTransmissionDescriptorAssociation
{Requirement II. 313}	Use Case 7.5.14:	MatrixFlowDomain::setTransmissionDescriptorAssociation
{Requirement II. 314}	Use Case 7.4.4:	ConnectionlessTechnologyManager::getFlowDomain
{Requirement II. 315}	Use Case 7.4.4:	Flow Domain::getAllCPTPs
{Requirement II. 316}	Use Case 7.4.4:	Flow Domain::getAllAssociatedMFDs
{Requirement II. 317}	Use Case 7.4.4:	ConnectionlessTechnologyManager::getFlowDomainsByUserLabel
{Requirement II. 318}	Use Case 7.4.4:	ConnectionlessTechnologyManager::getFlowDomain
{Requirement II. 319}	Use Case 7.5.16:	Flow Domain::getTransmissionParameters
{Requirement II. 320}		Not used.
{Requirement II. 321}	Use Case 9.3.1:	ConnectionlessTechnologyManager::createFlowDomain
{Requirement II. 322}	Use Case 9.3.1:	Flow Domain Creation Data.
{Requirement II. 323}	Use Case 9.3.3:	ConnectionlessTechnologyManager::modifyFlowDomain
{Requirement II. 324}	Use Case 9.3.3:	Flow Domain Modification Data.
{Requirement II. 325}	Use Case 9.3.2:	ConnectionlessTechnologyManager::deleteFlowDomain

Table 11.4: Category II requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirement II. 326}	Use Case 9.3.6:	Flow Domain::associateCPTPsWithFlowDomain
{Requirement II. 327}	Use Case 9.3.7:	Flow Domain::deassociateCPTPsFromFlowDomain
{Requirement II. 328}	Use Case 9.3.4:	Flow Domain::associateMFDsWithFlowDomain
{Requirement II. 329}	Use Case 9.3.5:	Flow Domain::deassociateMFDsFromFlowDomain
{Requirement II. 330}	Use Case 7.4.4:	ConnectionlessTechnologyManager::getAllTCProfiles
{Requirement II. 331}		ConnectionlessTechnologyManager::getTCProfile
{Requirement II. 332}	Use Case 7.4.4:	ConnectionlessTechnologyManager::getTCProfileAssociatedTPs
{Requirement II. 333}	Use Case 9.5.1:	ConnectionlessTechnologyManager::createTCProfile
{Requirement II. 334}	Use Case 9.5.1: Use Case 9.5.3:	TC Profile Creation Data.
{Requirement II. 335}	Use Case 9.5.3:	ConnectionlessTechnologyManager::modifyTCProfile
{Requirement II. 336}	Use Case 9.5.2:	ConnectionlessTechnologyManager::deleteTCProfile
{Requirement II. 337}	Use Case 9.5.4:	TransmissionParameters::setTransmissionParameters
{Requirement II. 338}	Use Case 7.4.4:	Flow Domain::getAllFDFrs
{Requirement II. 339}	Use Case 7.4.4:	Flow Domain::getAllFDFrsWithCPTP
{Requirement II. 340}	Use Case 7.4.4:	Flow Domain::getFDFr
{Requirement II. 341}	Use Case 7.4.4:	Flow Domain::getFDFrByUserLabel
{Requirement II. 342}	Use Case 7.4.4:	Flow Domain::getFDrWithFP
{Requirement II. 343}	Use Case 7.4.4:	FlowDomainFragment::getFDFrRoute
{Requirement II. 344}	Use Case 7.5.16:	FlowDomainFragment::getTransmissionParameter
{Requirement II. 345}	Use Case 9.6.1:	Flow Domain::createAndActivateFDFr
{Requirement II. 346}	Use Case 9.6.1:	FDFr Creation Data.
{Requirement II. 347}	Use Case 9.6.3:	Flow Domain::modifyFDFr
{Requirement II. 348}	Use Case 9.6.3:	FDFr Modification Data.
{Requirement II. 349}	Use Case 9.6.2:	Flow Domain::deactivateAndDeleteFDFr
{Requirement II. 350}	Use Case 9.6.3:	Flow Domain::modifyFDFr
{Requirement II. 351}	Use Case 9.6.3:	Flow Domain::modifyFDFr
{Requirement II. 352}	Use Case 7.5.16:	TransmissionDescriptor::getTransmissionParameters
{Requirement II. 353}	Use Case 7.5.15:	EMS::modifyTransmissionDescriptor
{Requirement II. 354}	Use Case 8.2.2:	ControlPlaneManager::getAllMLRAs
{Requirement II. 355}	Use Case 8.2.2:	ControlPlaneManager::getMLRA

