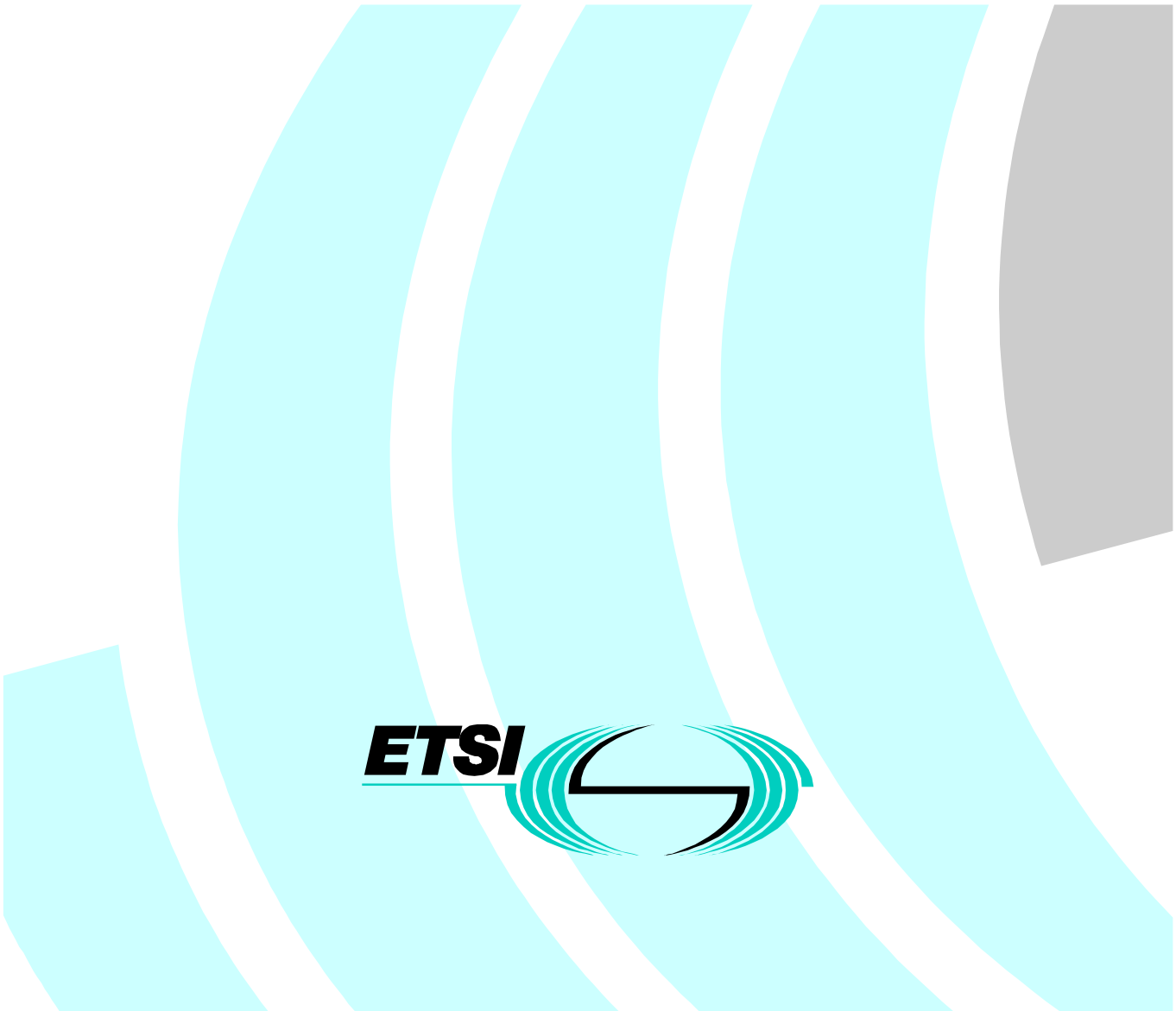
Draft ES 200 653 V1.2.1 (1999-02)

ETSI Standard

Telecommunications Management Network (TMN); Network level generic class library



Reference
RES/TMN-00030 (61o00iop.PDF)

Keywords
information model, interface, MO, TMN

ETSI

Postal address
F-06921 Sophia Antipolis Cedex - FRANCE

Office address
650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16
Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Internet
secretariat@etsi.fr
Individual copies of this ETSI deliverable
can be downloaded from
http://www.etsi.org
If you find errors in the present document, send your
comment to: editor@etsi.fr

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available **free of charge** from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://www.etsi.org/ipr).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This ETSI Standard (ES) has been produced by ETSI Technical Committee Telecommunications Management Network (TMN), and is now submitted for the ETSI standards Membership Approval Procedure.

Introduction

The present document provides a library of managed objects, for modelling the network level view described in ITU-T Recommendation M.3100 [10]. It identifies those Telecommunication Management Network (TMN) network level managed object classes that are generic (i.e. potentially apply to more than one specific information model).

These object classes are additional to those specified in I-ETS 300 293 [1] which enhances and extends ITU-T Recommendation M.3100 [10] in the area of the network element management view.

Whereas I-ETS 300 293 [1] concentrated on the network element view, the present document extend the library of generic object classes available in the area of network level modelling (i.e. the network level view).

Although the work on the development of network level view managed object classes is at an early stage in its evolution, the present document has been published to enable technology specific groups to profile the object classes in the present document to produce implementable models (e.g. technology specific models). The Technology specific groups are encouraged to document their models in the form of an Ensemble.

No conformance statements have yet been prepared for these object classes. These will be produced as part of the Ensemble process.

1 Scope

The present document describes the generic managed object class library for the network level view. It identifies those Telecommunication Management Network (TMN), as defined in ITU-T Recommendation M.3010 [8], network level managed object classes that are generic (i.e. potentially apply to more than one specific information model).

ITU-T Recommendation M.3100 [10] is extended by I-ETS 300 293 [1] in the area of the network element view, and this the present document in the area of the network level view.

The present document addresses generically the abstractions of those aspects of telecommunication resources required to manage the network (e.g. equipment, networks and telecommunication services). It also includes the abstractions related to the management services.

The present document does not address abstractions relevant to technology specific areas or implementation specific details.

The class library defined in the present document specifies the managed objects that define the management interfaces between a user and a service provider where these exist on separate systems. User and service provider refer to network capabilities and should not be confused with service management terminology. The use of the class library between the Network layer Operations System Function (OSFN) and the Network Element layer Operations System Function (OSFE) (see figure 2) is to support a network level view. Other uses of the class library across this interface are for further study.

The present document can be used for the definition of models to support TMN management services and/or management function sets using the TMN interface specification methodology (ITU-T Recommendation M.3020 [9]).

Following this methodology, the technique for the production of interfaces is divided into the following stages:

1) the definition of requirements upon which the managed object model will be based;

2) the translation of the above requirements into a generic object class library;

3) the specification of one or more interfaces;

4) the production of a set of conformance requirements.

The present document covers stages 1 and 2. Stages 3 and 4 are to be completed by technology groups for specific applications using profiling formats such as Ensembles and International Standardized Profiles (ISPs).

The purpose and field of application for the present document are as given in ITU-T Recommendation M.3100 [10].

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.
- A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same number.

[1] I-ETS 300 293: "Telecommunications Management Network (TMN); Generic managed objects".

[2] ETS 300 455-1: "Broadband Integrated Services Digital Network (B-ISDN); Broadband Virtual Path Service (BVPS); Part 1: BVPS for Permanent communications (BVPS-P)".

[3] ETS 300 469: "Broadband Integrated Services Digital Network (B-ISDN); Asynchronous Transfer Mode (ATM); Management of the network element view [ITU-T Recommendation I.751 (1996)]".

[4] ETR 037: "Network Aspects (NA); Telecommunications Management Network (TMN); Objectives, principles, concepts and reference configurations".

[5] ETR 046: "Network Aspects (NA); Telecommunications management networks modelling guidelines".

[6] ITU-T Recommendation G.803 (1993): "Architectures of transport networks based on the synchronous digital hierarchy (SDH)".

[7] ITU-T Recommendation G.805: "Generic functional architecture of transport networks".

[8] ITU-T Recommendation M.3010 (1992): "Principles for a telecommunications management network".

[9] ITU-T Recommendation M.3020 (1992): "TMN interface specification methodology".

[10] ITU-T Recommendation M.3100 (1992): "Generic network information model".

[11] ITU-T Recommendation M.3200 (1992): "TMN management services: overview".

[12] ITU-T Recommendation M.3400 (1992): "TMN management functions".

[13] ITU-T Recommendation X.721 | ISO/IEC 10165-2: (1992): "Information technology - Open Systems Interconnection - Structure of management information: Definition of management information".

[14] NMF Forum 25 (1992): "The Ensemble Concepts and Format".

[15] ITU-T Recommendation X.725: "Information technology - Open Systems Interconnection - Structure of management information: General Relationship Model".

[16] ITU-T Recommendation I.326: "Functional architecture of transport networks based on ATM".

[17] ITU-T Recommendation M.1400: "Designations for international networks".

[18] ITU-T Recommendation X.722 (1992): "Information technology - Open Systems Interconnection - Structure of Management Information: Guidelines for the definition of managed objects".

[19] ITU-T Recommendation X.208: "Specification of Abstract Syntax Notation One (ASN.1)".

[20] ITU-T Recommendation X.720: "Information technology - Open Systems Interconnection - Structure of management information: Management information model".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply.

a layer, or transport network layer: a layer, or transport network layer, is defined as ITU-T Recommendation G.805 [7] a topological component solely concerned with the generation and transfer of characteristic information.

partitioning: partitioning is defined in ITU-T Recommendation G.805 [7] as a framework for defining the network structure within a network layer.

profile: a profile of a managed object is the additional normative text which is required to restrict conditionality (e.g. specifies that a conditional package is or is not present) and specifies additional behaviour which may be required for a given implementation.

Ensemble: an Ensemble is the result of a particular profiling technique which provides a requirements-based view of a particular solution to a management problem. Ensembles are described in the NM Forum 25 specification document NMF Forum 25 [14].

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ABR	Available Bit Rate
ASN.1	Abstract Syntax Notation One
ATM	Asynchronous Transfer Mode
CBR	Constant Bit Rate
CP	Connection Point
CTP	Connection Termination Point
FCAPS	Fault, Configuration, Accounting, Performance, Security
GDMO	Guidelines for the Definition of Managed Objects
GOM	Generic Object Model
IA	Indirect Adapter
ISP	International Standard Profile
LLA	Logical Layered Architecture
LOS	Loss Of Signal
MSP	Multiplex Section Protection
NE	Network Element
NEF	Network Element Function
NMF	Network Management Forum
OS	Operations System
OSF	Operations System Function
PDH	Plesiochronous Digital Hierarchy
PNO	Public Network Operator
QoS	Quality of Service
RDN	Relative Distinguished Name
SDH	Synchronous Digital Hierarchy
SNC	Sub-Network Connection
SP	Service Provider
TIB	Table Information Based
TMN	Telecommunications Management Network
TP	Termination Point
TTP	Trail Termination Point
VBR	Variable Bit rate

4 General description of the class library

The class library specified in the present document is aimed at supporting the definition of interfaces for the network level view as defined in ITU-T Recommendation M.3100 [10].

"*There are several different viewpoints of management information which may be defined for management purposes, with the Network Element level viewpoint, the Network level viewpoint and the Service level viewpoint defined below. These viewpoints are not restrictive but define the levels of abstraction of particular types of interfaces. That is, object class definitions are not forced into this categorization but are constructed to meet the needs of exchanging management information across TMN interfaces. Objects defined for a given viewpoint may be used in others, and any object may be used by any interface which requires it. The definition of viewpoint is a means of generating requirements, hence there is no implicit definition of interfaces or storage requirements. This information is defined for the purpose of management via an open interface.*

The Network Element level viewpoint is concerned with the information that is required to manage a Network Element (NE). This refers to the information required to manage the NEF and the physical aspects of the NE. The information may be derived from open systems other than the NE.

The Network level viewpoint is concerned with the information representing the network, both physically and logically. It is concerned with how network element entities are related, topographically interconnected, and configured to provide and maintain end-to-end connectivity.

The Service level viewpoint is concerned with how Network level aspects (such as an end-to-end path) are utilized to provide a network service, and as such is concerned with the requirements of a network service (e.g. availability, cost, etc.) and how these requirements are met through the use of the network, and all related customer information."

The class library is a management information library which contains definitions of managed object classes expressed in GDMO templates, packages, attributes, name bindings, and actions. It represents an abstraction of the network and its network management capabilities.

This class library may be profiled to take into account specific types of network, for example:

a) an SDH ring;

b) an ATM sub-network;

c) a network containing two peer-to-peer OSs owned by different Public Network Operators (PNOs);

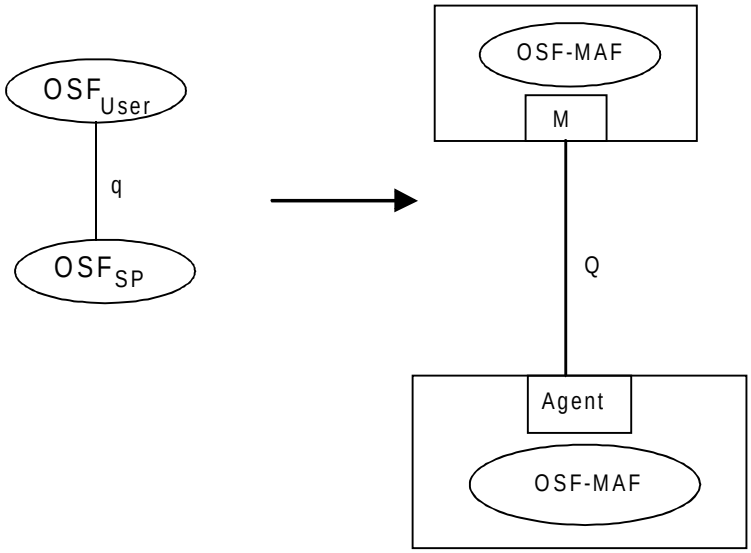
d) an Optical Access Network.

While it is the intention to extend the class library to cover a wide range of network technologies, the applicability of the present document, (i.e. the object classes) listed in the library, has not been checked for networks or technologies other than the ones listed as examples above.

The ITU-T Recommendation G.805 [7] functional architecture is used to describe the network resources for these networks. An enhanced functional architecture will be used if required for consideration of new types of network.

5 Functional architecture

A given q reference point may be characterized by an Operations System Function (OSF) which is a service provider and an OSF which is a service user. These two OSFs are represented by the OSF_{SP}, and the OSF_{User} respectively (see figure 1). Where the q reference point becomes an external interface, the OSF_{User} corresponds to the Manager (M), and the OSF_{SP} corresponds to the Agent (A).



MAF: Management Application Function
OSF_{SP}: Operations System Function in role of Service Provider
OSF_{User}: Operations System Function in role of service User

Figure 1: Service provider and service user roles of OSFs

If an OSF supports more than one q reference point, then the OSF may take on different roles for different q reference points. For example, OSFN is a service provider for the q3sn reference point and a user for the q3ne reference point.

For the purpose of the present document the element manager is represented by the Operations System Function, OSFE, within the element management layer (see figure 2).

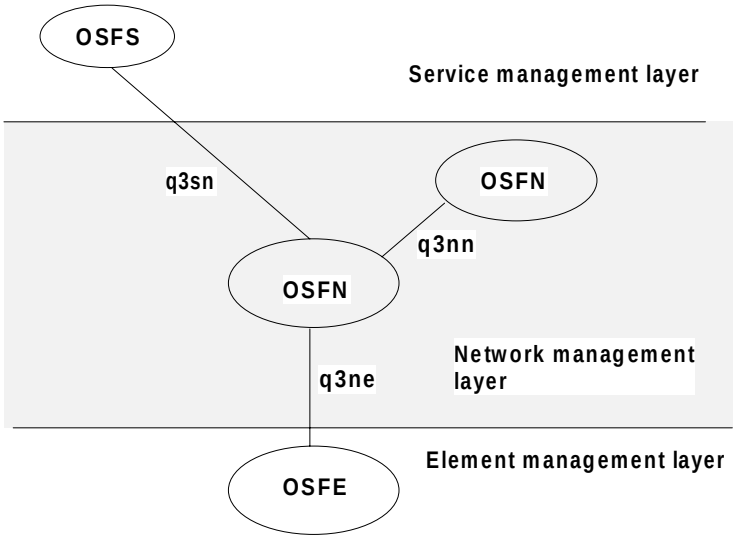


Figure 2: The TMN (management layer) view of this class library

Figure 3 which is based on figure B.3 of ETR 037 [4], clarifies the position of the reference points defined in figure 2. In this figure possible network level reference points have been high-lighted in bold.

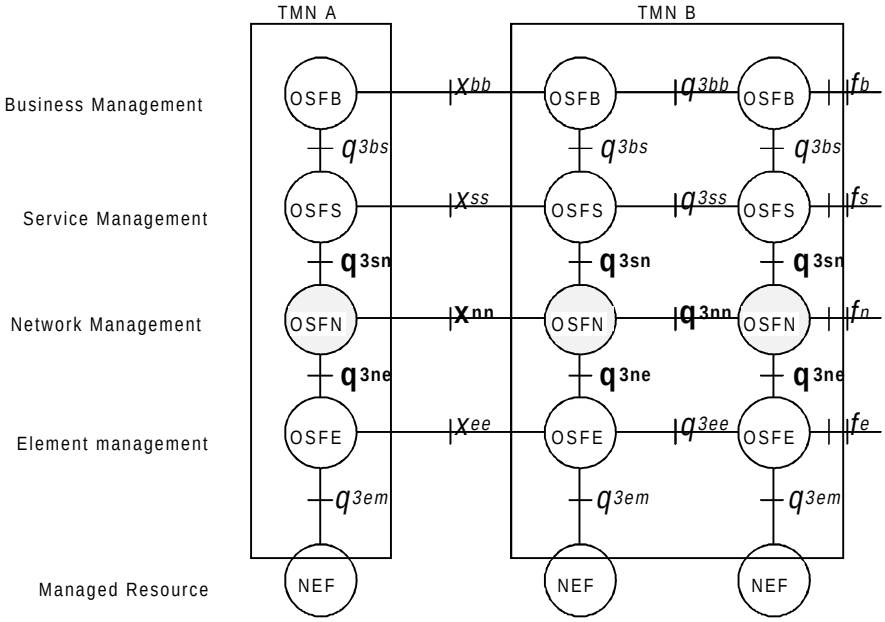


Figure 3: The reference points within the TMN architecture

The user OSF has the responsibility for a "larger" part of the network, which it undertakes by co-ordinating the activities of a number of service provider OSFs each of which has responsibility for a smaller parts of the network.

The service provider OSF is then responsible for the performance of the service (including, where appropriate, the maintenance of the service).

6 User guide to the network level view class library

6.1 Introduction

In order to successfully use the class library, the following points should be borne in mind:

- it is assumed that users of the class library will be following the TMN interface specification methodology (ITU-T Recommendation M.3020 [9]);
- technology specific groups should understand that the present document is a collection, or library, of managed object classes which may be applicable to their network management requirements. Where functionality required in a network management interface (in a given technology) can be modelled using the classes in this library, it is strongly recommended to use them. In order to satisfy specific technology requirements, specialization and profiling of the class library should be used. However, in the cases where the object classes of the library are not applicable to a given network management requirement of a particular interface, it is not intended to force such object classes to be used;
- the class library is aimed at satisfying the requirements of a wide range of groups. Accordingly there is a large amount of optionality in the classes. It is not the intention that the classes used across an interface should contain this degree of optionality;
- it is essential, therefore, that the classes are profiled, and a method such as Ensembles is strongly recommended so that the requirements behind this profiling are explicit. Profiling notes are included in the text of the classes to assist this process. All profiling notes are informative;
- ETR 046 [5] should be used when profiling these classes.

6.2 Relationship to ITU-T Recommendation M.3100

Where possible the modelling techniques in ITU-T Recommendation M.3100 [10] have been utilized to model a given requirement. Although the ITU-T Recommendation M.3100 [10] classes were primarily developed for interfaces to Network Elements, extensive use has been made of the ITU-T Recommendation M.3100 [10] modelling principles. In addition, some of the attributes and ASN.1 syntax definitions have been re-used.

6.3 Modelling technique

The class library can be used for the definition of models to support TMN management services and/or management function sets using the TMN interface specification methodology (ITU-T Recommendation M.3020 [9]), as illustrated in figure 4.

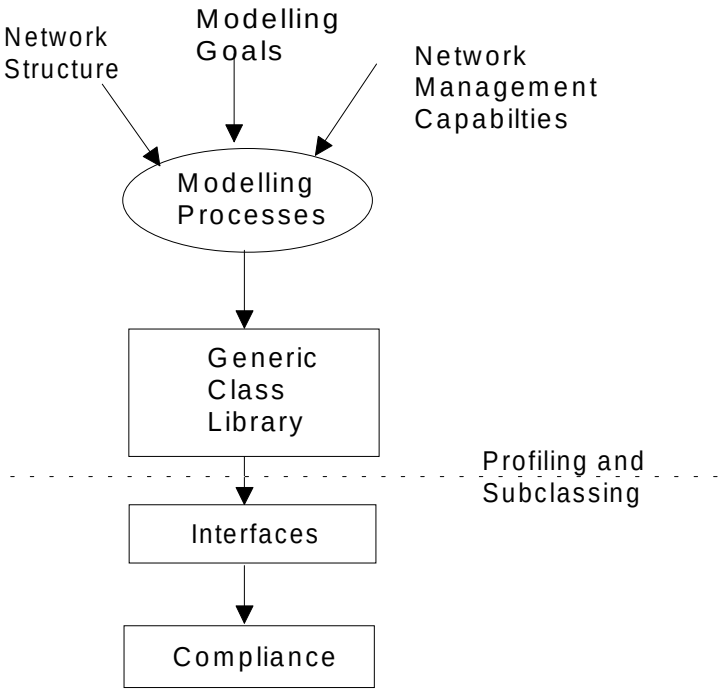


Figure 4: Model definition process

The class library may be specialized by technology groups using a profiling technique, such as the Ensemble technique given in I-ETS 300 293 [1], to produce a specific interface. This is illustrated in figure 5.

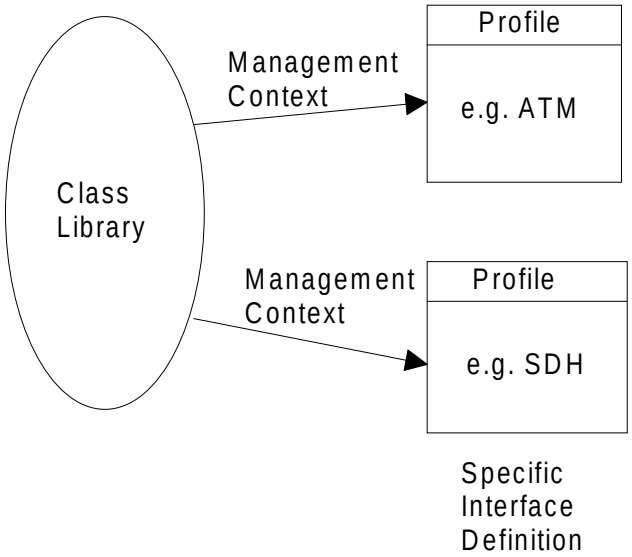


Figure 5: Use of profiles

For some applications it may be possible to use a profile of this class library, and instantiate the classes directly. However, for many applications there will be a need to add additional behaviour, and to add technology specific features. This may be done by inheritance or containment, as illustrated in figure 6.

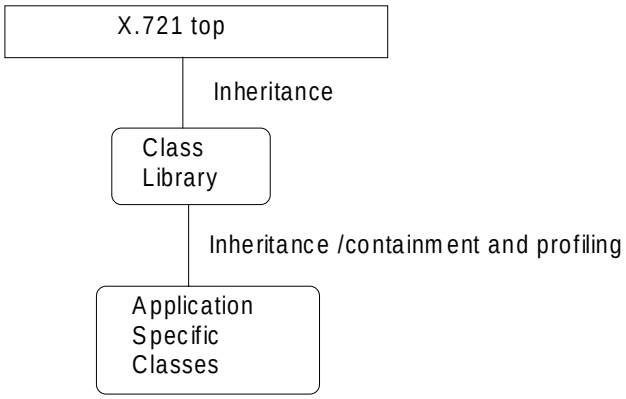


Figure 6: Derivation of application-specific classes

Since this class library only addresses configuration management aspects, it will be necessary to construct a complete object if other functions such as performance and testing need to be added. It is recommended that the composition is effected as part of an Ensemble. Two methods are available:

Method 1: Multiple inheritance

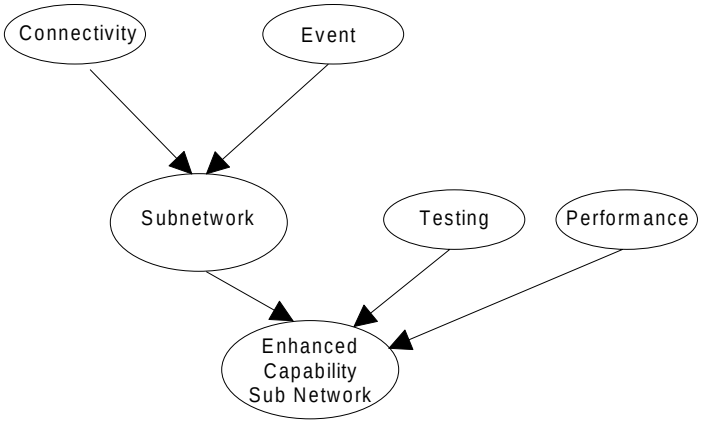


Figure 7: Composition of objects by multiple inheritance

In this method functions are defined as separate objects or packages which are incrementally inherited to produce the required capabilities.

Method 2: Naming

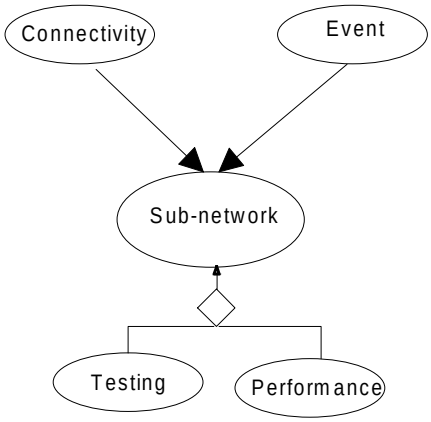


Figure 8: Composition of objects by naming

In this approach the original sub-network is formed from inheritance (or multiple packages) but subsequent functionality is added to by naming the appropriate object.

6.4 Using the TMN methodology

A pass through the TMN interface specification methodology (ITU-T Recommendation M.3020 [9]) should be made for each TMN management service or TMN management function set.

The following indicates the information which should be captured during task 1 and 2 of the methodology (this is taken from the proposed revision of ITU-T Recommendation M.3020 [9]).

6.4.1 Management service description

Use annex B of ITU-T Recommendation M.3200 [11], as a possible source.

6.4.2 Management goals

This subclause should give a clear description of the TMN users benefit, i.e. the reason for carrying out this management. Background and context should be added as necessary, but the explanatory and descriptive part and descriptions should be separated. Supporting background information, where required, should be placed in an annex.

6.4.3 Management context description

The objective of a TMN management context description is to capture, in a uniform way, relevant information on the management of a certain telecommunication area. The objective is to document the relevant information that leads to the definition of TMN management function sets and their corresponding functions. Management context can be described by using the orthogonal three components, roles, resources and functions.

6.4.4 Roles

This subclause should provide a description of roles identified for this management context (Maintenance, Provisioning, Installation, Testing, etc...). Roles should be listed in TIB B.

6.4.5 Resources

This subclause provides a description of the logical and physical telecommunication resources which shall be modelled by an(some) managed object(s). These should be clearly defined and stored in TIB B. Management layers of the network (Element, Network, Service, Business) can be used as classification guide. See ITU-T Recommendation M.3010 [8] for the description of these layers.

6.4.6 TMN management functions

This subclause should provide a description of TMN management functions (function sets/function set groups) to be used in achieving the management goals. They should be stored in TIB B. Guidelines for defining these TMN management functions are found in annex B of ITU-T Recommendation M.3020 [9].

6.4.7 Management scenarios

This subclause should provide examples of management interaction using TMN management information definition and TMN systems management services and messages.

Identification of management function sets (see ITU-T Recommendation M.3400 [12]).

Identification of management functions (related to ITU-T Recommendation M.3400 [12]).

Identification of applicable reference points (e.g. Q, X, F).

6.5 Documenting the model

Users of this class library are strongly recommended that, to assist readers understanding, the requirements for any profiling are explicitly documented along with the model.

A number of formats for documenting models exist. The Ensemble technique as defined by the Network Management Forum (NMF) is recommended. The Ensemble is described in annex F.

7 Normative requirements

NOTE: The mapping of the requirements to the managed object classes is given in annex B.

This clause gives the requirements that the class library satisfies, except for those marked for further study. These requirements comprise a set of modelling goals, a description of the resources to be managed, and the management capabilities which are supported, as illustrated in figure 4.

7.1 Modelling goals

The modelling goals listed below have been followed:

1) the managed object model shall support the concepts of network partitioning and network layering as defined in the network functional architecture (e.g. ITU-T Recommendations G.803 [6] for SDH, I.326 [16] for ATM, and G.805 [7], the generic architecture);

2) the service provider OSF shall manage one or several levels of partitioning (within a layer network) or one or several layer networks;

3) it shall be possible to manage client and server layers independently; for example to separate client layers in a service user from server layers in a service provider;

4) the model shall accommodate information not necessarily visible from the NE View, and information concerned with the management of associations between NEs;

5) the model shall provide support for requirements originating from access, switching and transport systems for a number of technologies (e.g. SDH, ATM, PDH, ISDN, B-ISDN, Optical Access), and shall not be restricted to a single technology. The model may be profiled and/or sub-classed to satisfy the requirements of a particular technology;

6) the managed object model shall accommodate the management layer concept of the TMN Logical Layered Architecture (LLA);

7) the managed object model shall allow for the management of a single or multiple LLA management layers by a single management system;

8) the managed object model shall accommodate intra-TMN (within one TMN and inter-TMN (between TMNs) management;

9) the managed object model shall accommodate different partitioning criteria, for example:

   a) geographic criteria/view;

   b) administrative domains;

   c) routing domains.

The managed object model shall allow overlapping and non-coincident management domains.

Different aspects of a sub-network shall be manageable by different OSFs.

Here "aspect" may include:

1) the functional decomposition (e.g. into the different FCAPS functional areas);

2) domain boundaries.

For example a given sub-network may be managed by one OSF for configuration, but may report events to a separate OSF.

This class library shall allow the management domains for different functions (e.g. routing) and maintenance to be different.

7.2 Resources

This subclause defines all the resources or components of resources that are to be the subject of this class library. There is a process of abstraction from these resources to produce the class library definitions.

The network resources to be managed are described below. These resources are based on a functional architecture. This architecture is defined by the entities and concepts defined within ITU-T Recommendation G.805 [7].

The resource definitions given below are extracted from ITU-T Recommendation G.805 [7] for the convenience of the reader. The resources described are:

- characteristic information;
- sub-networks;
- access groups;
- links;
- trails;
- connections (link connections and sub-network connections);
- tandem connections;
- tandem connection bundles;
- access points;
- connection points;
- adaptation function;
- trail termination function;
- termination connection points.

The following describes the layer network and the resources that make it up in a technology independent way (terms in *italics* refer to ITU-T Recommendation G.805 [7] entities described in other sections within this subclause).

7.2.1 Layer network

A layer network is defined by the complete set of like access *points* which may be associated for the purpose of transferring information. The information transferred is characteristic of the layer and is termed *characteristic information*. *Access point* associations may be made and broken by a layer management process thus changing its connectivity (i.e. the establishment or clearing down of *trails*). A separate, logically distinct layer network exists for each *access point* type. A layer network is made up of *sub-networks* and *links* between them. A layer network may serve a client layer network by transporting the *characteristic information* of the client layer within a signal of *characteristic information* of its own layer.

7.2.2 Characteristic information

Characteristic information is a signal of characteristic rate and format which is transferred within and between *sub-networks* and presented via an adaptation function to an *access point* for transport by a server *layer network*. (The adaptation function adapts the signal so that it may be transported by the server *layer network*, e.g. by multiplexing several client layer signals together.)

7.2.3 Sub-network

A sub-network describes the potential for *sub-network connections* across the sub-network. It can be partitioned into interconnected sub-networks and *links*. Each sub-network in turn can be partitioned into smaller sub-networks and links and so on. It is defined by the complete set of like *connection points* which may be associated for the purpose of transferring *characteristic information*. The *connection point* associations in a sub-network may be made and broken by a layer management process thus changing its connectivity (i.e. the establishment or clearing down of *sub-network connections*).

7.2.4 Link

A link describes the fixed relationship between a *sub-network* and another *sub-network* or *access group*. It is defined by the sub-set of *connection points* on one *sub-network* which are associated with a sub-set of *connection points* or *access points* on another *sub-network* or *access group* for the purpose of transferring *characteristic information*. The link represents the topological relationship between a pair of *sub-networks*.

7.2.5 Access point

An access *point* is where the adapted characteristic information from a client *layer network* enters the server *layer network*. It is the point where the adapted *characteristic information* is bound to a trail termination function, and thus the point where the adapted *characteristic information* enters the *trail*. (Trail termination generates the *characteristic information* of a *layer network* and ensures integrity of transport of that *characteristic information*.)

7.2.6 Access group

An access group is a group of co-located *access points* together with their associated trail termination functions. (Trail termination generates the *characteristic information* of a *layer network* and ensures integrity of transport of that *characteristic information*.).

NOTE: An access point does not have to belong to an access group.

7.2.7 Connection point

From ITU-T Recommendation G.805 [7]: A connection point is a "reference point" that consists of a pair of co-located "unidirectional connection points", and therefore represents the binding of two paired bi-directional "connections".

Unidirectional Connection point - a "reference point" that represents the binding of the output of a "unidirectional connection" to the input of another "unidirectional connection".

A connection point is where:

1) a link connection may be bound to another link connection;

2) a link connection may be bound to a sub-network connection;

3) a sub-network connection may be bound to another sub-network connection.

7.2.8 Trail

A trail in a server *layer network* is responsible for the integrity of transfer of *characteristic information* from one or more client *layer networks* between the server layer *access points*, utilizing the *characteristic information* of its own layer. It defines the association between *access points* in the same *layer network*. Trail termination functions at either end of the trail monitor the integrity of transfer by adding incremental information to the adapted *characteristic information* from the client *layer networks.* These trail termination functions are thought of as being part of the *trail*.

7.2.9 Link connection

A link connection is supported by a *trail* in the server *layer network*. It is capable of transferring information transparently across a *link* between two *connection points* or between a *termination connection point* and a *connection point* in the case of a *link connection* at the boundary of a *layer network.*

7.2.10 Sub-network connection

A sub-network connection is capable of transferring *characteristic information* across a *sub-network* transparently. It is delimited by *connection points* at the boundary of the *sub-network* and represents the association between *connection points* within the same *sub-network*. Sub-network connections are in general made up of a concatenation of lower level sub-network connections and *link connections* and can be viewed as an abstraction of this more detailed view.

A sub-network connection may be set-up between any two ports or groups of ports at the boundary of the same sub-network.

7.2.11 Tandem connection (for further study)

A tandem connection is an arbitrary series of *link connections* and *sub-network connections*.

7.2.12 Tandem connection bundle (for further study)

A parallel set of *tandem connections* with co-located end points.

7.2.13 Adaptation function

The Adaptation function is a "transport processing function" which adapts a server layer to the needs of a client layer. The "adaptation" function defines the "server/client" association between the "connection point" and "access point" and these points therefore delimit the "adaptation" function. "Adaptation" functions have been defined for many "client/server" interactions.

7.2.14 Trail termination function

The Trail termination function is a "transport processing function" which generates the "characteristic information" of a layer network and ensures integrity of that "characteristic information". The "trail termination" defines the association between the "access point" and "termination connection point" and these points therefore delimit the "trail termination".

The Trail termination source is a "transport processing function" which accepts adapted client layer network "characteristic information", adds "trail" overhead and assigns it to an associated "network connection" in the same "transport network layer".

The Trail termination sink is a "transport processing function" which terminates a "trail", extracts the "trail" overhead information, checks validity and passes the adapted client layer network "characteristic information" to the "adaptation" function.

7.2.15 Termination connection point

From ITU-T Recommendation G.805 [7]: **a Termination connection point** is a reference point that consists of a pair of co-located unit directional termination connection points, and therefore represents the binding of a trail termination to a bi-directional connection.

Unidirectional Termination connection point: a reference point that represents the following bindings: output of a trail termination source to the input of a unidirectional connection or; the output of a unidirectional connection to the input of a trail termination sink.

A termination point is where:

1) a link connection may be bound to a trail termination function (associated with an access point) forming the end of a trail;

2) a subnetwork connection may be bound to a trail termination function (associated with an access point) forming the end of a trail.

7.2.16 Connection modes and directionality

The **directionality** of a connection indicates whether transmission is uni-directional or bi-directional.

The **mode** of a connection indicates the type of transmission, that is, point to point, point to multi-point, multicast, broadcast or conference.

Mode	Description
Point-to-point	One A end and one Z end.
Point-to-multipoint	One A end and multiple Z ends. There is no traffic flow between Z ends.
Multicast	Multiple A ends and multiple Z ends. There is no traffic flow between A ends or between Z ends.
Conference	Multiple A ends send traffic to, and receive traffic from, all other A ends. There are no Z ends. Other conference types are for further study.
Broadcast	One A end and multiple undefined Z ends.

Where required, the designation of the Connectivity object should follow ITU-T Recommendation M.1400 [17]. The ITU-T Recommendation M.1400 [17] designation is independent of the aEnd NWTPs and the zEnd NWTPs.

The designation of A end and Z end is arbitrary, except that in the case of uni-directional transmission the A end termination shall send information, and the Z end termination shall receive information.

7.3 Management capabilities

7.3.1 Overview

This subclause defines the management functions that can be performed on the resources described above using the class library.

This subclause focuses on what can be performed, rather than how it is performed. Annex B describes how these functions can be performed using the managed objects described in the present document.

The network management capabilities represent the functionality (dynamic requirements) that the class library shall support. In this subclause the OSI FCAPS (Fault, Configuration, Accounting, Performance and Security management) structure will be used.

7.3.2 Configuration management

Configuration management consists of:

- configuration connection management (dynamic); and

- configuration resource management (static).

Where connection configuration management is concerned by the set-up, modification and release of sub-network connections and link connections, and where resource configuration management is concerned about provisioning including connection points, sub-networks, links, layered network domains, administration domains.

Configuration connection management (dynamic):

1) sub-network connection set-up;
2) the release of sub-network connections;
3) sub-network configuration;
4) scheduling;
5) trail set-up and release;
6) the setting-up of network connections, which comprises:
   a) the configuration of links;
   b) the provisioning of link connections;
   c) tandem connection provisioning and configuration;
7) the release of network connections.

Configuration resource management (static):

8) the provisioning of a layer network and characteristic information;
9) the provisioning of access points;
10) the provisioning of access groups;
11) the configuration of access groups;
12) the provisioning of connection points;
13) the configuration of connection points;
14) the provisioning of sub-networks;
15) link provisioning.

7.3.2.1 Sub-network connection set-up

Basic sub-network connection set-up covers the setting up of a sub-network connection, with a limited set of facilities, in response to a request containing only the minimum amount of information that is required to set-up a sub-network connection.

Sub-network connections which are set-up using this procedure are released by a request from the user:

1) a user will have the ability to request the immediate (that is, non-scheduled) setting-up of a sub-network connection between any two groupings of connection points in the same sub-network;
2) a user will have the ability to request the scheduled setting-up of a sub-network connection between any two groupings of connection points in the same sub-network;

3) a user shall have the ability to specify the following values for the different types of information within a basic sub-network connection set-up request:

   - mode;
   - directionality;
   - a-end of the sub-network connection;
   - z-end of the sub-network connection;
   - capacity;
   - user identifier (basic);
   - transaction identifier;
   - bandwidth allocation;
   - scheduling;
   - end PNOs.

   For each direction of an ATM layer connection, a specific ATM Layer Quality of Service (QoS) from those supported by the network is requested at connection setup time. This requested QoS is embodied in the traffic descriptor (which is being defined by technology specific groups) associated with the ATM connection. The network commits to meet the requested QoS as long as the end system complies with the negotiated traffic contract.

   The requested QoS could be either indicated by the objective of each individual parameter or by a QoS class specification where the actual default minimum performance objective for each of the parameters will be standardized by technology specific groups;

4) a user shall have the ability to request a particular quality of connectivity service for the sub-network connection;

5) a user shall have the ability to request a two phase sub-network connection setup, where the resources are initially reserved before they are activated;

6) a user shall be informed of the result of the set-up:

   - in the case of a successful set-up the user will be sent a sub-network connection identifier, and in the case of implicit creation, the identifiers of the connection point, or termination connection points;
   - in the case of an unsuccessful set-up the user will be sent a fault case or fault indication indicating why the request was unsuccessful;

7) in the case of an unsuccessful set-up attempt any resource which has been "reserved" during the attempted set-up shall be returned to the available pool.

7.3.2.2 Sub-network connection release

1) A user may request the release of a previously set-up sub-network connection.

2) A user may request the un-reservation of a previously reserved sub-network connection.

   In this case a sub-network connection has been reserved but has not yet been activated, that is, the un-reservation interrupts a set-up connection request.

3) Any resources associated with the sub-network connection shall be returned to the available pool when the sub-network connection is released un-reserved, cancelled or de-scheduled. This includes the deletion of connection points or termination connection points when implicit creation was used for the set-up.

4) A user shall be informed if a sub-network connection is released due to a management action.

5) A user may request the de-scheduling of a previously scheduled sub-network connection.

   A sub-network connection will be released automatically (that is, by a management action without an explicit request from the user) at the stop time specified in the set-up request.

7.3.2.3 Sub-network configuration

A user will have the ability to add and remove connection points and termination connection points to/from a sub-network. An access point will be visible from all the levels of sub-network partitioning in which it is contained. A connection point will be visible from a particular level of sub-network partitioning if it provides access to that sub-network (i.e. it will not be visible if it is internal to the sub-network).

A user may require more than one view of the resources. Therefore the user will have the ability to add and remove access and connection points to/from multiple sub-networks taking part in separate partitioning structures.

