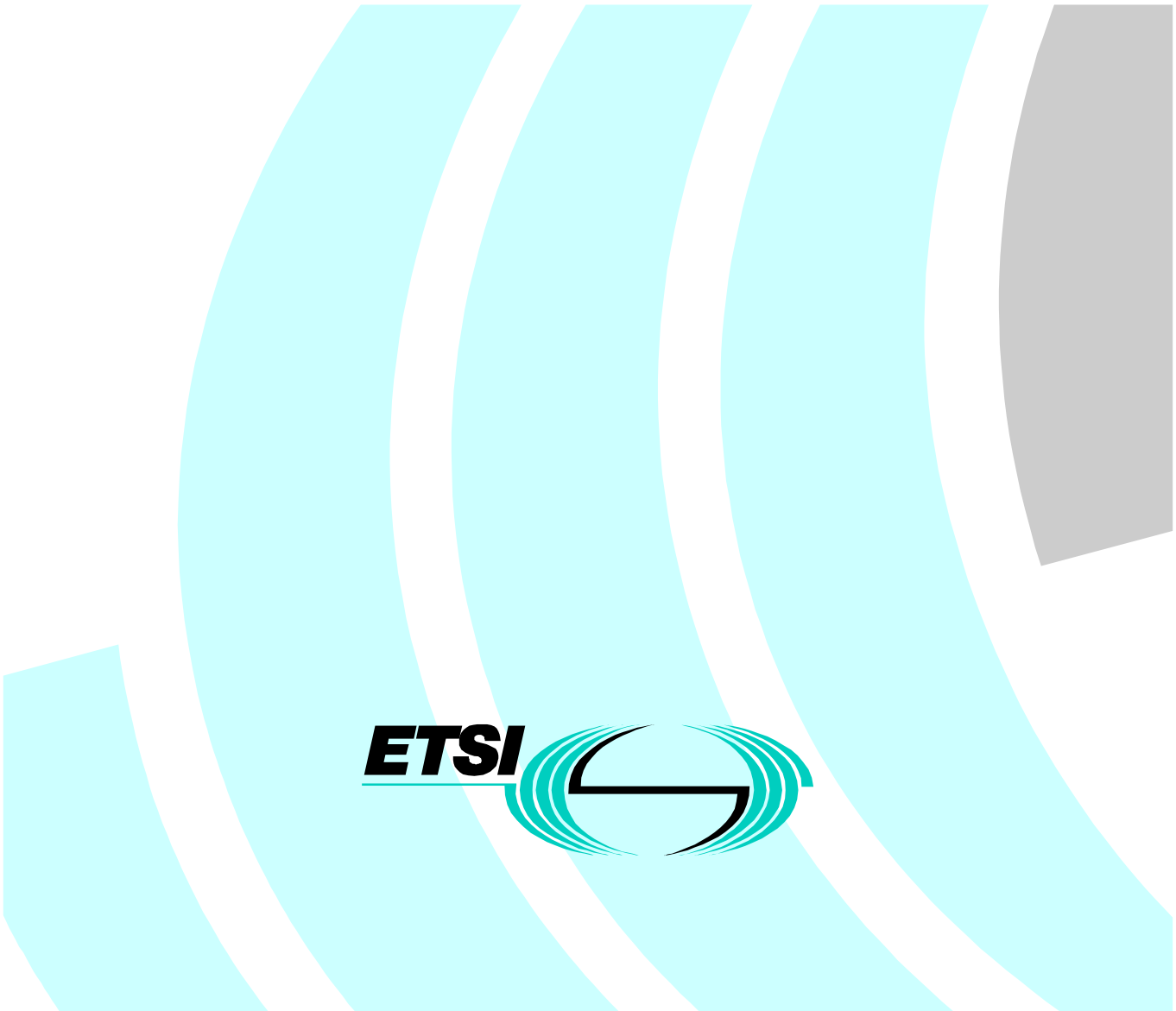
Final draft ETSI EN 300 820-1 V1.1.4 (2000-09)

European Standard (Telecommunications series)

Telecommunications Management Network (TMN); Asynchronous Transfer Mode (ATM) management information model for X interface between Operation Systems (OSs) of a Virtual Path (VP)/Virtual Channel (VC) cross connected networks; Part 1: Configuration Management



Reference
REN/TMN-GOM003-1

Keywords
ATM, B-ISDN, configuration, interface,
management, switching, TMN

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - 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

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 ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available 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 ETSI 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 European Standard (Telecommunications series) has been produced by ETSI Technical Committee Telecommunications Management Network (TMN), and is now submitted for the Voting phase of the ETSI standards Two-step Approval Procedure.