Table 11.4: Category II requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirement II. 356}	Use Case 8.2.3:	MultiLayerRoutingArea::getAllSubordinateMLRAs
{Requirement II. 357}	Use Case 8.2.2:	ControlPlaneManager::getAllTopLevelMLRAs
{Requirement II. 358}	Use Case 8.2.2:	ControlPlaneManager::getAllTopLevelMLRANames
{Requirement II. 359}	Use Case 8.2.1: Use Case 8.2.4: Use Case 8.2.5:	MultiLayerRoutingArea::getAllInternalMLSNPPLinks
{Requirement II. 360}	Use Case 8.2.4: Use Case 8.2.5:	MultiLayerRoutingArea::getAllMLSNPPLinks
{Requirement II. 361}	Use Case 8.2.1: Use Case 8.2.4: Use Case 8.2.5:	ControlPlaneManager::getMLSNPPLink
{Requirement II. 362}	Use Case 8.2.4:	ControlPlaneManager::getAllMLSNPPLinksWithTP
{Requirement II. 363}	Use Case 8.2.4:	MultiLayerRoutingArea::getAllEdgeMLSNPPLinks
{Requirement II. 364}	Use Case 8.2.4:	ControlPlaneManager::getAllMLSNPPLinksWithTNA
{Requirement II. 365}	Use Case 8.2.4:	ControlPlaneManager::getAllMLSNPPLinksWithMLRAs
{Requirement II. 366}	Use Case 8.2.5:	MultiLayerSNPPLink::getAvailableCapacity
{Requirement II. 367}	Use Case 8.3.12:	MultiLayerRoutingArea::getAllMLSNPPs
{Requirement II. 368}	Use Case 8.3.12:	MultiLayerRoutingArea::getMLSNPPLink
{Requirement II. 369}	Use Case 8.3.12:	ControlPlaneManager::getAllMLSNPPsWithTP
{Requirement II. 370}	Use Case 8.3.12:	ControlPlaneManager::getAllMLSNPPsWithTNA
{Requirement II. 371}	Use Case 8.3.12:	ControlPlaneManager::getAllMLSNPPs
{Requirement II. 372}		Deleted - requirement was already covered by {Requirement II. 390}.
{Requirement II. 373}	Use Case 8.3.1:	MultiLayerSNPPLink::assignSignallingController
{Requirement II. 374}	Use Case 8.3.6:	MultiLayerSNPPLink::deassignSignallingController
{Requirement II. 375}	Use Case 8.3.1: Use Case 8.3.2: Use Case 8.3.3:	MultiLayerSNPPLink::setSignallingProtocolAndParameters
{Requirement II. 376}	Use Case 8.3.4:	MultiLayerSNPPLink::enableSignalling
{Requirement II. 377}	Use Case 8.3.5:	MultiLayerSNPPLink::disableSignalling
{Requirement II. 378}	Use Case 8.3.8:	MultiLayerSNPPLink::setTNAName
{Requirement II. 379}	Use Case 8.3.7:	MultiLayerSNPP::assignTNA
{Requirement II. 380}		Flow Domain::getAllTopologicalLinks
{Requirement II. 381}	Use Case 8.2.6:	MultiLayerRoutingArea::getAllCallsAndTopLevelConnections