7.3.2.4 Scheduling

Inspired by the bandwidth scheduling requirements in ATM networks, (see ETS 300 455-1 [2]), a model is defined here that captures those requirements in a generic format so that all technologies needing scheduling of sub-network connections can apply this mechanism independently of whether these technologies allow for flexible bandwidth allocation or not. Schedules can be of five basic types (according to their periodicity):

- duration: one single slot, not periodic connection;
- dailySchedule: several day slots with different bandwidth each;
- weeklySchedule: several week slots with different bandwidth each;
- monthlySchedule: several month slots with different bandwidth each;
- occasional: several non-periodic slots with different bandwidth each.

Accordingly, each slot will have a start point in time, a stop point in time and the associated bandwidth (with the implicit and appropriate periodicity):

1) it shall be possible for a user to request the set-up of a connectivity resource (i.e. a trail, a network connection, a sub-network connection or a link connection), at a future date (that is, a scheduled set-up);

2) when requesting a scheduled connectivity resource a user shall be able to specify the start time, stop time, and frequency;

3) a user shall be informed of the result of the scheduling request;

4) resources which have been reserved from a scheduled request shall be available for use by other requests (both immediate and scheduled) prior to their use within the schedule;

5) in the case of a resource which has been reserved for a scheduled set-up becoming un-available prior to the set-up being performed, then the user shall be informed that the schedule request can no longer be met (in the case where other resources can not be substituted for the un-available resources);

6) a user shall be able to de-schedule a previously scheduled connectivity resource;

7) a user shall be able to request the modification (e.g. by the addition or deletion of time slots) of a previously scheduled connectivity resource;

8) the user may request the scheduling of a connectivity resource which uses resources which have not yet been installed.

NOTE: The scheduling of trails has not been implemented in this version of the class library.

7.3.2.5 Trail Set-Up and Release

A user will have the ability to request that a trail be set-up between access points or access groups. The user may specify the routing in terms of the sub-networks or links to be used, or in terms of particular link connections to be used, or may not specify a routing. The user may also specify that the trail is to be separate at some level from another configuration, may specify a particular QoS or a particular method of protection.

A user will be able to request the release of a previously set-up trail.

7.3.2.6 Network connection set-up

A user will have the ability to set up a connection between two termination connection points. The user may specify the routing in terms of the sub-networks or links to be used, or in terms of particular link connections to be used.

7.3.2.6.1 Link configuration

A user will have the ability to add and remove link connections to/from a particular link, or to request more link connections for the link. If a request for more link connections is made then these will be provided by a server layer network.

7.3.2.6.2 Link connection provision

A user will have the ability to request a link connection between two connection points, or to request more link connections for the link. Link connections are provided by a server layer network. It shall be possible to request a link connection some time before it is actually needed. The provider may have the ability to provide the underlying resource for the link connection just in time, but yet make the link connection visible across the management interface so that it is available for assignment in anticipation of the resource being available.

7.3.2.6.3 Tandem connection provision and configuration

For further study.

7.3.2.7 Network connection release

A user will be able to request the release of a previously set-up network connection.

7.3.2.8 Layer network provisioning and characteristic information

A user can request the provisioning or cessation of a layer network if this is supported by the service provider OSF.

7.3.2.9 Access point provisioning

A user is not allowed to request the creation or deletion of an access point. However, when an access point is created or deleted, a notification is sent to the user. The access points shall have a globally unique identifier. This identifier will contain sufficient information to allow the user to relate it to the overall network configuration process.

7.3.2.10 Access group provisioning

A user shall have the ability to create and delete access groups. The case of multiple users is for further study.

7.3.2.11 Access group configuration

A user will have the ability to add and remove access points to/from an access group.

7.3.2.12 Connection point provisioning

When a connection point is created or deleted, a notification is sent to the user. The connection points shall have a globally unique identifier. This identifier will contain sufficient information to allow the user to relate it to the overall network configuration process.

7.3.2.13 Connection point configuration

A user will have the ability to group connection points together. These groups may be associated with particular links leaving a sub-network. This allows an association between connection points and links, before the link connections bundled by the links and terminated by the connection points have been established.

7.3.2.14 Sub-network provisioning

A user will have the ability to request the creation or deletion of Sub-networks. Initially, the user will be presented with a default "view" of sub-networks provided by the provider This view will not necessarily be of the lowest possible level of partitioning, but will be appropriate for the task to be performed by the user. The user can create and delete sub-networks, specifying whether a new sub-network is to be a super- or sub- partition of existing sub-networks. These sub-networks may be overlapping.

7.3.2.15 Link provisioning

A user will have the ability to request that a link be set-up between two sub-networks.

8 Modelling guide to the class library

This clause gives further details of the modelling approach used in the production of the class library specified in the present document.

Terminology

Where possible, clear unambiguous terms are used. Existing terms are referenced to their source. Where new terms are introduced they are defined at first use, and summarized in the glossary.

However, it is not possible to use names which have not been used elsewhere in all cases. For example, ITU-T Recommendation M.3100 [10] uses the terms trail, and connectivity which are also used in the present document with different definitions. In some cases this may cause confusion when interpreting the output of syntax checkers, and the user is advised to be aware of this.

8.1 Guidelines

This class library follows the ETSI modelling guidelines. Additional information on how the managed object classes are composed, following these guidelines, is contained in annex B.

8.2 Mapping to requirements

The mapping of the requirements of clause 7 to the modelling implementation is given in clause B.1. An explanation of the modelling approach is provided together with tables giving a detailed mapping from the resources and management capabilities to the modelling constructs.

8.3 Representation of relationships

Relationships are described in clause B.2. These relationships exist between the information abstractions of the ITU-T Recommendation G.805 [7] resources, as defined by the managed objects. The relationship between these entities is summarized in the Entity-Relationship Diagram of figure B.1.15.

This representation is independent of implementation. For any given application there are a number of ways of implementing a relationship. The class library definitions allow entity relationships to be implemented in a flexible way. Once a relationship is defined in the Entity Relationship diagram it may be implemented in the following ways:

- Inheritance - see the inheritance diagram of figure B.20.
- Pointers - these are contained in conditional packages in the managed object definitions of subclause 9.1.
- Naming - as defined by the name bindings of subclause 9.4, and illustrated in the example naming schema of figures B.17, B.18 and B.19.

NOTE 1: For many relationships both pointer and name binding options are available.

Further details may be found in clause B.2

NOTE 2: This approach is the same as adopted in the ISO General Relationship Model where relationships are defined first, and a binding template produced to show how any given relationship is implemented for a particular application. The use of the General Relationship model will be considered in future versions of the class library.

8.4 Representation of state

There is an issue for the class library in the representation of the state of the network resource.

There are two requirements for the model:

- to be able to express the combined state of the resource, when there are multiple applications using it;
- to be able to represent a subset of the combined states for applications where a restricted number of applications use the resource.

To satisfy these objectives, a Status Condition is defined below. The Status Conditions are the requirements for the states which a user needs to see in the network resources of the provider. For example, if a user wishes to maintain a network resource the Maintenance Status Condition (14) is used. This is actually implemented as a particular combination of base states.

The Status Condition is not a state itself. It is a set of allowed combinations of base states. The base states are: the ISO Operational and Administrative states, the ISO Availability Status, the Assignment state, and the Lifecycle state.

The set of Status Conditions is not prescriptive, nor is it exhaustive: a subset of the Status Conditions may be used by any particular application, and new Status Conditions may be added (with the appropriate mappings) as new requirements emerge.

The behaviour of the resources is defined in terms of the Status Condition, but the GDMO definition is in terms of the base states, and the mapping is given in annex A.

NOTE: All five component states are needed to define the complete range of Status Conditions, but that a subset of the Status Conditions may be defined by using a smaller number of component states.

Further details are given in annex A.

8.5 Message sizes

Some of the relationships in the class library are implemented by unbounded lists. For example, the list of NWCTPs in a subnetwork. Potentially this list could have several hundred entries. This could give rise to a message which is too large for the stack limits on message size. Future issues of the class library will define ACTIONs so that the linked replies of the ACTION reply may prevent the message size limits being exceeded. For the current version, it is recognized that message sizes are a problem but that first implementations will not have extensive number of NWCTPs in a subnetwork, for example, so the issue will not arise in most cases.

8.6 Application notes

While it is the responsibility of technology specific groups to produce Ensembles for particular applications, guidance is given to these groups in the form of application notes to show how the GOM definitions are intended to be used. These may be found in clause C.2

8.7 Modelling of multipoint connections

Two alternative methods of representing multipoint connections are possible. The first follows the principles of ITU-T Recommendation M.3100 [10], and the second follows ITU-T Recommendation I.326 [16].

The first method is defined in annex D, and the second in annex E. The second is still under study.

9 Managed object class library for the network level view

All references to managed object classes refer by definition to sub-classes and allomorphic representations.

The GDMO definition of types is to be used in favour of the ASN.1 definition. For example only four values of the Availability Status are specified in the GDMO syntax while the IMPORTED ASN.1 allows the full range of the ISO attribute definition. Applying this rule, the additional ASN.1 values are not permitted.

9.1 Managed object class definitions

9.1.1 Access group

```
accessGroup MANAGED OBJECT CLASS
    DERIVED FROM    "Recommendation X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        accessGroupPackage PACKAGE
            BEHAVIOUR
            accessGroupBehaviour BEHAVIOUR
                 DEFINED AS "The Access Group object class is a class of managed objects which
                 groups Network Trail Termination Points for management purposes.";;
            ATTRIBUTES
                accessPointList                                          GET,
                accessGroupId                                            GET,
                signalid                                                 GET;;;
    CONDITIONAL PACKAGES
        linkPointerListPackage
            PRESENT IF "topology is modelled using links",
        topologicalGroupPointerPackage
            PRESENT IF "topology is modelled using topological points";
REGISTERED AS {es200653MObjectClass 1};
```

9.1.2 Admin domain

PROFILE NOTE: The systemTitle is used for naming when an instance of this object has to have a globally unique identifier.

```
adminDomain MANAGED OBJECT CLASS
    DERIVED FROM    "Recommendation X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        adminDomainPackage  PACKAGE
            BEHAVIOUR
        adminDomainBehaviour BEHAVIOUR
                DEFINED AS "This managed object represents the domain of resources to support a
                management function.";;;;
```

```
    CONDITIONAL PACKAGES
        adminDomainIdPackage
            PRESENT IF "an instance supports it",
        systemTitlePackage
            PRESENT IF "an instance supports it",
        "Recommendation M.3100 : 1992":userLabelPackage
            PRESENT IF "an instance supports it";
REGISTERED AS {es200653MObjectClass 2};
```

9.1.3 Allocation

PROFILE NOTE: Allocation is a managed object class for the representation of scheduling of the adaptation function of a trail, to provide link connections.

```
allocation MANAGED OBJECT CLASS
    DERIVED FROM "Recommendation X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Recommendation M.3100 : 1992":createDeleteNotificationsPackage,
        "Recommendation M.3100 : 1992":attributeValueChangeNotificationPackage,
        allocationPackage PACKAGE
            BEHAVIOUR
        allocationBehaviour BEHAVIOUR
                DEFINED AS "This MO books parts or all of the free time of its owning MO (Link
                connection or TandemConnection). If the booked time exceeds the live time of its
                owner the creation of the allocation will be rejected. If the creation of this MO
                intersects another allocation instance the creation will be rejected too. While the
                OS will be notified on creation of this MO instance, it will be not notified on
                deletion when it is the consequence of deleting its owner.";;

            ATTRIBUTES
                allocationId                                                    GET,
                clientPtr                                                       GET;;;
    CONDITIONAL PACKAGES
        administrativeStatePackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        assignmentStatePackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        "Recommendation X.721 | ISO/IEC 10165-2 : 1992":availabilityStatusPackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        lifecycleStatePackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        "Recommendation M.3100 : 1992":operationalStatePackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        durationSchedulingPackage
            PRESENT IF "the transport objects are scheduled to start at a specified time and stop
            at either specified time or function continuously",
        dailyBasisSchedulingPackage
            PRESENT IF " the transport objects are to be scheduled on a daily basis",
        weeklyBasisSchedulingPackage
            PRESENT IF " the transport objects are to be scheduled on a weekly basis",
        monthlyBasisSchedulingPackage
            PRESENT IF " the transport objects are to be scheduled on a monthly basis",
        occasionalSchedulingPackage
            PRESENT IF " the transport objects are to be occasionally scheduled";
REGISTERED AS   {es200653MObjectClass 3};
```

9.1.4 Basic layer network domain

```
basicLayerNetworkDomain MANAGED OBJECT CLASS
    DERIVED FROM    layerNetworkDomain;
        CHARACTERIZED BY
        basicTrailHandlerPackage,
        basicLayerNetworkDomainPackage PACKAGE
            BEHAVIOUR
        basicLayerNetworkDomainBehaviour BEHAVIOUR
                DEFINED AS "The Basic LayerNetworkDomain object class is a class of managed objects
                that manages the immediate setup and release of trails. It provides the following
                functionality: 1Immediate trail set-up; 2Trail release.";;;;
    CONDITIONAL PACKAGES
        addRemoveNWTTPsFromAccessGroupPackage
        PRESENT IF "the layer network domain has Access Groups";
REGISTERED AS {es200653MObjectClass 4};
```

9.1.5 Basic sub-network

PROFILE NOTE: The containedNWCTPList, if present, is used to indicate the CTPs which are part of a sub-network, at levels of partitioning other than the lowest level. (At the lowest level of partitioning the sub-networks name the CTPs) This allows higher level abstractions of the lowest level of partitioning to restrict the set of CTPs at that level to a desired sub-set of the lower level CTPs. CTPs from the lowest level of partitioning which are not visible at the boundary of the higher level sub-network may not be contained in the list. The actions to add/remove NWTPs from a NWGTP, add/remove NWCTPs from a Topological Point are defined as conditional packages, as not all Basic Sub-networks will support these classes.

```
basicSubNetwork MANAGED OBJECT CLASS
    DERIVED FROM    subNetwork;
    CHARACTERIZED BY
        basicConnectionPerformerPackage,
        subNetworkIdPackage,
        basicSubNetworkPackage PACKAGE
            BEHAVIOUR
        basicSubNetworkBehaviour BEHAVIOUR
                DEFINED AS "The Sub-network object class is a class of managed objects manages the
                setup and release of Sub-network Connections, under the control of a manager. It
                also manages the assignment of network termination points to Network GTPs.";; ;;
    CONDITIONAL PACKAGES
        activateSubNetworkConnectionPackage
            PRESENT IF "this sub-network supports a two-phase commit set-up process",
        addRemoveNWTPsFromNWGTPPackage
            PRESENT IF "this sub-network can contain NWGTPs",
        addRemoveNWCTPsFromTopologicalPtPackage
            PRESENT IF " this sub-network can contain Topological Points ",
        addDeletePackage
            PRESENT IF "this sub-network supports point to multipoint sub-network connections";
REGISTERED AS {es200653MObjectClass 5};
```

9.1.6 Connectivity

PROFILE NOTE: Status Conditions shall not be unnecessarily duplicated in Connectivity and Network Termination Point. It is expected that Status Condition will usually be present in Connectivity and its subclasses.
The aEndNWTPList will always be non-NULL. The zEndNWTPList is conditional as not all modes of transmission support Z ends.
The Signal Id shall match the Signal Id of the instance representing the network termination point.
This class is not instantiable.

```
connectivity MANAGED OBJECT CLASS
    DERIVED FROM    "Recommendation X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        connectivityPackage PACKAGE
            BEHAVIOUR
```

```
    connectivityBehaviour BEHAVIOUR
            DEFINED AS "The Connectivity object class is a class of managed objects which
            ensures the transfer of information between two or more network termination points.
            The directionality attribute indicates whether transmission is unidirectional or
            bi-directional. The mode attribute indicates the type of transmission, i.e. point
            to point, point to multi-point, multicast, broadcast or conference.

            These are defined as:
            -   point to point: there is one A end and one Z end;
            -   point to multipoint: there is one A end and multiple Z ends, and there is   no
            traffic flow between Z ends;
            -   multicast: there are multiple A ends and multiple Z ends, and there is no
                traffic flow between A ends or between Z ends;
            -   conference: the multiple A ends send traffic to, and receive traffic from,  all
            other A ends, there are no Z ends;
            -   broadcast: where there is one A end and no known Z ends.

            Where required, the designation of the Connectivity object should follow ITU-T
            Recommendation M.1400. The designation is stored in the User Label. The
            aEndNWTPList attribute and zEndNWTPList attribute are independent of the M.1400
            designation. For point to point unidirectional and bi-directional, the aEndNWTPList
            attribute shall identify a single A end network termination point, and the
            zEndNWTPList shall identify a single Z end network termination point. The
            zEndNWTPList attribute is required to support this case. For point to point
            unidirectional, the aEndNWTPList attribute shall identify the source end and the
            zEndNWTPList attribute shall identify the sink end. For point to multipoint
            unidirectional and bi-directional, the aEndNWTPList attribute shall identify a
            single A end network termination point, and the zEndNWTPList shall identify the Z
            end network termination points. The zEndNWTPList attribute is required to support
            this case. For multicast unidirectional and bi-directional, the aEndNWTPList
            attribute shall identify the A end network termination points, and the zEndNWTPList
            shall identify the Z end network termination points. The zEndNWTPList attribute is
            required to support this case. For broadcast unidirectional and bi-directional, the
            aEndNWTPList attribute shall identify a single A end network termination point.
            There are no known Z ends, so the zEndNWTPList attribute is not required to support
            this case. For conference, only bi-directional transmission is supported. The
            aEndNWTPList attribute shall identify the A end network termination points. There
            are no Z ends, so the zEndNWTPList attribute is not required to support this case.
            The Signal Id attribute describes the signal that is transferred across a
            Connectivity instance. The managed objects representing the network termination
            points, or NWGTPs, that are related by this instance shall have signal Ids that are
            compatible. The default value for the directionality attribute is bi-
            directional.";;

        ATTRIBUTES
            signalid                                                    GET,
            mode                                                        GET,
            aEndNWTPList                                                GET,
            "Recommendation M.3100 : 1992": directionality              GET;;;
CONDITIONAL PACKAGES
    "Recommendation M.3100 : 1992":createDeleteNotificationsPackage
        PRESENT IF "the objectCreation and objectDeletion notifications defined in
        Recommendation X.721 are supported by an instance of this managed object class",
    "Recommendation M.3100 : 1992":attributeValueChangeNotificationPackage
        PRESENT IF "the attributeValueChange notification defined in Recommendation X.721 is
        supported by an instance of this managed object class",
    "Recommendation M.3100 : 1992":stateChangeNotificationPackage
        PRESENT IF "the stateChange notification defined in Recommendation X.721 is supported
        by an instance of this managed object class",
    administrativeStatePackage
        PRESENT IF "The Status Condition described in the behaviour of this managed object
        class is composed using this state, as defined in annex A",
    assignmentStatePackage
        PRESENT IF "The Status Condition described in the behaviour of this managed object
        class is composed using this state, as defined in annex A",
    "Recommendation X.721 | ISO/IEC 10165-2 : 1992":availabilityStatusPackage
        PRESENT IF "The Status Condition described in the behaviour of this managed object
        class is composed using this state, as defined in annex A",
    lifecycleStatePackage
        PRESENT IF "The Status Condition described in the behaviour of this managed object
        class is composed using this state, as defined in annex A",
    "Recommendation M.3100 : 1992":operationalStatePackage
        PRESENT IF "The Status Condition described in the behaviour of this managed object
        class is composed using this state, as defined in annex A",
    "Recommendation M.3100 : 1992":tmnCommunicationsAlarmInformationPackage
        PRESENT IF "the communicationsAlarm notification (as defined in Recommendation X.721)
        is supported by this managed object",
    "Recommendation M.3100 : 1992":alarmSeverityAssignmentPointerPackage
        PRESENT IF "the communicationsAlarmInformationPkg package is present AND the managed
        object supports configuration of alarm severities",
    supportedByPackage
        PRESENT IF "an instance supports it",
```

```
        "Recommendation M.3100 : 1992":userLabelPackage
            PRESENT IF "an instance supports it",
        qualityOfConnectivityServicePackage
            PRESENT IF "an instance supports it",
        zEndNWTPListPackage
            PRESENT IF "an instance supports it";
REGISTERED AS {es200653MObjectClass 6};
```

9.1.7 Degenerate sub-network

```
degenerateSubNetwork MANAGED OBJECT CLASS
    DERIVED FROM subNetwork;
    CHARACTERIZED BY
    subNetworkIdPackage,
    degenerateSubNetworkPackage PACKAGE
            BEHAVIOUR
        degenerateSubNetworkBehaviour BEHAVIOUR
                DEFINED AS "This managed object represents sub-networks where it is not possible to
                flexibly assign Sub-network Connections.";;
;;
REGISTERED AS {es200653MObjectClass 7};
```

9.1.8 Instantiable basic connection performer

```
instantiableBasicConnectionPerformer MANAGED OBJECT CLASS
    DERIVED FROM "Recommendation X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY basicConnectionPerformerPackage,
        instantiableBasicConnectionPerformerPackage PACKAGE
            BEHAVIOUR
        instantiableBasicConnectionPerformerBehaviour BEHAVIOUR
                DEFINED AS "This object is used in the composition of the management capabilities
                of a sub-network";;
            ATTRIBUTES
                instantiableBasicConnectionPerformerId                      GET;;;
    CONDITIONAL PACKAGES
        activateSubNetworkConnectionPackage
            PRESENT IF "this sub-network supports a two-phase commit set-up process",
        addRemoveNWTPsFromNWGTPPackage
            PRESENT IF "this sub-network can contain NWGTPs",
        addRemoveNWCTPsFromTopologicalPtPackage
            PRESENT IF "this sub-network can contain Topological Points",
        addDeletePackage
            PRESENT IF "this sub-network supports point to multipoint sub-network connections";
REGISTERED AS {es200653MObjectClass 8};
```

9.1.9 Instantiable basic trail handler

```
instantiableBasicTrailHandler MANAGED OBJECT CLASS
    DERIVED FROM "Recommendation X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY basicTrailHandlerPackage,
        instantiableBasicTrailHandlerPackage PACKAGE
            BEHAVIOUR
        instantiableBasicTrailHandlerBehaviour BEHAVIOUR
                DEFINED AS "This object is used in the composition of the management capabilities
                of a layer network domain";;
            ATTRIBUTES
                basicTrailHandlerId                                         GET;;;
REGISTERED AS {es200653MObjectClass 9};
```

9.1.10 Layer network domain

PROFILE NOTE: A layer, or transport network layer: A layer, or transport network layer, is defined as ITU-T Recommendation G.805 [7] a topological component solely concerned with the generation and transfer of characteristic information.
The layer network may be characterized by the signal Id package or alternatively the layer network domain may be sub-classed for each characteristic information value.

```
layerNetworkDomain MANAGED OBJECT CLASS
    DERIVED FROM adminDomain;
    CHARACTERIZED BY
        layerNetworkDomainPkg   PACKAGE
            BEHAVIOUR
        layerNetworkDomainBehaviour BEHAVIOUR
```

```
                DEFINED AS "This managed object represents the part of the transport network layer
                which is managed by a management system. It represents the topological and
                connectivity aspects of the part transport network layer.";;;;
    CONDITIONAL PACKAGES
        signalidPackage
            PRESENT IF "an instance supports it";
REGISTERED AS {es200653MObjectClass 10};
```

9.1.11 Leg

PROFILE NOTE: This managed object represents a leg (branch) of a multipoint Sub-network Connection. (see annex D). This class is not used for multipoint implementations following ITU-T Recommendation I.326 [16] (see annex E).

```
leg MANAGED OBJECT CLASS
    DERIVED FROM
        "Recommendation X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Recommendation M.3100 : 1992":stateChangeNotificationPackage,
        legPackage PACKAGE
            BEHAVIOUR
        legBehaviour BEHAVIOUR
                DEFINED AS "A Leg has a single Z end. A Sub-network Connection of mode point to
                multipoint contains multiple Legs. The Status condition indicates the state of each
                Leg of the Sub-network Connection.";;
            ATTRIBUTES
                zEndNWTP                                                    GET,
                legId                                                       GET;;;
    CONDITIONAL PACKAGES
        administrativeStatePackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        assignmentStatePackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        "Recommendation X.721 | ISO/IEC 10165-2 : 1992":availabilityStatusPackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        lifecycleStatePackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        "Recommendation M.3100 : 1992":operationalStatePackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        "Recommendation M.3100 : 1992":createDeleteNotificationsPackage
            PRESENT IF "the objectCreation and objectDeletion notifications defined in
            Recommendation X.721 are supported by an instance of this managed object class",
    "Recommendation M.3100 : 1992":userLabelPackage
            PRESENT IF "an instance supports it";
REGISTERED AS {es200653MObjectClass 11};
```

9.1.12 Link

PROFILE NOTE: The topology view is represented using either links, access groups, and sub-networks, or by topological points, access groups and sub-networks.

Two types of link have been defined:

- externalLink: where the link spans sub-networks, or a sub-network and an access group, in different admin domains but the same layer domain. An example of this is a link between two administrations (PNOs);
- internalLink: where the link spans sub-networks, or a sub-network and an access group, in the same admin domain and same layer domain.

Note that a link only groups point-to-point link connections. Other groupings are for further study.

```
link MANAGED OBJECT CLASS
    DERIVED FROM "Recommendation X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Recommendation M.3100 : 1992":attributeValueChangeNotificationPackage,
        linkPackage PACKAGE
            BEHAVIOUR
        linkBehaviour BEHAVIOUR
```

```
                DEFINED AS "The Link object class is a class of managed objects which gives a
                topological description of the capacity between two adjacent Sub-networks or a sub-
                network and an Access Group, when NWTTPs lie outside the boundary of the largest
                sub-network.

                The use made of the individual attributes and notifications is detailed below:
                -   available link connections: the number of free Link Connections;
                -   a end point: the Sub-network or access group which terminates one end of    the
                Link;
                -   z end point: the Sub-network or access group which terminates the other     end
                of the Link;
                -   number of link connections: the total number of Link connections;
                -   signal Id: shows the signal Id of the Link Connections that provide the
                    capacity for the Link.
                    A Link shall be provided with capacity by Link connections of the same  signal
                Id;
                -   attribute value change notification: shall be emitted when the values   change
                of the following attributes: availableLink Connections, noOf   LinkConnections. ";;

            ATTRIBUTES
                availableLinkConnections                                    GET,
                aEndPoint                                                GET,
                linkId                                                   GET,
                zEndPoint                                                GET,
                noOfLinkConnections                                         GET,
                signalid                                                 GET;;;

    CONDITIONAL PACKAGES
        externalLinkPackage
            PRESENT IF "the link spans sub-networks, or a sub-network and an access group, in
            different admin domains but the same layer domain ",
        internalLinkPackage
            PRESENT IF "the link spans sub-networks, or a sub-network and an access group, in the
            same admin domain and same layer domain ",
        usageCostPackage
            PRESENT IF "the link has an allocated usage cost ";

REGISTERED AS {es200653MObjectClass 12};
```

9.1.13 Link connection

PROFILE NOTE: Each Link connection or Sub-network Connection in the sequence supporting a Trail may be a point to multipoint which gives rise to a "tree" of Link connections and Sub-network Connections which support the Trail. (see annex D). This mode is not used for multipoint implementations following ITU-T Recommendation I.326 [16] (see annex E).

Several Link connections can be bundled into a higher rate Trail. This higher rate Trail may be used to serve client Link connection(s).

A link connection may be a component of a sub-network connection and of a trail.

A single trail in a server layer may support a point to multi-point link connection in a client layer (see annex D). This mode is not used for multipoint implementations following ITU-T Recommendation I.326 [16] (see annex E).

```
linkConnection MANAGED OBJECT CLASS
    DERIVED FROM    connectivity;
        CHARACTERIZED BY
        linkConnectionPackage PACKAGE
            BEHAVIOUR
        linkConnectionBehaviour BEHAVIOUR
```

```
                DEFINED AS "The LinkConnection object class is a class of managed objects
                responsible for the transparent transfer of information between Network Connection
                Termination Points. A LinkConnection may be a component of a Trail. A sequence of
                one or more LinkConnections (and sub-network connections)may be linked together to
                form a Trail. A LinkConnection may be either uni- or bi-directional. A point to
                point unidirectional LinkConnection can be established between a Network CTP source
                or Network CTP bid; and a Network CTP sink or Network CTP bid. A point to point bi-
                directional LinkConnection can be established between a Network CTP bid; and a
                Network CTP bid. A point to multipoint unidirectional LinkConnection can be
                established between a Network CTP source or Network CTP bid; and a set whose
                members are Network CTP sinks or Network CTP bids. A point to multipoint bi-
                directional LinkConnection can be established between a Network CTP bid; and a set
                of Network CTP bids. A multicast unidirectional LinkConnection can be established
                between a set whose members are Network CTP sources or Network CTP bids; and a set
                whose members are Network CTP sinks or Network CTP bids. A multicast bi-directional
                LinkConnection can be established between a set of Network CTP bids; and a set of
                Network CTP bids. A broadcast unidirectional LinkConnection can be established from
                a Network CTP source or Network CTP bid. There are no known Z End terminations, so
                the zEndNWTPList attribute is not required to support this case. A broadcast bi-
                directional LinkConnection can be established from a Network CTP bid. There are no
                known Z End terminations, so the zEndNWTPList attribute is not required to support
                this case. A conference LinkConnection may only be bi-directional. It can be
                established between a set of Network CTP bids. There are no Z End terminations, so
                the zEndNWTPList attribute is not required to support this case. For all types of
                LinkConnection, the network termination point(s) pointed to by the A End attribute
                is related to the network termination point(s) pointed to by the Z End attribute in
                such a way that traffic can flow between the network termination points represented
                by these managed objects in a unidirectional or bi-directional manner as indicated
                by the directionality attribute. The following Status conditions are not valid for
                LinkConnection: In Service with spare capacity, Resource Failed with spare
                capacity, Shutting down with spare capacity.";;

            ATTRIBUTES
                "Recommendation M.3100 : 1992":connectionId                GET;;;
    CONDITIONAL PACKAGES
        serverTrailPackage
            PRESENT IF "an instance supports it",
        compositePointerPackage
            PRESENT IF "required to indicate a relationship from a link connection to a sub-
    network connection where the link connection is a component of that subnetwork
    connection",
        layerTrailPackage
            PRESENT IF "an instance supports it";
REGISTERED AS {es200653MObjectClass 13};
```

9.1.14 Network CTP bi-directional

```
networkCTPBidirectional MANAGED OBJECT CLASS
    DERIVED FROM
        networkCTPSink,
        networkCTPSource;
REGISTERED AS {es200653MObjectClass 14};
```

9.1.15 Network CTP sink

```
networkCTPSink MANAGED OBJECT CLASS
    DERIVED FROM
        networkTP;
    CHARACTERIZED BY
        networkCTPSinkPackage PACKAGE
            BEHAVIOUR
        networkCTPSinkBehaviour BEHAVIOURDEFINED AS "The Network CTP Sink object class is a class of
managed objects that terminates Link connections and/or originates Sub-network Connections. The
resource receivesinformation (traffic), via a Link connection, from an instance representing a
NetworkConnection Termination Point, and sends it on, via a Sub- network Connection, to instances
representing either NWCTP Sources or a NWTTP Sink in the same Sub-network.

                An instance of this class may only have connectivity relationships (link connection
                orsubnetworkconnection) with instances which represent Network Connection
                TerminationPoints, Source or Bi-directional, which are at the same layer. It may
                only besubnetworkconnected, via a Sub-network Connection, to instances representing
                multipleNWCTPs when it operates in broadcast mode i.e. the complete signal goes to
                eachand every downstream NWCTP.
```

```
                An instance of this class may be subnetworkconnected, via a Sub-network
                Connection,to a single instance which represents a Network Trail Termination Point,
                Sink orBi-directional, at the same layer. An instance of this class may not operate
                inbroadcast mode to a NWTTP. The Sub-network Connection Pointer attribute points to
                the managed object representing the relationship with the network termination
                point(s), within the same Sub-network, that receive(s) information (traffic) from
                this network termination point, or is null. The referenced managed object shall
                represent a Sub-network Connection. Where the NWCTP sink participates in many
                subnetwork connections for different subnetworks, the Sub-network Connection
                Pointer is null. Any network termination points identified by the related Sub-
                network Connection indicate that a relationship exists, but this does not indicate
                that information can flow between the network termination points. This capability
                is given by the Status.

                The Connectivity Pointer attribute points to the managed object representing the
                Connection which relates this instance to the instance representing the Network
                Connection Termination Point, Source or Bi-directional, that sends information
                (traffic) to this network termination point, or is null. The following Status
                conditions are not valid for NWCTPsink : In Service with spare capacity, Resource
                Failed with spare capacity, Shutting down with spare capacity.";;;;
    CONDITIONAL PACKAGES
        "Recommendation M.3100 : 1992":channelNumberPackage
            PRESENT IF " an instance supports it",
        "Recommendation M.3100 : 1992":ctpInstancePackage
            PRESENT IF "an instance supports it",
        networkCTPPackage
            PRESENT IF "an instance supports it",
        serverTTPPointerPackage
            PRESENT IF "an instance supports it";
REGISTERED AS {es200653MObjectClass 15};
```

9.1.16 Network CTP source

```
networkCTPSource MANAGED OBJECT CLASS
    DERIVED FROM
        networkTP;
    CHARACTERIZED BY
        networkCTPSourcePackage PACKAGE
            BEHAVIOUR
        networkCTPSourceBehaviour BEHAVIOUR
                DEFINED AS "The Network CTP Source object class is a class of managed objects that
                originates Link connections and/or terminates Sub-network Connections. The resource
                sends information (traffic), via a Link connection, to instances representing
                Network Connection Termination Points, and receives it, via a Sub-network
                Connection, from an instance representing either a NWCTP Sink or a NWTTP Source in
                the same Sub-network.

                An instance of this class may only have connectivity relationships (link connection
                or subnetworkconnection) with instances which represent Network Connection
                Termination Points, Sink or Bi-directional, which are at the same layer. It may
                only be connected, via a Link connection, to instances representing multiple NWCTPs
                when it operates in broadcast mode i.e. the complete signal goes to each and every
                Z end NWCTP. An instance of this class may be subnetworkconnected, via a Sub-
                network Connection, to a single instance which represents a Network Trail
                Termination Point, Source or Bi-directional, at the same layer.

                An instance of this class may not operate in broadcast mode to a NWTTP. The Sub-
                network Connection Pointer attribute points to the managed object representing the
                relationship with the network termination point, within the same Sub-network, that
                sends information (traffic) to this network termination point, or is null. The
                referenced managed object shall represent a Sub-network Connection. Where the NWCTP
                source participates in many subnetwork connections for different subnetworks, the
                Sub-network Connection Pointer is null.  Any network termination point identified
                by the related Sub-network Connection indicates that a relationship exists, but
                this does not indicate that information can flow between the network termination
                points. This capability is given in the admin state.

                The Connectivity Pointer attribute points to the managed object representing the
                Link connection which relates this instance to the instances representing the
                NetworkConnection Termination Point(s), Sink or Bi-directional, that receive
                information(traffic) from this network termination point at the same layer, or is
                null. Thereferenced managed object shall represent a Link connection. The following
                Status conditions are not valid for NWCTPsource : In Service with spare capacity,
                Resource Failed with spare capacity, Shutting down with spare capacity.";;;;
    CONDITIONAL PACKAGES
        "Recommendation M.3100 : 1992":channelNumberPackage
            PRESENT IF "an instance supports it",
        "Recommendation M.3100 : 1992":ctpInstancePackage
            PRESENT IF "an instance supports it",
        networkCTPPackage
            PRESENT IF "an instance supports it",
        serverTTPPointerPackage
```

```
            PRESENT IF "an instance supports it";
REGISTERED AS {es200653MObjectClass 16};
```

9.1.17 Network GTP

PROFILE NOTE: The use of the NWGTP is described in annex B.

```
networkGTP MANAGED OBJECT CLASS
    DERIVED FROM    "Recommendation X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Recommendation M.3100 : 1992":objectManagementNotificationsPackage,
        sncPointerPackage,
        networkGTPPackage   PACKAGE
            BEHAVIOUR
        networkGTPPackageBehaviour BEHAVIOUR
                DEFINED AS "This object class represents a group of network termination points
                treated as a single unit to terminate Sub-network Connections. When NWGTPs are used
                to relate one group of NWCTPs with another group of NWCTPs (with the same number of
                members) in the same Sub-network, there shall be the same number of NWCTPs in each
                group. The nth member of one group is related to the nth member of the other group.

                All the NWCTPs shall be in the same layer. The same rule applies when a group of
                NWTTPs are connected to a group of NWCTPs, where all the members of both groups are
                at the same layer.

                The instances which comprise the members of the Network Group Termination Point
                shall all be either Network Trail Termination Points, or Network Connection
                Termination Points, and shall all be capable of operating in the same direction.
                Valid combinations within the same Network Group Termination Point are:
                -   network connection termination point;
                -   sink/bi-directional;
                -   network trail termination point;
                -   sink/bi-directional;
                -   network connection termination point
                -   source/bi-directional;
                -   network trail termination point
                -   source/bi-directional;
                -   network connection termination point
                -   bi-directional only; and
                -   network trail termination point
                -   bi-directional only.

                The signal Id attribute describes the composition of the NWGTP. For NWGTPs with n
                members, each with the same signal Id, S, the signal Id for the NWGTP shall be
                taken to be a bundle of n times S. The network termination points listed in the
                tpsInNWGTPList attribute shall not be connected independently of the NWGTP.";;

            ATTRIBUTES
                "Recommendation M.3100 : 1992":gtpId                        GET,
                signalid                                                    GET,
                "Recommendation M.3100 : 1992":tpsInGtpList                 GET
;;;
    CONDITIONAL PACKAGES
        "Recommendation M.3100 : 1992":userLabelPackage
            PRESENT IF "an instance supports it";
REGISTERED AS {es200653MObjectClass 17};
```

9.1.18 Network TP

PROFILE NOTE: Status Condition shall be present in either Connectivity or Network Termination Point. It is expected that Status Conditions will usually be present in Connectivity and its subclasses. This class (but not its subclasses) is not instantiable.
Conditions for generation of state and attribute value change notifications are detailed in the subclasses.

```
networkTP MANAGED OBJECT CLASS
    DERIVED FROM
        "Recommendation X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Recommendation M.3100 : 1992":createDeleteNotificationsPackage,
        networkTPPackage PACKAGE
            BEHAVIOUR
        networkTPBehaviour BEHAVIOUR
```

```
                DEFINED AS "This managed object represents the termination of a transport entity,
                such as an instance representing a Trail or a Link connection. The sncPointer is
                used to point to a Sub-network Connection. However, not all network termination
                points will have a flexible connection, and it may be more appropriate to point to
                another network termination point, for example in a regenerator the two NWCTPs
                would point to each other as there is no flexibility between them. In this instance
                the networkTPPointer shall be used. Both pointers are conditional. The Connectivity
                Pointer attribute points to the managed object representing the Link connection
                which relates this instance to other instance(s) representing the Network
                Termination Point(s). The mode attribute may take on the following values: point to
                point, point to multipoint, multicast, conference,and broadcast. This is used for
                representation of modes following ITU-T Recommendation I.326. The default value for
                this attribute is point to point. ";;

            ATTRIBUTES
                mode                                                    GET,
                signalid                                                GET
;;;
    CONDITIONAL PACKAGES
        connectivityPointerPackage
            PRESENT IF "an instance supports it",
        neAssignmentPackage
            PRESENT IF "an instance supports it",
        "Recommendation M.3100 : 1992":tmnCommunicationsAlarmInformationPackage
            PRESENT IF "the communicationsAlarm notification (as defined in Recommendation X.721)
            is supported by this managed object",
        sncPointerPackage
            PRESENT IF "a NWTP may be flexibly connected to another NWTP",
        networkTPPointerPackage
            PRESENT IF "when there is no flexibility between NWTPs",
        "Recommendation M.3100 : 1992":attributeValueChangeNotificationPackage
            PRESENT IF "an instance supports it",
        "Recommendation M.3100 : 1992":userLabelPackage
            PRESENT IF "an instance supports it",
        administrativeStatePackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        assignmentStatePackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        "Recommendation X.721 | ISO/IEC 10165-2 : 1992":availabilityStatusPackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        lifecycleStatePackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        "Recommendation M.3100 : 1992":operationalStatePackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        "Recommendation M.3100 : 1992":stateChangeNotificationPackage
            PRESENT IF "an instance supports it",
        supportedByPackage
            PRESENT IF "an instance supports it";
REGISTERED AS {es200653MObjectClass 18};
```

9.1.19 Network TTP bi-directional

```
networkTTPBidirectional MANAGED OBJECT CLASS
    DERIVED FROM    networkTTPSink,
                    networkTTPSource;
REGISTERED AS {es200653MObjectClass 19};
```

9.1.20 Network TTP sink

```
networkTTPSink MANAGED OBJECT CLASS
    DERIVED FROM
        networkTP;
    CHARACTERIZED BY
        networkTTPSinkPackage PACKAGE
            BEHAVIOUR
        networkTTPSinkBehaviour BEHAVIOUR
                DEFINED AS "The Network TTP Sink object class is a class of managed objects that
                terminates Trails and Sub-network Connections in the Network viewpoint. An instance
                of this class may only have Trail relationships with Network Trail Termination
                Points, Source or Bidirectional, which are at the same layer.
```

```
                An instance of this class may be subnetworkconnected, via a Sub-network Connection,
                to a single Network Connection Termination Point Sink or Bidirectional, or a
                Network Trail Termination Point Source at the same layer. The Sub-network
                Connection Pointer attribute points to the managed object representing the
                relationship with one or more Network Connection Termination Points, within the
                same Sub-network, that send information (traffic) to this network termination
                point, or is null.

                Any network termination point identified by the related Sub-network Connection
                indicates that a relationship exists, but this does not indicate that information
                can flow between the network termination points. This capability is given in the
                state information.

                The Connectivity Pointer attribute points to the managed object representing the
                Trail which relates this instance to the instances representing the Network Trail
                Termination Points, that send information (traffic) to this network termination
                point at the same layer, or is null.";;;;

    CONDITIONAL PACKAGES
        "Recommendation M.3100 : 1992":supportableClientListPackage
            PRESENT IF "an instance supports it",
        "Recommendation M.3100 : 1992":ttpInstancePackage
            PRESENT IF "an instance supports it",
        clientCTPListPackage
            PRESENT IF "an instance supports it";
REGISTERED AS {es200653MObjectClass 20};
```

9.1.21 Network TTP source

```
networkTTPSource MANAGED OBJECT CLASS
    DERIVED FROM
        networkTP;
    CHARACTERIZED BY
        networkTTPSourcePackage PACKAGE
            BEHAVIOUR
        networkTTPSourceBehaviour BEHAVIOUR
                DEFINED AS "The Network TTP Source object class is a class of managed objects that
                originates Trails and Sub-network Connections in the Network viewpoint. An instance
                of this class may only have Trail relationships with Network Trail Termination
                Points, Sink or Bidirectional, which are at the same layer. An instance of this
                class may be subnetworkconnected, via a Sub-network Connection, to a single Network
                Connection Termination Point Source or Bidirectional, or a Network Trail
                Termination Point Sink at the same layer. It may also be connected, via a Sub-
                network Connection, to multiple instances of Network CTPs at the same layer when it
                is operating in the broadcast mode in order to transmit multiple copies of the same
                signal.

                The Sub-network Connection Pointer attribute points to the managed object
                representing the relationship with one or more Network Connection Termination
                Points, within the same Sub-network, that receive information (traffic) from this
                network termination point, or is null.

                Any network termination point identified by the related Sub-network Connection
                indicates that a relationship exists, but this does not indicate that information
                can flow between the network termination points. This capability is given in the
                Status.

                The Connectivity Pointer attribute points to the managed object representing the
                Trail which relates this instance to the instances representing the Network Trail
                Termination Points, that receive information (traffic) from this network
                termination point at the same layer, or is null. ";;;;

    CONDITIONAL PACKAGES
        "Recommendation M.3100 : 1992":supportableClientListPackage
            PRESENT IF "an instance supports it",
        "Recommendation M.3100 : 1992":ttpInstancePackage
            PRESENT IF "an instance supports it",
        clientCTPListPackage
            PRESENT IF "an instance supports it";
REGISTERED AS {es200653MObjectClass 21};
```

9.1.22 Node

```
node MANAGED OBJECT CLASS
    DERIVED FROM adminDomain;
    CHARACTERIZED BY
        adminDomainIdPackage,
        "Recommendation M.3100 : 1992": locationNamePackage,
        "Recommendation M.3100 : 1992": createDeleteNotificationsPackage,
        nodePackage PACKAGE
```

```
            BEHAVIOUR
        nodeBehaviour BEHAVIOUR
                DEFINED AS "The Node object class is a class of managed objects which represents
                logical collections of network termination points in a single geographical
                location.

                The Network Termination Points grouped together by node may be from different
                layers, and have different values of characteristic information. The Signal List
                attribute, if it is not NULL, indicates a list of signal types the node is capable
                of supporting.

                The unknown Status is used to indicate that the Manager has lost communications
                with the node and therefore the Status Condition of the related objects(for example
                termination points)may not be valid. The typeText attribute specifies the
                particular type of node. ";;

            ATTRIBUTES
                signalList                                                  GET
;;;
    CONDITIONAL PACKAGES
        "Recommendation M.3100 : 1992": attributeValueChangeNotificationPackage
            PRESENT IF "notification of changes in the signalList attribute are required",
        supportedByPackage
            PRESENT IF "an instance supports it",
        unknownStatusPackage
            PRESENT IF "an instance supports it",
        typeTextPackage
            PRESENT IF "an instance supports it";
REGISTERED AS {es200653MObjectClass 22};
```

9.1.23 Sub-network

PROFILE NOTE: The Sub-network object class represents the sub-network resource. It is not possible in all cases for subnetworks to be created and deleted by management action. In these cases the createDeleteNotificationsPackage will not be used.

```
subNetwork MANAGED OBJECT CLASS
    DERIVED FROM "Recommendation X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Recommendation M.3100 : 1992":createDeleteNotificationsPackage,
        subNetworkPackage PACKAGE
            BEHAVIOUR   subNetworkBehaviour BEHAVIOUR
                DEFINED AS " The Sub-network object class is a class of managed objects which
                represents logical collections of network termination points. The attribute
                ContainedSubNetworkList will be null if there are no contained Sub-networks. The
                attribute ContainedInSubNetworkList will also be null if there are no containing
                (parent) Sub-networks.";;;;

    CONDITIONAL PACKAGES
        "Recommendation M.3100 : 1992": stateChangeNotificationPackage
            PRESENT IF "an instance supports it",
        "Recommendation M.3100 : 1992":attributeValueChangeNotificationPackage
            PRESENT IF "an instance supports it",
        signalidPackage
            PRESENT IF "an instance supports it",
        "Recommendation M.3100 : 1992":userLabelPackage
            PRESENT IF "an instance supports it",
        subNetworkIdPackage
            PRESENT IF "an instance supports it",
        administrativeStatePackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        assignmentStatePackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        "Recommendation X.721 | ISO/IEC 10165-2 : 1992":availabilityStatusPackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        lifecycleStatePackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        "Recommendation M.3100 : 1992":operationalStatePackage
            PRESENT IF "The Status Condition described in the behaviour of this managed object
            class is composed using this state, as defined in annex A",
        supportedByPackage
            PRESENT IF "an instance supports it",
        containedNWCTPListPackage
            PRESENT IF "an instance supports it",
        containedNWTTPListPackage
            PRESENT IF "an instance supports it",
```

```
        containedLinkListPackage
            PRESENT IF "an instance supports it",
        containedSubNetworkListPackage
            PRESENT IF "an instance supports it ",
        containedInSubNetworkListPackage
            PRESENT IF "an instance supports it",
        linkPointerListPackage
            PRESENT IF "a topological view using links, sub-networks, and access groups is
            supported";
REGISTERED AS {es200653MObjectClass 23};
```

9.1.24 Sub-network connection

PROFILE NOTE: The Sub-network Connection object class is a class of managed objects that associates, across a subnetwork, the Network CTP(s), Network TTP(s), or Network GTP(s) object(s) identified in the A end attribute and the Network CTP(s), Network TTP(s), or Network GTP(s) object(s) listed in the Z end attribute of this managed object. The user label package should be made mandatory to assist in retrieving scheduling information when this is required.