The present document is part 1 of a multi-part deliverable covering the management information model for the X-type interface between Operations Systems (OSs) of a Virtual Path (VP)/Virtual Channel (VC) cross connected network, as identified below:

Part 1: "Configuration management";

Part 2: "Alarm management";

Part 3: "VP Performance management".

(VC Performance Management aspects are for further study).

Proposed national transposition dates	
Date of latest announcement of this EN (doa):	3 months after ETSI publication
Date of latest publication of new National Standard or endorsement of this EN (dop/e):	6 months after doa
Date of withdrawal of any conflicting National Standard (dow):	6 months after doa

1 Scope

The present document addresses the requirements of network and service providers of Asynchronous Transfer Mode (ATM) cross connected networks for establishing, maintaining and releasing Virtual Path (VP) and / or Virtual Channel (VC) connections (generally denoted as 'VP/VC connections' in the present document), which span several administrative ATM domains. These requirements are satisfied by the use of a standardized interface (the "X-interface") between Operation Systems belonging to different network operators.

Readers of the present document should be made aware that the abbreviation 'PNO' is taken to mean Providing Network Operator. In the previous version of the present document (and related documents), PNO was defined as Public Network Operator. The change in definition has been provided to reflect the change in market conditions for provision of interconnected telecommunications services. However, it is considered necessary to retain the abbreviation 'PNO' because it is found in many of the managed object definitions used to specify the X-interface. It would be disadvantageous to introduce major changes in these managed object definitions, which serve purely technical purposes for management of interconnections only.

The present document contains a general overview describing the different management areas that will be covered in the different X-interface ENs - configuration, alarm and performance - as well as the relationships between them.

The present document describes the configuration management area covering the following aspects:

- a management architecture that shows how the X-interface is to be used between service or network providers;
- the management services and functions needed to manage ATM connections, which span several administrative domains. These management services and functions cover the requirements for the X- interface;
- the management information crossing the X-interface. This management information specification uses the Guidelines for the Definition of Managed Objects GDMO formalism, described in ITU-T Recommendation X.722 [4].

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] ITU-T Recommendation M.3100: "Generic network information model".

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

[3] ITU-T Recommendation I.751 (1996): "Asyncronous Transfer Mode Management Of The Network Element View".

[4] ITU-T Recommendation X.722: "Guidelines for the definition of managed objects for ITU-T applications".

[5] ETSI EN 300 820-2: "Telecommunications Management Network (TMN); Asynchronous Transfer Mode (ATM) management information model for the X interface between Operation Systems (OSs) of a Virtual Path (VP)/Virtual Channel (VC) cross connected networks; Part 2: Alarm management".

[6] ITU-T Recommendation I.630 (1999): "Integrated Services Digital Network, Maintenance Principles: ATM Protection Switching" (Prepublished Recommendation).

[7] ITU-T Recommendation M.3010 (1996): "Principles for a Telecommunications management network".

[8] ETSI EN 300 820-3: "Telecommunications Management Network (TMN); Asynchronous Transfer Mode (ATM) management information model for the X interface between Operation Systems (OSs) of a Virtual Path (VP)/Virtual Channel (VC) cross connected networks; Part 3: VP Performance management".

[9] ETSI ES 200 653 (V1.2.1): "Telecommunications Management Network (TMN); Network level generic class library".

[10] ITU-T Recommendation X.721: "Information technology - Open Systems Interconnection - Structure of management information: Definition of management information".

[11] ISO/IEC 10165-2: "Information technology - Open Systems Interconnection - Structure of management information: Definition of management information ".

[12] ETSI ETS 300 469 (1997): "Broadband Integrated Services Digital Network (B-ISDN); Asynchronous Transfer Mode (ATM); Management of the network element view [ITU-T Recommendation I.751 (1996)]".

[13] ETSI EN 300 820-1 (V1.1.1): "Telecommunications Management Network (TMN); Management information model for the X-type interface between Operation Systems (OSs) of a Virtual Path (VP)/Virtual Channel (VC) cross connected network; Part 1: Configuration management aspects".

[14] ITU-T Recommendation I.356: "B-ISDN ATM Layer Cell Transfer Performance".

3 Definitions and abbreviations

3.1 Definitions

(Some definitions depend on the future acceptance of the "cascaded/mixed mode" as described in annex D. This dependence is already taken into account in these definitions).