Table 11.4: Category II requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirement II. 382}		MultiLayerSubnetwork::getAllCallsAndSNCs
{Requirement II. 383}		MultiLayerRoutingArea::getCallAndTopLevelConnections
{Requirement II. 384}		MultiLayerSubnetwork::getCallAndSNCs
{Requirement II. 385}	Use Case 8.3.11:	MultiLayerRoutingArea::getAllCallIdsWithSNPPOrTNAName
{Requirement II. 386}		MultiLayerRoutingArea::getAllCallsAndTopLevelConnections
{Requirement II. 387}		MultiLayerRoutingArea::getCall
{Requirement II. 388}		MultiLayerRoutingArea::getAllCallsAndTopLevelConnectionsWithME
{Requirement II. 389}		Call::getConnection
{Requirement II. 390}	Use Case 8.3.10:	MultiLayerRoutingArea::getRouteAndMLSNPPLinks
{Requirement II. 391}	Use Case 8.3.9:	MultiLayerRoutingArea::getAllSubordinateRAIdsWithConnection
{Requirement II. 392}	Use Case 7.6.1:	MultiLayerSubnetwork::establishCall
{Requirement II. 393}	Use Case 7.6.1:	CallCreateData
{Requirement II. 394}	Use Case 7.6.2:	MultiLayerSubnetwork::releaseCall
{Requirement II. 395}	Use Case 7.6.5:	MultiLayerSubnetwork::modifyCall
{Requirement II. 396}	Use Case 7.6.5:	CallModifyData
{Requirement II. 397}	Use Case 7.6.3:	Call::addConnections
{Requirement II. 398}	Use Case 7.6.3:	Connection Addition Data
{Requirement II. 399}	Use Case 7.6.4:	Call::removeConnections
{Requirement II. 400}	Use Case 7.6.7:	Associate Connection to Route Group in a Call
{Requirement II. 401}	Use Case 7.6.7:	Request diversity among Route Groups within a Call
{Requirement II. 402}	Use Case 7.6.7:	Call::modifyDiversityAndCorouting
{Requirement II. 403}	Use Case 7.6.7:	Diversity and co-routing modification parameters
{Requirement II. 404}	Use Case 7.6.6:	Routing constraints - list of nodes
{Requirement II. 405}	Use Case 7.6.6: Use Case 8.2.6: Use Case 8.3.9: Use Case 8.3.10:	Connection types (SC, SPC, PC)
{Requirement II. 406}	Use Case 7.6.6: Use Case 8.2.6:	Retrieve relationship between Connections, Route Groups, and Calls
{Requirement II. 407}	Use Case 7.6.6: Use Case 8.2.6: Use Case 8.3.10:	Call::getConnection

Table 11.4: Category II requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirement II. 408}	Use Case 7.6.6: Use Case 8.2.6: Use Case 8.3.9: Use Case 8.3.10:	Call::getConnection
{Requirement II. 409}	Use Case 7.6.6: Use Case 8.2.6: Use Case 8.3.10:	Call::getConnection
{Requirement II. 410}	Use Case 7.6.6: Use Case 8.2.6: Use Case 8.3.9: Use Case 8.3.10:	Call::getConnection
{Requirement II. 411}	Use Case 7.6.6: Use Case 8.2.6:	Call::getConnection
{Requirement II. 412}		MultiLayerSubnetwork::getAllCallsAndSNCsWithME
{Requirement II. 413}		MultiLayerSubnetwork::getAllCallsAndSNCsWithTP
{Requirement II. 414}		MultiLayerSubnetwork::getCall
{Requirement II. 415}		MultiLayerRoutingArea::getAllCallsAndTopLevelConnectionsAndSNCs
{Requirement II. 416}		MultiLayerRoutingArea::getAllCallsAndTopLevelConnectionsAndSNCsWithME
{Requirement II. 417}		MultiLayerRoutingArea::getAllCallsAndTopLevelConnectionsAndSNCsWithTP

11.3.3 Category III

Table 11.5 provides a mapping between the Category III requirements in Section 4.3, the Use Cases in Section 7 and the UML Class operations in Section 10.1.1. The nature of Category III requirements means that they do not have a corresponding UML Class operation and so a comment has been added to summarize the requirement (these comments are in a **bold font**).