Point-to-point and point-to-multipoint subnetwork connections may be set up as described in annex D. Multicast subnetwork connections are also defined in annex D.

To support point-to-multipoint following ITU-T Recommendation I.326 [16], only point-to-point subnetwork connections are used (see annex E).

```
subNetworkConnection MANAGED OBJECT CLASS
    DERIVED FROM    connectivity;
    CHARACTERIZED BY
        subNetworkConnectionPackage PACKAGE
            BEHAVIOUR
        subNetworkConnectionBehaviour BEHAVIOUR
                DEFINED AS "The Sub-network Connection object class is a class of managed objects
                that associates the Network CTP(s), Network TTP(s), or Network GTP(s) object
                identified in the A end attribute and the Network CTP(s), Network TTP(s), or
                Network GTP(s) object(s) listed in the Z end attribute of this managed object. The
                Sub-network Connection may be set up between network termination points (or NWGTPs)
                specified explicitly, or between Topological Points or Access Groups from which any
                idle network termination point or NWGTP may be used. If the managed objects listed
                in the A End and Z End attributes represent Network GTPs, the nth element of the A
                end NWGTP is related to the nth element of every Z end NWGTP (for every n).

                There shall be n elements in each NWGTP involved in the Sub-network Connection. For
                a NWGTP with n elements, the Signal Id shall be taken to be a bundle of n times the
                characteristic information of the individual elements, all of which are the same. A
                point to point unidirectional Sub-network Connection can be established between one
                of Network CTP sink, Network CTP bid, Network TTP source, Network TTP bid or
                Network GTP; and one of Network CTP source, Network CTP bid, Network TTP sink,
                Network TTP bid or Network GTP. A point to point bi-directional Sub-network
                Connection can be established between one of Network CTP bid, Network TTP bid or
                Network GTP; and one of Network CTP bid, Network TTP bid or Network GTP. A point to
                multipoint unidirectional Sub-network Connection can be established between one of
                Network CTP sink, Network CTP bid, Network TTP source, Network TTP bid or Network
                GTP; and a set whose members are Network CTP sources, Network CTP bids, Network TTP
                sinks, Network TTP bids or Network GTPs.

                A point to multipoint bi-directional Sub-network Connection can be established
                between one of Network CTP bid, Network TTP bid or Network GTP; and a set whose
                members are Network CTP bids, Network TTP bids or Network GTPs. For all types of
                Sub-network Connection, the network termination point(s) or NWGTP object(s) pointed
                to by the A End attribute is related to the network termination point(s) or NWGTP
                object(s) pointed to by the Z End attribute in such a way that traffic can flow
                between the network termination points represented by these managed objects in a
                unidirectional or bi-directional manner as indicated by the directionality
                attribute. A sub-network connection may be established in any of the following
                Status Conditions:

                -   planned (1);
                -   in service, reserved (4);
                -   in service with no spare capacity (8);
                -   in service with no spare capacity, under test (9).

                Status Condition (4) is the default. Other Status Conditions shall be explicitly
                expressed in set-up sub-network connection ACTION.

                The compositePointerPackage is supported where the Sub-network Connection is a
                component of another Sub-network Connection within the same layer.
```

```
                The componentListPackage is supported where the Sub-network Connection is made up
                of a number of component Sub-network Connections, and Connections, within the same
                layer.";;

            ATTRIBUTES
                subNetworkConnectionId                                          GET;;;
    CONDITIONAL PACKAGES
        compositePointerPackage
            PRESENT IF "an instance supports it.",
        componentPointerPackage
            PRESENT IF "an instance supports it.",
        "Recommendation M.3100 : 1992":userLabelPackage
            PRESENT IF "an instance supports it.",
        durationSchedulingPackage
            PRESENT IF "The sub network connection is to be immediately set up",
        dailyBasisSchedulingPackage
            PRESENT IF "The sub network connection is to be scheduled on a daily basis",
        weeklyBasisSchedulingPackage
            PRESENT IF "The sub network connection is to be scheduled on a weekly basis",
        monthlyBasisSchedulingPackage
            PRESENT IF "The sub network connection is to be scheduled on a monthly basis",
        occasionalSchedulingPackage
            PRESENT IF "The sub network connection is to be occasionally scheduled";
REGISTERED AS {es200653MObjectClass 24};
```

9.1.25 Sub-network pair

```
subNetworkPair MANAGED OBJECT CLASS
    DERIVED FROM adminDomain;
    CHARACTERIZED BY
        subNetworkPairPackage PACKAGE
            BEHAVIOUR
        subNetworkPairBehaviour BEHAVIOUR
                DEFINED AS "This managed object represents a collection of Trail objects
                originating and terminating in a given pair of Sub-networks.";;

            ATTRIBUTES
                aEndPoint                                                       GET,
                zEndPoint                                                       GET,
                trailList                                                       GET,
                subNetworkPairId                                                GET,
                signalid                                                        GET
;;;
REGISTERED AS {es200653MObjectClass 25};
```

9.1.26 Topological point

PROFILE NOTE: This managed object class is used if a topology view using topological points, sub-networks, and access groups is supported.

```
topologicalPoint MANAGED OBJECT CLASS
    DERIVED FROM    "Recommendation X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        "Recommendation M.3100 : 1992":createDeleteNotificationsPackage,
        topologicalGroupPointerPackage,
        topologicalPointPackage PACKAGE
            BEHAVIOUR
                topologicalPointBehavior BEHAVIOUR
                 DEFINED AS "The Topological Point object class is a class of managed objects which
                 contains Network Connection Termination Points for the purpose of representing
                 topology.";;

            ATTRIBUTES
                signalid                                                        GET,
                nWCTPsInTopologicalPointList                                    GET,
                totalNWCTPCount                                                 GET,
                connectedNWCTPCount                                             GET,
                idleNWCTPCount                                                  GET,
                topologicalPointId                                              GET;;;
    CONDITIONAL PACKAGES
        "Recommendation M.3100 : 1992":userLabelPackage
            PRESENT IF "an instance supports it";
REGISTERED AS {es200653MObjectClass 26};
```

9.1.27 Trail

```
trail MANAGED OBJECT CLASS
    DERIVED FROM    connectivity;
    CHARACTERIZED BY
        trailPackage PACKAGE
            BEHAVIOUR
        trailBehaviour BEHAVIOUR

                DEFINED AS "Trail is a class of managed objects in layer networks which
                isresponsible for the integrity of transfer of characteristic information from one
                or more other layer networks. A Trail is composed of two or more Network Trail
                Termination Points and one or more Link connection or Sub-network Connections, and
                associated Network Connection Termination Points.

                A point to point unidirectional Trail can be established between a Network TTP
                source or Network TTP bid; and a Network TTP sink or Network TTP bid.

                A point to point bi-directional Trail can be established between a Network TTP bid;
                and a Network TTP bid.

                A point to multipoint unidirectional Trail can be established between a Network TTP
                source or Network TTP bid; and a set whose members are Network TTP sinks or Network
                TTP bids.

                A point to multipoint bi-directional Trail can be established between a Network TTP
                bid; and a set of Network TTP bids.

                A multicast unidirectional Trail can be established between a set whose members are
                Network TTP sources or Network TTP bids; and a set whose members are Network TTP
                sinks or Network TTP bids.

                A multicast bi-directional Trail can be established between a set of Network TTP
                bids; and a set of Network TTP bids.

                A broadcast unidirectional Trail can be established from a Network TTP source or
                Network TTP bid. There are no known Z End terminations, so the zEndNWTPList
                attribute is not required to support this case.

                A broadcast bi-directional Trail can be established from a Network TTP bid.
                There are no known Z End terminations, so the zEndNWTPList attribute is not
                required to support this case.

                A conference Trail may only be bi-directional. It can be established between a set
                of Network TTP bids. There are no Z End terminations, so the zEndNWTPList attribute
                is not required to support this case.

                For all types of Trail, the termination point(s) pointed to by the A End
                attribute is related to the network termination point(s) pointed to by the Z End
                attribute in such a way that traffic can flow between the network termination
                points represented by these managed objects in a unidirectional or bi-directional
                manner as indicated by the directionality attribute.

                The layerConnectionListPackage lists the subnetwork connections and link
                connections (in the same layer) which compose the trail."
                ;;

            ATTRIBUTES
                "Recommendation M.3100 : 1992":trailId                    GET;;;
    CONDITIONAL PACKAGES
        layerConnectionListPackage
            PRESENT IF "there is a requirement to view the sequence of subnetwork connections and
        link connections which make up the trail in the same layer.",
        clientConnectionListPackage
            PRESENT IF "there is a requirement to view the link connection(s) in a higher layer
    which are supported by this trail.";
REGISTERED AS {es200653MObjectClass 27};
```

9.2 Package definitions

9.2.1 Activate sub-network connection package

```
activateSubNetworkConnectionPackage PACKAGE
    ACTIONS
        activateSubNetworkConnection;
REGISTERED AS {es200653Package 1};
```

9.2.2 Add delete package

```
addDeletePackage    PACKAGE
            BEHAVIOUR
        addDeletePackageBehaviour BEHAVIOUR
                DEFINED AS "The action AddToSubNetworkConnection adds a leg to a Sub-network
                Connection, and DeleteFromSubNetworkConnection deletes a leg from it.";;
    ACTIONS
        addToSubNetworkConnection,
        deleteFromSubNetworkConnection;
REGISTERED AS {es200653Package 2};
```

9.2.3 Add remove NWCTPs from topological Pt package

```
addRemoveNWCTPsFromTopologicalPtPackage PACKAGE
    ACTIONS
        addNWCTPsToTopologicalPt,
        removeNWCTPsFromTopologicalPt;
REGISTERED AS {es200653Package 3};
```

9.2.4 Add remove NWTPs from NWGTP package

```
addRemoveNWTPsFromNWGTPPackage  PACKAGE
    ACTIONS
        addNWTPsToNWGTP,
        removeNWTPsFromNWGTP;
REGISTERED AS {es200653Package 4};
```

9.2.5 Add remove NWTTPs from access group package

```
addRemoveNWTTPsFromAccessGroupPackage   PACKAGE
    ACTIONS
        addNWTTPsToAccessGroup,
        removeNWTTPsFromAccessGroup;
REGISTERED AS {es200653Package 5};
```

9.2.6 Admin Domain Id Package

```
adminDomainIdPackage    PACKAGE
            ATTRIBUTES
                adminDomainId   GET;
REGISTERED AS { es200653Package 51};
```

9.2.7 Administrative state package

```
administrativeStatePackage  PACKAGE
            ATTRIBUTES
                "Recommendation X.721 | ISO/IEC 10165-2 : 1992":administrativeState GET-REPLACE;
REGISTERED AS {es200653Package 6};
```

9.2.8 Assignment state package

```
assignmentStatePackage  PACKAGE
            ATTRIBUTES
                assignmentState                                         GET;
REGISTERED AS {es200653Package 7};
```

9.2.9 Basic connection performer package

```
basicConnectionPerformerPackage PACKAGE
            BEHAVIOUR
        basicConnectionPerformerBehaviour BEHAVIOUR
                DEFINED AS "The Basic Connection Performer object class provides basic connection
                set-up functionality. The action SetupSubNetworkConnection sets up a Sub-network
                Connection, and releaseSubNetworkConnection removes the Sub-network connection .";;

    ACTIONS
        setupSubNetworkConnection,
        releaseSubNetworkConnection
;
REGISTERED AS {es200653Package 8};
```

9.2.10 Basic trail handler package

PROFILE NOTE: Where the trail is setup between accessGroups, the directionality is specified from the ConnectivityDirectionality defined in the setupTrailInformation of the set up trail request.

```
basicTrailHandlerPackage PACKAGE
            BEHAVIOUR
        basicTrailHandlerBehaviour BEHAVIOUR
                DEFINED AS "Immediate trail set-up. When it receives the setupTrail request the
                agent has the responsibility to:

                1)find a route for the trail;
                2)set-up any required sub-network connections;
                3)ensure that the trail object instance has been created with the correct
                initial values.
                4)Inform the service user of the result of its request.

                Trail release:
                When it receives the releaseTrail request the agent has the responsibility
                to:
                1)Release any used sub-network connections;
                2)Update network resource usage (configuration) information;
                3)Inform the service user of the result of its request.";;

    ACTIONS
        setupTrail,
        releaseTrail;
REGISTERED AS {es200653Package 9};
```

9.2.11 Client connection list package

```
clientConnectionListPackage PACKAGE
            ATTRIBUTES
                clientLinkConnectionList                                        GET;
REGISTERED AS {es200653Package 10};
```

9.2.12 Client CTP list package

```
clientCTPListPackage    PACKAGE
            ATTRIBUTES
                clientCTPList                                                GET;
REGISTERED AS {es200653Package 11};
```

9.2.13 Component pointer package

```
componentPointerPackage PACKAGE
            BEHAVIOUR
        componentPointerPackageBehaviour BEHAVIOUR
                DEFINED AS "This package identifies a sequence of instances of Link connection
        and; Sub-network Connection managed objects which are components of a Sub-network
                Connection, within a given layer.";;

            ATTRIBUTES
                componentPointers                                            GET;
REGISTERED AS {es200653Package 12};
```

9.2.14 Composite pointer package

```
compositePointerPackage PACKAGE
            BEHAVIOUR
        compositePointerPackageBehaviour BEHAVIOUR
                DEFINED AS "This package identifies an instance of the Sub-network Connection
                managed object class. Within a given layer, a given subnetwork connection is
                composed of a sequence of link connections and subnetwork connections. This pointer
                points from one these componens to the composite sub-network connection."
                ;;

            ATTRIBUTES
                compositePointer                                         GET;
REGISTERED AS {es200653Package 13};
```

9.2.15 Connectivity pointer package

```
connectivityPointerPackage  PACKAGE
            BEHAVIOUR
connectivityPointerPackageBehaviour BEHAVIOUR
                DEFINED AS "This package identifies an instance of a Link connection or Trail
                managed object class which is terminated by the Network Termination Point."
                ;;

            ATTRIBUTES
                connectivityPointer                                     GET;
REGISTERED AS {es200653Package 14};
```

9.2.16 Contained in sub network list package

```
containedInSubNetworkListPackage    PACKAGE
            ATTRIBUTES
                containedInSubNetworkList               GET-REPLACE ADD-REMOVE;
REGISTERED AS {es200653Package 15};
```

9.2.17 Contained link list package

```
containedLinkListPackage    PACKAGE
            ATTRIBUTES
                containedLinkList                       GET-REPLACE ADD-REMOVE;
REGISTERED AS {es200653Package 16};
```

9.2.18 Contained network CTP list package

```
containedNWCTPListPackage   PACKAGE
            ATTRIBUTES
                containedNWCTPList                      GET-REPLACE ADD-REMOVE;
REGISTERED AS {es200653Package 17};
```

9.2.19A Contained network TTP list package

```
containedNWTTPListPackage   PACKAGE
            ATTRIBUTES
                containedNWTTPList                      GET-REPLACE ADD-REMOVE;
REGISTERED AS {es200653Package 18};
```

9.2.19 Contained sub network list package

```
containedSubNetworkListPackage  PACKAGE
            ATTRIBUTES
                containedSubNetworkList                 GET-REPLACE ADD-REMOVE;
REGISTERED AS {es200653Package 19};
```

9.2.20 Daily basis scheduling package

```
dailyBasisSchedulingPackage PACKAGE
            BEHAVIOUR
        dailyBasisSchedulingPackageBehaviour BEHAVIOUR
                DEFINED AS "This package is instantiated when the setup action which created the
                sub-network connection requests a daily schedule. It contains the attributes
                describing this scheduling and the action which enables any subsequent modification
                of the schedule."
                ;;

            ATTRIBUTES
                reservationBegin                                        GET,
                reservationEnd                                          GET,
                dailySchedule                                           GET;
    ACTIONS
        changeDailyScheduling;
REGISTERED AS {es200653Package 20};
```

9.2.21 Duration scheduling package

```
durationSchedulingPackage PACKAGE
            BEHAVIOUR
        durationSchedulingPackageBehaviour BEHAVIOUR
                DEFINED AS "This package is instantiated when the setup action which entailed the
                creation of the connection request an immediate connection. It contains the
                attributes describing the bandwidth and the action which enables modification of
                the bandwdth."
                ;;

            ATTRIBUTES
                bidirectionalTrafficDescriptor                              GET;
    ACTIONS
        changeDurationScheduling;
REGISTERED AS {es200653Package 21};
```

9.2.22 External link package

```
externalLinkPackage PACKAGE
            BEHAVIOUR
        externalLinkPackageBehaviour BEHAVIOUR
                DEFINED AS "The external link represents the view of a link exported to another
                admin. domain e.g. another operator, and therefore provides a restricted view to
                that of an internal link which exists within a management domain. If the number of
                Link connections in a Link is changed, either as a result of an internal Agent
                operation or a fault, then the relevant attributes shall be changed accordingly."
                ;;
REGISTERED AS {es200653Package 22};
```

9.2.23 Void

9.2.24 Internal link package

```
internalLinkPackage PACKAGE
            BEHAVIOUR
        internalLinkPackageBehaviour BEHAVIOUR
                DEFINED AS "If the number of Link connections in a Link is changed, either as a
                result of a SET operation or a fault, then the relevant attributes shall be changed
                accordingly."
                ;;

            ATTRIBUTES
                linkConnectionList                                  GET-REPLACE ADD-REMOVE;
REGISTERED AS {es200653Package 24};
```

9.2.25 Layer connection list package

```
layerConnectionListPackage  PACKAGE
            ATTRIBUTES
                layerLinkConnectionList                                         GET;
REGISTERED AS {es200653Package 25};
```

9.2.26 Layer trail package

```
layerTrailPackage   PACKAGE
            ATTRIBUTES
                layerTrail                                                  GET;
REGISTERED AS {es200653Package 26};
```

9.2.27 Lifecycle state package

```
lifecycleStatePackage   PACKAGE
            ATTRIBUTES
                lifecycleState                                              GET;
REGISTERED AS {es200653Package 27};
```

9.2.28 Link pointer list package

```
linkPointerListPackage  PACKAGE
            BEHAVIOUR
        linkPointerListPackageBehaviour BEHAVIOUR
                DEFINED AS "This package identifies instances of the link managed object
                class.";;

            ATTRIBUTES
                linkPointerList                                          GET;
REGISTERED AS {es200653Package 28};
```

9.2.29 Monthly basis scheduling package

```
monthlyBasisSchedulingPackage PACKAGE
            BEHAVIOUR
        monthlyBasisSchedulingPackageBehaviour BEHAVIOUR
                DEFINED AS "This package is instantiated when the setup action which created the
                sub-network connection requests a monthly schedule. It contains the attributes
                describing this scheduling and the action which enables any subsequent modification
                of the schedule."
                ;;

            ATTRIBUTES
                reservationBegin                                         GET,
                reservationEnd                                           GET,
                monthlySchedule                                          GET;
    ACTIONS
        changeMonthlyScheduling;
REGISTERED AS {es200653Package 29};
```

9.2.30 NE assignment package

```
neAssignmentPackage PACKAGE
            BEHAVIOUR
        neAssignmentPackageBehaviour BEHAVIOUR
                DEFINED AS "The NE Assignment package provides a pointer from the lowest level
                Network TP in the partitioning hierarchy to a NE TP which represents the
                functionality which supports the Network TP. The sub-partition pointer for a NWCTP
                which utilizes the NE assignment pointer will be NULL."
                ;;

            ATTRIBUTES
                neAssignmentPointer                                      GET;
REGISTERED AS {es200653Package 30};
```

9.2.31 Network CTP package

```
networkCTPPackage   PACKAGE
            BEHAVIOUR
        networkCTPPackageBehaviour BEHAVIOUR
                DEFINED AS "The Network CTP package identifies instances of the Network CTP
                managed object class at higher and lower levels of sub-network partitioning
                (within a given layer) by the use of partitioning pointers. The Super Partition
                pointer is a pointer to a Network CTP which is in a higher level partition. This
                pointer will only be present for the Network CTPs in the lower partition which have
                a direct correspondence to the Network CTPs at the higher level. The higher level
                Network CTPs have an inverse pointer, the sub partition pointer to the lower level.
                Where the lowest level of NWCTP points to a NE CTP via the NE assignment pointer,
                the value of the sub-partition pointer is null."
                ;;

            ATTRIBUTES
                superPartitionPointer                                    GET,
                subPartitionPointer                                      GET;
REGISTERED AS {es200653Package 31};
```

9.2.32 Network TP pointer package

```
networkTPPointerPackage PACKAGE
            BEHAVIOUR
        networkTPPointerPackageBehaviour BEHAVIOUR
                DEFINED AS "This package defines a pointer to an instance of a network
                termination point. Needs further definition."
                ;;
```

```
            ATTRIBUTES
                networkTPPointer                                        GET;
REGISTERED AS {es200653Package 32} ;
```

9.2.33 Occasional scheduling package

```
occasionalSchedulingPackage PACKAGE
            BEHAVIOUR
        occasionalSchedulingPackageBehaviour BEHAVIOUR
                 DEFINED AS "This package is instantiated when the setup action which entailed the
                 creation of the connection convey an occasional schedule. It contains the
                 attributes describing the scheduling and the action which enables to modify the
                 schedule."
                 ;;

            ATTRIBUTES
                reservationBegin                                        GET,
                reservationEnd                                          GET,
                occasionalSchedule                                      GET;
    ACTIONS
        changeOccasionalScheduling;
REGISTERED AS {es200653Package 33} ;
```

9.2.34 Quality of connectivity service package

```
qualityOfConnectivityServicePackage PACKAGE
            ATTRIBUTES
                qualityOfConnectivityService                            GET;
REGISTERED AS {es200653Package 34};
```

9.2.35 Server trail package

```
serverTrailPackage  PACKAGE
            ATTRIBUTES
                serverTrail                                             GET;
REGISTERED AS {es200653Package 35};
```

9.2.36 Server TTP package

```
serverTTPPointerPackage PACKAGE
            ATTRIBUTES
                serverTTPPointer                                        GET;
REGISTERED AS {es200653Package 36};
```

9.2.37 Signal Id package

```
signalidPackage PACKAGE
            ATTRIBUTES
                signalid                                                GET;
REGISTERED AS {es200653Package 37};
```

9.2.38 SNC pointer package

```
sncPointerPackage   PACKAGE
            BEHAVIOUR
        sncPointerPackageBehaviour BEHAVIOUR
                DEFINED AS "This package defines a pointer to instance(s) of the Sub-network
                Connection managed object class, within a given layer.";;

            ATTRIBUTES
                subNetworkConnectionPointer                             GET;
REGISTERED AS {es200653Package 38};
```

9.2.39 Sub-network Id package

```
subNetworkIdPackage PACKAGE
            ATTRIBUTES
                subNetworkId                                            GET;
REGISTERED AS {es200653Package 39};
```

9.2.40 Supported by package

```
supportedByPackage  PACKAGE
            BEHAVIOUR
            supportedByPackageBehaviour BEHAVIOUR
                DEFINED AS "This package identifies resources that are required to support the
                operation of a particular package.";;

            ATTRIBUTES
                "Recommendation M.3100 : 1992":supportedByObjectList        GET;
REGISTERED AS {es200653Package 40};
```

9.2.41 System title package

```
systemTitlePackage PACKAGE
            ATTRIBUTES
                "Recommendation X.721 | ISO/IEC 10165-2 : 1992":systemTitle GET;
REGISTERED AS {es200653Package 41};
```

9.2.42 Topological group pointer package

```
topologicalGroupPointerPackage  PACKAGE
            BEHAVIOUR
        topologicalGroupPointerPackageBehaviour BEHAVIOUR
                 DEFINED AS "This package identifies an instance of a Topological Point or Access
                 Group managed object class.";;

            ATTRIBUTES
                topologicalGroupPointer                                     GET;
REGISTERED AS {es200653Package 43};
```

9.2.43 Type text package

```
typeTextPackage PACKAGE
            ATTRIBUTES
                typeText                                                    GET;
REGISTERED AS {es200653Package 44};
```

9.2.44 Unknown status package

```
unknownStatusPackage    PACKAGE
            ATTRIBUTES
                "Recommendation X.721 | ISO/IEC 10165-2 : 1992":unknownStatus   GET;
REGISTERED AS {es200653Package 45};
```

9.2.45 Usage cost package

```
usageCostPackage    PACKAGE
            ATTRIBUTES
                usageCost                                                   GET;
REGISTERED AS {es200653Package 46};
```

9.2.46 Weekly basis scheduling package

```
weeklyBasisSchedulingPackage PACKAGE
            BEHAVIOUR
        weeklyBasisSchedulingPackageBehaviour BEHAVIOUR
                 DEFINED AS "This package is instantiated when the setup action which created the
                 sub-network connection requests a weekly schedule. It contains the attributes
                 describing this scheduling and the action which enables any subsequent modification
                 of the schedule.";;

            ATTRIBUTES
                reservationBegin                                            GET,
                reservationEnd                                              GET,
                weeklySchedule                                              GET;
    ACTIONS
        changeWeeklyScheduling;
REGISTERED AS {es200653Package 47};
```

9.2.47 Z end NWTP list package

```
zEndNWTPListPackage PACKAGE
            ATTRIBUTES
                zEndNWTPList                                                GET;
REGISTERED AS {es200653Package 48};
```

9.3 Attribute definitions

9.3.1 Access group Id

```
accessGroupId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ES200653.NameType;
    MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
            BEHAVIOUR
    accessGroupIdBehaviour BEHAVIOUR
                DEFINED AS "The Access Group Id is an attribute type whose distinguished value can
                be used as an RDN when naming an instance of the Access Group object class.";;
REGISTERED AS {es200653Attribute 1};
```

9.3.2 Access point list

```
accessPointList ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ES200653.TPList;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
            BEHAVIOUR
    accessPointListBehaviour BEHAVIOUR
                DEFINED AS "The Access Point List attribute lists all the Network Trail Termination
                Points within an instance of the managed object class Access Group.";;
REGISTERED AS {es200653Attribute 2};
```

9.3.3 Admin domain Id

```
adminDomainId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ES200653.NameType;
    MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
            BEHAVIOUR
    adminDomainIdBehaviour BEHAVIOUR
                DEFINED AS "The Admin Domain Id is an attribute type whose distinguished value can
                be used as an RDN when naming an instance of the Admin Domain object class.";;
REGISTERED AS {es200653Attribute 3};
```

9.3.4 A end NWTP list

```
aEndNWTPList ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ES200653.TPList;
    MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
            BEHAVIOUR
    aEndNWTPListBehaviour BEHAVIOUR
                DEFINED AS "The value of this attribute identifies one or more network
                termination points of an instance of a sub-class of the Connectivity object
                class. The attribute cannot be null.";;
REGISTERED AS {es200653Attribute 4};
```

9.3.5 A end point

```
aEndPoint ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ES200653.ObjectInstance;
    MATCHES FOR EQUALITY;
            BEHAVIOUR
    aEndPointBehaviour BEHAVIOUR
                DEFINED AS "The A End Point attribute is used to indicate the terminating sub-
                network or Access Group either at one end of a Sub-network Pair, or at one end of a
                Link. The attribute cannot be null.";;
REGISTERED AS {es200653Attribute 5};
```

9.3.6 Allocation Id

```
allocationId ATTRIBUTE
WITH ATTRIBUTE SYNTAX   ES200653.NameType;
MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
        BEHAVIOUR
        allocationIdBehaviour BEHAVIOUR
                DEFINED AS "The allocationId attribute is an attribute type whose distinguished
                value can be used as an RDN when naming an instance of the Allocation managedobject
                class.";;
REGISTERED AS {es200653Attribute 6};
```

9.3.7 Assignment state

```
assignmentState ATTRIBUTE
WITH ATTRIBUTE SYNTAX   ES200653.AssignmentState;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        assignmentStateBehaviour BEHAVIOUR
                DEFINED AS "This attribute provides the assignment state of a resource. The states
                have the following meanings:
                free:the resource currently has no users,
                reserved:the resource is reserved for use by a user and may not be used by another
                user.
                NB This is not used for scheduling. partially
                assigned:capacity on the resource is used or reserved but capacity is still
                available for other users,
                assigned:the resource is in use and there is no spare capacity.";;
REGISTERED AS {es200653Attribute 7};
```

9.3.8 Available Link connections

```
availableLinkConnections ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.Count;
MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR
        availableLinkConnectionsBehaviour BEHAVIOUR
                DEFINED AS "This attribute indicates the number of available Link Connections
                contained in a Link.";;
REGISTERED AS {es200653Attribute 8};
```

9.3.9 Basic trail handler Id

```
basicTrailHandlerId ATTRIBUTE
WITH ATTRIBUTE SYNTAX   ES200653.NameType;
MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
REGISTERED AS {es200653Attribute 9};
```

9.3.10 Bi-directional traffic descriptor

```
bidirectionalTrafficDescriptor ATTRIBUTE
WITH ATTRIBUTE SYNTAX   ES200653.BidirectionalTrafficDescriptor;
MATCHES FOR EQUALITY;
REGISTERED AS {es200653Attribute 10};
```

9.3.11 Client link connection list

```
clientLinkConnectionList ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.ObjectList;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR
        clientLinkConnectionListBehaviour BEHAVIOUR
                DEFINED AS "This attribute defines the list of Link Connections which are clients
                of a Trail, or bundle (i.e. a number of Trails in parallel) of Trails in another
                layer. Usually a single Trail in a higher order layer will support a number of Link
                Connections in a lower order layer. Alternatively, a bundle (i.e. a number of
                Trails in parallel) of Trails in a lower order layer may support a Link Connection
                (or Link Connections) in a higher order layer.";;
REGISTERED AS {es200653Attribute 11};
```

9.3.12 Client CTP list

```
clientCTPList ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.ObjectList;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR
        clientCTPListBehaviour BEHAVIOUR
                DEFINED AS "This attribute defines the CTP or list of CTPs which are clients of a
                TTP or TTPs in another layer. Usually a single TTP in a higher order layer will
                support a number of CTPs in a lower order layer. Alternatively, where concatenation
                is used, a number of TTPs in a lower order layer may serve a CTP or CTPs in a
                higher order layer.";;
REGISTERED AS {es200653Attribute 12};
```

9.3.13 Client pointer

```
clientPtr ATTRIBUTE
WITH ATTRIBUTE SYNTAX   ES200653.ClientPtr;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        clientPtrBehaviour BEHAVIOUR
                DEFINED AS "This attribute points to the client of this allocation e.g. a client
                Tandem Connection or the client trail.";;
REGISTERED AS {es200653Attribute 13};
```

9.3.14 Component pointers

PROFILE NOTE: A composite subnetwork connection is made up of (i.e. partitioned into) a sequence of subnetwork connections and link connections, within the same layer. These subnetwork connections and link connections are components of the composite subnetwork connection. The component pointer is contained in the composite subnetwork connection, and points to each of the link connections and subnetwork connections in the sequence. Further details may be found in annex B.

```
componentPointers ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.ComponentPointers;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR
        componentPointersBehaviour BEHAVIOUR
                DEFINED AS "This attribute is used where the Sub-network Connection is made up of a
                number of component Sub-network Connections and Link connections within the same
                layer.";;
REGISTERED AS {es200653Attribute 14};
```

9.3.15 Composite pointer

PROFILE NOTE: A composite subnetwork connection is made up of (i.e. partitioned into) a sequence of subnetwork connections and link connections, within the same layer. These subnetwork connections and link connections are components of the composite subnetwork connection. The composite pointer is contained in each of the link connections and subnetwork connections and points from each of them to the composite subnetwork connection. Further details may be found in annex B.

```
compositePointer ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ES200653.CompositePointer;
    MATCHES FOR EQUALITY;
        BEHAVIOUR
        compositePointerBehaviour BEHAVIOUR
                DEFINED AS "This attribute is used where the connectivity instance is a component
                of a Sub-network Connection within the same layer.";;
REGISTERED AS {es200653Attribute 15};
```

9.3.16 Connected NWCTP count

```
connectedNWCTPCount ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.Count;
MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR
        connectedNWCTPCountBehaviour BEHAVIOUR
                DEFINED AS "This attribute indicates the number of NWCTPs associated with a
                Topological Point that have been connected.";;
REGISTERED AS {es200653Attribute 16};
```

9.3.17 Link connection list

```
linkConnectionList ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.ConnectionList;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR
        connectionListBehaviour BEHAVIOUR
                DEFINED AS "This attribute defines the list of Link connections which comprise a
                Link in a given layer.";;
REGISTERED AS {es200653Attribute 17};
```

9.3.18 Connectivity pointer

```
connectivityPointer ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.ConnectivityPointer;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        connectivityPointerBehaviour BEHAVIOUR
                DEFINED AS "This attribute points to the Link connection or Trail terminated by the
                Network Termination Point.";;
REGISTERED AS {es200653Attribute 18};
```

9.3.19 Contained in sub network list

```
containedInSubNetworkList ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.SubNetworkList;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR
        containedInSubNetworkListBehaviour BEHAVIOUR
                DEFINED AS "This attribute defines the list of parent Sub-networks which contain
                the Sub-network in a given layer.";;
REGISTERED AS {es200653Attribute 19};
```

9.3.20 Contained link list

```
containedLinkList ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.LinkList;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR
        containedLinkBehaviour BEHAVIOUR
                DEFINED AS "This attribute is used to describe the internal topology of a sub-
                network (in a given layer). This topology comprises links and sub-networks. The
                links are listed in this attribute.";;
REGISTERED AS {es200653Attribute 20};
```

9.3.21 Contained network CTP list

```
containedNWCTPList ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.NWCTPList;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR
        containedNWCTPListBehaviour BEHAVIOUR
                DEFINED AS "This attribute defines the list of Network CTPs which are contained in
                a Sub-network in a given layer.";;
REGISTERED AS {es200653Attribute 21};
```

9.3.22 Contained network TTP list

```
containedNWTTPList ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.NWTTPList;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR
        containedNWTTPListBehaviour BEHAVIOUR
                DEFINED AS "This attribute defines the list of Network TTPs which are contained in
                a Sub-network in a given layer.";;
REGISTERED AS {es200653Attribute 22};
```

9.3.23 Contained sub network list

```
containedSubNetworkList ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.SubNetworkList;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR
        containedSubNetworkListBehaviour BEHAVIOUR
                DEFINED AS "This attribute is used to describe the internal topology of a sub-
                network (in a given layer). This topology comprises links and sub-networks. The
                sub-networks are listed in this attribute.";;
REGISTERED AS {es200653Attribute 23};
```

9.3.24 Daily schedule

```
dailySchedule ATTRIBUTE
WITH ATTRIBUTE SYNTAX   ES200653.DailySchedule;
MATCHES FOR EQUALITY;
REGISTERED AS {es200653Attribute 24};
```

9.3.25 Void

9.3.26 Idle NWCTP count

```
idleNWCTPCount ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.Count;
MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR
        idleNWCTPCountBehaviour BEHAVIOUR
                DEFINED AS "This attribute indicates the number of NWCTPs associated with a
                Topological Point that have a status condition of In Service with Spare Capacity
                (6).";;
REGISTERED AS {es200653Attribute 26};
```

9.3.27 Instantiable basic connection performer Id

```
instantiableBasicConnectionPerformerId ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.NameType;
MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
        BEHAVIOUR
        instantiableBasicConnectionPerformerIdBehaviour BEHAVIOUR
                DEFINED AS "The instantiable Basic Connection Performer Id is an attribute type
                whose distinguished value can be used as an RDN when naming an instance of the
                Degenerate SubNetwork object class.";;
REGISTERED AS {es200653Attribute 27};
```

9.3.28 Layer link connection list

```
layerLinkConnectionList ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.LayerConnectionList;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR
        layerLinkConnectionListBehaviour BEHAVIOUR
                DEFINED AS "This attribute defines the list of Link Connections and subnetwork
                connections in a given layer which may compose a Trailin the same layer. This
                composition of Connectivity instances may be a simple sequence or, in the
                multipoint case a, a tree structure.";;
REGISTERED AS {es200653Attribute 28};
```

9.3.29 Layer trail

```
layerTrail ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.RelatedObjectInstance;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        layerTrailBehaviour BEHAVIOUR
                DEFINED AS "This attribute defines the Trail or concatenated Trail which a Link
                connection forms a part of within a given layer. It may be null.";;
REGISTERED AS {es200653Attribute 29};
```

9.3.30 Leg Id

```
legId ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.NameType;
MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
        BEHAVIOUR
        legIdBehaviour BEHAVIOUR
                DEFINED AS "The Leg Id is an attribute type whose distinguished value can be used
                as an RDN when naming an instance of the Leg object class.";;
REGISTERED AS {es200653Attribute 30};
```