Table 11.5: Category III requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirement III. 001}	Use Case 7.2.6: Use Case 7.2.7:	NMS detects when an EMS is no longer available.
{Requirement III. 002}	Use Case 7.2.6: Use Case 7.2.7:	EMS detects when an NMS is no longer available.

11.3.4 Category IV

Table 11.6 provides a mapping between the Category IV requirements in Section 4.4, the Use Cases in Section 7 and the UML Class operations in Section 10.1.1. The nature of Category IV requirements means that they do not have a corresponding UML Class operation and so a comment has been added to summarize the requirement (these comments are in a **bold font**).

Table 11.6: Category IV requirements

Requirement	Use Case	UML Class Operation or Comment
{Requirements IV. 001}		Interface must be scalable.
{Requirements IV. 002}		Interface model shall be concise.
{Requirements IV. 003}	Use Case 7.2.1:	EMS to support multiple NMSs.
{Requirements IV. 004}		NMS to support multiple EMSs.

11.3.5 Category V

There are currently no Category V requirements.

12 Summary and Open Issues

12.1 Summary

This version (3.1) of the Business Agreement specification adds the following:

- Control Plane Management (see SD1-45_ASONControlPlaneManagement-Primer.pdf for an overview) and
- Connectionless Technology (e.g., Ethernet) Management (see SD1-44_ConnectionlessTechnologyManagement.pdf for an overview)

capabilities to the MTNM model.

Note that SONET/SDH, Asynch/PDH, ATM, DSL, DWDM and point-to-point Ethernet have already been included in the previous MTNM specification (i.e. version 3.0).

12.2 Open Issues

None.

Appendix A: Terminology, Acronyms and Abbreviations

A.1 Terminology

Term	Definition	TMF or Outside Source
Active Alarm	A clearable raised alarm is considered to be active for as long as no associated clear notification has been received.	
Actor	The term actor is a keyword which refers to a coherent set of roles that an entity (human or non-human) outside of the system being modeller plays when interacting with one or more use cases. Systems can also be related to actors in that for two systems interacting with each other, each is an actor to the other. Actors represent system users. They help delimit the system and give a clearer picture of what the system should do. It is important to note that an actor interacts with, but has no control over the use cases. An actor is someone or something that: Interacts with or uses the system Provides input to and receives information from the system Is external to the system and has no control over the use cases	
Inner topological link	A topological link that exists within a subnetwork.	
Internal topological link	A topological link that exists within the same managed element.	
Network Access Domain	A Network Access Domain (NAD) represents a domain to which certain transmission network resources may be assigned.	
Network Management System	The Network Management System represents the hardware and software components used by the SP or Network Provider to manage their networks as a whole. The NMS provides a end-to-end network view of the entire network enabling management of the NEs contained in the network. These NEs managed across the network are typically provided by multiple vendors. The NMS performs management functions across the Network Management Layer (NML) of the TMN. Some examples of these management functions include connection management and circuit fault correlation.	

Term	Definition	TMF or Outside Source
NML-EML Interface	The NML-EML Interface represents the communication data and exchange mechanism between the management system(s) that deploy the NML and EML functions of the TMN. A Network Management System (NMS) that performs NML functionalities may communicate with one or more Element Management Systems (EMSs) that performs EML functionalities via the NML-EML Interface.	
Off-network topological link	A topological link whose zEnd TP is outside of the EMS' span of control and is reported as a remote address if the EMS knows about the remote end.	
System	The term system refers to a stereotyped package that represents the entire system being modelled.	W1 - UML Modeling for the W1 - UML Modeling for the Telecom Environment Telecom Environment, Part 1: Introduction to UML for Communications, 7/18/2000, Al Vincent (TeleManagement Forum)
Top level topological link	A topological link that exists between subnetworks.	
Trail	A topological link or an SNC between two TPs that encapsulate a trail termination function at the layer of the TL or SNC.	

A.2 Abbreviations and Acronyms

Abbreviation/ Acronym	Abbreviation/ Acronym Spelled Out	Definition	TMF or External Source
15min	15 minute	Used to identify a 15 minute window size.	MTNM
24hr	24 hour	Used to identify a 1 day window size.	MTNM
CTP	Connection Termination Point	A CTP represents the actual or potential end point of a subnetwork connection, or represents an ATM network interface. (It is a combination of the ITU-T Connection Termination Point function and Trail Termination Point function.)	MTNM
EMS	Element Management System	It represents the management system located at the "south" side of the interface. See also chapter "Terminology Used In This Document".	MTNM
ME	Managed Element	A ME is an abstract class used to represent Network Elements visible across the interface.	MTNM

Abbreviation/ Acronym	Abbreviation/ Acronym Spelled Out	Definition	TMF or External Source
NA	Not Applicable	In the PM context the non applicable granularity is used for current instantaneous measurements.	MTNM
NMS	Network Management System	It represents the management system located at the "north" side of the interface. See also chapter Terminology Used In This Document".	MTNM
PM	Performance Management	Is the comprehensive term for all functionalities that are used to provide the Network Management System with information about the performance of the services in the Subnetwork.	MTNM
PMD	Performance Monitoring Data	Data that is retrieved via the interface using Performance Management	MTNM
TCA	Threshold Crossing Alert	Functionality that provides automatic notifications when performance of a service is outside a predefined range.	MTNM
TP	Termination Point	A TP is a logical abstraction of an endpoint (actual or potential) of a topological link, or a subnetwork connection.	MTNM
PTP	Physical Termination Point	A PTP is an actual or potential endpoint of a topological link. Essentially, this is a representation of a physical port.	MTNM
UTC	Universal Time Coordinated	Standardised time format used for time stamping.	ITU-T Recommendation X.680

Appendix B: References

B.1 References

Reference	Description	Brief Use Summary
Project Charter	MTNM Project Charter v3.0	Used for objectives and scope and for the project benefits.
eTOM	TMF Enhanced Telecom Operations Map: A high-level view of end-to-end service fulfillment, service assurance and billing, TMF-GB910,TMF, Morris-town, 1998	Used for process definitions.
TMF Project Management, Process and Methodology Documents	TMF Team Leader Handbook, TMF Process Guides, TMF Templates, TMF Template Guides, TMF Modeling Methodology Manual	Used to provide method and consistency to work and documents
ITU-T G.805	ITU recommendations, Generic functional archi-tecture of transport networks, March 2000	Used in defining the information model.
TMF 608	MTNM Information Agreement A protocol neutral Interface Information model for the NML-EML interface.	Used for the UML models for the inter-face.
SD1-5	Service Category and Conformance Definition val-ues	Referred for more details on the relevant subject.
SD1-16	Layered Parameters	Referred for more details on the relevant subject.
SD1-17	Layer Rates	Referred for more details on the relevant subject.
SD1-18	Functional Modeling Concepts	Referred for the functional encapsulation and modeling principles at the interface, with exhaustive explanations and exam-ples.
SD1-24	Native EMS Name	Referred for more details on the relevant subject.
SD1-26	OMG Service Usage	Referred for more details on the relevant subject.
SD1-28	Performance Parameters	Referred for more details on the relevant subject.
SD1-30	Performance Management Data	Referred for more details on the relevant subject.
SD1-33	Probable Causes	Referred for more details on the relevant subject.
SD1-36	SNC Types and Protection	Referred for more details on the relevant subject.

Reference	Description	Brief Use Summary
SD1-37	PM Threshold Types	Referred for more details on the relevant subject.
SD1-44	Connectionless Technology Management	Referred for more details on the relevant subject.
SD1-45	ASON Control Plane Management - Primer	Referred for more details on the relevant subject.

B.2 Source or Use

None.

B.3 IPR Releases and Patent Disclosures

None.