9.3.31 Lifecycle state

```
lifecycleState ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.LifecycleState;
MATCHES FOR EQUALITY;
REGISTERED AS {es200653Attribute 31};
```

9.3.32 Link Id

```
linkId ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.NameType;
MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
        BEHAVIOUR
        linkIdBehaviour BEHAVIOUR
                DEFINED AS "The Link Id is an attribute type whose distinguished value can be used
                as an RDN when naming an instance of the Link object class.";;
REGISTERED AS {es200653Attribute 32};
```

9.3.33 Link pointer list

```
linkPointerList ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.LinkPointerList;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        linkPointerBehaviour BEHAVIOUR
                DEFINED AS "This attribute points to the links terminated by the sub-network or the
                link terminated by an access group";;
REGISTERED AS {es200653Attribute 33};
```

9.3.34 Mode

```
mode ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.Mode;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        modeBehaviour BEHAVIOUR
                DEFINED AS "The Mode attribute indicates the type of transmission supported by an
                instance of Connectivity, or its subclasses. It may take any of the following
                values: point to point:there is one A end and one Z end; point to multipoint:there
                is one A end and multiple Z ends, and there is no traffic flow between Z ends;
                multicast:there are multiple A ends and multiple Z ends, and there is no traffic
                flow between A ends or between Z ends; conference:the multiple A ends send traffic
                to, and receive traffic from, all other A ends, there are no Z ends;
                broadcast:there is one A end and no known Z ends.";;
REGISTERED AS {es200653Attribute 34};
```

9.3.35 Monthly schedule

```
monthlySchedule ATTRIBUTE
WITH ATTRIBUTE SYNTAX   ES200653.MonthlySchedule;
MATCHES FOR EQUALITY;
REGISTERED AS {es200653Attribute 35};
```

9.3.36 NE assignment pointer

```
neAssignmentPointer ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.RelatedObjectInstance;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        neAssignmentPointerBehaviour BEHAVIOUR
                DEFINED AS "The NE Assignment Pointer attribute points from the lowest level
                Network TP in the partitioning hierarchy to a NE TP which represents the
                functionality which supports the Network TP. The sub-partition pointer for a NWCTP
                which utilizes the NE assignment pointer will be NULL.";;
REGISTERED AS {es200653Attribute 36};
```

9.3.37 Network TP pointer

```
networkTPPointer ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.RelatedObjectInstance;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        networkTPPointerBehaviour BEHAVIOUR
                DEFINED AS "The Network TP Pointer attribute points to a network termination
                point.";;
REGISTERED AS {es200653Attribute 37};
```

9.3.38 No of link connections

```
noOfLinkConnections ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.Count;
MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR
        noOfLinkConnectionsBehaviour BEHAVIOUR
                DEFINED AS "This attribute indicates the total number of Link connections contained
                in a Link.";;
REGISTERED AS {es200653Attribute 38};
```

9.3.39 NWCTPs in topological point list

```
nWCTPsInTopologicalPointList ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.TPList;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR
        nWCTPsInTopologicalPointListBehaviour BEHAVIOUR
                DEFINED AS "This attribute lists the NWCTPs that are represented by a Topological
                Point.";;
REGISTERED AS {es200653Attribute 39};
```

9.3.40 Occasional schedule

```
occasionalSchedule ATTRIBUTE
WITH ATTRIBUTE SYNTAX   ES200653.OccasionalSchedule;
MATCHES FOR EQUALITY;
REGISTERED AS {es200653Attribute 40};
```

9.3.41 Quality of connectivity service

```
qualityOfConnectivityService ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.QofConnectivityService;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        qualityOfConnectivityServiceBehaviour BEHAVIOUR
                DEFINED AS "This attribute indicates the quality of service for Connectivity and
                its subclasses, and requires further definition.";;
REGISTERED AS {es200653Attribute 41};
```

9.3.42 Reservation begin

```
reservationBegin ATTRIBUTE
WITH ATTRIBUTE SYNTAX   ES200653.StartTime;
MATCHES FOR EQUALITY;
REGISTERED AS {es200653Attribute 42};
```

9.3.43 Reservation end

```
reservationEnd ATTRIBUTE
WITH ATTRIBUTE SYNTAX   ES200653.StopTime;
MATCHES FOR EQUALITY;
REGISTERED AS {es200653Attribute 43};
```

9.3.44 Server trail

```
serverTrail ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.ObjectList;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        serverTrailBehaviour BEHAVIOUR
                DEFINED AS "This attribute defines the Trail which may serve a Link connection in
                another layer. Usually a single Trail in a higher order layer will support a number
                of Link connections in a lower order layer. Alternatively, a number of concatenated
                Trails in a lower order layer may support a Link connection in a higher order
                layer.";;
REGISTERED AS {es200653Attribute 44};
```

9.3.45 Server TTP Pointer

```
serverTTPPointer ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.ObjectList;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        serverTTPPointerbehaviour BEHAVIOUR
                DEFINED AS "This attribute defines the TTP which may serve a CTP in another layer.
                Usually a TTP or TTPs in a higher order layer will serve a CTP or CTPs in a lower
                order layer.";;
REGISTERED AS {es200653Attribute 45};
```

9.3.46 Signal Id

```
signalid ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.SignalId;
MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
        BEHAVIOUR
        signalidBehaviour BEHAVIOUR
                DEFINED AS "This attribute defines the characteristic information of the layer (in
                the G.805 sense) to which the entity under consideration belongs. It is used to
                determine whether sub-network connection/connectivity is possible. The signal Id
                may be a simple rate and format or may be a bundle of entities with the same
                characteristic information which form an aggregate signal.";;
REGISTERED AS {es200653Attribute 46};
```

9.3.47 Signal list

```
signalList ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.SignalList;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
REGISTERED AS {es200653Attribute 47};
```

9.3.48 Sub network connection Id

```
subNetworkConnectionId ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.NameType;
MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
        BEHAVIOUR
        subNetworkConnectionIdBehaviour BEHAVIOUR
                DEFINED AS "The Sub-network Connection Id is an attribute type whose distinguished
                value can be used as an RDN when naming an instance of the sub-network Connection
                object class.";;
REGISTERED AS {es200653Attribute 48};
```

9.3.49 Sub network connection pointer

```
subNetworkConnectionPointer ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.SubNetworkConnectionPointerList;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        subNetworkConnectionPointerBehaviour BEHAVIOUR
                DEFINED AS "The Sub-network Connection Pointer attribute points to the ordered list
                of sub-network Connection(s) which have a relationship with the network termination
                point or NWGTP. For a network Termination Point within a NWGTP, the
                subNetworkConnectionPointer points to the NWGTP. When no sub- network connection is
                present this pointer points to a sub-network or is NULL.This list has a single
                entry for point to point applications, and may have mutliple entries for point to
                multipoint applications.";;
REGISTERED AS {es200653Attribute 49};
```

PROFILE NOTE: A NWCTP may be part of many sub-networks. When no sub-network connection is present the pointer will usually point to a sub-network at the lowest level of partitioning in the Agent.

9.3.50 Sub network Id

```
subNetworkId ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.NameType;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        subNetworkIdBehaviour BEHAVIOUR
                DEFINED AS "The Sub-network Id is an attribute type whose distinguished value can
                be used as an RDN when naming an instance of the Sub-network object class.";;
REGISTERED AS {es200653Attribute 50};
```

9.3.51 Sub network pair Id

```
subNetworkPairId ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.NameType;
MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
        BEHAVIOUR
        subNetworkPairIdBehaviour BEHAVIOUR
                DEFINED AS "The Sub-network Pair Id is an attribute type whose distinguished value
                can be used as an RDN when naming an instance of the Sub-network Pair object
                class.";;
REGISTERED AS {es200653Attribute 51};
```

9.3.52 Sub partition pointer

```
subPartitionPointer ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.RelatedObjectInstance;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        subPartitionPointerBehaviour BEHAVIOUR
                DEFINED AS "The Sub Partition Pointer is a pointer to a Network CTP which is in a
                lower level partition. Where the lowest level of NWCTP points to a NE CTP via the
                NE Assignment Pointer, the value of the Sub Partition Pointer is null.";;
REGISTERED AS {es200653Attribute 52};
```

9.3.53 Super partition pointer

```
superPartitionPointer ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.RelatedObjectInstance;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        superPartitionPointerBehaviour BEHAVIOUR
                DEFINED AS "The Super Partition Pointer is a pointer to a Network CTP which is in a
                higher level partition. It will only be present for those Network CTPs in the lower
                partition which have a direct correspondence to the Network CTPs at the higher
                level. It can be null.";;
REGISTERED AS {es200653Attribute 53};
```

9.3.54 Topological group pointer

```
topologicalGroupPointer ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.RelatedObjectInstance;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        topologicalGroupPointerBehaviour BEHAVIOUR
                DEFINED AS "The Topological Group Pointer is an attribute type which identifies an
                instance of the Topological Point managed object class or identifies an instance of
                the Access Group managed object class .";;
REGISTERED AS {es200653Attribute 54};
```

9.3.55 Topological point Id

```
topologicalPointId ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.NameType;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        topologicalPointIdBehaviour BEHAVIOUR
                DEFINED AS "The Topological Point Id is an attribute type whose distinguished value
                can be used as an RDN when naming an instance of the Topological Point object
                class.";;
REGISTERED AS {es200653Attribute 55};
```

9.3.56 Total NWCTP count

```
totalNWCTPCount ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.Count;
MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR
        totalNWCTPCountBehaviour BEHAVIOUR
                DEFINED AS "This attribute indicates the total number of NWCTPs associated with a
                Topological Point.";;
REGISTERED AS {es200653Attribute 56};
```

9.3.57 Trail list

```
trailList ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.TrailList;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR
        trailListBehaviour BEHAVIOUR
                DEFINED AS "This attribute defines the list of Trails originating and terminating
                in a given pair of Sub-networks associated with a Sub-Network Pair.";;
REGISTERED AS {es200653Attribute 57};
```

9.3.58 Type text

```
typeText ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.TypeText;
MATCHES FOR EQUALITY;
REGISTERED AS {es200653Attribute 58};
```

9.3.59 Usage cost

```
usageCost ATTRIBUTE
WITH ATTRIBUTE SYNTAX   ES200653.UsageCost;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        usageCostBehaviour BEHAVIOUR
                DEFINED AS "This attribute contains the costs for a transport entity. It is to be
                used as selection/routingcriteria.";;
REGISTERED AS {es200653Attribute 59};
```

9.3.60 Weekly schedule

```
weeklySchedule ATTRIBUTE
WITH ATTRIBUTE SYNTAX   ES200653.WeeklySchedule;
MATCHES FOR EQUALITY;
REGISTERED AS {es200653Attribute 60};
```

9.3.61 Z end point

```
zEndPoint ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.ObjectInstance;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        zEndPointBehaviour BEHAVIOUR
                DEFINED AS "The Z End Point attribute is used to indicate the terminating sub-
                network or Access Group either at one end of a Sub-network Pair, or at one end of a
                Link. The attribute cannot be null.";;
REGISTERED AS {es200653Attribute 61};
```

9.3.62 Z end NWTP

```
zEndNWTP ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.ConnectivityEndPoint;
MATCHES FOR EQUALITY;
        BEHAVIOUR
        zEndNWTPBehaviour BEHAVIOUR
                DEFINED AS "The value of this attribute identifies the Z end network termination
                point of an instance of a Leg contained in a Sub-network Connection. The attribute
                cannot be null.";;
REGISTERED AS {es200653Attribute 62};
```

9.3.63 Z end NWTP list

```
zEndNWTPList ATTRIBUTE
WITH ATTRIBUTE SYNTAX ES200653.TPList;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR
        zEndNWTPListBehaviour BEHAVIOUR
                DEFINED AS "The value of this attribute identifies one or more network termination
                points of an instance of a sub-class of the Connectivity object class.";;
REGISTERED AS {es200653Attribute 63};
```

9.4 Name bindings

PROFILE NOTE: The set of name bindings defines the MIB for a particular interface. A name binding (as discussed in annex B) is both the implementation of a relationship and part of the construction of the MIB for a particular interface.
Since the Generic class library is not specific to any given interface, it is not possible in the class library to give a definitive set of name bindings. In particular the choice as to how a given relationship is implemented (e.g. by pointers or name bindings) is the responsibility of the application groups. Hence these name bindings are not exhaustive, nor are they prescriptive, and additional or alternative name bindings may be defined in ensembles for particular applications.
Example schema are presented in annex B.

9.4.1 Access group

```
accessGroup-adminDomain NAME BINDING
    SUBORDINATE OBJECT CLASS    accessGroup AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   adminDomain AND SUBCLASSES;
    WITH ATTRIBUTE  accessGroupId;
    CREATE
        WITH-REFERENCE-OBJECT;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 1};
```

9.4.2 Admin domain

```
adminDomain-system NAME BINDING
    SUBORDINATE OBJECT CLASS    adminDomain AND SUBCLASSES;
    NAMED BY
SUPERIOR OBJECT CLASS "Recommendation X.721 | ISO/IEC 10165-2 : 1992":system AND SUBCLASSES;
    WITH ATTRIBUTE  adminDomainId;
    CREATE
```

```
        WITH-REFERENCE-OBJECT;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 2};

adminDomain-adminDomain NAME BINDING
    SUBORDINATE OBJECT CLASS    adminDomain AND SUBCLASSES;
    NAMED BY
SUPERIOR OBJECT CLASS adminDomain AND SUBCLASSES;
    WITH ATTRIBUTE  adminDomainId;
    CREATE
        WITH-REFERENCE-OBJECT;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 31};
```

9.4.3 Allocation

```
allocation-trail NAME BINDING
    SUBORDINATE OBJECT CLASS    allocation AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   trail AND SUBCLASSES;
    WITH ATTRIBUTE  allocationId;
    CREATE
        WITH-REFERENCE-OBJECT;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 3};
```

9.4.4 Degenerate sub-network

```
degenerateSubNetwork-adminDomain NAME BINDING
    SUBORDINATE OBJECT CLASS    degenerateSubNetwork AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   adminDomain AND SUBCLASSES;
    WITH ATTRIBUTE  subNetworkId;
    CREATE
        WITH-REFERENCE-OBJECT;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 4};
```

9.4.5 Instantiable basic connection performer

```
instantiableBasicConnectionPerformer-subNetwork NAME BINDING
    SUBORDINATE OBJECT CLASS instantiableBasicConnectionPerformer AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   subNetwork AND SUBCLASSES;
    WITH ATTRIBUTE  instantiableBasicConnectionPerformerId;
    CREATE
        WITH-REFERENCE-OBJECT;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 5};
```

9.4.6 Instantiable basic trail handler

```
instantiableBasicTrailHandler-layerNetworkDomain NAME BINDING
    SUBORDINATE OBJECT CLASS    instantiableBasicTrailHandler AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   layerNetworkDomain AND SUBCLASSES;
    WITH ATTRIBUTE  basicTrailHandlerId;
    CREATE
        WITH-REFERENCE-OBJECT;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 6};
```

9.4.7 Leg

```
leg-subNetworkConnection NAME BINDING
    SUBORDINATE OBJECT CLASS    leg AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   subNetworkConnection AND SUBCLASSES;
    WITH ATTRIBUTE  legId;
REGISTERED AS {es200653NameBinding 7};
```

9.4.8 Link

```
link-adminDomain NAME BINDING
    SUBORDINATE OBJECT CLASS    link AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   adminDomain AND SUBCLASSES;
    WITH ATTRIBUTE  linkId;
    CREATE
        WITH-REFERENCE-OBJECT;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 8};


link-system NAME BINDING
    SUBORDINATE OBJECT CLASS link AND SUBCLASSES;
    NAMED BY
    SUPERIOR OBJECT CLASS "Recommendation X.721 | ISO/IEC 10165-2 : 1992":system AND SUBCLASSES;
    WITH ATTRIBUTE  linkId;
REGISTERED AS {es200653NameBinding 9};
```

9.4.9 Link connection

```
linkConnection-adminDomain NAME BINDING
    SUBORDINATE OBJECT CLASS    linkConnection AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   adminDomain AND SUBCLASSES;
    WITH ATTRIBUTE  "Recommendation M.3100 : 1992":connectionId;
    CREATE
        WITH-REFERENCE-OBJECT;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 10};


linkConnection-linkOne NAME BINDING
    SUBORDINATE OBJECT CLASS    linkConnection AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   link AND SUBCLASSES;
    WITH ATTRIBUTE  "Recommendation M.3100 : 1992":connectionId;
    CREATE
        WITH-REFERENCE-OBJECT;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 11};


linkConnection-linkTwo NAME BINDING
    SUBORDINATE OBJECT CLASS    linkConnection AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   link AND SUBCLASSES;
    WITH ATTRIBUTE  "Recommendation M.3100 : 1992":connectionId;
REGISTERED AS {es200653NameBinding 12};

--Two bindings for link connection to link are defined. This is to reflect the fact that the link
may be within a TMN or betrween TMNs. Each case has different CREATE/DELETE behaviour because of the
different management capabilities of the two cases.
```

9.4.10 Network CTP sink

```
networkCTPSink-subNetwork NAME BINDING
    SUBORDINATE OBJECT CLASS    networkCTPSink AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   subNetwork AND SUBCLASSES;
    WITH ATTRIBUTE  "Recommendation M.3100 : 1992":cTPId;
            BEHAVIOUR
        networkCTPSink-subNetworkBehaviour BEHAVIOUR
```

```
                DEFINED AS "The subordinate managed object is automatically instantiated or deleted
                when the superior managed object is instantiated, or when additional resources
                (including planned resources) are added to, or removed from, the sub-network,
                according to the configuration of the Sub-network.";;
REGISTERED AS {es200653NameBinding 13};

networkCTPSink-adminDomain NAME BINDING
    SUBORDINATE OBJECT CLASS    networkCTPSink AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   adminDomain AND SUBCLASSES;
    WITH ATTRIBUTE  "Recommendation M.3100 : 1992":cTPId;
    CREATE
        WITH-REFERENCE-OBJECT;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 14};


networkCTPSink-networkTTPSink NAME BINDING
    SUBORDINATE OBJECT CLASS    networkCTPSink AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   networkTTPSink AND SUBCLASSES;
    WITH ATTRIBUTE  "Recommendation M.3100 : 1992":cTPId;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 15};
```

9.4.11 Network CTP source

```
networkCTPSource-subNetwork NAME BINDING
    SUBORDINATE OBJECT CLASS    networkCTPSource AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   subNetwork AND SUBCLASSES;
    WITH ATTRIBUTE  "Recommendation M.3100 : 1992":cTPId;
            BEHAVIOUR
        networkCTPSource-subNetworkBehaviour BEHAVIOUR
                DEFINED AS "The subordinate managed object is automatically instantiated or deleted
                when the superior managed object is instantiated, or when additional resources
                (including planned resources) are added to, or removed from, the sub-network,
                according to the configuration of the sub-network.";;
REGISTERED AS {es200653NameBinding 16};


networkCTPSource-adminDomain NAME BINDING
    SUBORDINATE OBJECT CLASS    networkCTPSource AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   adminDomain AND SUBCLASSES;
    WITH ATTRIBUTE  "Recommendation M.3100 : 1992":cTPId;
    CREATE
        WITH-REFERENCE-OBJECT;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 17};


networkCTPSource-networkTTPSource NAME BINDING
    SUBORDINATE OBJECT CLASS    networkCTPSource AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   networkTTPSource AND SUBCLASSES;
    WITH ATTRIBUTE  "Recommendation M.3100 : 1992":cTPId;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 18};
```

9.4.12 Network GTP

```
networkGTP-subNetwork NAME BINDING
    SUBORDINATE OBJECT CLASS    networkGTP AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   subNetwork AND SUBCLASSES;
    WITH ATTRIBUTE  "Recommendation M.3100 : 1992":gtpId;
            BEHAVIOUR
    networkGTP-networkBehaviour BEHAVIOUR
```

```
                DEFINED AS "The subordinate managed object is automatically created by invoking the
                action addNWTPsToNWGTP. It is automatically deleted when the last contained NWCTP
                is removed using the action removeNWTPsFromNWGTP."
                ;;
REGISTERED AS {es200653NameBinding 19};
```

9.4.13 Network TTP sink

```
networkTTPSink-adminDomain NAME BINDING
    SUBORDINATE OBJECT CLASS    networkTTPSink AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   adminDomain AND SUBCLASSES;
    WITH ATTRIBUTE  "Recommendation M.3100 : 1992":tTPId;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 20};


networkTTPSink-subNetwork NAME BINDING
    SUBORDINATE OBJECT CLASS    networkTTPSink AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   subNetwork AND SUBCLASSES;
    WITH ATTRIBUTE  "Recommendation M.3100 : 1992":tTPId;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 21};
```

9.4.14 Network TTP source

```
networkTTPSource-adminDomain NAME BINDING
    SUBORDINATE OBJECT CLASS    networkTTPSource AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   adminDomain AND SUBCLASSES;
    WITH ATTRIBUTE  "Recommendation M.3100 : 1992":tTPId;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 22};


networkTTPSource-subNetwork NAME BINDING
    SUBORDINATE OBJECT CLASS    networkTTPSource AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   subNetwork AND SUBCLASSES;
    WITH ATTRIBUTE  "Recommendation M.3100 : 1992":tTPId;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 23};
```

9.4.15 Node

```
node-adminDomain NAME BINDING
    SUBORDINATE OBJECT CLASS    node AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   adminDomain AND SUBCLASSES;
    WITH ATTRIBUTE  adminDomainId;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 24};
```

9.4.16 Sub-network

```
subNetwork-adminDomain NAME BINDING
    SUBORDINATE OBJECT CLASS    subNetwork AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   adminDomain AND SUBCLASSES;
    WITH ATTRIBUTE  subNetworkId;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 25};

subNetwork-system NAME BINDING
    SUBORDINATE OBJECT CLASS    subNetwork AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS
        "Recommendation X.721 | ISO/IEC 10165-2 : 1992":system AND SUBCLASSES;
    WITH ATTRIBUTE  subNetworkId;
REGISTERED AS {es200653NameBinding 26};
```

9.4.17 Sub-network connection

```
subNetworkConnection-subNetwork NAME BINDING
    SUBORDINATE OBJECT CLASS    subNetworkConnection AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   subNetwork AND SUBCLASSES;
    WITH ATTRIBUTE  subNetworkConnectionId;
            BEHAVIOUR
        subNetworkConnection-subNetworkBehaviour BEHAVIOUR
                DEFINED AS "There is no creation or deletion behaviour because this is performed by
                actions.";;
REGISTERED AS {es200653NameBinding 27};
```

9.4.18 Sub-network pair

```
subNetworkPair-adminDomain NAME BINDING
    SUBORDINATE OBJECT CLASS    subNetworkPair AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   adminDomain AND SUBCLASSES;
    WITH ATTRIBUTE  subNetworkPairId;
    CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 28};
```

9.4.19 Topological point

```
topologicalPoint-subNetwork NAME BINDING
    SUBORDINATE OBJECT CLASS    topologicalPoint AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   subNetwork AND SUBCLASSES;
    WITH ATTRIBUTE  topologicalPointId;
            BEHAVIOUR
        topologicalPoint-subNetworkBehaviour BEHAVIOUR
                DEFINED AS "The subordinate managed object is automatically created by invoking the
                action addNWCTPsToTopologicalPoint. It is automatically deleted when the last
                contained NWCTP is removed using the action removeNWCTPsFromTopologicalPoint.";;
REGISTERED AS {es200653NameBinding 29};
```

9.4.20 Trail

```
trail-adminDomain NAME BINDING
    SUBORDINATE OBJECT CLASS    trail AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   adminDomain AND SUBCLASSES;
    WITH ATTRIBUTE  "Recommendation M.3100 : 1992":trailId;
            BEHAVIOUR
        trail-adminDomainBehaviour BEHAVIOUR
                DEFINED AS "There is no creation or deletion behaviour because this is performed by
                actions.";;
REGISTERED AS {es200653NameBinding 30};
```

9.5 Actions

9.5.1 Activate sub network connection

```
activateSubNetworkConnection ACTION
            BEHAVIOUR
        activateSubNetworkConnectionBehaviour BEHAVIOUR
                DEFINED AS "This action is the second half of the two-stage process to set up sub-
                network Connections. It activates a Sub-network Connection which has already been
                set up and has a Status Condition of In Service Reserved (4). This action (if
                successful) changes the Status condition to In Service with no spare capacity (8).
                If the transactionId parameter is used, it shall be the same as the transactionId
                used in the original SetupSubNetworkConnection action. The Status condition of all
                network termination points, Link Connections and sub-network connections involved
                in the Sub-network Connection being activated will be the same as that of the
                composite Sub-network Connection. If any of the underlying resources supporting the
                Sub-network Connection have a Status condition of Resource Failed (10), Resource
                Failed , Reserved (10a) or Resource Failed with no spare capacity (10c), the Sub-
                network Connection shall have the same Status condition.";;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX ES200653.ActivateSubNetworkConnectionInformation;
    WITH REPLY SYNTAX           ES200653.ActivateSubNetworkConnectionResult;
REGISTERED AS {es200653Action 1};
```

9.5.2 Add to sub network connection

```
addToSubNetworkConnection ACTION
            BEHAVIOUR
        addToSubNetworkConnectionBehaviour BEHAVIOUR
                DEFINED AS "This action is used to add one or more legs to an existing sub-network
                Connection of type point to multipoint or multicast. If the action is used on a
                point to point Sub-network Connection, the Sub-network Connection becomes point to
                multipoint. Additional Z End network termination points shall be provided, and Leg
                objects will be created for each Z End, including the Z End of the original point
                to point Sub-network Connection. For addition to a point to point or point to
                multipoint Sub-network Connection, Z End network termination points shall be
                provided. One additional Leg object will be created for each new Z End network
                termination point. For addition to a multicast Sub-network Connection, either or
                both A and Z End network termination points may be provided. If A End network
                termination points are added, then one new Sub-network Connection object will be
                created for each A End. Each new Sub-network Connection will be contained by the
                parent Multicast sub-network Connection object, and will have the same set of Z
                Ends as the existing Sub-network Connections contained in the Multicast Sub-network
                Connection. If Z End network termination points are added, then each new Z End
                shall be added to each existing Sub-network Connection contained by the Multicast
                Sub-network Connection. Additional Leg objects shall be created for each Z End
                which is new or is in a new Sub-network Connection. Supplied network termination
                points or NWGTPs shall support a similar Signal Id to that of the network
                termination points already in the Sub-network Connection. The result, if
                successful, always returns the network termination points or NWGTPs involved in the
                Sub-network Connection. If a Topological Point is involved in the Sub-network
                Connection, its attributes idleNWCTPCount, and connectedNWCTPCount will be updated
                as a result of this action.";;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX ES200653.AddToSubNetworkConnectionInformation;
    WITH REPLY SYNTAX           ES200653.AddToSubNetworkConnectionResult;
REGISTERED AS {es200653Action 2};
```

9.5.3 Add NWCTPs to topological Pt

```
addNWCTPsToTopologicalPt ACTION
            BEHAVIOUR
        addNWCTPsToTopologicalPtBehaviour BEHAVIOUR
                DEFINED AS "This action is used to arrange Network Connection Termination Points
                into Topological Points. If one of the Topological Point instances does not exist,
                then a new one is automatically created and its identity returned in the action
                result. Otherwise the NWCTPs are added to those already in the Topological
                Point(s)."
                ;;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX ES200653.AddNWCTPsToTopologicalPtInformation;
    WITH REPLY SYNTAX           ES200653.AddNWCTPsToTopologicalPtResult;
REGISTERED AS {es200653Action 3};
```

9.5.4 Add NWTPs to NWGTP

```
addNWTPsToNWGTP ACTION
            BEHAVIOUR
        addNWTPsToNWGTPBehaviour BEHAVIOUR
                DEFINED AS "This action is used to arrange network termination points into Network
                Group Termination Points. If the NWGTP instance does not exist then a new one is
                automatically created and its identity returned in the action result. Members of
                the NWGTP shall be all NWTTPs or all NWCTPs, and shall all be capable of operating
                in the same direction."
                ;;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX     ES200653.AddNWTPsToNWGTPInformation;
    WITH REPLY SYNTAX           ES200653.AddNWTPsToNWGTPResult;
REGISTERED AS {es200653Action 4};
```

9.5.5 Add NWTTPs to access group

```
addNWTTPsToAccessGroup ACTION
            BEHAVIOUR
        addNWTTPsToAccessGroupBehaviour BEHAVIOUR
                DEFINED AS "This action is used to arrange Network Trail Termination Points into
                Access Groups. If one of the Access Group instances does not exist then a new one
                is automatically created and its identity returned in the action result. Otherwise
                the NWTTPs are added to those already in the Access Group(s)."
                ;;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX     ES200653.AddNWTTPsToAccessGroupInformation;
    WITH REPLY SYNTAX           ES200653.AddNWTTPsToAccessGroupResult;
REGISTERED AS {es200653Action 5};
```

9.5.6 Change daily scheduling

```
changeDailyScheduling ACTION
            BEHAVIOUR
        changeDailySchedulingBehaviour BEHAVIOUR
                DEFINED AS "This action enables to request a change of the bandwidth of a daily
                scheduled sub-network connection. This request is immediately applicable. A two
                phase modification process is for further study."
                ;;

    MODE CONFIRMED ;
    WITH INFORMATION SYNTAX     ES200653.ChangeDailySchedulingInfo;
    WITH REPLY SYNTAX           ES200653.ChangeDailySchedulingResult;
REGISTERED AS {es200653Action 6};
```

9.5.7 Change duration scheduling

```
changeDurationScheduling ACTION
            BEHAVIOUR
        changeDurationSchedulingBehaviour BEHAVIOUR
                DEFINED AS "This action enables to request a change of the bandwidth of an
                immediate sub-network connection. This request is immediately applicable. A two
                phase modification process is for further study."
                ;;

    MODE CONFIRMED ;
    WITH INFORMATION SYNTAX     ES200653.ChangeDurationSchedulingInfo;
    WITH REPLY SYNTAX           ES200653.ChangeDurationSchedulingResult;
REGISTERED AS {es200653Action 7};
```

9.5.8 Change monthly scheduling

```
changeMonthlyScheduling ACTION
            BEHAVIOUR
    changeMonthlySchedulingBehaviour BEHAVIOUR
                DEFINED AS "This action enables to request a change of the bandwidth of a monthly
                scheduled sub-network connection. This request is immediately applicable. A two
                phase modification process is for further study."
                ;;
```

```
    MODE CONFIRMED ;
    WITH INFORMATION SYNTAX     ES200653.ChangeMonthlySchedulingInfo;
    WITH REPLY SYNTAX           ES200653.ChangeMonthlySchedulingResult;
REGISTERED AS {es200653Action 8};
```

9.5.9 Change occasional scheduling

```
changeOccasionalScheduling ACTION
            BEHAVIOUR
    changeOccasionalSchedulingBehaviour BEHAVIOUR
                DEFINED AS "This action enables to request a change of the bandwidth of an
                occasionally scheduled sub-network connection. This request is immediately
                applicable. A two phase modification process is for further study."
                ;;

    MODE CONFIRMED ;
    WITH INFORMATION SYNTAX     ES200653.ChangeOccasionalSchedulingInfo;
    WITH REPLY SYNTAX           ES200653.ChangeOccasionalSchedulingResult;
REGISTERED AS {es200653Action 9};
```

9.5.10 Change weekly scheduling

```
changeWeeklyScheduling ACTION
            BEHAVIOUR
    changeWeeklySchedulingBehaviour BEHAVIOUR
                DEFINED AS "This action enables to request a change of the bandwidth of a weekly
                scheduled sub-network connection. This request is immediately applicable. A two
                phase modification process is for further study.";;

    MODE CONFIRMED ;
    WITH INFORMATION SYNTAX     ES200653.ChangeWeeklySchedulingInfo;
    WITH REPLY SYNTAX           ES200653.ChangeWeeklySchedulingResult;
REGISTERED AS {es200653Action 10};
```

9.5.11 Delete from sub network connection

```
deleteFromSubNetworkConnection ACTION
            BEHAVIOUR
    deleteFromSubNetworkConnectionBehaviour BEHAVIOUR
                DEFINED AS "This action is used to delete a leg from a Sub-network Connection,
                providing it is not the last remaining leg in the Sub-network Connection. In that
                instance, the action ReleaseSubNetworkConnection shall be used. To delete a leg
                from a point to multipoint Sub-network Connection, Z End network termination points
                shall be provided. To delete a leg from a multicast Sub-network Connection, either
                or both A and Z End network termination points may be provided. To delete a leg
                from a conference Sub-network Connection, A End network termination points shall be
                provided. The Sub-network Connections pointed to by the compositePointer attribute
                will also be cleared down by this action. If a Topological Point is involved in the
                Sub-network Connection, its attributes idleNWCTPCount, and connectedNWCTPCount will
                be updated as a result of this action. ";;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX     ES200653.DeleteFromSubNetworkConnectionInformation;
    WITH REPLY SYNTAX           ES200653.DeleteFromSubNetworkConnectionResult;
REGISTERED AS {es200653Action 11};
```

9.5.12 Release sub network connection

PROFILE NOTE: A brach of a connection may refer to the leg of a multipoint subnetwork connection (see annex D) or a subnetwork connection of a multipoint connection (see annex E).

```
releaseSubNetworkConnection ACTION
            BEHAVIOUR
        releaseSubNetworkConnectionBehaviour BEHAVIOUR
                DEFINED AS "This action is used to release Sub-network Connection(s). If the
                connection is more complex than point to point, all branchesof the connection
                willbe disconnected. The Sub-network Connection pointed to by the
                compositePointerattribute will also be cleared down by this action. If a
                Topological Point is involved in the Sub-network Connection, its attributes
                idleNWCTPCount, and connectedNWCTPCount will be updated as a result of this action.
                If implicit TP creation is used, the associated TPs will be deleted when the sub-
                network connection is released.";;
```

```
    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX     ES200653.ReleaseSubNetworkConnectionInformation;
    WITH REPLY SYNTAX           ES200653.ReleaseSubNetworkConnectionResult;
REGISTERED AS {es200653Action 12};
```

9.5.13 Release trail

```
releaseTrail ACTION
            BEHAVIOUR
        releaseTrailBehaviour BEHAVIOUR
                DEFINED AS "This action is used to release a Trail. The link connections pointed to
                by the clientConnectionList and the sub-network connections pointed to by the layer
                connection list package will also be released by this action. The
                connectivityPointer in the disconnected network trail termination points will be
                set to NULL as a result of this action."
                ;;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX     ES200653.ReleaseTrailInformation;
    WITH REPLY SYNTAX           ES200653.ReleaseTrailResult;
REGISTERED AS {es200653Action 13};
```

9.5.14 Remove NWCTPs from topological Pt

```
removeNWCTPsFromTopologicalPt ACTION
            BEHAVIOUR
        removeNWCTPsfromTopologicalPtBehaviour BEHAVIOUR
                DEFINED AS "This action is used to remove Network Connection Termination Points
                from Topological Points. Removing the last NWCTP from a Topological Point has the
                effect of deleting the Topological Point object. If the Topological Point is
                deleted, its name will be sent back in the action result."
                ;;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX     ES200653.RemoveNWCTPsFromTopologicalPtInformation;
    WITH REPLY SYNTAX           ES200653.RemoveNWCTPsFromTopologicalPtResult;
REGISTERED AS {es200653Action 14};
```

9.5.15 Remove NWTPs from NWGTP

```
removeNWTPsFromNWGTP ACTION
            BEHAVIOUR
        removeNWTPsFromNWGTPBehaviour BEHAVIOUR
                DEFINED AS "This action is used to remove network termination points from Network
                Group Termination Points. This action will fail if the NWGTP is involved in a Sub-
                network Connection. Removing the last network termination point from a NWGTP has
                the effect of deleting the NWGTP object. If the NWGTP is deleted, its name will be
                sent back in the action result."
                ;;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX     ES200653.RemoveNWTPsFromNWGTPInformation;
    WITH REPLY SYNTAX           ES200653.RemoveNWTPsFromNWGTPResult;
REGISTERED AS {es200653Action 15};
```

9.5.16 Remove NWTTPs from access group

```
removeNWTTPsFromAccessGroup ACTION
            BEHAVIOUR
        removeNWTTPsFromAccessGroupBehaviour BEHAVIOUR
                DEFINED AS "This action is used to remove Network Trail Termination Points from
                Access Groups. Removing the last NWTTP from an Access Group has the effect of
                deleting the Access Group object. If the Access Group is deleted, its name will be
                sent back in the action result."
                ;;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX     ES200653.RemoveNWTTPsFromAccessGroupInformation;
    WITH REPLY SYNTAX           ES200653.RemoveNWTTPsFromAccessGroupResult;
REGISTERED AS {es200653Action 16};
```

9.5.17 Setup sub-network connection

PROFILE NOTE: There are five basic forms of multipoint connection- point-to-point, point-to-multipoint, multicast, broadcast and conference.

This action may be used to set up any of the first three types; the setup action for broadcast and conference Multipoint Connections requires further study. The setup is effected by creation of a point-to-point, point-to-multipoint, or multicast subnetwork connection. This is described in annex D.

An alternative approach, following ITU-T Recommendation I.326 [16] using point-to-point subnetwork connections and a multipoint root is described in annex E. If the approach of annex E is used this action may only be used to set up point-to-point subnetwork connections. The setupMultipointConnection action is used for the other modesin this case.

Timeout and holdtime are defined as INTEGER time intervals. It is the responsibility of application groups to determine what the unit of time interval is (e.g. milliseconds, seconds).