Appendix C: Contacts

MTNM Team Leader	
Keith Dorking, Ciena Corporation	
Phone :	+1-678-867-5007
Fax :	+1-678-867-5010
email :	kdorking@ciena.com

MTNM Business Agreement (TMF 513) Editor	
Keith Dorking, Ciena Corporation	
Phone :	+1-678-867-5007
Fax :	+1-678-867-5010
email :	kdorking@ciena.com

MTNM Information Agreement (TMF 608) Editor	
Bernd Zeuner, T-Systems	
Phone :	+49-6151 937-3709
Fax :	+49-6151 937 19875
email :	bernd.zeuner@t-systems.com

MTNM Solution Set (TMF 814) Editor	
Gerard Vila, Alcatel-Lucentl	
Phone :	+33 1 30 77 78 68
Fax :	+33
email :	gerard.vila@alcatel-lucent.fr

MTNM Implementation Statement Template and Guidelines (TMF 814A) Editor	
H Kam Lam, Alcatel-Lucent	
Phone :	+1-732 949 8338
Fax :	+1-732 615 4610

email :	hklam@lucent.com

TMF Senior Technical Manager	
Stephen Gaito, TM Forum	
Phone :	+44-1666-335-007
Fax :	
email :	isgaito@tmforum.org
MTNM email address : mtnm@lists.tmforum.org	

Administrative Appendix

This Appendix provides additional background material about the TeleManagement Forum and this document.

About This Document

The Multi-Technology Network Management (MTNM) NML-EML Interface Business Agreement is being issued as Member Evaluation Version 3.1. It can be considered valid until further noticed by the Telemanagement Forum ("TM Forum"), at which time the TM Forum expects to update it to reflect comments from implementation experience, as well as to reflect additional member comment.

The Multi-Technology Network Management (MTNM) NML-EML Interface Business Agreement Version 3.1 supersedes the TM Forum Multi-Technology Network Management (MTNM) NML-EML Interface Business Agreement Version 3.0 in its entirety.

The purpose of an Evaluation Version is to encourage input based on experience of members and the public as they begin to use the document. Following the Evaluation Period, documents that are seen to deliver value are candidates for formal approval by the TM Forum. All documents approved by the TM Forum undergo a formal review and approval process.

This document will continue under formal change control. Supporting work will be issued as revisions to this document. A document of this type is a "living document," intended to capture and communicate current knowledge and practices. Further inputs will be made because of detailed work ongoing in the TM Forum and the industry.

The specification of the NML-EML interface is contained in a set of four documents:

- TMF513 - MTNM Business Agreement (this document)
- TMF608 - MTNM Information Agreement
- TMF814 - MTNM Solution Set
- TMF814A - MTNM Implementation Statement Template and Guidelines

In addition to these specification documents there are a number of supporting documents that are provided as part of the NML-EML Interface specification. These supporting documents contain details on various specific aspects of the NML-EML Interface and are intended to help clarify some of the specific aspects of the interface.

The details of the NML-EML Interface are specified in the above mentioned documents and it is expected that anyone wishing to understand the details of this interface be familar with these documents. Specifically it is expected that TMF 513 which specifies the requirements and the use cases should be considered as the entry point into the document set.

TMF 513 defines the requirements for the interface both in terms of the objects and operations that the interface is required to support and the behavior of the interface. TMF 513 contains a mapping to TMF 608 this mapping links the requirements to the classes and operations specified in UML information model.

TMF 608 specifies the details of the classes their relationships and their operations behavior that are required to support the requirements of the NML-EML Interface. Hence TMF 513 and TMF 608 fully define the NML-EML Interface.

TMF 814 specifies a CORBA IDL implementation of the NML-EML Interface. In defining a specific implementationthat meets the specification defined in TMF 513 and TMF 608 it is necessary to consider implementation specific issues such as efficiency, compatibility etc.