Where the subnetworkConnection is setup between accessGroups and/or topological points, the directionality is specified from the ConnectivityDirectionality defined in the SetupSubnetworkConnectionInformation.

```
setupSubNetworkConnection ACTION
    BEHAVIOUR
        setupSubNetworkConnectionBehaviour BEHAVIOUR
                DEFINED AS "This action is used to set up a Sub-network Connection between
                network termination points or network GTPs. The termination points to be
                connected can be specified in one of two ways:
                (1)by explicitly identifying the network termination points or NWGTPs,
                (2)by specifying one or more Topological Points or Access Groups from which any
                idle network termination point or NWGTP may be used.
                The result, if successful, always returns an explicit list of NWTPs or NWGTPs.
                A sub-network connection may be established in any of the following Status
                Conditions:

                -planned (1)
                -in service, not allocated (2)
                -in service, reserved (4-in service with no spare capacity (8
                -in service with no spare capacity, under test (9).

                Status Condition (8) is the default. Other Status Conditions shall be explicitly
                expressed in set-up sub-network connection action.

                If it is set up as In Service Reserved, this permits all resources involved in the
                Sub-network Connection to be reserved in sequence, and when all have been reserved
                the entire Sub-network Connection may be activated by invoking
                the action ActivateSubNetworkConnection.The Status condition of all network
                termination points, Link connections and Sub-network Connections involved in the
                Sub-network Connection being set up will be the same as that of the composite Sub-
                network Connection.

                A single Sub-network Connection object will be created if any of
                ptoPUnidirectional, ptoPBidirectional, ptoMultipointUni or ptoMultipointBidir modes
                are selected in this action. The Sub-network Connection object will have one A End
                and one or more Z Ends.

                For a point-to-point subnetwork connection, the z end is indicated by the
                zEndNWTPList. For a point-to-multipoint subnetwork connection, the zEndNWTPList is
                NULL, and the zEnds are indicated by the ZEndNWTP pointer of the leg.

                One Leg object will be created for each Z End in a point to multipoint Sub-network
                Connection. The Sub-network Connection object points to the NWTPs or NWGTPs
                involved in the Sub-network Connection. The subNetworkConnectionPointer in the
                NWTPs or NWGTPs points to the Sub-network
                Connection object.

                If a Topological Point is involved in the Sub-network Connection, its attributes
                idleNWCTPCount, and connectedNWCTPCount will be updated as a result of this action.

                This action will fail if any of the network termination points specified is already
                involved in a Sub-network Connection or if a NWTP which is part of an existing
                NWGTP is specified.
```

```
                The Sub-network Connection will have a directionality (unidirectional or bi-
                directional) as specified in the action parameter sncDirectionality. The
                sncDirectionality parameter also specifies the end points of the Sub-network
                Connection.

                If any of the underlying resources supporting the Sub-network Connection have a
                Status condition of Resource Failed with no spare capacity (10c) or Resource
                Failed, Reserved (10a), the Sub-network Connection shall have the same Status
                condition.

                If the Sub-network Connection parameters cannot be met by the server, the action
                response will indicate where possible, these parameters, and the values which can
                be actually be achieved by the server.

                If used, the quality of connectivity service specifies one pre-determined set of
                transport parameters which the server may offer. Where a particular quality of
                transport service level is not available from the server, the action response will
                indicate the next lowest level in the pre-defined set of levels which is possible.

                The optional timeout and holdtime parameters are used as part of a two-phase
                set-up process.

                Timeout is the time allowed to the agent sub-network to respond to the set-up
                request from the manager. This avoids the manager being slowed down by waiting for
                unacceptable periods of time for an agent response.

                Holdtime is the time interval which the agent sub-network waits for an activate
                ACTION once it has entered the reserved state. This allows the agent to free
                resources if the manager is slow to complete the two phase process.

                If they are used, transactionId and the identifier of the client will be passed to
                the server and will be logged by the server against the identifier of the created
                Sub-network Connection.

                When a bandwidth-scheduled sub-network connection is requested, the bandwidth
                scheduling parameter is used. The sub-network, will create a subNetworkConnection
                object instance. That object will have instantiated the package associated for the
                type of scheduling requested (e.g. weeklySchedulePkg if it requested for a weekly
                scheduled connection). That package will contain the schedule itself and the
                appropriate actions to modify the bandwidth schedule (add, delete and modify time
                slots) without the need of clearing down the connection and re-establishing the
                sub-network connection

                 StartTime          Condition
                 NULL               duration schedule is only vaild CHOICE (i.e.set-up is
                                    immediate and has no defined end)
                 NULL               reservation period begins immediately, and terminates at
                                    StopTime
                 GeneralizedTime    reservation period begins at StartTimeand has no defined end

                The sub-network shall guarantee that resources will be available when the
                sub-network connection is due to be activated.

                The action replies for set-up includes full information about the reasons in case
                the request could not be satisfied (lack of resources, overlapping time slots,
                etc.).

                The "in traffic" condition of the subNetworkConnection is driven by the schedule.
                A scheduled connection is set-up in the In Service, Not allocated (4) Status
                Condition. When the schedule indicates that the sub-network connection is to be put
                in traffic, the Status Condition changes to In Service with no spare capacity
                (8) (preceded by the In Service with no spare capacity, under test (9) Status
                Condition if an initial test is made).

                In a two-phase set-up comprising reservation and activation, the he sub-network
                connection is set-up in the In Service, Reserved (4) Status Condition at the time
                dictated by the schedule, pending an Activate Action from the manager.

                The default value of the implicit creation of TPs parameter is FALSE. That is, by
                default, the sub-network requires NWTPs to be in existence before a sub-network
                connection can be made. Only if the implicit creation parameter is set to be TRUE
                in the set-up sub-network connection request, will implicit NWTP creation occur.
                The identities of the created NWTPs are returned in the result.

                The EndPno parameter is used when it is necessary to specify a destination PNO when
                a step-by-step set-up process is used for inter TMN applications."
                ;;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX     ES200653.SetupSubNetworkConnectionInformation;
    WITH REPLY SYNTAX           ES200653.SetupSubNetworkConnectionResult;
REGISTERED AS {es200653Action 17};
```

9.5.18 Setup trail

```
setupTrail ACTION
    BEHAVIOUR
        setupTrailBehaviour BEHAVIOUR
                DEFINED AS "This action is used to set up a Trail between network trail
                termination points or network GTPs. The trail termination points to be connected
                can be specified in one of three ways: (1) by explicitly identifying the network
                trail termination points or NWGTPs, (2) by specifying one or more Access Groups
                from which any idle network trail termination point or NWGTP may be usedThe result,
                if successful, always returns an
                explicit list of NWTTPs or NWGTPs.The Trail is set up with the service state. In
                Service with no spare capacity.A single Trail object will be created if any
                ofptoPUnidirectional, ptoPBidirectional, ptoMultipointUni or ptoMultipointBidir
                modes are selected in this action. The Trail object will have one A End and one or
                more Z Ends. The Trail object points to the NWTPs or NWGTPs involved in the Trail.
                The connectivityPointer in the NWTTPs points to the Trail object.
                This action will fail if any of the network termination points specified is
                already involved in a Trail or if a NWTTP which is part of an existing NWGTP is
                specified. The Trail will have a directionality (unidirectional or bi-directional)
                as specified in the action parameter directionality. The identifier of the client
                will be passed to the server and will be logged by the server against the
                identifier of the created Trail."
                ;;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX     ES200653.SetupTrailInformation;
    WITH REPLY SYNTAX           ES200653.SetupTrailResult;
REGISTERED AS {es200653Action 18};
```

9.6 ASN.1 Syntax

```
ES200653 {ccitt(0) identified-organization(4) etsi(0) ets(653) informationModel(0) asn1Module(2) es200653(0)}

DEFINITIONS IMPLICIT TAGS ::= BEGIN

--EXPORTS everything


IMPORTS
AdditionalInformation, AdministrativeState, AvailabilityStatus, OperationalState FROM Attribute-ASN1Module{joint-iso-ccitt ms(9) smi (3) part2
(2) asn1Module(2) 1}
Bundle, CharacteristicInformation, Directionality, NameType, UserLabel, LogicalProblem, ResourceProblem, ProblemCause, ObjectList,
RelatedObjectInstance FROM ASN1DefinedTypesModule {ccitt(0) recommendation(0)
m(13) gnm(3100) informationModel(0) asn1Modules(2) asn1DefinedTypesModule(0)}
ObjectInstance FROM CMIP-1 {joint-iso-ccitt ms(9) cmip(1) modules(0) protocol(3)}
DistinguishedName FROM InformationFramework {joint-iso-ccitt ds(5) modules(1)
informationFramework(1)}
StopTime, Time24 FROM Attribute-ASN1Module {joint-iso-ccitt ms (9) smi (3) part2 (2)
    asn1Module (2) 1}

TrafficDescriptor FROM ASN1TypeModule {ccitt (0) administration (2) etsi (0) ets (469) informationModel (0) asn1Module (2) asn1TypesModule
(0)}
;

gomNLVClassLibrary OBJECT IDENTIFIER ::= {ccitt(0) identified-organization(4) etsi(0) ets(653) informationModel(0)}

es200653MObjectClass OBJECT IDENTIFIER ::= {gomNLVClassLibrary managedObjectClass(3)}

es200653Attribute OBJECT IDENTIFIER ::= {gomNLVClassLibrary attribute(7)}

es200653NameBinding OBJECT IDENTIFIER ::= {gomNLVClassLibrary nameBinding(6)}

es200653Package OBJECT IDENTIFIER ::= {gomNLVClassLibrary package(4)}

es200653Action OBJECT IDENTIFIER ::= {gomNLVClassLibrary action(9)}

es200653Notification OBJECT IDENTIFIER ::= {gomNLVClassLibrary notification(10)}

ActivateSubNetworkConnectionInformation ::= SEQUENCE {
    snc                 ObjectInstance,
    transactionIdTransactionId      OPTIONAL,
    userId          UserId          OPTIONAL
}
```

```
ActivateSubNetworkConnectionResult ::= CHOICE {
    failed                  [0] EXPLICIT Failed,
    sncActivated            [1] SEQUENCE {
                                snc  [0] ObjectInstance,
                        transactionId[1] TransactionIdOPTIONAL
}
}


AddNWCTPsToTopologicalPtInformation ::= SEQUENCE OF SEQUENCE {
    nwCTPs                      SET OF ObjectInstance,
    topologicalPoint            ObjectInstance OPTIONAL
}


AddNWCTPsToTopologicalPtResult ::= SEQUENCE OF CHOICE {
    failed                      [0] EXPLICIT Failed,
    addedNWCTPs                 [1] SEQUENCE {
                                topologicalPoint  ObjectInstance,
                                addedNWCTPs         SET OF ObjectInstance}
}
-- the nth element of the "SEQUENCE OF" is related to the nth element in the "SEQUENCE OF"
--in the "addNWCTPsToTopologicalPtInformation" type.

AddNWTPsToNWGTPInformation ::= SEQUENCE OF SEQUENCE {
    nwTPs                       SET OF ObjectInstance,
    nwGTP                       ObjectInstance OPTIONAL
}


AddNWTPsToNWGTPResult ::= SEQUENCE OF CHOICE {
    failed                      [0] EXPLICIT Failed,
    addedNWTPs                  [1] SEQUENCE {
                                nwGTP           ObjectInstance,
                                addedNWTPs          SET OF ObjectInstance}
}
-- the nth element of the "SEQUENCE OF" is related to the nth element in the "SEQUENCE OF"
--in the "addNWTPsToNWGTP"Information" type.

AddNWTTPsToAccessGroupInformation ::= SEQUENCE OF SEQUENCE {
    nwTTPs                      SET OF ObjectInstance,
    accessGroup                 ObjectInstance OPTIONAL
}


AddNWTTPsToAccessGroupResult ::= SEQUENCE OF CHOICE {
    failed                      [0] EXPLICIT Failed,
    addedNWTTPs                 [1] SEQUENCE {
                                accessGroup ObjectInstance,
                                addedNWTTPs     SET OF ObjectInstance}
}
-- the nth element of the "SEQUENCE OF" is related to the nth element in the "SEQUENCE OF"
--in the "addNWTTPsToAccessGroup" type.


Address ::= GraphicString

--the length of this string should be limited in application specific definitions

AddToSubNetworkConnectionInformation ::= SEQUENCE {
    implicitTPCreation      BOOLEAN,
    nWTP                    CHOICE {

    aEnds                   [0] SET OF ConnectivityEndPoint,

    zEnds                   [1] SET OF ConnectivityEndPoint,

    aAndZEndNWTPs               [2] SEQUENCE OF SET OF ConnectivityEndPoint},
    existingsubNetworkConnection        ObjectInstance
}


AddToSubNetworkConnectionResult ::= CHOICE {
    failed                      [0] EXPLICIT Failed,
    success                     [1] PtoMpSNCSetupResult
```

```
}


AssignmentState ::= ENUMERATED{
    free                    (0),
    reserved                (1),
    partiallyAssigned       (2),
    assigned                (3)}


BandwidthScheduling ::= SEQUENCE {
        startTime StartTime ,
        stopTime StopTime ,
        CHOICE {
        durationSchedule    [0] BidirectionalTrafficDescriptor ,
        dailySchedule   [1] DailySchedule ,
        weeklySchedule  [2] WeeklySchedule ,
        occasionalSchedule  [3] OccasionalSchedule,
        monthlySchedule     [4] MonthlySchedule }}

BidirectionalTrafficDescriptor ::= SEQUENCE {
        aToZ                    TrafficDescriptor ,
        zToA                    TrafficDescriptor}


Broadcast ::= ConnectivityEndPoint
-- single A end, no Z ends known

ChangeDailySchedulingInfo ::= SEQUENCE {
        changeSchedule              DailyScheduleModification OPTIONAL ,
        changeReservationBegin          [10] StartTime OPTIONAL,
        changeReservationEnd            [11] StopTime OPTIONAL}


ChangeDailySchedulingProblem ::= CHOICE {
        problemMultipoint               [1] ChangeMpDailySchedulingProblem ,
        oldNewScheduleTypeMismatch          [10] NULL ,
        insufficientBandwidthAtTheServer    [20] InsufficientBWAtTheServer ,
        networkProblem                  [30] ProblemCause ,
        numberOfSlotsTooLarge           [40] INTEGER ,
        slotDurationTooSmall            [41] Minutes ,
        overlappingDaySlots             [42] OverlappingDaySlots ,
        invalidDaySlot                  [46] DaySlot,
        beginEndTimeInconsistency       [49] NULL ,
        invalidReservationBegin         [50] StartTime ,
        invalidReservationEnd       [51] StopTime ,
        invalidScheduling           [52] NULL}

ChangeDailySchedulingResult ::= CHOICE {
        success  [0] NULL ,
        problem [1] ChangeDailySchedulingProblem,
        generalFailure                  [2] NULL}

ChangeDaySlot ::= SEQUENCE {
        slotId Time24 ,
        newSlot DaySlot}

ChangeDurationSchedulingInfo ::= BidirectionalTrafficDescriptor


ChangeDurationSchedulingProblem ::= CHOICE{
        resultMultipoint        [2] ChangeMpDurationSchedulingProblem,
        insufficientBandwidthAtTheServer    [20] InsufficientBWAtTheServer ,
        networkProblem                  [30] ProblemCause ,
        invalidDurationBw       [45] StartTime ,
        invalidScheduling               [52] NULL}

ChangeDurationSchedulingResult ::= CHOICE {
        success                 [0] NULL ,
        problem                 [1] ChangeDurationSchedulingProblem,
        generalFailure              [2] NULL}
```

```
ChangeMonthlySchedulingInfo ::= SEQUENCE {
        changeSchedule                  MonthlyScheduleModification OPTIONAL ,
        changeReservationBegin              [10] StartTime OPTIONAL,
        changeReservationEnd                [11] StopTime OPTIONAL}

ChangeMonthlySchedulingProblem ::= CHOICE {
        problemMultipoint                   [1] ChangeMpMonthlySchedulingProblem ,
        oldNewScheduleTypeMismatch              [10] NULL ,
        insufficientBandwidthAtTheServer            [20] InsufficientBWAtTheServer ,
        networkProblem                  [30] ProblemCause ,
        numberOfSlotsTooLarge               [40] INTEGER ,
        slotDurationTooSmall                [41] Minutes ,
        overlappingMonthSlots               [42] OverlappingMonthSlots ,
        invalidMonthSlot                    [46] DaySlot,
        beginEndTimeInconsistency               [49] NULL ,
        invalidReservationBegin             [50] StartTime ,
        invalidReservationEnd               [51] StopTime ,
        invalidScheduling                   [52] NULL}

ChangeMonthlySchedulingResult ::= CHOICE {
        success                 [0] NULL,
        problem                 [1] ChangeMonthlySchedulingProblem,
        generalFailure              [2] NULL}

ChangeMonthSlot ::= SEQUENCE {
        slotId TimeMonth ,
        newSlot MonthSlot}

ChangeMpDailySchedulingProblem ::= SEQUENCE {
        newScheduling DailyScheduling ,
        conflictingLegs SET OF LegChangeSlotProblem}

ChangeMpDurationSchedulingProblem ::= SEQUENCE {
        newScheduling BidirectionalTrafficDescriptor,
        conflictingLegs SET OF LegChangeSlotProblem}

ChangeMpMonthlySchedulingProblem ::= SEQUENCE {
        newScheduling MonthlyScheduling ,
        conflictingLegs SET OF LegChangeSlotProblem}

ChangeMpOccasionalSchedulingProblem ::= SEQUENCE {
        newScheduling OccasionalScheduling ,
        conflictingLegs SET OF LegChangeSlotProblem}

ChangeMpWeeklySchedulingProblem ::= SEQUENCE {
        newScheduling WeeklyScheduling,
        conflictingLegs SET OF LegChangeSlotProblem}

ChangeOccasionalSchedulingInfo ::= SEQUENCE {
        changeSchedule OccasionalScheduleModification OPTIONAL ,
        changeReservationBegin [10] StartTime OPTIONAL,
        changeReservationEnd [11] StopTime OPTIONAL}

ChangeOccasionalSchedulingProblem ::= CHOICE {
        problemMultipoint                   [1] ChangeMpOccasionalSchedulingProblem ,
        insufficientBandwidthAtTheServer            [20] InsufficientBWAtTheServer ,
        networkProblem                  [30] ProblemCause ,
        numberOfSlotsTooLarge               [40] INTEGER ,
        slotDurationTooSmall                [41] Minutes ,
        overlappingOccasionalSlots              [42] OverlappingOccasionalSlots ,
        invalidOccasionalSlot               [48] OccasionalSlot,
        beginEndTimeInconsistency               [49] NULL ,
        invalidReservationBegin             [50] StartTime ,
        invalidReservationEnd               [51] StopTime ,
        invalidScheduling                   [52] NULL}

ChangeOccasionalSchedulingResult ::= CHOICE {
        success                 [0] NULL ,
        problem                 [1] ChangeOccasionalSchedulingProblem,
        generalFailure              [2] NULL}

ChangeOccasionalSlot ::= SEQUENCE {
        slotId StartTime ,
        newSlot OccasionalSlot}
```

```
ChangeWeeklySchedulingInfo ::= SEQUENCE {
        changeSchedule WeeklyScheduleModification OPTIONAL ,
        changeReservationBegin          [10] StartTime OPTIONAL,
        changeReservationEnd            [11] StopTime OPTIONAL}

ChangeWeeklySchedulingProblem ::= CHOICE {
        problemMultipoint               [1] ChangeMpWeeklySchedulingProblem ,
        insufficientBandwidthAtTheServer    [20] InsufficientBWAtTheServer ,
        networkProblem              [30] ProblemCause ,
        numberOfSlotsTooLarge           [40] INTEGER ,
        slotDurationTooSmall            [41] Minutes ,
        overlappingWeekSlots            [43] OverlappingWeekSlots ,
        invalidWeekSlot             [47] WeekSlot,
        beginEndTimeInconsistency           [49] NULL ,
        invalidReservationBegin         [50] StartTime ,
        invalidReservationEnd           [51] StopTime ,
        invalidScheduling               [52] NULL}

ChangeWeeklySchedulingResult ::= CHOICE {
        success                     [0] NULL,
        problem                     [1] ChangeWeeklySchedulingProblem,
        generalFailure                  [2] NULL}

ChangeWeekSlot ::= SEQUENCE {
        slotId TimeWeek ,
        newSlot WeekSlot}


ClientPtr::= ObjectInstance

ComponentPointers ::= SET OF ObjectInstance

CompositePointer ::= RelatedObjectInstance

Conference ::= SET OF ConnectivityEndPoint
-- all A ends, no Z ends


ConnectionList ::= SET OF ObjectInstance

ConnectivityDirectionality ::= CHOICE {
    ptoPUnidirectional              [0] PtoPoint,
    ptoPBidirectional               [1] PtoPoint,
    ptoMultipointUni                [2] PtoMultipoint,
    ptoMultipointBidir              [3] PtoMultipoint,
    multicastUni            [4] Multicast,
    multicastBidir              [5] Multicast,
    broadcastUni                [6] Broadcast,
    broadcastBi             [7] Broadcast,
    conference              [8] Conference
}


ConnectivityEndPoint ::= CHOICE {
    none                    [0] NULL,
    sncTp                   [1] ObjectInstance,
    topologicalPoint            [2] ObjectInstance,
    accessGroup             [3] ObjectInstance
}
-- This allows a network termination point or GTP to be chosen explicitly (using the sncTPchoice)
--or a Topological Point or Access Group may be selected, and hence any idle NWTP within them.

ConnectivityPointer ::= RelatedObjectInstance

Count ::= INTEGER

DailySchedule ::= SEQUENCE OF DaySlot

DailyScheduleModification ::= SET OF DaySlotModification

DailyScheduling ::= SEQUENCE {
        reservationBegin StartTime ,
        reservationEnd      StopTime ,
        schedule        DailySchedule}
```

```
DaySlot ::= SEQUENCE {
        slotBegin Time24 ,
        slotEnd Time24 ,
        bandwidth BidirectionalTrafficDescriptor}

DaySlotModification::= CHOICE {
        deleteSlot [0] Time24 ,
        createSlot [1] DaySlot ,
        changeSlot [2] ChangeDaySlot}


DeletedLeg ::= SEQUENCE {
        legId NameType ,
        zEnd ObjectInstance}

DeleteFromSubNetworkConnectionInformation ::= SEQUENCE {
    nWTPs       CHOICE {
                    aEnds   [0] SET OF ConnectivityEndPoint,
                    zEnds   [1] SET OF ConnectivityEndPoint,
                    aAndZEndNWTPs   [2] SEQUENCE OF SET OF ConnectivityEndPoint},
    existingSubNetworkConnection        ObjectInstance
}


DeleteFromSubNetworkConnectionResult ::= CHOICE {
        legsDeleted             [0] DeleteLegsResult ,
        multipointConnectionDeleted     [1] DeleteLegsResult }


DeleteLegProblem ::= CHOICE {
        noSuchTp                [0] ObjectInstance ,
        connectionTpMismatch            [1] ObjectInstance}


DeleteLegsResult ::= SEQUENCE {
        multipointConnection            ObjectInstance ,
        aEnd                    ObjectInstance ,
        deletedLegs             SET OF DeletedLeg ,
        failures                SET OF DeleteLegProblem}



EndPNOs ::= SEQUENCE{
nearEndPnoSubnetworkIdGraphicString OPTIONAL,
CHOICE {
    farEndPnoSubnetworkId   [0]GraphicString,
    destinationAddress      [1]Address}OPTIONAL
}


Failed ::= CHOICE {
    logicalProblem                  [1] EXPLICIT LogicalProblem,
    resourceProblem                 [2] EXPLICIT ResourceProblem,
    noSuchConnection                    [10] ObjectInstance
}


Format ::= OBJECT IDENTIFIER

Holdtime ::=INTEGER

Implicit ::= BOOLEAN (TRUE)

InsufficientBWAtTheServer ::= SEQUENCE {
        serverTTP   ObjectInstance ,
        conflictingSlot   SET OF SlotId OPTIONAL}

LayerConnectionList ::= Tree


LegChangeSlotProblem ::= SEQUENCE {
        legId NameType ,
        slotId SlotId OPTIONAL}
```

```
LegDescription ::= SEQUENCE {
        legId           NameType ,
        zEnd            ObjectInstance ,
        statusCondition     SetupStatus,
        slotProblems    SET OF SlotId OPTIONAL}


LegResult ::= CHOICE {
        success                 [0] LegDescription ,
        failure                 [1] LegSetupProblem}

LegSetupProblem ::= CHOICE {
        noSuchSncTp                 [10] NULL ,
        noSuchServerTTP                 [11] NULL ,
        sncTpAlreadyConnected               [12] NULL ,
        noMoreAvailableTpInServerTTP            [13] NULL ,
        invalidSncTpParameter               [14] NULL ,
        networkProblem              [30] NULL}


LegSetupResult ::= SEQUENCE {
        sncTP ConnectivityEndPoint ,
        legResult LegResult}

LifecycleState ::= ENUMERATED{
    planned                     (0),
    inService                       (1),
    decommissioned                  (2)}


LinkList ::= SET OF ObjectInstance

LinkPointerList ::= SET OF ObjectInstance

Minutes ::= INTEGER

Mode ::= ENUMERATED {
    pointToPoint                        (0),
    pointToMultipoint                       (1),
    multicast                       (2),
    broadcast                   (3),
    conference                  (4)
}


MonthDay ::= INTEGER (1..31)

MonthlySchedule ::= SEQUENCE OF MonthSlot

MonthlyScheduleModification ::= SET OF MonthSlotModification

MonthlyScheduling ::= SEQUENCE {
        reservationBegin StartTime ,
        reservationEnd StopTime ,
        schedule MonthlySchedule}

MonthSlot ::= SEQUENCE {
        slotBegin TimeMonth ,
        slotEnd TimeMonth ,
        bandwidth BidirectionalTrafficDescriptor}

MonthSlotModification::= CHOICE {
        deleteSlot                  [0] TimeMonth ,
        createSlot                  [1] MonthSlot ,
        changeSlot                  [2] ChangeMonthSlot}



Multicast ::= SEQUENCE {
    aEnds       SET OF ConnectivityEndPoint,
    zEnds       SET OF ConnectivityEndPoint
}
-- multiple A ends, multiple Z ends
```

```
NWCTPList ::= SET OF ObjectInstance

NWTTPList ::= SET OF ObjectInstance


OccasionalSchedule ::= SEQUENCE OF OccasionalSlot

OccasionalScheduleModification ::= SET OF OccasionalSlotModification

OccasionalScheduling ::= SEQUENCE {
        reservationBegin StartTime ,
        reservationEnd StopTime ,
        schedule OccasionalSchedule}

OccasionalSlot ::= SEQUENCE {
        slotBegin StartTime ,
        slotEnd StopTime ,
        bandwidth BidirectionalTrafficDescriptor}

OccasionalSlotModification::= CHOICE {
        deleteSlot                  [0] StartTime ,
        createSlot                  [1] OccasionalSlot ,
        changeSlot                  [2] ChangeOccasionalSlot}

OverlappingDaySlots ::= SEQUENCE {
        slot1   DaySlot,
        slot2   DaySlot}

OverlappingMonthSlots ::= SEQUENCE {
        slot1   MonthSlot,
        slot2   MonthSlot}

OverlappingOccasionalSlots ::= SEQUENCE {
        slot1   OccasionalSlot,
        slot2   OccasionalSlot}

OverlappingWeekSlots ::= SEQUENCE {
        slot1   WeekSlot,
        slot2   WeekSlot}

--ProblemCause is imported from ITU-T Recommendation M.3100
-- The following values are used for integerValue of ProblemCause:
-- noSuchTPInstance             0
-- noSuchTopologicalPtInstance          1
-- noSuchAccessGroupInstance            2
-- noSuchSNCInstance            3
-- noNWCTPInTopologicalPoint            4
-- noNWTTPInAccessGroup                 5
-- nwCTPAlreadyInTopologicalPoint               6
-- nwTTPAlreadyInAccessGroup            7
-- sncAlreadyInSNC              8


PtoMpSNCReleaseResult ::= SEQUENCE {
        sNConnection ObjectInstance ,
        aEnd ObjectInstance OPTIONAL,
        zEnds SET OF ObjectInstance OPTIONAL}

PtoMpSNCSetupResult ::= SEQUENCE {
        sNConnection ObjectInstance ,
        legs SET OF LegSetupResult}

PtoMultipoint ::= SEQUENCE {
    aEnd            ConnectivityEndPoint,
    zEnds       SET OF ConnectivityEndPoint
}
-- single A end, multiple Z ends

PtoPoint ::= SEQUENCE {
    aEnd            ConnectivityEndPoint,
    zEnd            ConnectivityEndPoint
}
-- single A and Z ends
```

```
PtoPSNCReleaseResult ::= SEQUENCE {
        connection      ObjectInstance ,
        aEnd        [0] ObjectInstance OPTIONAL,
        zEnd        [1] ObjectInstance OPTIONAL}


PtoPSNCSetupResult ::= SEQUENCE {
        connection ObjectInstance ,
        aEnd ObjectInstance ,
        zEnd ObjectInstance
}



QofConnectivityService ::=ObjectInstance

ReleaseSubNetworkConnectionInformation ::= SEQUENCE {
        snc     ObjectInstance,
        userId      UserId      OPTIONAL
}


ReleaseSubNetworkConnectionResult ::= CHOICE {
        failure                 [0] EXPLICIT Failed,
        pointToPointResult              [1] PtoPSNCReleaseResult,
        multipointResult            [2] PtoMpSNCReleaseResult
}



ReleaseTrailInformation ::= SEQUENCE {
    trailId             ObjectInstance,
    userId          UserId  OPTIONAL
}




ReleaseTrailResult ::= CHOICE {
    unknown             NULL,
    integerValue        INTEGER
}
-- The following values are used for integerValue of releaseTrailResult:
-- The trail has been released                          0
-- The trail has not been released                      1
-- The identified trail was not recognized                  2
-- The service user which issued the release trail request is not authorized to do so   3
-- The user Id was not recognized                       4


RemoveNWTTPsFromAccessGroupInformation ::= SEQUENCE OF SEQUENCE {
    nWTTPs              SET OF ObjectInstance,
    accessGroup             ObjectInstance
}


RemoveNWTTPsFromAccessGroupResult ::= SEQUENCE OF CHOICE {
    failed                      [0] EXPLICIT Failed,
    removedNWTTPs                   [1] SEQUENCE {
                                accessGroup     ObjectInstance,
                                removedNWTTPs   SET OF ObjectInstance}
}
-- the nth element of the "SEQUENCE OF" is related to the nth element in the "SEQUENCE
-- OF" in the "removeNWTTPsFromAccessGroup" type.

RemoveNWTPsFromNWGTPInformation ::= SEQUENCE OF SEQUENCE {
    nwTPs               SET OF ObjectInstance,
    nwGTP               ObjectInstance
}
```

```
RemoveNWTPsFromNWGTPResult ::= SEQUENCE OF CHOICE {
    failed                      [0] EXPLICIT Failed,
    removedNWTPs                    [1] SEQUENCE {
                            nwGTP       ObjectInstance,
                            removedNWTPs  SET OF ObjectInstance}
}
-- the nth element of the "SEQUENCE OF" is related to the nth element in the "SEQUENCE OF"
--in the " RemoveNWTPsFromNWGTPInformation " type.



RemoveNWCTPsFromTopologicalPtInformation ::= SEQUENCE OF SEQUENCE {
    nWCTPs                      SET OF ObjectInstance,
    topologicalPoint        ObjectInstance
}


RemoveNWCTPsFromTopologicalPtResult ::= SEQUENCE OF CHOICE {
    failed                      [0] EXPLICIT Failed,
    removedNWCTPs                   [1] SEQUENCE {
                            topologicalPoint  ObjectInstance,
                            removedNWCTPs     SET OF ObjectInstance}
}
-- the nth element of the "SEQUENCE OF" is related to the nth element in the "SEQUENCE
--OF" in the "removeNWCTPsFromTopologicalPtInformation" type.



--ResourceProblem is imported from ITU-T Recommendation M.3100
--The semantics for each integer valueis defined by the application.

SetupStatus  ::= SET {
lifecycleState          [0] LifecycleState,
assignmentState         [1] AssignmentState,
availabilityStatus  [2]AvailabilityStatus       --see X.721
}

SetupSubNetworkConnectionInformation ::= SEQUENCE {
    sncDirectionality           ConnectivityDirectionality,
    statusCondition             [0] SetupStatus         OPTIONAL,
    signalid                [1] SignalId  OPTIONAL,
    qofConnectivityService          [2] QofConnectivityService    OPTIONAL,
    transactionId           [3] TransactionIdOPTIONAL,
    userId                  [4] UserId    OPTIONAL,
    timeout                 [5] Timeout  OPTIONAL ,
    holdtime                    [6] HoldtimeOPTIONAL,
    bandwidthScheduling         [7] BandwidthScheduling OPTIONAL,
    implicitTPCreation              [8] Implicit   OPTIONAL,
    endPNOs                     [9] EndPNOs     OPTIONAL
    }

SetupSubnetworkConnectionProblem ::= CHOICE {
    logicalProblem                  [0] EXPLICIT LogicalProblem,
    resourceProblem                 [1] EXPLICIT ResourceProblem,
    parameterProblem                    [2] SET OF ENUMERATED {
            sncDirectionalityRelatedFailure     (0),
            stateRelatedFailure             (1),
            signalidRelatedFailure          (2),
            qofServiceRelatedFailure        (3),
            transactionIdRelatedFailure         (4),
            senderRelatedFailure        (5)},
```

```
    noSuchSncTp                     [10] ObjectInstance ,
    noSuchServerTTP                 [11] ObjectInstance ,
    sncTpAlreadyConnected           [12] ObjectInstance ,
    noMoreAvailableTpInServerTTP    [13] ObjectInstance ,
    invalidSncTpParameter           [14] ObjectInstance ,
    insufficientBandwidthAtTheServer    [20] InsufficientBWAtTheServer ,
    networkProblem                  [30] ProblemCause ,
    noLegsSetup                 [31] LegSetupProblem,
    numberOfSlotsTooLarge           [40] INTEGER ,
    slotDurationTooSmall            [41] Minutes ,
    overlappingDaySlots             [42] OverlappingDaySlots ,
    overlappingMonthSlots           [43] OverlappingMonthSlots ,
    overlappingOccasionalSlots      [44] OverlappingOccasionalSlots ,
    overlappingWeekSlots            [45] OverlappingWeekSlots ,
    invalidDurationBW               [46] BidirectionalTrafficDescriptor ,
    invalidDaySlot                  [47] DaySlot ,
    invalidMonthSlot                [48] MonthSlot ,
    invalidOccasionalSlot           [49] OccasionalSlot ,
    invalidWeekSlot                 [50] WeekSlot ,
    beginEndTimeInconsistency       [51] NULL ,
    invalidReservationBegin         [52] StartTime ,
    invalidReservation              [53] StopTime ,
    invalidScheduling               [54] NULL
}

-- a logical problem indicates for example that an object instance was specified which does not
--exist
-- a resource problem - these need to be defined
-- a parameter problem indicates that one of the parameters requested in the setup request
--was not available, or that the failure is related to that parameter.


SetupSubNetworkConnectionResult ::= SEQUENCE{
        transactionId       TransactionId       OPTIONAL,
        CHOICE {
        pointToPointResult          [1] PtoPSNCSetupResult ,
        multipointResult        [2] PtoMpSNCSetupResult ,
        generalFailure          [3] NULL,
        problem             [4] SetupSubnetworkConnectionProblem}
        }

SetupTrailInformation ::= SEQUENCE {
    trailEndPoints          ConnectivityDirectionality,
    userId              [0] UserId          OPTIONAL,
    userLabel           [1] UserLabel       OPTIONAL,
    additionalInformation   [2] AdditionalInformation OPTIONAL
}

SetupTrailResult ::= SEQUENCE {
    setupTrailResultCode    SetupTrailResultCode,
    trailId                 ObjectInstance,
    aEnds                   SET OF ObjectInstance,
    zEnds                   SET OF ObjectInstance
}

SetupTrailResultCode ::= CHOICE {
    unknown             NULL ,
    integerValue    INTEGER
}
-- The following values are used for integerValue of SetupTrailResultCode :
-- Trail setup successful                                   0
-- End point identifiers Parameter value error                  1
-- (requested end point identifiers not recognized)
-- End point identifiers Parameter value error                  2
-- (requested end points not available)
-- Directionality Parameter value error
-- (requested directionality not supported)                 3
-- Mode Parameter value error - requested mode not supported        4
-- User identifier Parameter value error- requested User identifier not recognized      5
-- No route between the specified end-point identifiers can be found    6
```

```
SignalId ::= CHOICE {
    simple      [0] CharacteristicInformation,
    bundle      [1] Bundle,
    none        [2] NULL,
    complex     [3] SEQUENCE OF Bundle,
    extended        [4]SEQUENCE OF SEQUENCE{
                characteristicInformation  CharacteristicInformation,
                format          Format,
    variable    [5] BidirectionalTrafficDescriptor}
}
-- The use of signal Id is described in Clause B.1.8. For unidirectional variable
-- cases one of the traffic descriptors is NULL.

SignalList    ::= SET OF SignalId

SlotId ::= CHOICE {
        duration                [0] NULL ,
        daySlotId                   [1] Time24 ,
        weekSlotId              [2] TimeWeek ,
        monthSlotId             [3] TimeMonth,
        occasionalSlot              [4] StartTime}
StartTime ::= StopTime

--StartTime uses the same syntax as StopTime to allow for a Null value of the StartTime, for example where a set-up is immediately activated on
receipt of the setup request.

SubNetworkConnectionPointerList ::= SEQUENCE OF RelatedObjectInstance

SubNetworkList ::= SET OF ObjectInstance

Subtree ::= CHOICE {
    singleConnectivityInstance          [0] ObjectInstance,
    multicast                   [1] SET OF Subtree
}
TimeMonth ::= SEQUENCE {
                monthDay MonthDay ,
                time Time24}

Timeout ::= INTEGER

TimeWeek ::= SEQUENCE {
                weekDay WeekDay ,
                time Time24}

TPList ::= SET OF ObjectInstance


TrailList ::= SET OF ObjectInstance


TransactionId ::= SEQUENCE {
    localId                         [0] INTEGER,
    globalRef                           [1] CHOICE {
    dnGlobalRef         DistinguishedName,
    oidGlobalRef                        OBJECT IDENTIFIER} OPTIONAL
}


Tree ::= SET OF Subtree

TypeText    ::= GraphicString
--Note that the length of this string shall be limited in Technology specific applications.

UsageCost::=INTEGER(0..255)
```

```
UserId ::= GraphicString
--Note that the length of this string shall be limited in Technology specific applications.
WeekDay ::= ENUMERATED {
        sunday          (0) ,
        monday          (1) ,
        tuesday         (2) ,
        wednesday       (3) ,
        thursday        (4) ,
        friday          (5) ,
        saturday        (6)}

WeeklySchedule ::= SEQUENCE OF WeekSlot

WeeklyScheduleModification ::= SET OF WeekSlotModification

WeeklyScheduling ::= SEQUENCE {
        reservationBegin StartTime ,
        reservationEnd StopTime ,
        schedule WeeklySchedule}

WeekSlot ::= SEQUENCE {
        slotBegin TimeWeek ,
        slotEnd TimeWeek ,
        bandwidth BidirectionalTrafficDescriptor}

WeekSlotModification ::= CHOICE {
        deleteSlot                      [0] TimeWeek ,
        createSlot                      [1] WeekSlot ,
        changeSlot                      [2] ChangeWeekSlot}


END
```

Annex A (normative): Definition of status conditions for the network level view

To reflect the state of the information object a Status Condition is defined below. The Status Conditions are the requirements for the states which a service user OSF needs to see in the network resources of the service provider OSF. For example, if a user wishes to maintain a network resource the Maintenance Status Condition (14) is used. This is actually implemented as a particular combination of base states (as detailed below), but the particular implementation is not an issue for the Status Condition.

The Status Conditions refer to the state of network resource and how that resource is used. The states of the management system are not reflected in the states of the network resource. For example if the resource was no longer capable of performing new configuration requests, but still carried traffic normally, it would have a Status Condition of In Service with spare capacity, Degraded.

The Status Condition is not a state itself. It is composed of a set of allowed combinations of base states as shown in the table below. The base states are: the ISO Operational and Administrative states, the ISO Availability Status, the Assignment state, and the Lifecycle state. It is important to note that the base states are an implementation of the Status Condition requirements.

The set of Status Conditions is not prescriptive, nor is it exhaustive: a subset of the Status Conditions may be used by any particular application, and new Status Conditions may be added (with the appropriate mappings) as new requirements emerge.

The Status Condition reflects the combined state behaviour of the resource, as viewed by the managing applications. For this reason the behaviour of the managed object class is expressed in terms of the Status Conditions, and not the component states.

The Status Condition reflects the state of the resource at the instant it is accessed. It noes not contain any future or history data - these are part of the scheduling function.

If no scheduling function is present, the Status Condition may reflect the previous state of the system e.g. the states Resource Failed, Reserved; Resource Failed with spare capacity; Resource Failed, with no spare capacity.

The behaviour of the resources is defined in terms of the Status Condition, but the GDMO definition is in terms of the base states, and the mapping is given in this annex.

NOTE: All five component states are needed to define the complete range of Status Conditions, but that a subset of the Status Conditions may be defined by using a smaller number of component states.

Operational State denotes the ability of the resource to supply its normal service. In this context normal service is the ability to carry traffic. Failures in management capability will not result in a disabled state but may be expressed as a degraded value of the availability Status.

When the administrative state has the value "locked", the resource is not able to carry traffic.

The state transition diagrams for any particular subset of the Status Conditions, will be defined by applications.

A.1 Status condition values

A.1.1 Planned

A resource would take this state when it is planned for use and the underlying resources are not present.

This Status Condition is defined by the following component states:

Lifecycle State, Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.1.1 Under commission

A resource would take this state when the underlying resources are present and undergoing commissioning, or have not been configured.

This Status Condition is defined by the following component states:

Lifecycle State, Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.1.2 Planned and allocated for use

A resource would take this state when it is planned and reserved for use. The underlying resources, however, are not installed yet.

This Status Condition is defined by the following component states:

Lifecycle State, Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.1.3 Under commission and allocated for use

A resource would take this state when it is planned and reserved for use. The underlying resources are present and undergoing commissioning, or have not been configured.

This Status Condition is defined by the following component states:

Lifecycle State, Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.2 In service, not allocated

A resource would take this state when supporting resources have been installed and this resource has not yet been allocated for use.

NOTE: Installed means configured and/or commissioned.

This Status Condition is defined by the following component states:

Operational State, Assignment State.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.2.1 In service, not allocated, degraded

A resource would take this state when supporting resources have been installed and this resource has not yet been allocated for use. It is degraded in some respect e.g. it can still carry traffic unimpaired but can not offer full management capabilities.

NOTE: Applications shall specify the particular degradation expressed by this state in each case.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.3 In service, not allocated, under test

A resource would take this state when supporting resources have been installed and this resource has not yet been allocated for use. Additionally the resource is under test.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.3.1 In service, not allocated, under test, degraded

A resource would take this state when supporting resources have been installed and this resource has not yet been allocated for use. Additionally the resource is under test. It is degraded in some respect e.g. it can still carry traffic unimpaired but can not offer full management capabilities.

NOTE: Applications shall specify the particular degradation expressed by this state in each case.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.4 In service, reserved

A resource would take this state when supporting resources have been installed and the resource has been reserved for use. Another manager could not reserve this resource.

This Status Condition is defined by the following component states:

Operational State, Assignment State.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.4.1 In service, reserved, degraded

A resource would take this state when supporting resources have been installed and the resource has been reserved for use as part of a two phase assignment process. The holdtime and time-outs will be specified by applications Another manager could not reserve this resource. It is degraded in some respect e.g. it can still carry traffic unimpaired but can not offer full management capabilities.

NOTE: Applications shall specify the particular degradation expressed by this state in each case.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.5 In service, reserved, under test

A resource would take this state when supporting resources have been installed and the resource has been reserved for use as part of a two phase assignment process. The holdtime and time-outs will be specified by applications. Another manager could not reserve this resource. Additionally the resource is under test.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.5.1 In service, reserved, under test, degraded

A resource would take this state when supporting resources have been installed and the resource has been reserved for use as part of a two phase assignment process. The holdtime and time-outs will be specified by applications

Another manager could not reserve this resource. Additionally the resource is under test. It is degraded in some respect e.g. it can still carry traffic unimpaired but can not offer full management capabilities.

NOTE: Applications shall specify the particular degradation expressed by this state in each case.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.6 In service with spare capacity

A resource would take this state when supporting resources have been installed, the resource has been allocated for use and there is spare capacity on the resource.

This Status Condition is defined by the following component states:

Operational State, Assignment State.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.6.1 In service with spare capacity, degraded

A resource would take this state when supporting resources have been installed, the resource has been allocated for use and there is spare capacity on the resource. It is degraded in some respect e.g. it can still carry traffic unimpaired but can not offer full management capabilities.

NOTE: Applications shall specify the particular degradation expressed by this state in each case.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.7 In service with spare capacity, under test

A resource would take this state when supporting resources have been installed, the resource has been allocated for use and there is spare capacity on the resource. Additionally the resource is under test.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.7.1 In service with spare capacity, under test, degraded

A resource would take this state when supporting resources have been installed, the resource has been allocated for use and there is spare capacity on the resource. Additionally the resource is under test. It is degraded in some respect e.g. it can still carry traffic unimpaired but can not offer full management capabilities.

NOTE: Applications shall specify the particular degradation expressed by this state in each case.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.8 In service with no spare capacity

A resource would take this state when supporting resources have been installed, the resource has been allocated for use and there is no spare capacity on the resource.

This Status Condition is defined by the following component states:

Operational State, Assignment State.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.8.1 In service, with no spare capacity, degraded

A resource would take this state when supporting resources have been installed, the resource has been allocated for use and there is no spare capacity on the resource. It is degraded in some respect e.g. it can still carry traffic unimpaired but can not offer full management capabilities.

NOTE: Applications shall specify the particular degradation expressed by this state in each case.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.9 In service, with no spare capacity, under test

A resource would take this state when supporting resources have been installed, the resource has been allocated for use and there is no spare capacity on the resource. Additionally the resource is under test.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.9.1 In service with no spare capacity, under test, degraded

A resource would take this state when supporting resources have been installed, the resource has been allocated for use and there is no spare capacity on the resource. Additionally the resource is under test. It is degraded in some respect e.g. it can still carry traffic unimpaired but can not offer full management capabilities.

NOTE: Applications shall specify the particular degradation expressed by this state in each case.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.10 Resource failed

The resource takes on this state when it is in service but is no longer capable of providing it's normal function.

This Status Condition is defined by the following component states:

Operational State, Assignment State.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.10.1 Resource failed, reserved

The resource takes on this state when it is in service but is no longer capable of providing it's normal function, and has been reserved.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.10.2 Resource failed, with spare capacity

The resource takes on this state when it is in service but is no longer capable of providing it's normal function, and has been partially assigned.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.10.3 Resource failed, with no spare capacity

The resource takes on this state when it is in service but is no longer capable of providing it's normal function and has been assigned.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.11 Resource failed, under test

The resource takes on this state when it is in service but is no longer capable of providing it's normal function. Additionally the resource is under test.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.12 Shutting down, with spare capacity

A resource would take this state when it has been marked for removal from service. Usage is limited to current instances of use, and when all current users have terminated their use of the resource, the managed object will automatically transit to the Temporarily Out of Service Status Condition. When a resource is dependent on other resources which are in the shutting down state, it may only enter the shutting down state by explicit management action. It is degraded in a non-traffic affecting respect e.g. it can not offer full management capabilities.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Administrative State.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.12.1 Shutting down, with spare capacity, degraded

A resource would take this state when it has been marked for removal from service. Usage is limited to current instances of use, and when all current users have terminated their use of the resource, the managed object will automatically transit to the Temporarily Out of Service Status Condition. When a resource is dependent on other resources which are in the shutting down state, it may only enter the shutting down state by explicit management action. It is degraded in some respect e.g. it can still carry traffic unimpaired but can not offer full management capabilities.

NOTE: Applications shall specify the particular degradation expressed by this state in each case. This Status Condition is defined by the following component states:

Operational State, Assignment State, Administrative State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.13 Shutting down, with no spare capacity

A resource would take this state when it has been marked for removal from service. Usage is limited to current instances of use, and when all current users have terminated their use of the resource, the managed object will automatically transit to the Temporarily Out of Service Status Condition. When a resource is dependent on other resources which are in the shutting down state, it may only enter the shutting down state by explicit management action.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Administrative State.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.13.1 Shutting down, with no spare capacity, degraded

A resource would take this state when it has been marked for removal from service Usage is limited to current instances of use, and when all current users have terminated their use of the resource, the managed object will automatically transit to the Temporarily Out of Service Status Condition. When a resource is dependent on other resources which are in the shutting down state, it may only enter the shutting down state by explicit management action. It is degraded in some respect e.g. it can still carry traffic unimpaired but can not offer full management capabilities.

NOTE: Applications shall specify the particular degradation expressed by this state in each case.

This Status Condition is defined by the following component states:

Operational State, Assignment State, administrative State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.13.2 Shutting down, reserved

A resource would take this state when it has been marked for removal from service. Usage is limited to current instances of use, and when all current users have terminated their use of the resource, the managed object will automatically transit to the Temporarily Out of Service Status Condition. When a resource is dependent on other resources which are in the shutting down state, it may only enter the shutting down state by explicit management action.

Prior to entering this Status Condition, the resource had been reserved by the manager as part of a two-phase commit process.

NOTE: Applications shall specify the particular degradation expressed by this state in each case.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Administrative State.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.14 Maintenance

A resource would take this state when performing non-intrusive testing, for example. Additional users, and changes to the configuration are undesirable, but traffic shall still flow through the resource for the purposes of the test.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Administrative State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.14.1 Temporarily out of service, degraded

A resource would take this state when it has been taken out of traffic by being locked. The resource is still capable of providing service.

It is degraded in some respect e.g. it can still carry traffic unimpaired but can not offer full management capabilities.

NOTE: Applications shall specify the particular degradation expressed by this state in each case.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Administrative State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.15 Temporarily out of service under test

A resource would take this state when it has been taken out of traffic by being locked. The resource is still capable of providing service. Additionally the resource is under test.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Administrative State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.15.1 Temporarily out of service under test, degraded

A resource would take this state when it has been taken out of traffic by being locked. The resource is still capable of providing service.

Additionally the resource is under test. It is degraded in some respect e.g. it can still carry traffic unimpaired but can not offer full management capabilities.

NOTE: Applications shall specify the particular degradation expressed by this state in each case.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Administrative State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.15.2 Temporarily out of service

A resource would take this state when it has been taken out of traffic by being locked. The resource is still capable of providing service.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Administrative State.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.16 Resource faulty and temporarily out of service

A resource would take this state when it becomes incapable of performing its normal function.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Administrative State.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.17 Resource faulty and temporarily out of service, under test

A resource would take this state when it becomes incapable of performing its normal function. Additionally the resource is under test.

This Status Condition is defined by the following component states:

Operational State, Assignment State, Administrative State, Availability Status.

The values of these states and the default values of the other component states (if present) are given in table A.1.

A.1.18 Decommissioned

A resource would take this state when it is decommissioned.

This Status Condition is defined by the following component states:

Lifecycle State.

The values of these states and the default values of the other component states (if present) are given in table A.1.

Table A.1

Base State Status Condition	Lifecycle State			Operational state		Assignment state				Administrative state			Availability Status			
	P	IS	D	E	Dis	F	R	PA	A	U	L	SD	IT	D	NI	-
1	✔				✔	✔				D					✔	
1a	✔				✔	✔				D			✔			
1b	✔				✔		✔			D					✔	
1c	✔				✔		✔			D			✔			
2	D			✔		✔				D						D
2a	D			✔		✔				D				✔		
3	D			✔		✔				D			✔			
3a	D			✔		✔				D			✔	✔		
4	D			✔			✔			D						D
4a	D			✔			✔			D				✔		
5	D			✔			✔			D			✔			
5a	D			✔			✔			D			✔	✔		
6	D			✔				✔		D						D
6a	D			✔				✔		D				✔		
7	D			✔				✔		D			✔			
7a	D			✔				✔		D			✔	✔		
8	D			✔					✔	D						D
8a	D			✔					✔	D				✔		
9	D			✔					✔	D			✔			
9a	D			✔					✔	D			✔	✔		
10	D				✔	✔				D						D
10a	D				✔		✔			D						D
10b	D				✔			✔		D						D
10c	D				✔				✔	D						D
11	D				✔	✔				D			✔			
12	D			✔				✔				✔				D
12a	D			✔				✔				✔		✔		
13	D			✔					✔			✔				D
13a	D			✔					✔			✔		✔		
13b	D			✔			✔					✔				D
14	D			✔		✔						✔	✔			D
14a	D			✔		✔					✔			✔		
15	D			✔		✔					✔		✔			
15a	D			✔		✔					✔		✔	✔		
15b	D			✔		✔					✔					
16	D				✔	✔					✔					D
17	D				✔	✔					✔		✔			
18			✔		D	D					D					D

DAVID, THIS SHOULD BE INCLUDED IN THE FIGURE

Legend:

P	Planned	U	Unlocked.
IS	In Service	L	Locked.
D	Decommisioned	SD	Shutting Down.
E	Enabled	IT	Under Test.
Dis	Disabled	D	Degraded.
F	Free	NI	Not Installed.
R	Reserved	-	Empty Set.
PA	Partially Assigned	✔	A tick indicates that this is a valid base state value for the particular Status Condition.
A	Assigned		
	Component state not required to define the Status Condition.		
D	Default value taken by a component state, if present, though not required to define this Status Condition.		

Annex B (informative): Description of the modelling processes

B.1 Mapping of requirements to the model

B.1.1 Modelling goals

A number of the requirements are only partly supported or not supported at all in the current version of the class library. Examples include layering, inter TMN management, conflict resolution mechanisms when overlapping domains are supported, Tandem Connection/Tandem Connection Bundle, and scheduling of resources which have not been installed yet. Additional mechanisms (e.g. security) may be needed to support these modelling goals.

B.1.2 Layering and partitioning

Two fundamental relationships in the Functional Architecture are partitioning and layering. These are illustrated in ITU-T Recommendation G.805 [7], as shown in figure B.4a.

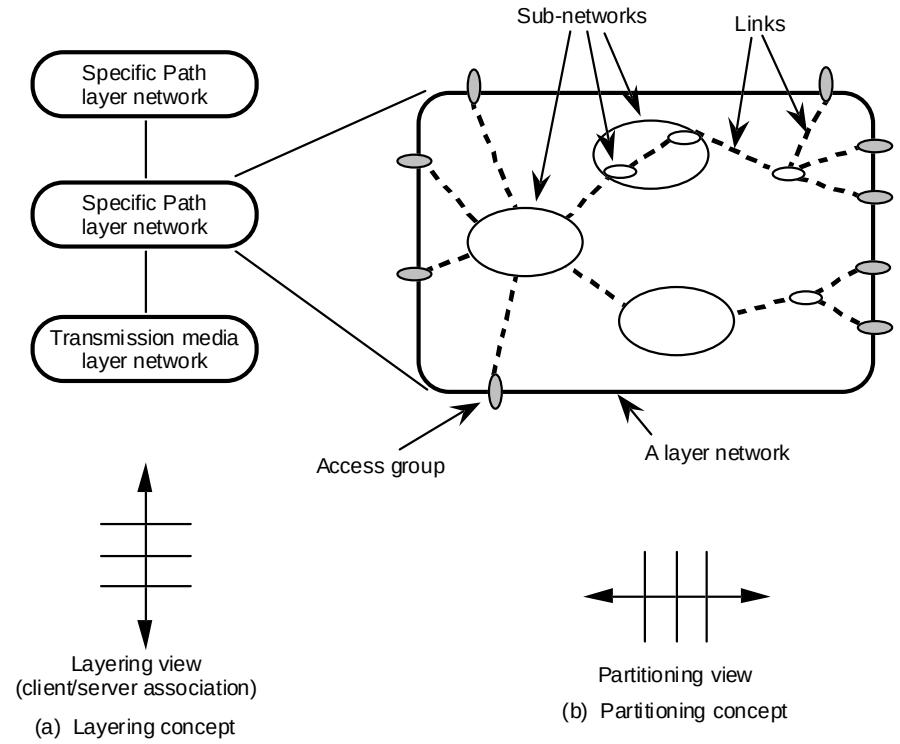
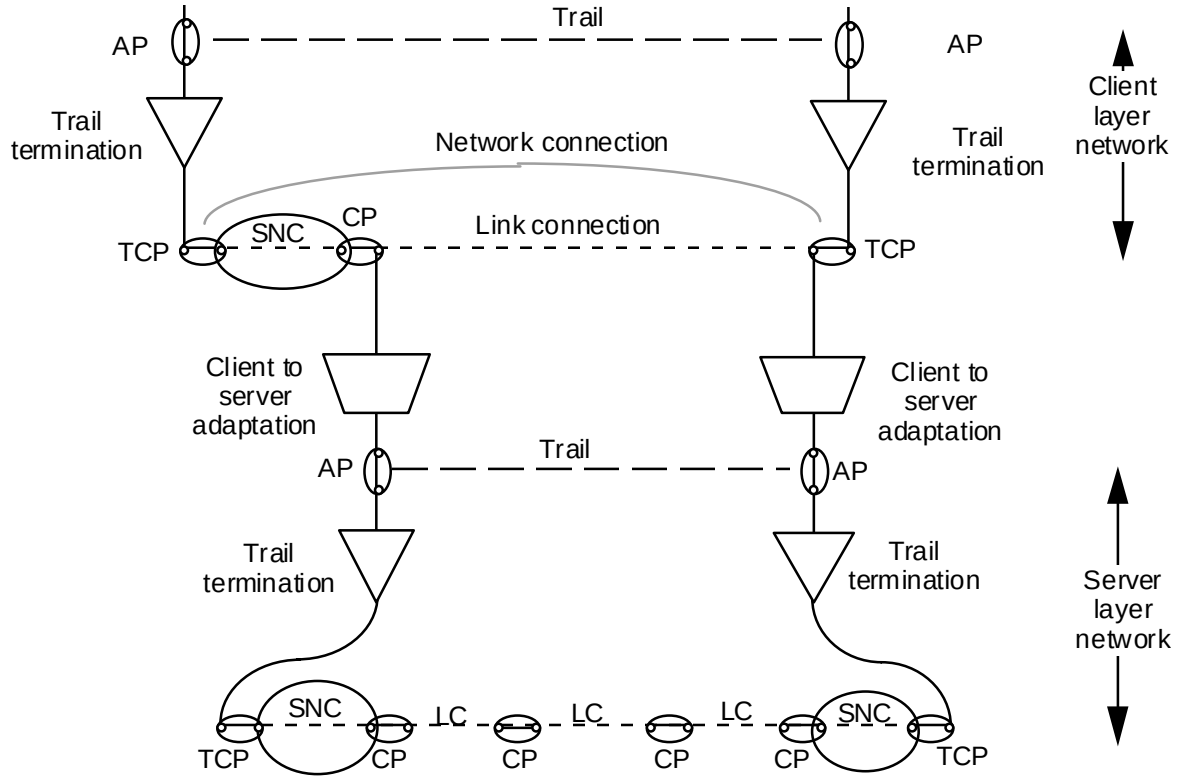


Figure B.1: Orthogonal Views of Layering and Partitioning

B.1.3 Layering

Consider the client-server architecture of ITU-T Recommendation G.805 [7] as illustrated in figure B.2. This figure shows the functional components used to describe the client-server relationship. A connection in a client layer is served by a trail in a server layer. The trail is composed of a sequence of link connections and connection points in the same layer.



AP	Access point
LC	Link connection
SNC	Sub-network connection
TCP	Termination connection point

Figure B.2: Example of functional model fragment illustrating use of some architectural components

For the purposes of the Managed Object Representation, not all the entities in ITU-T Recommendation G.805 [7] are modelled as separate classes. It is necessary to perform an abstraction of the ITU-T Recommendation G.805 [7] entities to produce a Managed Object description which only represents the features which need to be managed. This abstraction is based on the requirements identified above. The abstraction used in the class library is illustrated in figure B.3.

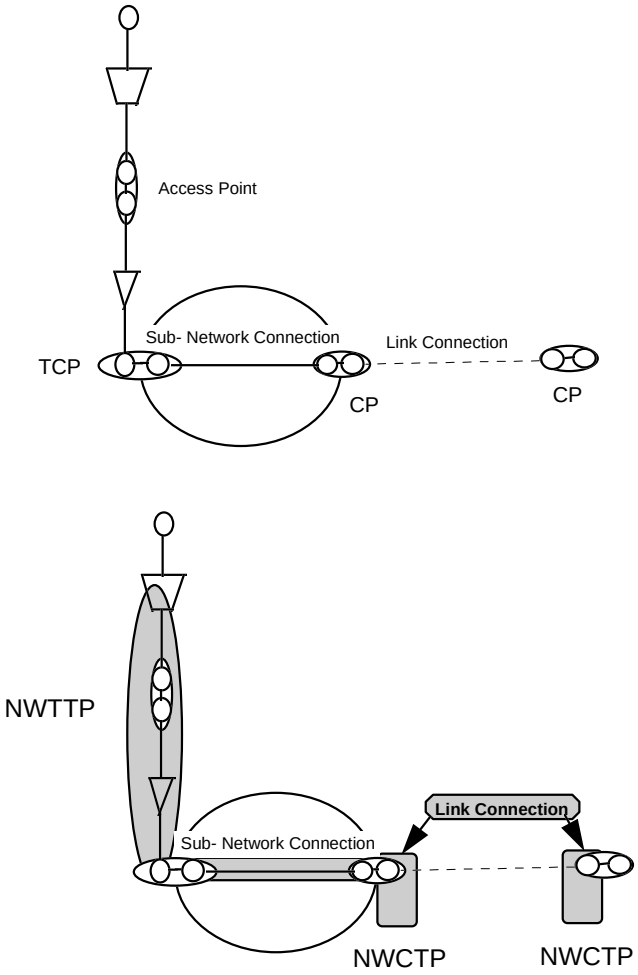


Figure B.3

The case shown in figure B.3 is where the NWTTP is on the boundary of a sub-network. The case where a link connection exists between the sub-network and the NWTTP is illustrated in figure B.4a.

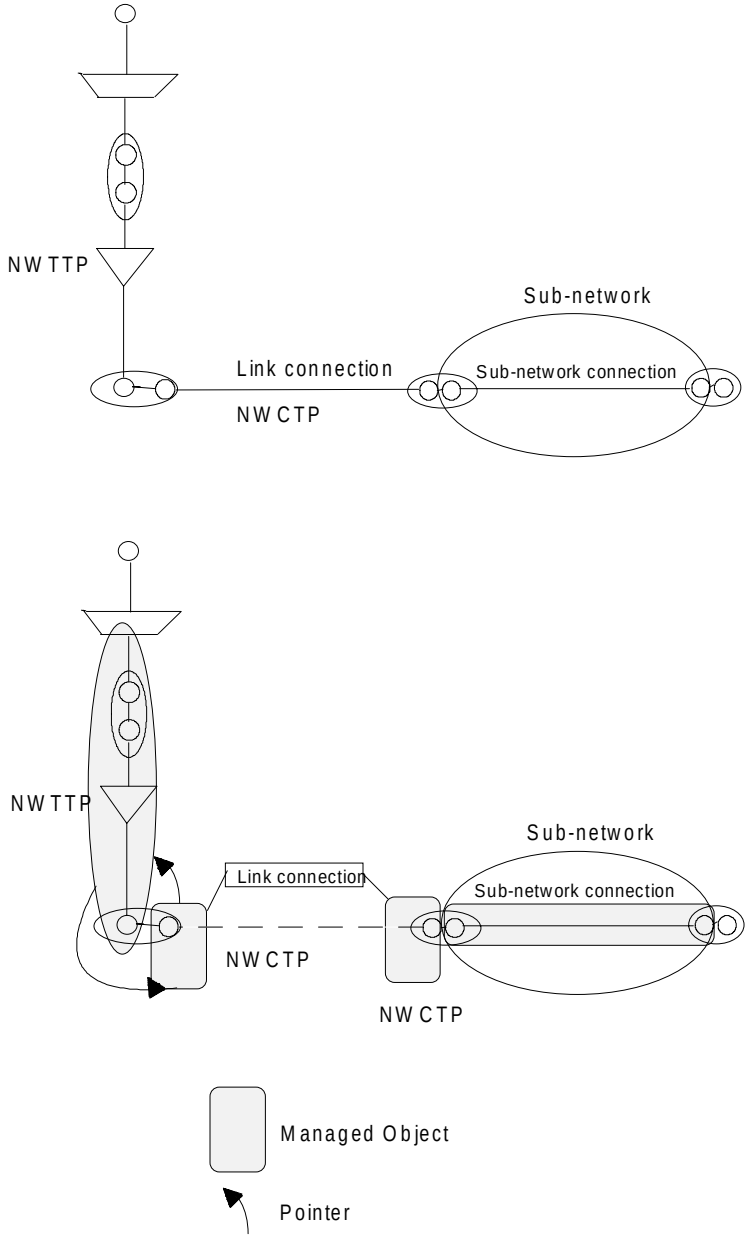


Figure B.4a: Mapping of ITU-T Recommendation G.805 [7] Entities to Managed Objects

There are two options for representing the points at the edge of the sub-network (the CTP). These are as link points or sub-network points.

Link points represent the ability of a sub-network to terminate a link connection (with its underlying resources), while sub-network points represents the capability of a sub-network to make connections across the sub-network. Thus a sub-network could have a high connecting capacity across it (due to the capacity of the underlying sub-networks), but these points can not all be used because some will not have link connections associated with them due to the number of connections that can be supported by the server trails in lower layers.

It is not very useful to model a high number of sub-network points not currently being used - so NWCTPs are used to model link points. These points represent the capability of terminating a link connection prior to the link connection actually being established. However the NWCTP does not reflect the state of the connectivity resource - this is expressed by the Status condition for that resource. For most applications the NWCTP will only carry very limited Status Conditions.

B.1.4 Partitioning

As described in ITU-T Recommendation G.805 [7], another important concept needed to describe a network is partitioning. Partitioning allows a hierarchy of sub-networks (and by implication sub-network connections), with successive layers abstracting the detail of the sub-networks in lower levels. This is illustrated in ITU-T Recommendation G.805 [7]. As a consequence, sub network connections may be composed of a sequence of connections and sub-network connections.

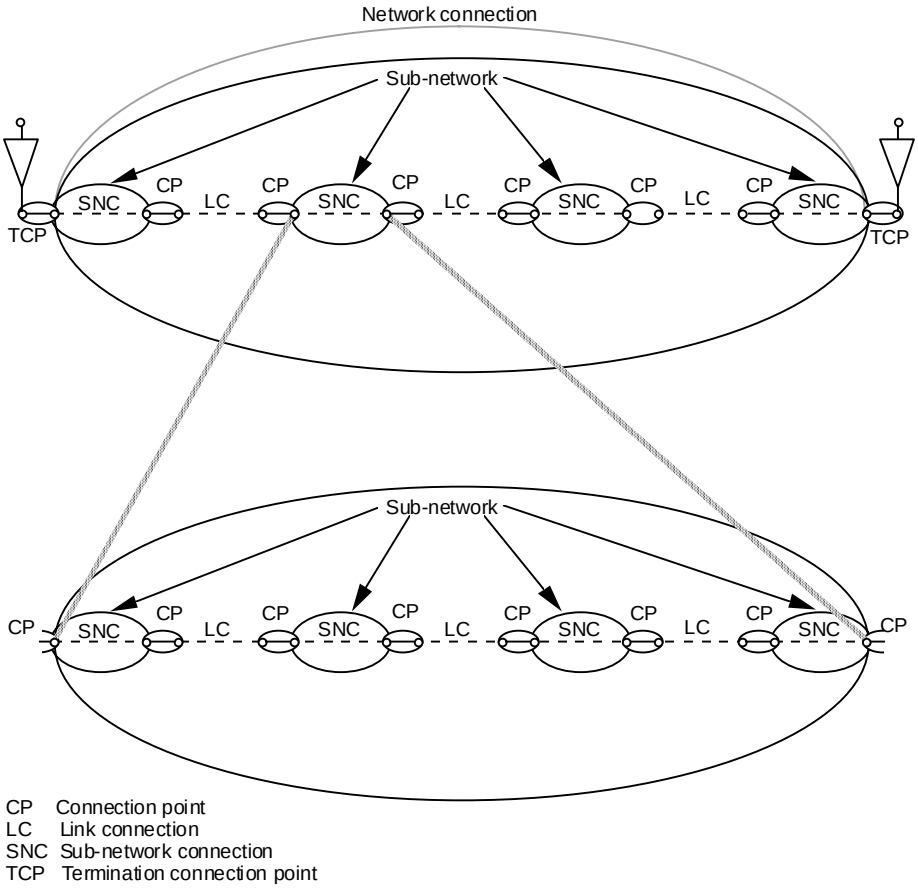


Figure B.4b: Partitioning of a network connection into sub-network connections

The representation of a sub-network connection across a sub-network at a single level of partitioning is given in figure B.5.

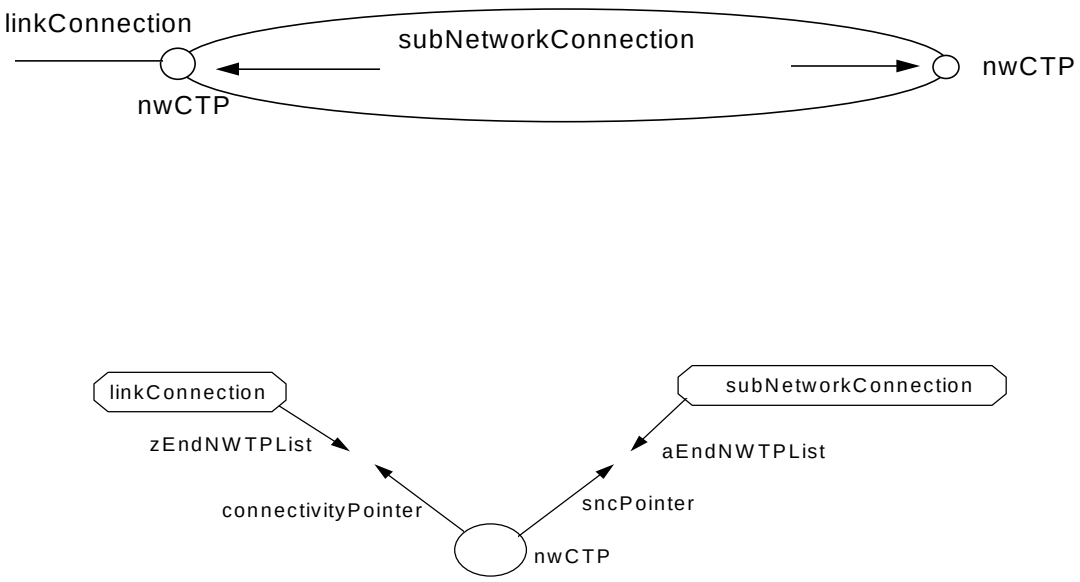


Figure B.5: Representation of a sub-network connection at a Single Level of Partitioning

For more complex cases, it may be seen that there is a hierarchy of levels of partitioning depending on the level of detail required For multiple levels of partitioning the "NWCTPs" at higher levels are in fact pointers to the single NWCTP which terminates the connection. This removes the need to duplicate NWCTPs (including their pointers)at every level of partitioning, at the expense of some ease of navigation from the NWCTP to the sub-network connections at the various levels of partitioning. However navigation starting from the sub-network is not impaired. This is illustrated in figure B.6.

The lowest level of partitioning of the hierarchy may correspond to a subnetwork, or may correspond to a cross-connection matrix. The subNetworkConnectionPointer of the NWCTPpoints to the subNetwork Connection Instance, otherwise it is null.

NOTE: If a subnetwork connection is set up at a higher level of partitioning, this implies that a sequence of subnetwork connections and link connections shall be set up in the lower level to support this subnetwork connection.

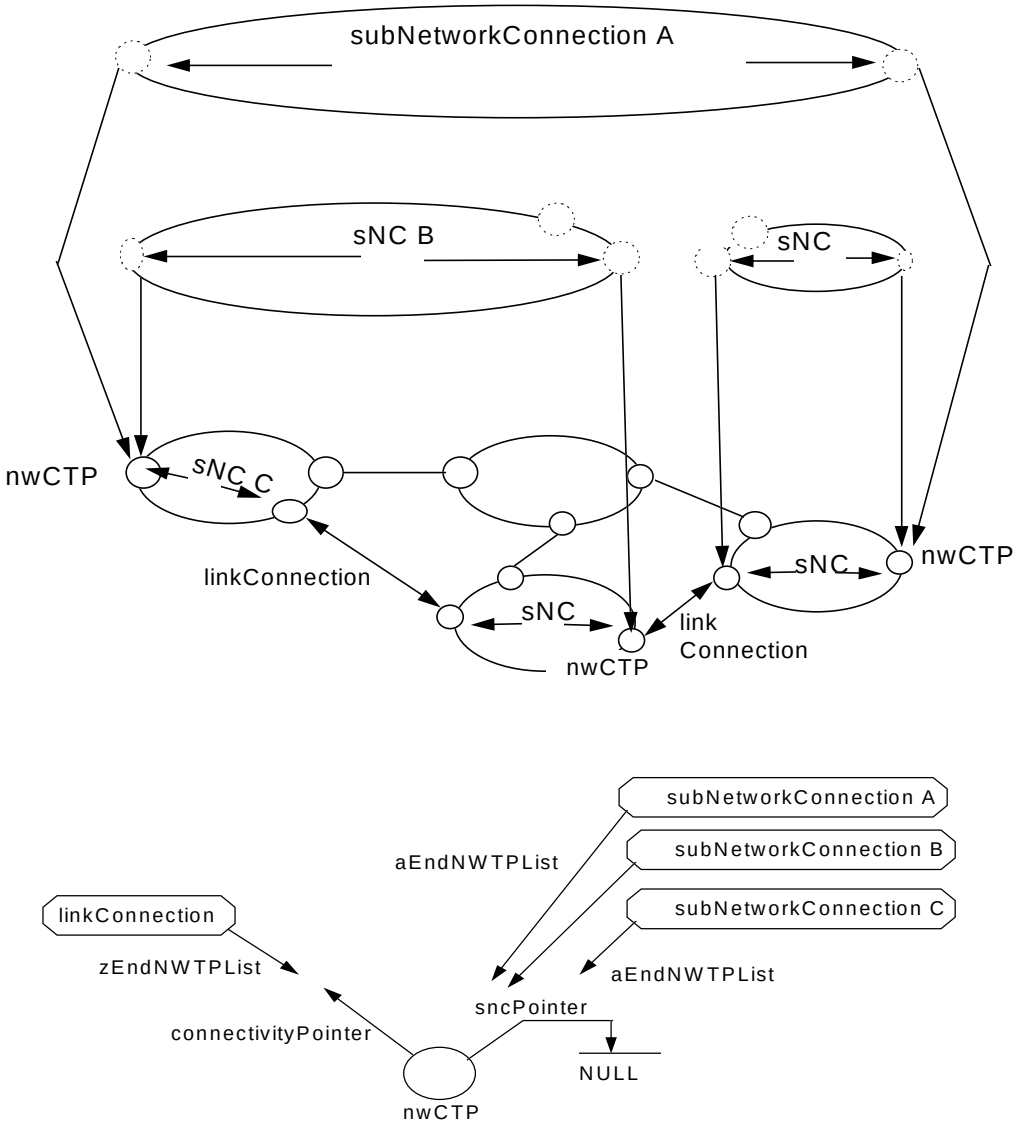


Figure B.6: Representation of a sub-network connection for Multiple Levels of Partitioning

This partitioning scheme has the advantage that it easily lends itself to support non-coincident and overlapping sub-networks at higher levels of partitioning as illustrated in figure B.7.

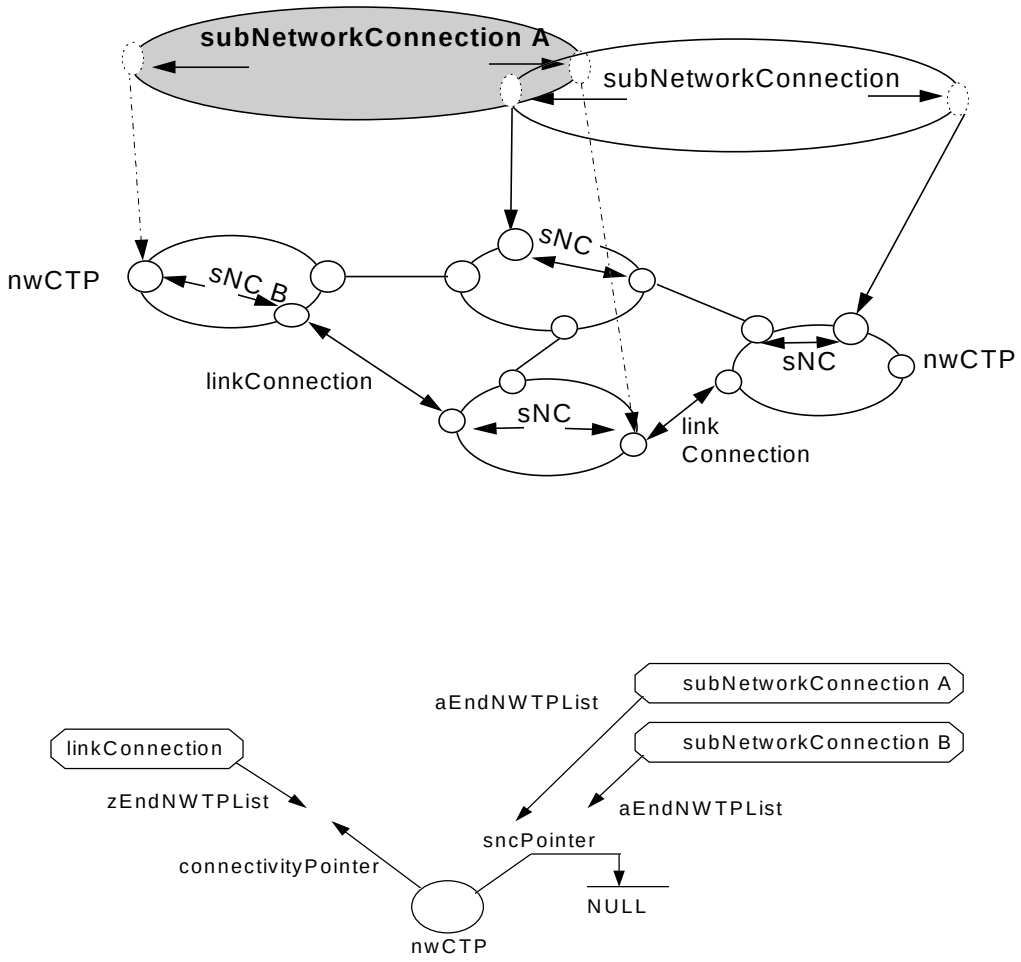


Figure B.7: Non- coincident and Overlapping Sub-networks

For the cases where navigation is of primary importance, it is possible to define a NWCTP at each level of partitioning and to use sub and super partitioning pointers. However, the use of this technique is deprecated except when the navigation requirement is sufficiently strong.

B.1.5 Topological view

According to ITU-T Recommendation G.805 [7], topological relationships within a layer network are expressed through the associations which exist between sub-networks, links and access groups. The library has endeavoured to capture these relationships through the following object classes: sub-network, link, access group and topological point.

There are two scenarios: firstly, when a link is used to terminate an access group at one end and a sub-network at the other (figure B.8). The other scenario is when two sub-networks are associated together by a link.

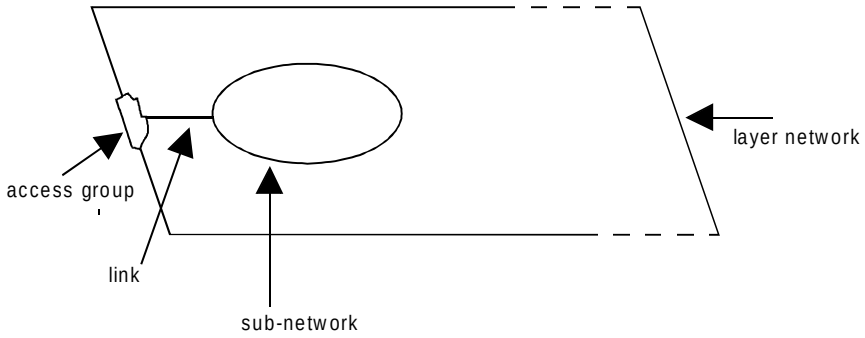


Figure B.8: Link Terminating a sub-network and Access Group

The support for the above topological description is depicted in figure B.9. Here, there exists a fixed relationship using the network TP pointer between the NWTTPs that form the Access Group and NWCTPs which terminate the Link. The access group is at the boundary of the layer network, and is modelled as being named from layerNetworkDomain. The NWCTPs which are bound to the NWTTPs are also named from layerNetworkDomainThe resulting relationship between the Link and Access Group and sub-network is captured by using the Link Pointer. Two name bindings exist for the link connections: one to link, and the other to layerNetworkDomain. The choice of name binding is application dependent.

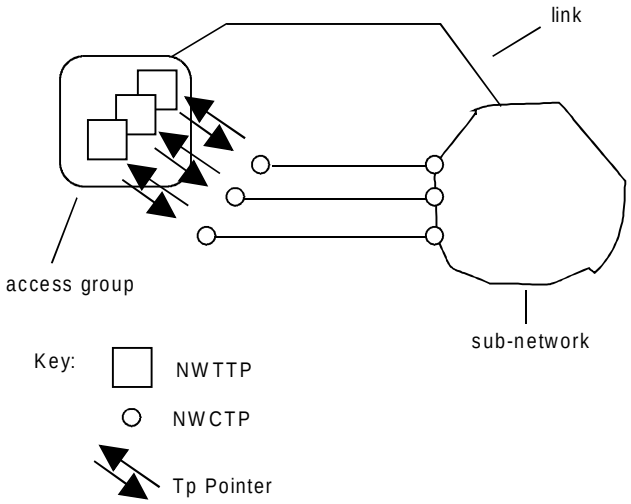
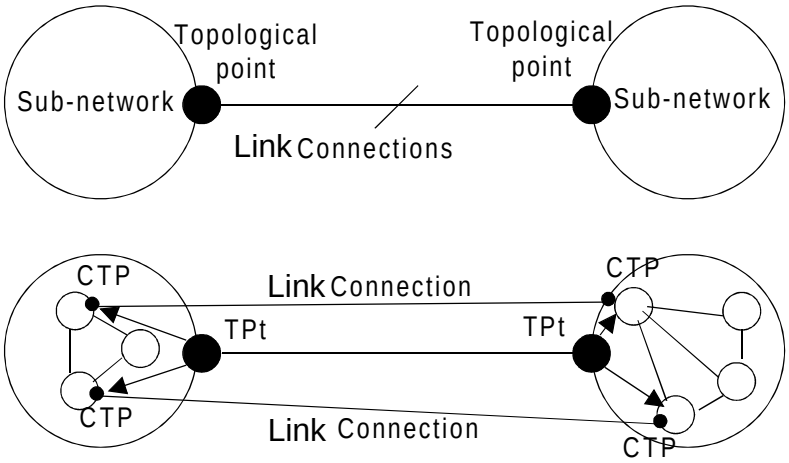


Figure B.9: Relationship Between sub-network and Remote NWTTPs

When modelling the topological relationship between sub-networks there are two possible approaches that can be used. They use either the link or the topological point objects to reflect capacity between the two sub-networks, but not both as this results in redundancy. If a link object is instantiated the link pointer attribute in sub-network is used to point to the link. When topological points are instantiated at the boundary of the two sub-networks forming the association, they point to each other. This pointer gives a direct relationship between two sub-networks without the need to carry out extensive operations to traverse the MIB. The two alternatives are depicted in figures B.10 and B.11.



Figures B.10 and B.11: Association of sub-networks with Topological Pointsand association of sub-networks with links

B.1.6 Administrative domains

Management domains are required for a number of purposes besides routing. This gives rise to a need for an managed object class other than sub-network to reflect the domains for other applications.

The Admin Domain class is used for generic division of the network for purposes such as defining maintenance zones etc. and specific sub-classes are introduced for each application such as representing parts of a layer network.

B.1.7 Layer networks

The class, "Layer Network Domain" is introduced for the purpose of representing the part of the layer network managed by a management system. Where a Service Provider OSF manages part of a network via another Service Provider OSF, then the Layer Network Domain presented to the first Service Provider OSF shall include that part of the network which it manages indirectly.

Example of the resulting naming trees are given in clause B.2.

B.1.8 Resources

A summary of the representation of resources by the managed objects of the class library is given in table B.1.

Table B.1

Resource	Managed Object Representation	Notes
Layer Network	Layer Network Domain	Represents part of Layer Network within domain of OSF
Characteristic Information	Signal Id (attribute)	
Sub-networks	Sub-network, degenerate sub-network, node	
Access Groups	Access Group	
Links	Link	Two types: internal link and external link
Trails	Trail	
Link Connections	Link Connection	
Sub-network Connections	Sub-network connection	
Tandem Connections	Tandem Connection (For further study)	see candidate classes
Tandem Connection Bundles	Signal Id (attribute) (For further study)	
Access Points	NWTTP	
Connection Points	NWCTP	
Adaptation Function	NWTTP,NWCTP	
Trail Termination Function	NWTTP	
Termination Connection Points	NWTTP	

Directionality and Mode

The modelling of directionality and mode is illustrated in figure B.12.

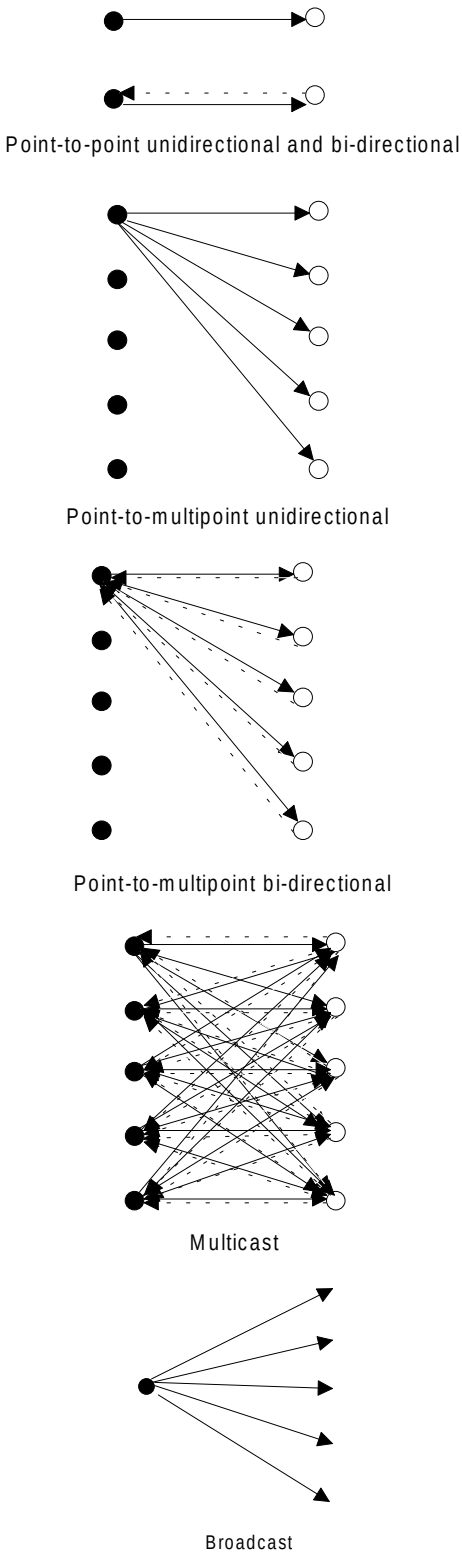


Figure B.12: Modes and directionality of connectivity

Table B.2: Combinations of directionality and mode

Mode	Uni-directional		Bi-directional	
	Source	Sink	Source	Sink
point-to-point	Network CTP source or Network CTP bid	Network CTP sink or Network CTP bid	Network CTP bid	Network CTP bid
point to multi-point	Network CTP source or Network CTP bid	Set whose members are Network CTP sinks or Network CTP bids	Network CTP bid	Set of Network CTP bid
multicast	Set whose members are Network CTP sources or Network CTP bids	Set whose members are Network CTP sinks or Network CTP bids	Set of Network CTP bid	Set of Network CTP bid
conference	Not Valid		Set of Network CTP bid	There are no known Z end terminations
broadcast	Network CTP source or Network CTP bid	There are no known Z end terminations	-	-

Signal Id

Signal Id is used to represent the characteristic information of a ITU-T Recommendation G.805 [7] layer.

It may be used in three ways.

The first is a simple identification of the layer such as VP layer or VC layer for ATM. Two signals with the same signal Id do not necessarily (and probably won't) have the same bandwidth.

The second way, more commonly used in circuit switched networks, is to indicate that two signals may be subnetwork connected

The following rules may be used:

- characteristicInformation shall match *exactly* for subNetworkConnection to be possible (this means that if a layerNetworkDomain only supports type simple, then no need to check Cis);
- in a bundling factor, bundling factor shall match exactly (and CI) for subNetworkconnection;
- if extended is used, then there are special rules regarding the use of format (e.g. possible to connect a 64KCTP with CAS (channel associated signalling) to one without CAS; can only connect voice ports to 64K with CAS [CAS channel is allocated in underlying 2MB]).