Document Lifecycle

TM Forum Business Agreements

A Business Agreement has two parts, i.e., the Business Problem Statement and the Business Requirement Model. The Problem Statement is customer oriented and written in business terms. It defines the problem, the scope and the objectives, recognizes the customer point of view, and identifies both process and information needs or issues. The Business Requirement Model consists of implementation independent description that includes categorized Requirement Statements, a set of Use Cases, and static and dynamic UML diagrams.

It is expected to be in draft form before the start of implementation of applications, systems and system interfaces. Several of the components of the BA are produced using Paradigm Plus UML diagrams. TM Forum uses the Paradigm Plus tool to produce its Business Requirement Models.

For a description of the relationship between the different documents produced by TMF projects, please see TMF Process documentation, templates and guides. For a description of TMF methodology for protocol neutral modeling, please see TMF methodology documentation.

Use and Extension of a TM Forum Business Agreement

This document defines the business problem and requirement model for NML-EML Interface. The Business Agreement is used to gain consensus on the business requirements for exchanging information among processes and systems in order to solve a specific business problem. The Business Agreement should feed the development of Information Agreement(s), which is a technology-neutral model of one or more interfaces. While the Business Agreement contains sufficient information to be a “stand alone” document, it is better read together with the Information Agreement document (TMF 608). Reviewing the two documents together helps in gaining a full understanding of how the technology neutral information model is defined for this requirement model. An initial Business Agreement may only deal with a subset of the requirements. It is acceptable for subsequent issues of the document to add additional requirements not addressed by earlier releases of the Business Agreement. Business Agreements are the basis for requirement traceability for information models.

It is expected that this document will be used:

- As the foundation for a TM Forum Information Agreement(s)
- To facilitate requirement agreement between Service Providers and vendors
- As input to a service Provider's Request for Information / Request for Proposal (RFI/RFP-RFX)
- As input for vendors developing Commercial Off The Shelf (COTS) products

As a source of requirements for other bodies working in this area.

Document History

Version	Date	Description of change
3.1	March 07	Added support for Connectionless Technology management (refer to Section 5 and Section 9). Added support for Control Plane connection management (refer to Section 6 and Section 8) Changes to align with ETSI Equipment Model TS 102 359: • added a manufacturerDate attribute to Managed Element (ME), Equipment, and Equipment Holder objects. • added a manufacturer attribute to the Equipment Holder object. • added a location attribute to the Equipment Holder object. Changed the Subnetwork object to MultiLayer Subnetwork (MLSN). Added the following attributes to the Subnetwork Connection (SNC) object: • Route Group Id • Bundle Id Updated the Transmission Descriptor (TMD) requirements to support the Matrix Flow Domain (MFD). Updated the Topological Link (TL) Inventory requirements (refer to {Requirement II. 380}). Updated the Supporting Termination Point (TP) Inventory requirements to support the Matrix Flow Domain (MFD). Updated the Inventory Notifications requirements (refer to Section 4.2.1.2) to support the new objects. Updated Verification of Transmission Descriptor (TMD) Assignment (refer to Section 4.2.2.1.2.2) requirement to support the Matrix Flow Domain (MFD). Updated Transmission Descriptor (TMD) Management (refer to Section 4.2.2.6) requirement to support the Matrix Flow Domain (MFD). Updated Maintenance Commands to include new types of operations. Corrected a problem with the re-use of requirement number II. 223 in release 3.0. For the Alarm Root Cause Indication the requirement number has been changed from II. 223 to II. 298.

Acknowledgements

The Multi-Technology Network Management (MTNM), NML-EML Interface, is a genuinely collaborative effort. The TeleManagement Forum would like to thank the following people for contributing their time and expertise to the production of this document. It is just not possible to recognize all the organizations and individuals that have contributed or influenced the MTNM Interface. We apologize to any person or organization we inadvertently missed in these acknowledgments.