The third way is variable. This is a traffic descriptor which defines the bandwidth characteristics of the signal. This may be changed during the lifetime of the connectivity resource.

B.1.9 Event reporting

Event Forwarding Discriminators (EFDs) may be present in the Network OSF (SP). These may be named from System following ISO Event Reporting. However the number of EFDs, and their detailed use is not specified in the present document.

This specification may be provided by technology specific groups.

B.1.10 Scheduling

The scheduling mechanism operates on two levels: scheduling of sub-network connections and scheduling of link connections. Scheduling of sub-network connections concerns scheduling within a layer, while scheduling of link connections involves configuration of the adaptation function across a layer boundary.

For scheduling of link connections (i.e. between layers), an example is that a Trail may be configured, via the adaptation function, to provide a particular link connection from Monday to Wednesday and a different link connection from Thursday to Friday. This is effected by creating an allocation on the trail which serves the link connection. This is illustrated in figure B.13.

For scheduling of sub-network connections (i.e. within a layer) an example is that during the period Monday to Wednesday (when the link connections are available), the capacity of the layer may be scheduled to different users for different periods, e.g. a user may require a network connection on Monday morning. This would be set up by creating a sub-network connection with the appropriate schedule.

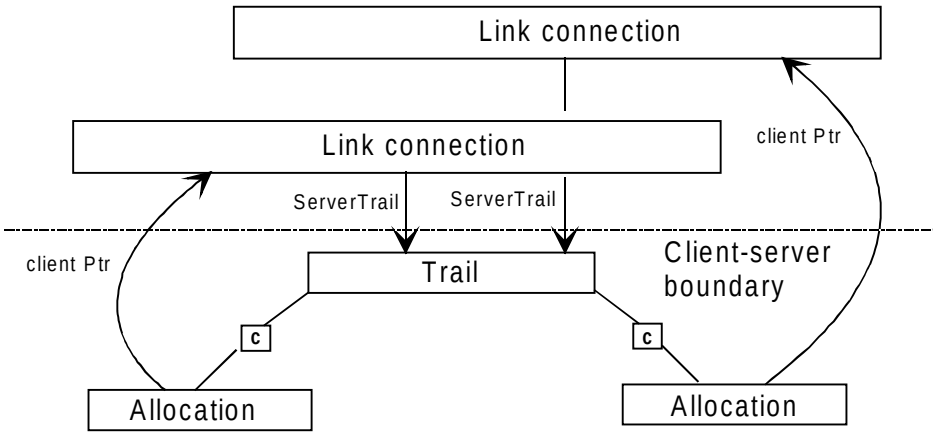


Figure B.13: Scheduling of link connections

The detailed operation of the scheduling of sub-network connections is as follows:

When a user OSF requests a bandwidth-scheduled sub-network connection to a service provider OSF, the former will specify the requested bandwidth in the appropriate parameter in the setup SubNetwork Connection action directed to the basicSubNetwork object in the domain of the service provider OSF.

The service provider OSF, will (if everything is OK) create a subNetworkConnection object instance. That object will have instantiated the package associated for the type of scheduling requested by the user OSF (e.g. weeklySchedulePkg if it requested for a weekly scheduled connection). The package will contain the schedule itself and the appropriate actions to modify the bandwidth schedule (add, delete and modify time slots). This avoids the need to release the sub-network connection and re-establishing again, avoiding the recalculation of the resource availability and reservation for all unmodified time slots (this is particularly of use when thinking of semi-permanent connections extending through several administrative domains, or even different PNO networks).

It is important to understand that the service provider OSF is delegated with the responsibility of the resource planning in the time, so if it acknowledges a request, it shall guarantee that resources will be available when the time slots come.

Results of above mentioned actions (set-up and modifications) include full information about the reasons in case the request could not be satisfied (lack of resources, overlapping time slots...).

The "in traffic" condition of the subNetworkConnection is driven by the schedule. A scheduled connection is set-up in the In Service, Not allocated (4) Status Condition. When the schedule indicates that the sub-network connection is to be put in traffic, the Status Condition changes to In Service with no spare capacity (8) (preceded by the In Service with no spare capacity, under test (9) Status Condition if an initial test is made).

In a two-phase set-up comprising reservation and activation, the sub-network connection is set-up in the In Service, Reserved (4) Status Condition at the time dictated by the schedule, pending an Activate Action from the manager.

As an example, if the schedule dictated "Tuesdays from 9 a.m. to 10 a.m.", the In Service, Not allocated (4) Status Condition will not change until the next Tuesday at 9 a.m. In other words the service provider OSF is delegated the responsibility of controlling the "in traffic" condition in accordance with the requested schedule.

B.1.11 Mapping of management capabilities to the class library

A summary of the representation of management capabilities by the managed objects of the class library is given in table B.3.

Table B.3

Management Capability	Implementation
Static Configuration	
1 The provisioning of a layer network and characteristic information	Outside scope of class library. Characteristic layers represented by creation of Layer Network Domains.
2 The provisioning of access points	CREATE/DELETE NWTTP
3 The provisioning of access groups	Automatically CREATED when addNWTTPsToAccessGroup ACTION is invoked.
4 The configuration of access groups	ACTION on Layer Network Domain
5 The provisioning of connection points	Creation of NWCTPs when sub-network created.
6 The configuration of connection points	ACTION e.g. addNWTPsToNWGTP
7 The provisioning of sub-networks	CREATE/DELETE Subnetwork
8 Link Provisioning	CREATE/DELETE link
Dynamic configuration management consists of:	
1 The setting-up of sub-network connections	ACTIONs on Basic Connection sub-network
2 The release of sub-network connections	ACTION on Basic sub-network
3 Sub-network Configuration	containedNWCTPList GET-REPLACE ADD-REMOVE; containedSubNetworkListGET-REPLACE ADD-REMOVE containedLinkListGET-REPLACE ADD-REMOVE
4 The scheduling of sub-network connections. The scheduling of trails	Set-Up sub-network connection ACTION on Basic Sub-Network. Modicication by ACTION on sub-network connection FFS
5. Trail set-up and release	ACTION on Basic Trail Handler
6 The setting-up of tandem connections, (for further study) which comprises,: 6.1 The configuration of links 6.2 The provisioning of link connections 6.3 Tandem Connection provisioning and configuration 7 The release of network connections	connectionList GET-REPLACE ADD-REMOVE CREATE/DELETE Link Connection CREATE/DELETE Tandem Connection connect/disconnectAll ACTIONs
7. Link Provisioning.	CREATE/DELETE Internal Link, External Link
9. Network restoration (including path restoration)	For further study
10. Network protection (including path protection)	For further study
11. The testing of a sub-network connection	For further study
12. Scheduling of trails	For further study

B.1.12 Composition of resources and capabilities

There is a need for a composition technique so that the functional capabilities and the resources to which the functions apply can be modelled in a flexible manner. Two methods are possible, as discussed in clause 7. These have been applied to the modelling of sub-network s as illustrated below.

The sub-network represents the resource and capabilities are added by mandatory packages. Optional features are defined using conditional packages.

Alternatively capabilities may be added by naming. In this case the capabilities form part of an instantiated object.

The two techniques may be combined. For example, a basic sub-network may be defined by inclusion of the mandatory package but may be further extended, when new capabilities are defined, by use of naming.

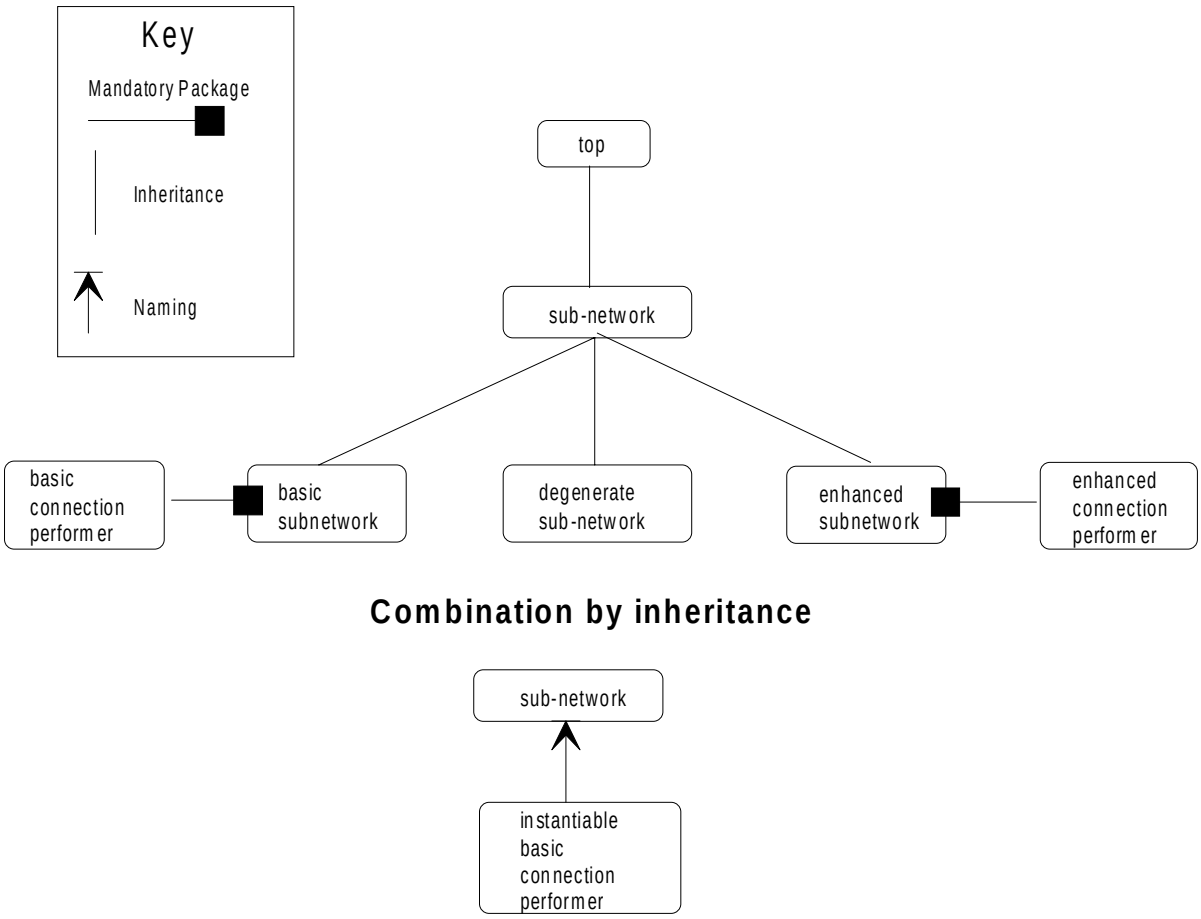


Figure B.14: Modelling of sub-networks

B.2 Relationships

Overview of methods for representing relationships

The class library is an abstraction of the Network Resources (as defined by the Functional Architecture, which is based on the generic aspects of ITU-T Recommendation G.805 [7] with appropriate extension for other technologies). The relationships between the Network Resources are illustrated by the Entity-Relationship diagram in subclause B.2.1.

The abstraction for management purposes produces a set of managed object classes with relationships between these classes. The managed object classes form the class library. The relationships defined are the superset of relationships between the classes. These objects and relationships are illustrated in subclause B.2.2. This is an informal representation. An alternative to this diagram would be to represent the Relationships using the General Relationship Model.

When implementing the class library the application group shall choose (by selection of name bindings or profiling pointers in conditional packages) those relationships which are needed. For example if a topological point view of topology is required, relationships involving links will not be used.

The application group shall also choose the relationship binding to implement any given relationship defined in the Entity-Relationship diagram, and chosen by the application group. Two types of relationship are supported by the class library: functional composition ("is a") and general association ("has a").

For functional composition two mechanisms are available: inheritance and name binding, as discussed in clause 7 and subclause B.1.10. An inheritance diagram for the class library is given in subclause B.2.4.

For general association there are several types of relationship but the most pertinent is the "contains" type which has two possible representations: by pointers and by name bindings. General associations which are not of the "contains" type are implemented by pointers (in conditional packages).

The choice of which of the "contains" type relationships are represented by name bindings is particularly important since this governs the structure of the MIB, and the operation of the associated CMIP scoping and filtering mechanisms. The class library supports this choice by providing both name bindings and pointer (in conditional packages) implementations. The application groups may define different relationship bindings (either by new name bindings or by adding new pointers through specialization) if those provided by the class library are not appropriate for the application. However, in order to achieve the goal of maximum compatibility, the naming schema defined in this class library are strongly recommended to the application groups.

Examples of naming schema, for the guidance of the user, are given in subclause B.2.3.

B.2.1 Resource relationship diagram

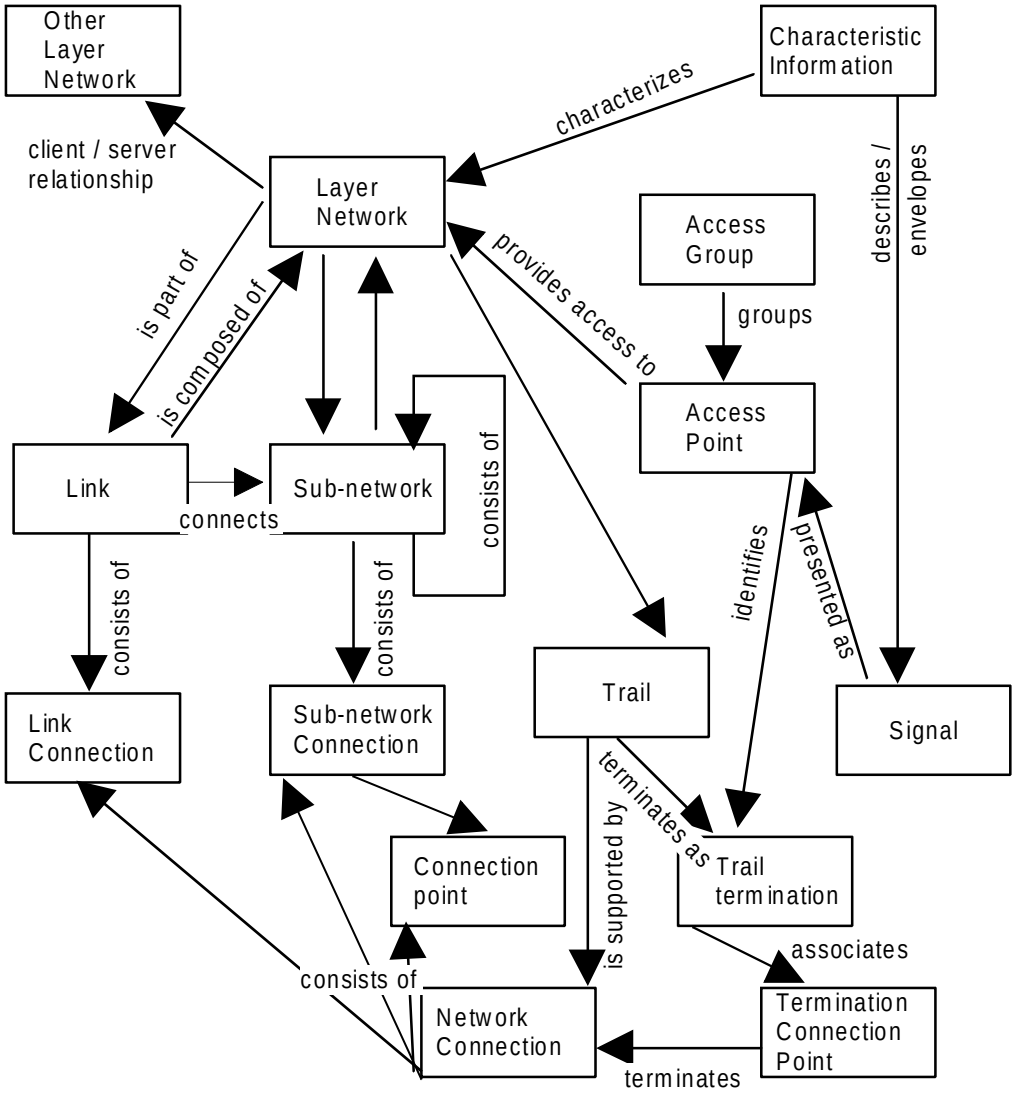
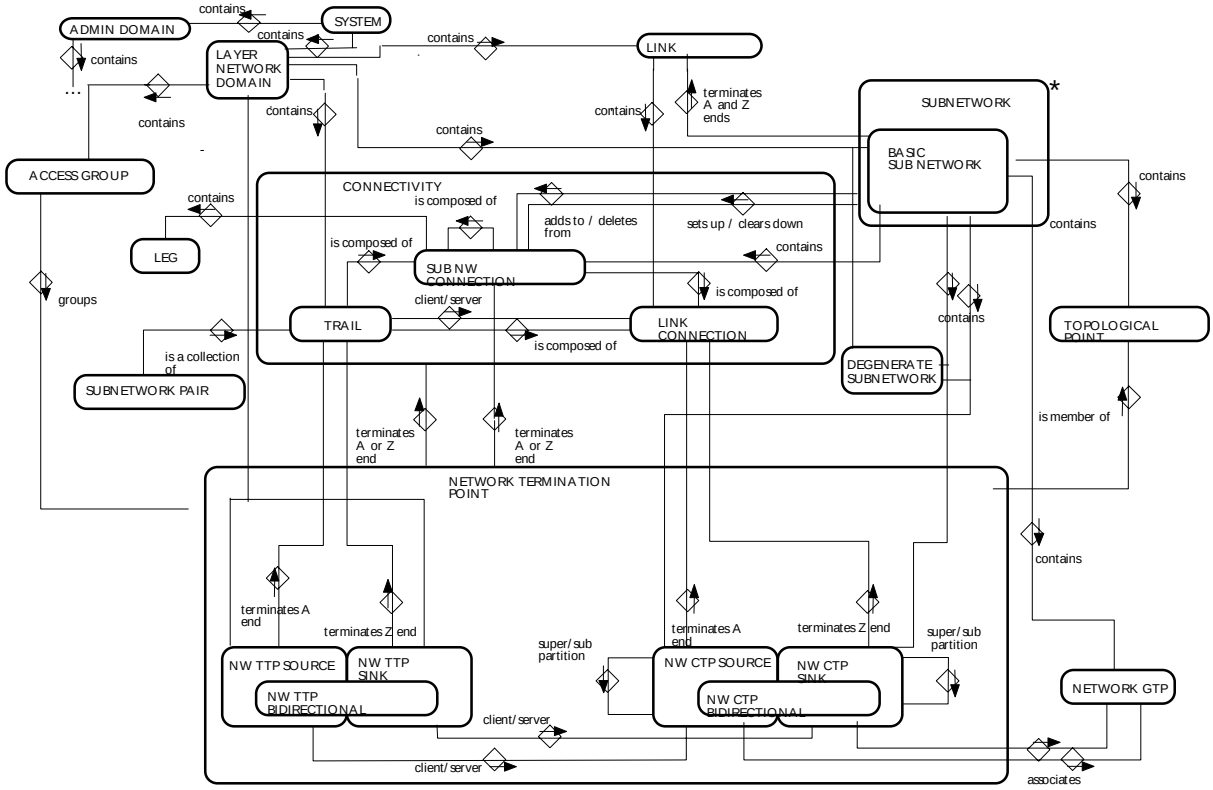


Figure B.15: Resource relationship diagram

B.2.2 Entity relationship diagram

The entity relationship diagram for the components of the class library is given below. This diagram shows all the possible relationships. Not all of these relationships will be used by any given application and an application has a choice of bindings for most of the relationships as discussed in clause B.2.



Relationships Diagram

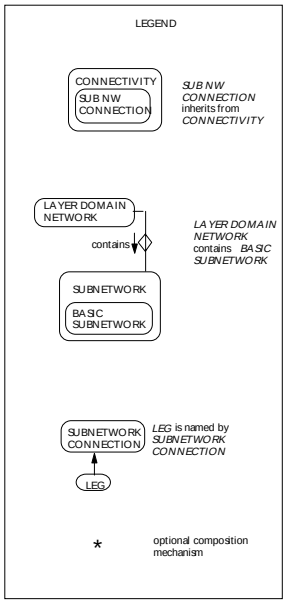


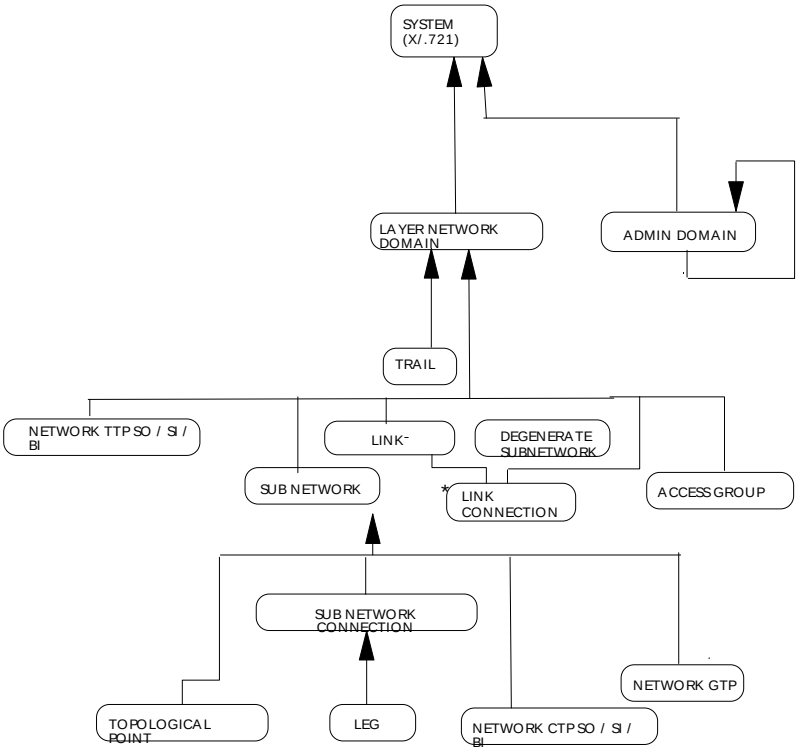
Figure B.16: Managed object entity relationships

B.2.3 Object naming

Definition of the naming relationships implicitly species the construction of the MIB for a particular interface. The class library contains optionality which allows some relationships (as described in subclause B.2.0) to be expressed by naming or by pointers. It is the role of the application specific profile to select which method is used in each case. Hence the naming tree is application specific.

Examples of naming schema which can be used in conjunction with the class library are given below:

Example Schema 1:



Naming Diagram

NOTE 1: In the NE view, the naming relationship is used to define the client-server relationship between two ITU-T Recommendation G.805 [7] layers. This method is most suited to describing a tightly coupled multiplexing hierarchy, but does not allow an OS to manage a layer network independently of other layer networks. In the latter case it is better to describe the client-server interaction using a relationship, (i.e. using a pointer or a relationship object).

NOTE 2: The "ITU-T Recommendation X.721 [13] System" class is used at the top of the naming tree which represents the MIB for a Service Provider OSF. Event Forwarding Discriminators, Logs, etc. may be named from system as in the ITU-T X.700 series of Recommendations.

Figure B.17: Naming diagram for example schema 1

Example Schema 2:

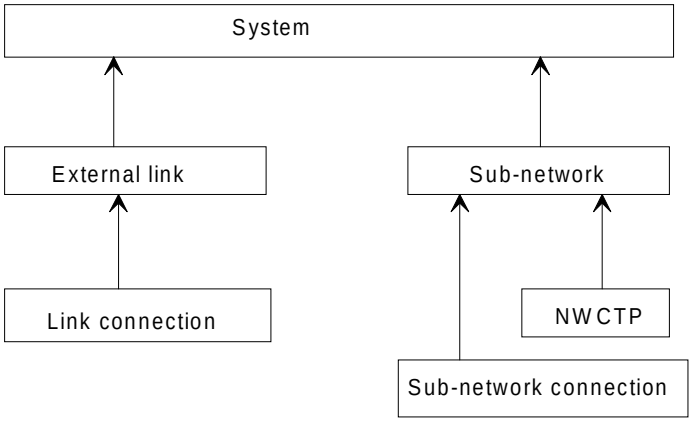


Figure B.18: Naming diagram for example schema 2

Example Schema 3:

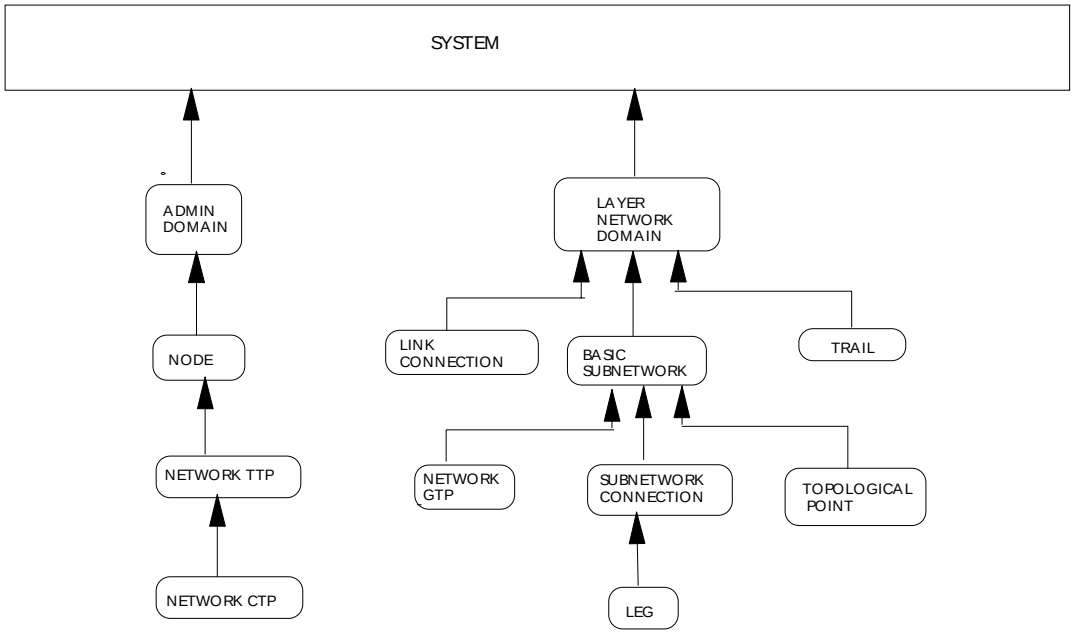


Figure B.19: Naming diagram for example schema 3

B.2.4 Inheritance diagram

The inheritance diagram for this class library is given in figure B.20.

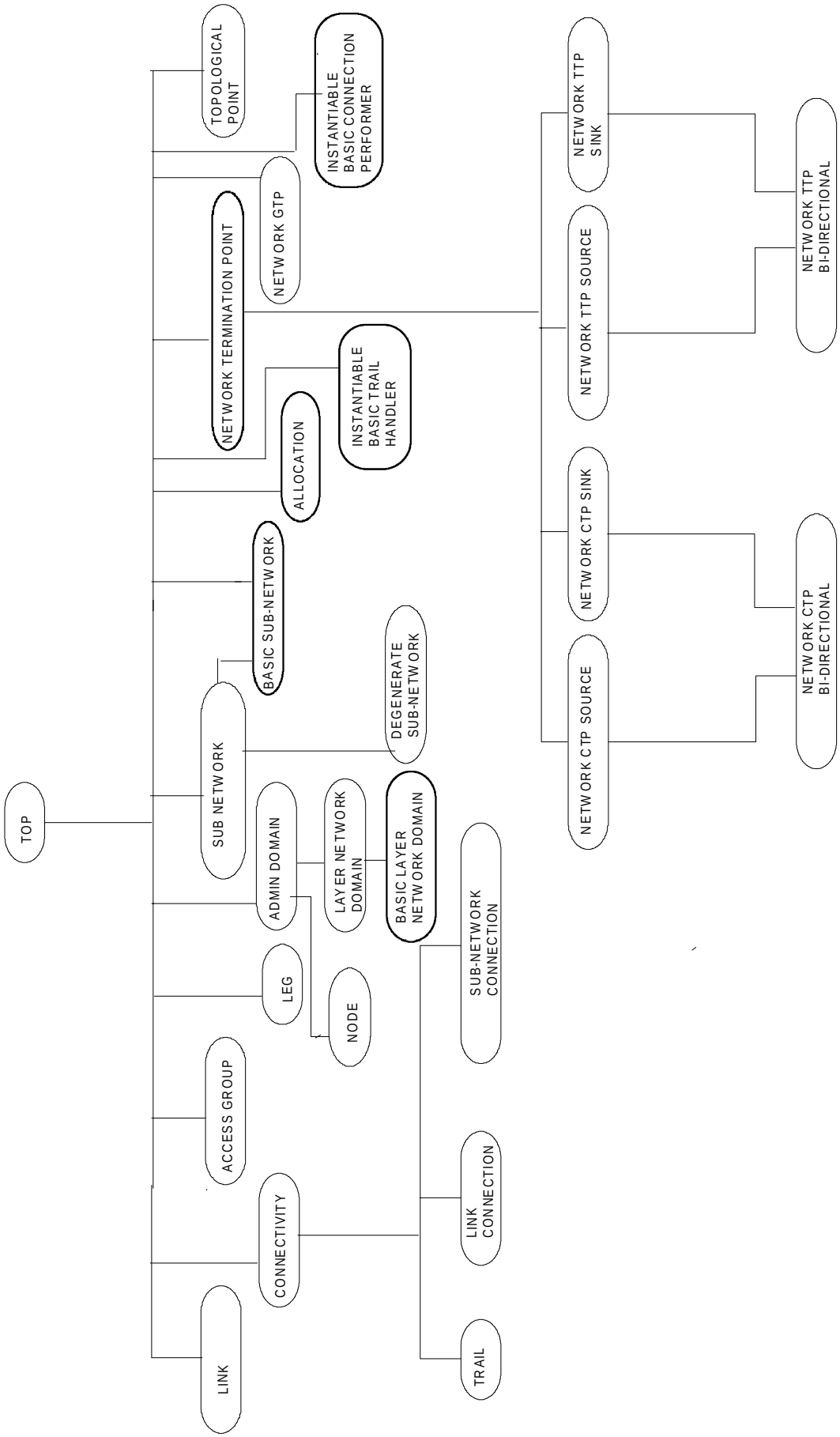


Figure B.20: Inheritance diagram for the class library

Annex C (informative): Profiling guide

As discussed in clause 1, the Generic Object Model comprises a Generic class library with a profiling guide to allow Technology Groups to select the Generic features which are most appropriate to an application, and add features which are most appropriate to their requirements.

The following are the minimum requirements for a profile for a particular interface:

- requirements detailing the interfaces considered;
- a set of use cases including network examples; that is, particular networks which an implementation of the class library needs to represent;
- a definition of the functional scope of the model (e.g. configuration);
- a definition of the range of applicable transport technologies;
- standardized profiles, which select options (conditional packages), and potentially add sub-classes;
- domain examples, which detail the domain structure;
- conformance statements.

NOTE: Not all interfaces which can be derived from the class library will be the subject of standardization.

C.1 Removal of optionality

As described in clause 7, the User Guide, the class library contains a high degree of optionality. This exists to provide different bindings for relationships, and for the selection of features which may be used by the application. Following the approach above, the optionality is removed in the profile described above by:

- selection of conditional packages by use of the CHARACTERIZED BY clause where there is subclassing;
- selection of conditional packages by explicit statement of which conditional packages are mandatory and which are not used in the application;
- selection of which bindings are used for relationships, by producing Entity-Relationship diagrams and the naming tree for the application.

C.2 Application notes

C.2.1 Support for ATM requirements

A single setupSubNetworkConnection action accommodates circuit switched and flexible bandwidth requirements.

The ATM known requirements are accommodated by specifying scheduling and bandwidth of the connections. There is a further requirement on implicit TP creation and deletion with the set-up and release of the connections. Implicit TP creation and deletion is modelled as an optional parameter to capture this operating mode.

The ATM requirements are described below.

C.2.1.1 Scheduling and bandwidth allocation

Bandwidth scheduling can be of five basic types (see requirements and annex B for further discussion):

- duration: one single slot, non periodic connection;
- dailySchedule: several day slots each with different bandwidth;
- weeklySchedule: several weekSlots each with different bandwidth;
- monthlySchedule: several monthSlots each with different bandwidth;
- occasional: several non-periodic slots each with different bandwidth.

Accordingly, each slot will have a start point in time, a stop point in time and the associated bandwidth (with the implicit and appropriate periodicity).

If we consider using bandwidth with a null value, this perfectly applicable to other technologies (SDH etc.).

C.2.1.2 Implicit TP creation and deletion

In ATM the number of possible termination points in a physical interface is enormous (4096x65536). CTPs can not be instantiated before the need to setup a Sub-Network Connection.

A optional parameter defines whether or not there is implicit TP creation and deletion.

C.2.1.3 Quality of service negotiation

For each direction of an ATM layer connection, a specific ATM Layer QoS from those supported by the network is requested at connection setup time. This requested QoS is embodied in the traffic descriptor (which is being defined by technology specific groups) associated with the ATM connection. The network commits to meet the requested QoS as long as the end system complies with the negotiated traffic contract.

The requested QoS could be either indicated by the objective of each individual parameter or by a QoS class specification. The actual default minimum performance objective for each of the parameters (either explicitly or as part of a QoS) will be standardized by technology specific groups.

It is expected that the technology specific groups will consider the following the following ATM parameter:

Service Type (e.g. CBR, VBR, VBR, ABR);

and related parameters from the following list:

- peak-to-peak cell delay variation;
- maximum cell transfer delay;
- cell loss ratio;
- cell error ratio;
- Peak Cell Rate (PCR);
- Cell Delay Variation Tolerance (CDVT);
- Sustainable Cell Rate (SCR); and
- Burst Tolerance (BT).

C.2.2 Support for inter-TMN requirements

Support is required for setting up Network Connections between two administrations which have separate TMNs. This connection will involve an originating subnetwork within one TMN and a destination subnetwork in another TMN. The connection may traverse one or more subnetworks belonging to third party TMNs.

The setting up of a network connection between Co-operative Administrations consists of the setting up of several sub-network connection until getting the Destination User. So, the control of a Network Connection across the different Sub-networks will always involve:

1) one originating sub-network:

   - the manager requests a "Set-Up Sub Network Connection" ACTION on its own sub-network between two "Access Points" (NWTTPs);

   - the "SNC Directionality" parameter in the ACTION INFO " will have two "snc TP", that is to say, two object Instances;

   - the "Far End PNO sub-network ID" and "Destination Address" will not be present;

2) zero, one or more transit sub-networks:

   - the manager requests a "Set-Up Sub-network Connection" ACTION between a A end Access Point and the next Sub-network on the chosen path;

   - the "A end NWTP" parameter of the "SNC Directionality" will have the object Instance of the A end Access Point, and the "Z end NWTP" one should have a NULL value;

   - the "far End Sub-network Id" parameter will be present;

3) one destination sub-network:

   - the manager requests a "Set up Sub-network Connection" ACTION between an A end NWTP (Object instance) and the Destination User;

   - the "destination Address" parameter will be present.

NOTE: For some co-operative interfaces, the "near End Pno Sub-network Id" parameter is not needed.

C.2.3 Alarm reporting

For further study. This may include the following:

Many TP-alarms and alarm Status changes are potentially redundant. However a network level TP does not report the same alarms and Status changes as its network element level counterpart. At the network level interface, an abstraction of the element level view is provided. For example, in figure C.1, a LOS alarm from two (bi-directional) trail termination points at the network element level is associated with a trail at the network level.

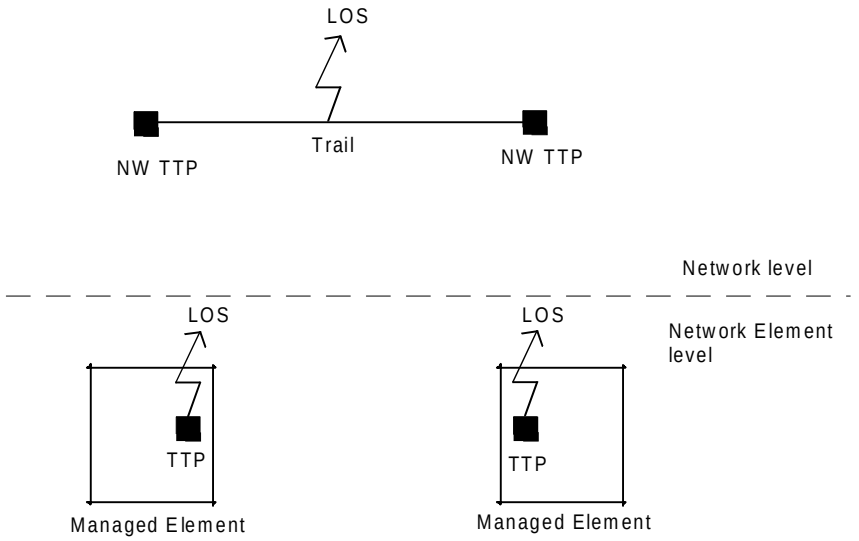


Figure C.1: Alarms at network and network element level

The presence of network TPs does not mean that these will generate alarms in addition to the connections and trails, nor that they will generate the same notifications as at the Network Element level. Furthermore, there is no need to report all consequential alarms a failure of a network resource. It should be noted that for some failures of network resources e.g. an STM16 failure, there will not be secondary alarm reports (e.g. AIS received notification) because these can be suppressed at source. Careful design of the network level model together with use of Event Forwarding Discriminators allows the flow of notifications to be minimized. Table C.1 shows how profiles may be used to remove unnecessary notifications from the GOM when they are not needed for a particular ensemble.

Table C.1: Network TP

ATTRIBUTES	Base document	Profile support	Use in profile
createDeleteNotifications	M	Y	
connectivityPointer	C	Y	
neAssignment	C	N	
tmnCommunicationsAlarmInformation	C	N	Notification is carried by Connectivity object.
State	C	Y	
stateChangeNotification	C	Y	
sncPointerPackage	C	Y	
networkTPPointerPackage	C	N	
attributeValueChangeNotification	C	N	

At present, much of the alarm behaviour for the GOM is for further study. This work shall take into account which alarms and event reports are/are not required at the network level and provide definitions accordingly. The propagation and inhibition of alarms between network layers is of particular interest. The class library does not currently address the representation of the state of a resource where alarm reporting has been disabled.

Annex D (informative): Additional candidate class definitions

The requirements, managed object classes, and other definitions in this annex are provided for information only.

D.1 Requirements

D.1.1 Fault management

Fault management requirements are for further study but may include:

a) alarm surveillance (including alarm suppression);

There shall be a facility to report alarms optionally against termination point and connectivity managed objects. Network View alarms may be at a higher level of abstraction than NE View alarms;

b) fault localization (including alarm correlation);

c) test management (including intrusive type testing).

D.1.2 Configuration

a) Network restoration (including path restoration).

For further study.

b) Network protection (including path protection).

For further study.

c) The testing of a sub-network connection.

For further study.

d) Enhanced Sub-network Connection Set-Up.

Only basic Sub- Network Connection is supported by the class library at this stage.

This subclause covers the setting-up of a sub-network connection in response to a request containing more than a minimum of information. Within an enhanced sub-network connection set-up request a user will have the ability to specify additional values for the different types of information in addition to these values specified in Basic Sub-network Connection Set-Up. The additional types of information are:

- routing criteria;
- links and sub-networks at the next level of partitioning to be used;
- the individual connection points within the sub-networks at the next level of partitioning to be used;
- diversity criteria;
- use of resources which have not yet been installed;
- a set-up or reservation using a best attempt policy;
- it shall be possible for a user to request the modification of stop time for an existing (or already scheduled) sub-network connection. A request for an earlier stop time shall always be accepted. However, the fulfilment of a request for a later stop time will be dependent on the availability of resources;
- it shall be possible to ensure that set-up requests from non-authorized users can be identified;

- during the processing of a set-up request users shall only be allocated resources to which they have they are allowed access;
- the reservation of a sub-network connection using the routing criteria specified in the enhanced set-up request shall be supported;
- as user may specify the "use" of the sub-network connection:
  - ordinary traffic (protected);
  - ordinary traffic (not protected);
  - shadow traffic (e.g. the standby side in MSP 1+1, use as part of reconfiguration);
  - protecting (e.g. the standby side in MSP 1:1);
- the user may request policy based routing.

D.1.3 Performance management

These requirements are for further study.

D.1.4 Accounting

These requirements are for further study.

D.1.5 Security

These requirements are for further study.

D.1.6 Viewing requirements

These requirements are for further study but shall include:

1) viewing of provisioning state;
2) viewing of network topology;
3) viewing of network connectivity.

D.2 Connectivity classes

D.2.1 Types of sub-network connection

This subclause aims to explain the different types of sub-network connection which may be modelled using the existing class library, and also those options covered by use of the additional definitions provided below.

The simplest type of Sub-network Connection is point to point, as shown in figure D.1.

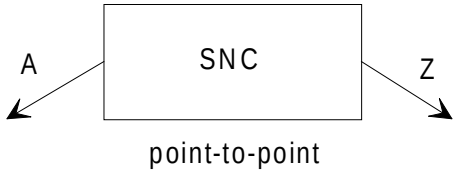


Figure D.1: Point to point sub-network connection

A point-to-multipoint Sub-network Connection has one A end and multiple Z ends. A Leg object connects each Z end to the parent Sub-network Connection which permits the service state of each leg to be independent of the others.

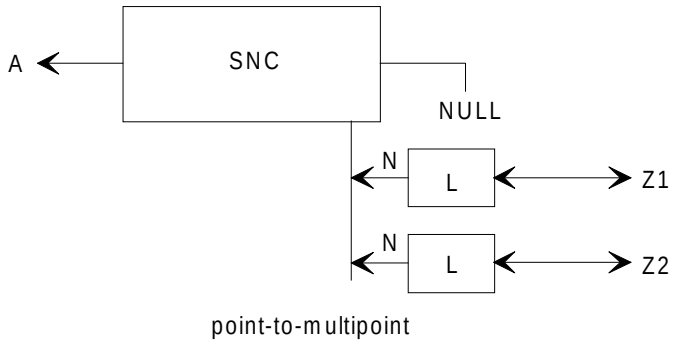


Figure D.2: Point-to-multipoint sub-network connection

Both point-to-point and point-to-multipoint Sub-network Connections may be modelled using the managed object class subNetworkConnection. The attribute mode is used to identify between the two types of Sub-network Connection. More complex connections are described below.

A multicast Sub-network Connection has multiple A ends, and multiple Z ends, and may be considered to be a set of superimposed point-to-multipoint Sub-network Connections, each with the same set of Z ends. To model a multicast Sub-network Connection, a Multicast Sub-network Connection object is used which contains a number of Sub-network Connections of type point to multipoint. There is one contained Sub-network Connection for each A end. The managed object class multicastSubNetworkConnection is defined in annex B as a candidate class for the library.

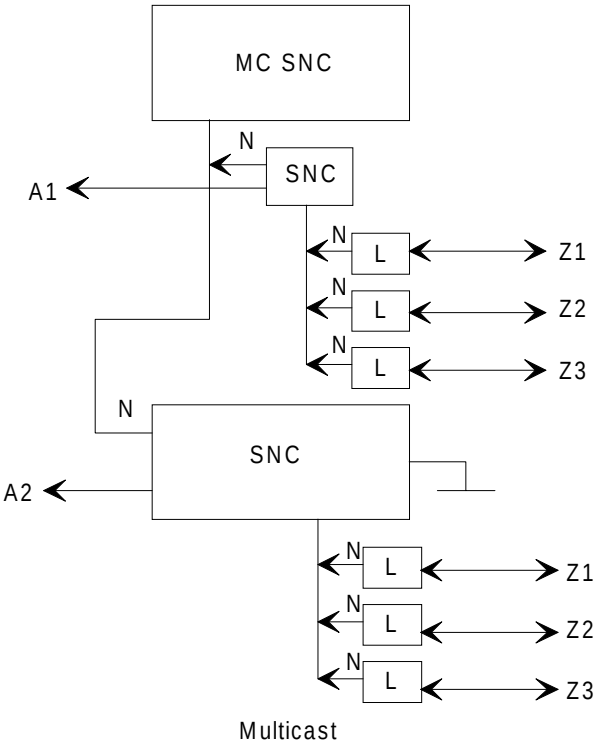


Figure D.3: Multicast sub-network connection composed of point to multipoint sub-network connections

Broadcast and conference Sub-network Connections require further study.

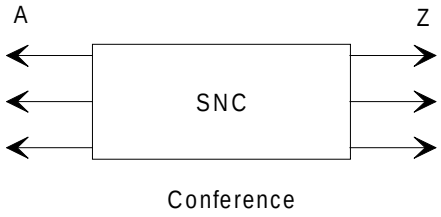


Figure D.4: Broadcast and conference sub-network connections

D.2.1.1 Multicast sub-network connection

```
multicastSubNetworkConnection MANAGED OBJECT CLASS
    DERIVED FROM    "Recommendation X.721 | ISO/IEC 10165-2 : 1992":top;
    CHARACTERIZED BY
        multicastSubNetworkConnectionPackage PACKAGE
            BEHAVIOUR
                multicastSubNetworkConnectionBehaviour BEHAVIOUR
                 DEFINED AS "The Multicast Sub-network Connection object class is a class of managed
                 objects which models a sub-network connection of mode multicast as a number of
                 point to multipoint SubNetworkConnections. When a Multicast Sub-Network Connection
                 is created, a separate (point to multipoint) Sub-network Connection will be created
                 for each A End. Each Sub-network Connection will have the same set of Z Ends. The
                 MulticastSubNetworkConnection thus contains a number of SubnetworkConnections.

                 A multicast unidirectional Subnetwork Connection can be established between a set
                 whose members are Network CTP sinks, Network CTP bids, Network TTP sources, Network
                 TTP bids or Network GTPs; and a set whose members are Network CTP sources, Network
                 CTP bids, Network TTP sinks, Network TTP bids or Network GTPs.

                 A multicast bi-directional Subnetwork Connection can be established between a set
                 whose members are Network CTP bids, Network TTP bids or Network GTPs; and a set
                 whose members are Network CTP bids, Network TTP bids or Network GTPs.

                 The MulticastSubNetworkConnection will contain one SubNetworkConnection for each A
                 End identified in the aEndNWTPList attribute. Each SubNetworkConnection will have
                 the same set of Z Ends, as identified in the zEndNWTPList attribute.";;
            ATTRIBUTES
                multicastSubNetworkConnectionId                         GET;;;
REGISTERED AS {es200653MObjectClass 28};


multicastSubNetworkConnectionId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX ES200653.NameType;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
        multicastSubNetworkConnectionIdBehaviour    BEHAVIOUR
                 DEFINED AS "The Multicast Subnetwork Connection Id is an attribute type whose
                 distinguished value can be used as an RDN when naming an instance of the Multicast
                 Subnetwork Connection object class.";;
REGISTERED AS {es200653Attribute 64};


subNetworkConnection-multicastSubNetworkConnection NAME BINDING
    SUBORDINATE OBJECT CLASS    subNetworkConnection AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   multicastSubNetworkConnection AND SUBCLASSES;
    WITH ATTRIBUTE      subNetworkConnectionId;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 31};


multicastSubNetworkConnection-subNetwork NAME BINDING
    SUBORDINATE OBJECT CLASS    multicastSubNetworkConnection AND SUBCLASSES;
    NAMED BY
        SUPERIOR OBJECT CLASS   subNetwork AND SUBCLASSES;
    WITH ATTRIBUTE      multicastSubNetworkConnectionId;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {es200653NameBinding 32};
```

```
setupMCastSubNetworkConnection ACTION
    BEHAVIOUR
        setupMCastSubNetworkConnectionBehaviour BEHAVIOUR
                DEFINED AS "This action is used to set up a Multicast Subnetwork Connection which
                contains a number of Subnetwork Connections. The number of contained Subnetwork
                Connections will be equal to the number of A ends. Each contained Subnetwork
                Connection will be of mode point to mulitpoint, and will have one A end and the
                same set of Z ends as all the other Subnetwork Connections in the Multicast
                Subnetwork Connection.

                If a Topological Point is involved in any of the contained Subnetwork Connections,
                its attributes idleNWCTPCount, connectedNWCTPCount and NWCTPsInTopologicalPointList
                will be updated as a result of this action.

                This action will fail if any of the network termination points specified is already
                involved in a Subnetwork Connection or if a NWTP which is part of an existing NWGTP
                is specified. The contained Subnetwork Connections will all have the same
                directionality (unidirectional or bi-directional) as specified in the action
                parameter sncDirectionality. The sncDirectionality parameter also specifies the end
                points of the Multicast Subnetwork Connection, and hence the end points of the
                contained Subnetwork Connections.

                The contained Subnetwork Connections shall have Status conditions of In Service
                Assigned, In Service Busy or In Service Reserved. If any of the underlying
                resources supporting a Subnetwork Connection have a Status condition of Unavailable
                Faulty Assigned or Unavailable Faulty Reserved, that Subnetwork Connection shall
                have the same Status condition.

                If the Subnetwork Connection parameters cannot be met by the server, the action
                response will indicate the values for the parameters which can be achieved by the
                server.

                The quality of service specifies one pre-determined set of transport parameters
                which the server may offer. Where a particular quality of transport service level
                is not available from the server, the action response will indicate the next lowest
                level in the pre-defined set of levels which is possible.

                The transactionId and the identifier of the client will be passed to the server and
                will be logged by the server against the identifier of the created Multicast
                Subnetwork Connection.";;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX     ES200653.SetupSubNetworkConnectionInformation;
    WITH REPLY SYNTAX           ES200653.SetupSubNetworkConnectionResult;
REGISTERED AS {es200653Action 19};


addToMCastSubNetworkConnection ACTION
    BEHAVIOUR
        addToMCastSubNetworkConnectionBehaviour BEHAVIOUR
                DEFINED AS "This action is used to add one or more legs to an existing Multicast
                Subnetwork Connection. Either or both A and Z End network termination points may be
                provided. If A End network termination points are added, then one new Subnetwork
                Connection object will be created for each A End. Each new Subnetwork Connection
                will be contained by the parent Multicast Subnetwork Connection object, and will
                have the same set of Z Ends as the existing Subnetwork Connections contained in the
                Multicast Subnetwork Connection. If Z End network termination points are added,
                then each new Z End shall be added to each existing Subnetwork Connection contained
                by the Multicast Subnetwork Connection. Additional Leg objects shall be created for
                each Z End which is new or is in a new Subnetwork Connection.

                Supplied network termination points or NWGTPs shall support a similar Signal Id to
                that of the network termination points already in the Subnetwork Connection. The
                result, if successful, always returns the network termination points or NWGTPs
                involved in the Subnetwork Connection.

                If a Topological Point is involved in the Subnetwork Connection, its attributes
                idleNWCTPCount, connectedNWCTPCount and NWCTPsInTopologicalPointList will be
                updated as a result of this action.";;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX     ES200653.AddToSubNetworkConnectionInformation;
    WITH REPLY SYNTAX           ES200653.AddToSubNetworkConnectionResult;
REGISTERED AS {es200653Action 20};
```

```
releaseMCastSubNetworkConnection ACTION
    BEHAVIOUR
        releaseMCastSubNetworkConnectionBehaviour BEHAVIOUR
                DEFINED AS "This action is used to release a Multicast Subnetwork Connection. This
                action will also release all of the contained Subnetwork Connections and all legs
                of the connections will be disconnected. The Subnetwork Connections pointed to by
                the compositePointer attribute will also be cleared down by this action.

                If a Topological Point is involved in any of the Subnetwork Connections, its
                attributes idleNWCTPCount, connectedNWCTPCount and NWCTPsInTopologicalPointList
                will be updated as a result of this action. The subNetworkConnectionPointer in the
                disconnected network termination points or NWGTPs will be set to NULL as a result
                of this action.";;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX     ES200653.ReleaseSubNetworkConnectionInformation;
    WITH REPLY SYNTAX           ES200653.ReleaseSubNetworkConnectionResult;
REGISTERED AS {es200653Action 21};


deleteFromMCastSubNetworkConnection ACTION
    BEHAVIOUR
        deleteFromMCastSubNetworkConnectionBehaviour BEHAVIOUR
                DEFINED AS "This action is used to delete part of a Multicast Subnetwork
                Connection. Network termination points representing A or Z Ends, or both may be
                deleted. If only Z ends are to be deleted, this will result in the specified Z Ends
                being deleted from each contained Subnetwork Connection and the corresponding Leg
                objects being removed. If A Ends are specified, then the Subnetwork Connections
                which connect to those A Ends will be removed, and their contained Leg objects will
                also be removed.

                The Subnetwork Connections pointed to by the compositePointer attribute will also
                be cleared down by this action.

                If a Topological Point is involved in the Multicast Subnetwork Connection, its
                attributes idleNWCTPCount, connectedNWCTPCount and NWCTPsInTopologicalPointList
                will be updated as a result of this action. The subNetworkConnectionPointer in the
                disconnected network termination points or NWGTPs will be set to NULL as a result
                of this action.";;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX     ES200653.DeleteFromSubNetworkConnectionInformation;
    WITH REPLY SYNTAX           ES200653.DeleteFromSubNetworkConnectionResult;
REGISTERED AS {es200653Action 22};
```

D.2.2 Tandem connection

```
tandemConnection                MANAGED OBJECT CLASS
    DERIVED FROM                connectivity,
    CHARACTERIZED BY            tandemConnectionPackage,
    BEHAVIOUR DEFINITION        tandemConnectionBehviour;
    ATTRIBUTES
        tandemConnectionId                                          GET,
        allocationPtrList                                           GET,
    PACKAGES
        relationshipChangeNotificationPackage,
    CONDITIONAL PACKAGES
        monitoringPackage   PACKAGE
        PRESENT IF "both aEndTP and zEndTP provide the facility to be monitored"
    ACTIONS
        connectAll,
        disconnectAll;
REGISTERED AS   { es200653MObject30}
        tandemConnectionBehaviour BEHAVIOUR
                DEFINED AS "The tandemConnection is used to comprise these connections belonging to
                a service. A tandemConnection can be triggered to initiate monitoring if the a/zEnd
                TPs of the concerning connections support monitoring."

tandemConnectionId  ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   ES200653.NameType;
    MATCHES FOR Equality;
REGISTERED AS {es200653Attribute 65}

allocationPtrList   ATTRIBUTE
    WITH ATTRIBUTE SYNTAX       ES200653.AllocationPtrList;
    MATCHES FOR Equality;
    BEHAVIOUR   allocationPtrListBehaviour
                DEFINED AS "This attribute points to the server connectivity instances."

AllocationPtrList::= SET OF OBJECTINSTANCE
```

```
monitoringPackage   PACKAGE
    ATTRIBUTE
        monitoring                                                  GET-REPLACE,
    BEHAVIOUR   monitoringPackageBehaviour
                DEFINED AS "With the contained attribute tandem Connection monitoring can be
                switched on and of"

monitoring::= ENUMERATE {
            off(0),
            on(1);
            }

connectAll  ACTION
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX     ES200653.ConnectAllInformation;
    WITH REPLY SYNTAX           ES200653.ConnectAllResult;
    BEHAVIOUR   connectAllBehaviour
                DEFINED AS "This action is used to connect all connectivities contained within the
                transport object. On success the result is empty, on failure the result contains
                these connectivity instances which failed."

disconnectAll   ACTION
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX     ES200653.ConnectAllInformation;
    WITH REPLY SYNTAX           ES200653.ConnectAllResult;
    BEHAVIOUR   connectAllBehaviour
                DEFINED AS "This action is used to disconnect all connectivities contained within
                the transport object. On success the result is empty, on failure the result
                contains these connectivity instances which failed."

ConnectAllInformation::=    NULL
ConnectAllResult::= SET OF SEQUENCE {
                objectInstance  OBJECTINSTANCE,
                ProblemCause    OPTIONAL;
                }
```

D.3 Alarm reporting

This attribute contains the time an alarm condition shall persist until a communications alarm is generated.

```
alarmPersistenceTime    ATTRIBUTE
    WITH ATTRIBUTE SYNTAX       ES200653.AlarmPersistenceTime;
    MATCHES FOR EQUALITY;
    BEHAVIOUR   alarmPersistenceTimeBehaviour
                DEFINED AS "This attribute determines the time an alarm has to be permanently
                persistent until a communicationsAlarm notification is sent, or a protection
                switching takes place. time = (value of alarmPersistemceTime) * 100 ms. The value 0
                indicates the alarm is sent after the shortest possible time needed for identifying
                the alarm.
REGISTERED AS {es200653Attribute 66}

AlarmPersistenceTime::= INTEGER(0..255)
```

This attribute contains a SET whether primary and secondary alarms should be suppressed. This attribute enables an OS to suppress alarms more preferment than with a CMISE filter.

```
inhibitNWCommunicationsAlarm    ATTRIBUTE
    WITH ATTRIBUTE SYNTAX       ES200653.inhibitNWCommunicationsAlarm;
    MATCHES FOR Equality;
    BEHAVIOUR   inhibitNWCommunicationsAlarmBehaviour
                DEFINED AS "This attribute contains a SET whether primary and secondary alarms
                should be suppressed."
REGISTERED AS {es200653Attribute 67}

inhibitNWCommunicationsAlarm::= SET OF {
                            primary     [1]     INTEGER(1),
                            secondary   [2]     INTEGER(2);
                                }
```

Annex E (informative): Representation of multipoint connections following ITU-T Recommendation I.326

E.1 Introduction

Two alternative methods of representing multipoint connections are possible. The first follows the principles of ITU-T Recommendation M.3100 [10], and the second follows ITU-T Recommendation I.326 [16].

The first method is defined in annex D, and the second in this annex.

This approach is still under study, and only candidate solutions are presented here.

E.2 Summary of I.326 model

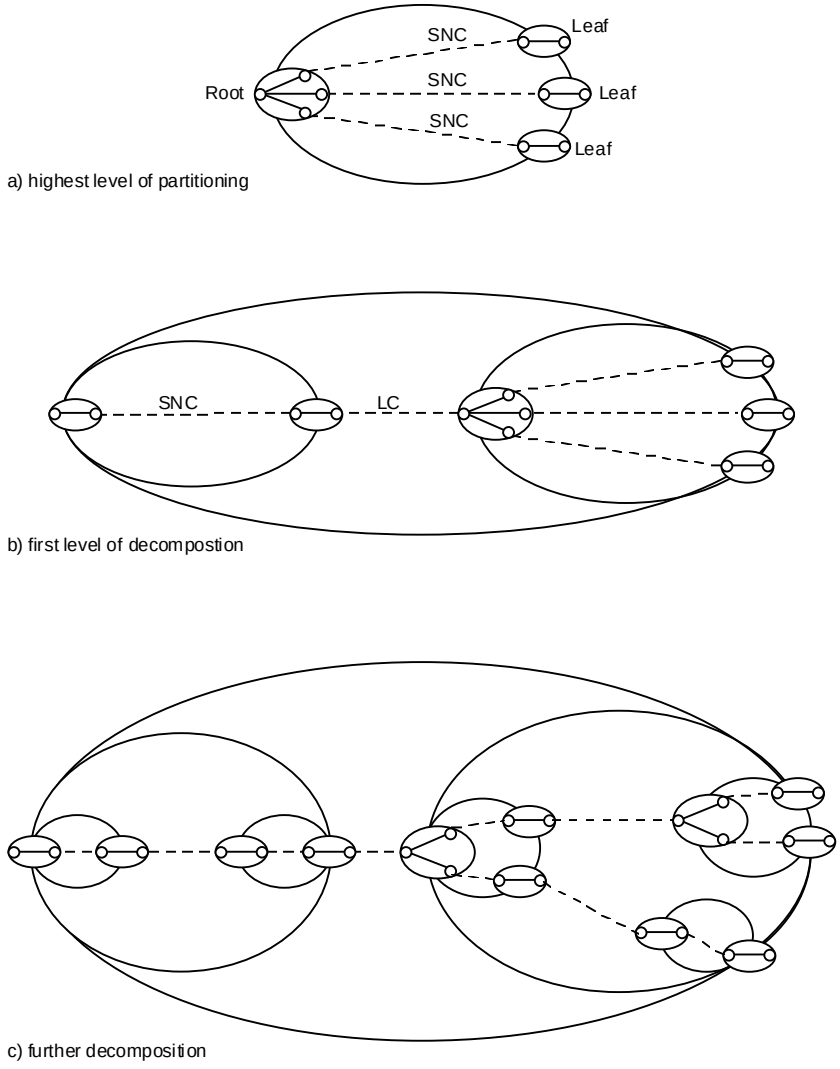


Figure E.1: Decomposition of a Multipoint connection

Only point-to-point subnetwork connections and link connections are used.

The multipoint capability is given by a multipoint networkTP which may point to number of connectivity instances.

E.3 Modelling implications

The addToSubNetworkConnection and deleteFromSubnetworkConnection Actions are no longer required.

New Actions addToMultipointConnection, and deleteFromMultipointConnection are required instead.

The release subnetworkConnection action is modified to cover both cases.

The Leg object (and associated attributes) is not required.

The mode attribute in Connectivity is defaulted to point-to-point. The mode attribute in the networkTP represents the mode of the multipoint connection.

Only point-to-point modes in the ConnectivityDirectionality syntax will be supported.

The root is modelled as networkTP. A multipoint termination point is modelled from the existing Network TP by allowing the subnetwork connection pointer to be multi-valued.

Since each branch of the multipoint is a sub-network connection, it may carry an individual status condition, schedule, quality of network service, and bandwidth allocation. In some applications this information will be identical for each branch of the multipoint connection. In this case the root (networkTP) will be subclassed to contain this information, and the subnetwork connections will not contain this. This extension is for further study.

Alternative Modelling Approaches

There is a problem with using networkTP as the root object. As figure E.1 shows a root may be partitioned into a single networkTP at a lower level of partitioning. Each subnetwork connection of the branch points to the lower level TP, but since the snc pointer of the lower level TP is null for multiple partitioning levels, this TP can no longer maintain the integrity of the multipoint via a multi-valued snc pointer. Effectively the root is a part of the higher level subnetwork and can't be referenced from a lower level. To solve this problem it is proposed to use a new root object as described below:

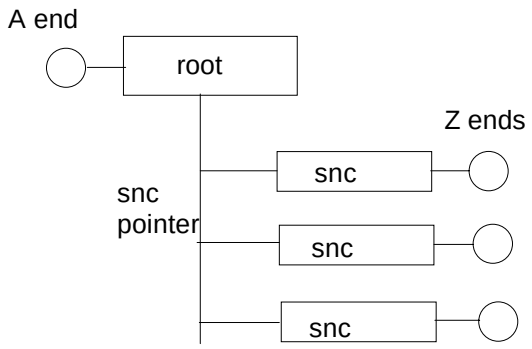


Figure E.2: Modelling of the root of a multipoint connection

PROFILE NOTE: The connectivityPointer package is not used.

```
root MANAGED OBJECT CLASS
    DERIVED FROM
        networkTP;
    CHARACTERIZED BY
        sncPointerPackage
        "Recommendation M.3100 : 1992":createDeleteNotificationsPackage,
        rootPackage PACKAGE
            BEHAVIOUR
        rootBehaviour BEHAVIOUR
                DEFINED AS "This managed object represents the root of a multipoint connection
                defined according to ITU-T Recommendation I.326. The root is the Aend of the
                muliple subnetwork connections which make up the multipoint connection" ";;

            ATTRIBUTES
                aEndNWTPList                                                GET,
;;;
REGISTERED AS {es200653.MObjectClass ??};
```

NOTE: Root has a namebinding to subnetwork.

A second alternative is to allow the subnetwork connections to point directly to the Aend:

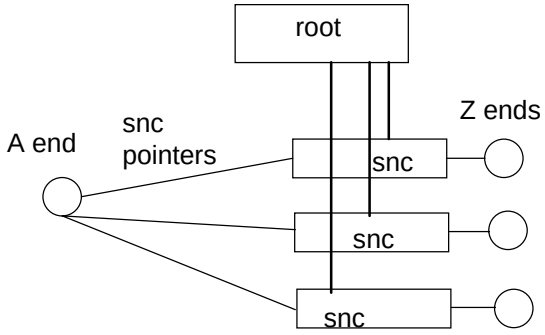


Figure E.3: Modelling of the root of a multipoint connection

These solutions will be resolved in a later version of the present document.

E.4 Candidate actions

The following candidate actions are used to support the above approaches.

Add to multipoint connection

```
addToMultipointConnection ACTION
            BEHAVIOUR
        addToMultipointConnectionBehaviour BEHAVIOUR
                DEFINED AS "This action is used to add one or more legs to an existing sub-network
                Connection of type point to multipoint or multicast. If the action is used on a
                point to point Sub-network Connection, the Sub-network Connection becomes point to
                multipoint. Additional Z End network termination points shall be provided, and Leg
                objects will be created for each Z End, including the Z End of the original point
                to point Sub-network Connection. For addition to a point to point or point to
                multipoint Sub-network Connection, Z End network termination points shall be
                provided. One additional Leg object will be created for each new Z End network
                termination point.

                For addition to a multicast Sub-network Connection, either or both A and Z End
                network termination points may be provided. If A End network termination points are
                added, then one new Sub-network Connection object will be created for each A End.
                Each new Sub-network Connection will be contained by the parent Multicast sub-
                network Connection object, and will have the same set of Z Ends as the existing
                Sub-network Connections contained in the Multicast Sub-network Connection. If Z End
                network termination points are added, then each new Z End shall be added to each
                existing Sub-network Connection contained by the Multicast Sub-network Connection.
                Additional Leg objects shall be created for each Z End which is new or is in a new
                Sub-network Connection.

                Supplied network termination points or NWGTPs shall support a similar Signal Id to
                that of the network termination points already in the Sub-network Connection. The
                result, if successful, always returns the network termination points or NWGTPs
                involved in the Sub-network Connection.

                If a Topological Point is involved in the Sub-network Connection, its attributes
                idleNWCTPCount, and connectedNWCTPCount will be updated as a result of this
                action.";;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX ES200653.AddToMultipointConnectionInformation;
    WITH REPLY SYNTAX          ES200653.AddToMultipointConnectionResult;
REGISTERED AS {es200653Action 30};
```

Delete from multipoint connection

```
deleteFromMultipointConnection ACTION
            BEHAVIOUR
    deleteFromMultipointConnectionBehaviour BEHAVIOUR
                DEFINED AS "This action is used to delete a leg from a Sub-network Connection,
                providing it is not the last remaining leg in the Sub-network Connection. In that
                instance, the action ReleaseSubNetworkConnection shall be used. To delete a leg
                from a point to multipoint Sub-network Connection, Z End network termination points
                shall be provided. To delete a leg from a multicast Sub-network Connection, either
                or both A and Z End network termination points may be provided. To delete a leg
                from a conference Sub-network Connection, A End network termination points shall be
                provided. The Sub-network Connections pointed to by the compositePointer attribute
                will also be cleared down by this action.

                If a Topological Point is involved in the Sub-network Connection, its attributes
                idleNWCTPCount, and connectedNWCTPCount will be updated as a result of this action.
                ";;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX     ES200653.DeleteFromMultipointConnectionInformation;
    WITH REPLY SYNTAX           ES200653.DeleteFromMultipointConnectionResult;
REGISTERED AS {es200653Action 31};
```

Setup multipoint connection

PROFILE NOTE: There are five basic forms of multipoint connection- point-to-point, point-to-multipoint, multicast, broadcast and conference.

This action may be used to set up any of the first three types; the setup action for broadcast and conference Multipoint Connections requires further study.

This approach, following ITU-T Recommendation I.326 [16], uses point-to-point subnetwork connections and a multipoint root, as described in annex E, to set up a multipoint connection. If a point-to-point connection is required, the setupSubNetworkConnection action is used.

An alternative approach where the setup is effected by creation of a point to point, point to multipoint, or multicast subnetwork connection is described in annex D.

Timeout and holdtime are defined as INTEGER time intervals. It is the responsibility of application groups to determine what the unit of time interval is (e.g. milliseconds, seconds).

Where the subnetworkConnection is setup between accessGroups and/or topological points, the directionality is specified from the ConnectivityDirectionality defined in the SetupMultipointConnectionInformation.

```
setupMultipointConnection ACTION
    BEHAVIOUR
        setupMultipointConnectionBehaviour BEHAVIOUR
                DEFINED AS "This action is used to set up a Multipoint Connection between network
                termination points or network GTPs. The termination points to be connected can be
                specified in one of two ways:

                (1) by explicitly identifying the network termination points or NWGTPs,
                (2) by specifying one or more Topological Points or Access Groups from which    any
                idle network termination point or NWGTP may be used.

                The result, if successful, always returns an explicit list of NWTPs or NWGTPs. The
                multiple subnetwork connections of a Multipoint connection may be established in
                any of the following Status Conditions:

                -   planned (1);
                -   in service, not allocated (2);
                -   in service, reserved (4);
                -   in service with no spare capacity (8);
                -   in service with no spare capacity, under test (9).

                Status Condition (8) is the default. Other Status Conditions shall be explicitly
                expressed in set-up Multipoint connection action.
```

If it is set up as In Service Reserved, this permits all resources involved in the Multipoint Connection to be reserved in sequence, and when all have been reserved the entire Multipoint Connection may be activated by invoking the action ActivateMultipointConnection. The Status condition of all network termination points, Link connections and subnetwork Connections involved in the Multipoint Connection being will be the same.

One subnetwork connection object will be created for each Z End (leaf)in a Multipoint Connection. A root (networkTP) will be created for each Aend. If a Topological Point is involved in the Multipoint Connection, its attributes idleNWCTPCount, and connectedNWCTPCount will beupdated as a result of this action.

This action will fail if any of the network termination points specified is already involved in asubnetwork Connection or if a NWTP which is part of an existing NWGTP is specified.

The subnetwork connections will have a directionality (unidirectional or bi-directional) as specified in the action parameter sncDirectionality. The sncDirectionality parameter also specifies the end points of the multiple subnetwork connections.

If any of the underlying resources supporting one of the multiple subnetwork connections have a Status condition of Resource Failed with no spare capacity (10c) or Resource Failed, Reserved (10a), the subnetwork Connection shall have the same Status condition.

If used, the quality of connectivity service specifies one pre-determined set of transport parameters which the server may offer. The optional timeout and holdtime parameters are used as part of a two-phase set-up process.

Timeout is the time allowed to the agent multipoint to respond to the set-up request from the manager. This avoids the manager being slowed down by waiting for unacceptable periods of time for an agent response.

Holdtime is the time interval which the agent multipoint waits for an activate ACTION once it has entered the reserved state. This allows the agent to free resources if the manager is slow to complete the two phase process.

If they are used, transactionId and the identifier of the client will be passed to the server and will be logged by the server against the identifier of the created subnetwork Connections. When a bandwidth-scheduled multipoint connection is requested, the bandwidth scheduling parameter is used. The subnetwork, will create multiple subNetworkConnection object instances. These objects will have instantiated the package associated for the type of scheduling requested (e.g. weeklySchedulePkg if it requested for a weekly scheduled connection). That package will contain the schedule itself and the appropriate actions to modify the bandwidth schedule (add, delete and modify time slots) without the need of clearing down the connection and re-establishing the multiple subnetwork connections.

StartTime	StopTime	Condition
NULL	NULL	duration schedule is only vaild CHOICE (i.e.set-up is immediate and has no defined end)
NULL	GeneralizedTime	reservation period begins immediately, and terminates at StopTime
GeneralizedTime	NULL	reservation period begins at StartTimeand has no defined end

The subnetwork shall guarantee that resources will be available when the multipoint connection is due to be activated.

The ACTION replies for set-up includes full information about the reasons in case the request could not be satisfied (lack of resources, overlapping time slots, etc.).

The "in traffic" condition of the subNetworkConnection is driven by the schedule. A scheduled connection is set-up in the In Service, Not allocated (4) Status Condition. When the schedule indicates that the subnetwork connection is to be put in traffic, the Status Condition changes to In Service with no spare capacity (8) (preceded by the In Service with no spare capacity, under test (9) Status Condition if an initial test is made).

In a two-phase set-up comprising reservation and activation, the subnetwork connections are set-up in the In Service, Reserved (4) Status Condition at the time dictated by the schedule, pending an Activate Action from the manager.

```
                The default value of the implicit creation of TPs parameter is FALSE. That is, by
                default, the subnetwork requires NWTPs to be in existence before a multipoint
                connection can be made. Only if the implicit creation parameter is set to be TRUE
                in the set-up multipoint connection request, will implicit NWTP creation occur.

                The identities of the created NWTPs are returned in the result. The EndPno
                parameter is used when it is necessary to specify a destination PNO when a step-by-
                step set-up process is used for inter TMN applications.";;

    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX     ES200653.SetupMultipointConnectionInformation;
    WITH REPLY SYNTAX           ES200653.SetupMultipointConnectionResult;
```

REGISTERED AS {es200653Action 32};

E.5 Terminology

The terminology used in this class library and in ITU-T Recommendation I.326 [16] is slightly different. The mapping between the terms is given in the table below.

Further details may be found in subclause B.1.8, and for ITU-T Recommendation I.326 [16] the modes are detailed below.

Network Level View class library Mode	ITU-T Recommendation I.326 [16] Mode	Directionality
point-to-point	point-to-point	uni or bi-directional
point-to-multipoint	composite	bi-directional
point-to-multipoint	merge	unidirectional
multicast	full multipoint	bi-directional
conference	full multipoint	bi-directional
broadcast	broadcast	unidirectional

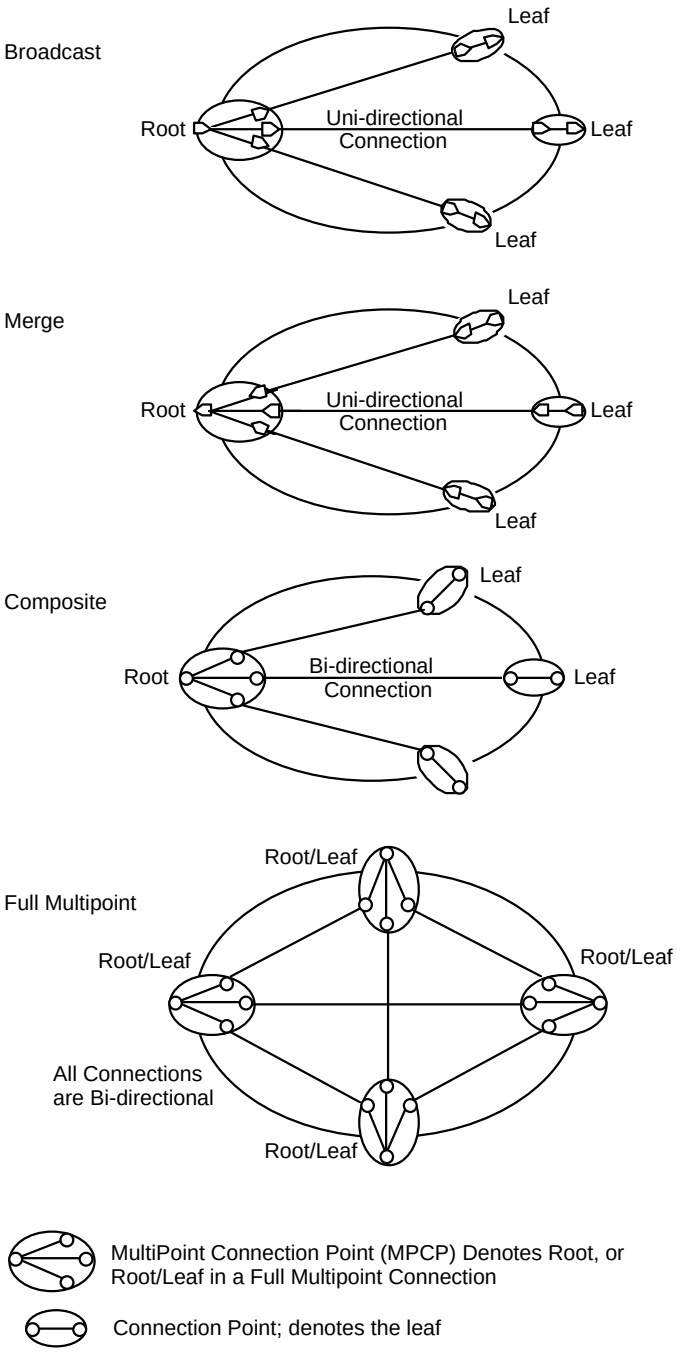


Figure E.4: Connection Modes in ITU-T Recommendation I.326 [16]

Annex F (informative): The ensemble technique

F.1 Introduction

There is a dilemma for the Generic Model between creating a description which is sufficiently wide to cope with all the applications envisaged, but allowing a very precise description for the individual applications. The ensemble concept has been defined by the NMF Forum 25 specification [14], and the format of an Ensemble is illustrated in table F.1.

Table F.1

```
                ENSEMBLE STRUCTURE

        Requirements
        Constraints

        Scenarios
        Identify services,
            resources,
            abstractions,
            functions


        Identify entities
        E-R diagrams, relationships

        Identify management information elements
                ISPs etc.

        Implementation
        MANAGED OBJECT CLASSS, PICS, etc.
        Real object on real systems
```

F.2 Use of ensembles in ETSI

An Ensemble describes a particular solution to a particular problem, sufficient to permit interworking. This format is particularly useful for the technology-specific applications being progressed by other groups.

Use of the Ensemble method of documentation eases the design process when specializing class library, and permits a more rigorous definition of the management problem.

One of the tests of the generic model is how well it satisfies the detailed requirements of the other groups. As more Ensembles are created by the technology groups, ETSI will concentrate on adapting the candidate classes to meet these requirements, and extracting the generic features.

Annex G (informative): Alternative modelling approach

G.1 Matrix

```
matrix                                      MANAGED OBJECT CLASS
    DERIVED FROM                        subNetworkBaseNI
    CHARACTERIZED BY                matrixPackage PACKAGE
    BEHAVIOUR                               matrixBehaviour;
        ATTRIBUTES
            matrixId                                                    GET,
            netAdress                                                   GET,
            linkPtrList                                                 GET,
            trailPtrList                                                GET;
        CONDITIONAL PACKAGES
            userListPackage
                PRESENT IF "this matrix object is instantiated on the G.805 circuit layer";
            getTpIdActionPackage
                PRESENT IF "the connection/tandemConnection/trail instances don"t provide
                a/zEndTP identifier"
        ACTIONS
            connect,
            disconnect;
REGISTERED AS {es200653MObjectClass 29};
        matrixBehaviour  BEHAVIOUR
                DEFINED AS "The matrix managed object class represents a network element the
                characteristic information it supports on a G.805 network layer. The
                supportedByObjectlist attribute inherited from its base class points to the
                representation of a network element managed by an element manager. The userLabel
                contains the user friendly name of the network element pointed to by the
                supportedByObjectList attribute by default. The statePackage represents the actual
                state of the network element An attributeValueChance notification will be issued
                when a link/trail instance will be added or removed. A stateChange notification
                will be issued when the state changes. On creation/deletion of a matrix instance a
                create/delete notification will be generated. The actions connect/disconnect
                connects the participated transport objects. The action getTpId requires a
                transport object identifier and returns the distinguished name of the supporting TP
                represented by the element manager. The userList contains the userFriendly name of
                a port assigned to a customer"
```

G.2 Connectivity

```
Make ATTRIBUTE "a/zENDNWTPList" CONDITIONAL:
CONDITIONAL PACKAGES
    aEndNWTPList        PACKAGE     PRESENT IF "an instance supports it";

aEndNWTPListPackage PACKAGE
    ATTRIBUTE
        aEndNWTPList                                                    GET;
REGISTERED AS {es200653Package 49}
```

G.3 Attribute definitions

```
Matrix ID
matrixId    ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   ES200653.NameType
    MATCHERS FOR EQUALITY
    BEHAVIOR
        matrixIdBehaviour BEHAVIOUR
                DEFINED AS "The Matrix Id is an attribute type whose distinguished value can be
                used as an RDN when naming an instance of the Matrix object class."
REGISTERED AS {es200653Attribute 68}
```

```
Link Pointer List
linkPtrList ATTRIBUTE
    WITH ATTRIBUTE SYNTAX       ES200653.PtrList
    MATCHERS FOR EQUALITY
    BEHAVIOR
        linkPtrListBehaviour BEHAVIOUR
                DEFINED AS "The linkPtrList contains all instances of link managed object
                terminated at this matrix instance"
REGISTERED AS {es200653Attribute 69}

Trail Pointer List
trailPtrList    ATTRIBUTE
    WITH ATTRIBUTE SYNTAX       ES200653.PtrList
    MATCHERS FOR EQUALITY
    BEHAVIOR
        trailPtrListBehaviour BEHAVIOUR
                DEFINED AS "The trailPtrList contains all instances of trail managed object
                terminated at this matrix instance"
REGISTERED AS {es200653Attribute 70}

User List

userList    ATTRIBUTE
    WITH ATTRIBUTE SYNTAX       ES200653.userList
    MATCHERS FOR SETINTERSECTION,SET COMPARISON
    BEHAVIOR
        userListBehaviour BEHAVIOUR
                DEFINED AS "The userList contains a set of user friendly names which have a port at
                this matrix instance"
REGISTERED AS {es200653Attribute 71}

Network Adress
netAdress   ATTRIBUTE
    WITH ATTRIBUTE SYNTAX       ES200653.netAdress
    MATCHES FOR EQUALITY
    BEHAVIOUR   netAdressBehaviour BEHAVIOUR
                DEFINED AS "The netAdress attribute contains the network Adress of a certain
                network element within an element manager."
```

G.4 Package definitions

```
User List Package
userListPackage PACKAGE
    ATTRIBUTES
        userList                                                    GET;
REGISTERED AS {es200653Package 50}
```

G.5 Actions definitions

```
Connect
connect ACTION
    BEHAVIOUR
        connectBehaviour BEHAVIOUR
                DEFINED AS "This action is used to connect a connection/connection or
                connection/trail relation. It invokes the element manager to connect the TPs
                assigned to the transport objects involved. If the underlying resource is not in
                the appropriate state this action fails. On success the result is NULL, on fail the
                result contains these MANAGED OBJECT CLASS instances on which the action failed."
    MODE    CONFIRMED,
    WITH INFORMATION SYTAX      ES200653.connectionInformation,
    WITH REPLY SYNTAX           ES200653.connectioniResult;
REGISTERED AS {es200653Action 23}

Disconnect
disconnect  ACTION
    BEHAVIOUR
        disconnectBehaviour BEHAVIOUR
                DEFINED AS "This action is used to disconnect a connection/connection or
                connection/trail relation. It invokes the element manager to disconnect the TPs
                assigned to the transport objects involved. If the underlying resource is not in
                the appropriate state this action fails. On success the result is NULL, on fail the
                result contains these MANAGED OBJECT CLASS instances on which the action failed."
    MODE    CONFIRMED,
    WITH INFORMATION SYTAX      ES200653.connectionInformation,
    WITH REPLY SYNTAX           ES200653.connectionResult;
REGISTERED AS {es200653Action 24}
```

```
Get TP Id
getTPId ACTION
    BEHAVIOUR
        getTPIdBehaviour BEHAVIOUR DEFINED AS
"This action is used to retrive the distinguished name of TPs terminated at transport objects."
    MODE    CONFIRMED,
    WITH INFORMATION SYTAX      ES200653.getTPIdInformation,
    WITH REPLY SYNTAX           ES200653.getTPIdResult;
REGISTERED AS {es200653Action 25}
```

G.6 ASN.1 productions

```
connectionInformation::=    SET OF SEQUENCE{
        nearEndConnectivityObject   [1] OBJECTINSTANCE,
        farEndConnectivityObjects   [2] SET OF OBJECTINSTANCE
        }

connectionReply::=  CHOICE {
        NULL,
        connectionInformation;
        }

getTPIdInformation::=   SET OF OBJECTINSTANCE
getTPIdReply::= SET OF TransportObjectPair
netAdress::=    PrintableString[0..80]
```

EXAMPLE: AE_NAME / AE_TITLE / AE_QUAL / NetworkAdress / TSAP/ SSAP/ PSAP
agent {1 3 6666 3 9} 1/ 4700040006000808002B06251201/ 1302 /
0003 / 0001

NOTE: The maximum string length and formatting is for further study.

```
TransportobjectPair::=  SEQUENCE OF {
        connectivityObject  [1]     OBJECTINSTANCE
        tpObject            [2]     OBJECTINSTANCE;
        }
```

History

Document history		
Edition 1	May 1996	Publication as I-ETS 300 653
V1.2.1	February 1999	Membership Approval Procedure MV 9914: 1999-02-02 to 1999-04-02