Key individuals that reviewed, provided input, managed, edited and determined how to utilize inputs coming from all over the world, and really made this document happen were:

- Steve Arnott, Marconi Communications
- Beau Atwater, Telcordia Technologies
- François Bèdard, Nortel Networks

- Ian Best, TMForum
- Malcom Betts, Nortel Networks
- Teddy Bezoza, AT&T
- Alex Bojanic, Tellabs
- Harrie Buiting, Alcatel-Lucent
- Debbie Burkett TMForum
- Henri Campagno, France Telecom
- Lily Chen, Verizon
- Lee Chow, Tellabs
- Merav Cohen-Sarussi, ECI Telecom
- Nigel Davis, Nortel Networks
- Keith Dorking, Ciena
- Shane Edmonds, Alcatel-Lucent
- Felix Flemisch, Siemens
- Marianne Franklin, Lucent Technologies
- Steve Fratini, Telcordia Technologies
- Einat Gabo-Rafaeli, ECI Telecom
- Steve Gaito, TM Forum
- Salim Galou, Fujitsu
- Elizabetta Gardelli, Siemens
- CB Genrich, Hitatchi
- Noam Goldberg, ECI Telecom
- Floyd Goldstein, Alcatel-Lucent
- Alison Greig, Nortel Networks
- Gary Grimes, Marconi Communications
- Milton Hall, Telcordia Technologies
- Dragan Jovanovic, Acterna
- Roshan Joyce, Fujitsu
- Nathan Kohn, Alcatel-Lucent
- Hing-Kam Lam, Alcatel-Lucent
- Bertrand LeBorgne, Alcatel-Lucent

- François Leroux, Fujitsu
- Tom Ligda, Cisco Systems
- Peter Litz, Siemens
- Astrid Lozano, Nortel Networks
- Menachem Malkosh, ECI Telecom
- Andrea Mazzini, Alcatel-Lucent
- Srini Nangavaram, Alcatel-Lucent
- Frank O’Rourke, Marconi Communications
- Sandip Patel, Alcatel-Lucent
- Noga Peled, ECI Telecom
- Edgar Riemann, Siemens
- Jeffrey Ross, Fujitsu
- Victor Santos, Acterna
- Horst Schukat, Siemens
- Eric Seldner, Telcordia Technologies
- Charles Shreidan Nortel Networks
- Sidney Shiba, Fujitsu
- Wendy Teller, Tellabs
- Niranjan Tripathy, Fujitsu
- Alexander Tschersich, Siemens
- Gerard Vila, Alcatel-Lucent
- Xuli Wu, Tellabs
- Bernd Zeuner, Deutsche Telekom
- Yakov Zimmerman, ECI Telecom

Although not directly used within this document, access to documentation and work from standards bodies and other forums have contributed to the evolution of the Multi-Technology Network Management (MTNM) Business Agreement. This access was via public information or TM Forum member knowledge. This list of standards bodies and forums is not inclusive and does not imply review and concurrence by these organizations or their representatives. It is important however to acknowledge the work and their influence on the TeleManagement Forum work:

- American National Standards Institute (ANSI)

- ATM Forum
- DSL Forum
- European Telecommunications Standards Institute (ETSI)
- Institute of Electrical and Electronics Engineers (IEEE)
- International Telecommunications Union - Telecommunication Standardization Sector (ITU-T)
- Internet Engineering Task Force (IETF)
- Metro Ethernet Forum (MEF)
- MPLS and Frame Relay Alliance
- Object Management Group (OMG)
- Optical Interworking Forum (OIF)
- Telcordia Technologies

About TeleManagement Forum

TeleManagement Forum is an international consortium of communications service providers and their suppliers. Its mission is to help service providers and network operators automate their business processes in a cost- and time-effective way. Specifically, the work of the TM Forum includes:

- Establishing operational guidance on the shape of business processes.
- Agreeing on information that needs to flow from one process activity to another.
- Identifying a realistic systems environment to support the interconnection of operational support systems.
- Enabling the development of a market and real products for integrating and automating telecommunications operations processes.

The members of TM Forum include service providers, network operators and suppliers of equipment and software to the communications industry. With that combination of buyers and suppliers of operational support systems, TM Forum is able to achieve results in a pragmatic way that leads to product offerings (from member companies) as well as paper specifications.