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

A PNO: PNO where the Initiator starts reserving the end-to-end VP or VC connection

NOTE 1: If the termination point of the connection is at a User access point, this is considered to be the **A User** Access point. The A PNO can be the Initiating PNO, but this is not always the case. It is the Consumer of other PNO's parts of the VP/VC connection.

ATM Accesspoint: means by which a subnetwork offers external interfaces to other subnetworks

NOTE 2: It is associated with an end point of an IPPL.

ATM Interconnection Gateway: ATM access point in one subnetwork, which is associated with an ATM access point in another subnetwork for the purpose of topological interconnection

connection: "transport entity" which is capable of transferring information transparently between "connection points (CP)"

NOTE 3: A "connection" defines the association between the "connection points" and the "connection points" delimit the "connection".

consumer and provider roles of a PNO: with respect to a particular End-to-end connection, a Consumer PNO is a PNO that has delegated the management of a subnetwork connection plus the outgoing link connection to another PNO (being a Provider PNO)

NOTE 4: If, in future, the "cascaded/mixed" mode should be accepted (annex D) a PNO can have both roles at the same time, if it is providing part of the End-to-end connection (being a Provider), and at the same time asks another PNO to provide a part of the End-to-end connection (being a Consumer).

end: point where a connection terminates and which is associated with an address

NOTE 5: The address should be specified in accordance with ITU-T Recommendation M.1400 [2].

end-to-end Connection: overall Connection which can be one of the following types: User-to-user VPC / VCC, Network-to-user VPC, User-to-network VPC or Network-to-network VPC)

NOTE 6: These types are defined in the remainder of this subclause.

NOTE 7: For a given end-to-end connection, any PNO may act in any of the I, A, T or Z PNO roles according to the interconnection requirements to provide the service.

EXAMPLE 1: Figures 2 and 3 provide some examples of end-to-end connections.

Inter PNO Physical Link (IPPL): represents a physical link that offers bidirectional transmission capabilities and connects two subnetworks. Each InterPNOPhysicalLink is terminated by two ATM Access Points which are in charge of emitting failures related to the link or to the access point itself. An IPPL can be realized by any transmission capability (SDH, PDH etc.)

I PNO: initiating PNO. It is the PNO requesting for a particular ATM end-to-end connection. It starts requesting in the subnetwork of the A PNO and ends in the subnetwork of the Z PNO. It controls the end-to-end connection

link: "topological component" which describes the fixed relationship between a "sub-network" and another "sub-network" or "access group". It is defined by an access point on one sub-network, which is associated with an access point on another subnetwork

link connection: 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

NOTE 8: Figures 2 and 3 show how a link connection is a part of an end-to-end connection.

network-to-network VPC: VP "transport entity" formed by the series of "connections" between "termination CPs", starting at an ATM Interconnection Gateway and ending at an ATM Interconnection Gateway. This involves the ATM resources of more than one PNO. The Network-to-network connections that are relevant to the X Interface are of the VP type

(Figure 2 shows how a Network-to-network connection is build up by a series of other connections)

network-to-user VPC: VP "transport entity" formed by the series of "connections" between "termination CPs", starting at an ATM Interconnection Gateway and ending at the Z User Accesspoint in the network of the Z PNO. This involves the ATM resources of more than one PNO. The Network-to-user connections that are relevant to the X Interface are of the VP type

PNO: providing Network Operator. An operator able to provide network resources to customers

pnoVcSubnetwork: topological component used to effect routing and management of ATM cells. It describes the potential for setting up "ATM-VC connections" across the subnetwork. The pnoVcSubnetworks are delineated by ATM AccessPoints and interconnected by "inter-PNO Physical links"

NOTE 9: A pnoVcSubnetwork can be partitioned into interconnected "sub-networks" and "links", but this partitioning is not shown over X Interface. In the context of the present document, one pnoVcSubnetwork represents an ATM network belonging to one PNO.

pnoVpSubnetwork: topological component used to effect routing and management of ATM cells. It describes the potential for setting up "ATM-VP connections" across the subnetwork. The pnoVpSubnetworks are delineated by ATM AccessPoints and interconnected by "inter-PNO Physical links"

NOTE 10: A pnoVpSubnetwork can be partitioned into interconnected "sub-networks" and "links", but this partitioning is not shown over Xinterface. In the context of the present document, one pnoVpSubnetwork represents an ATM network belonging to one PNO.

sub-network: "topological component" used to effect routing and management. It 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. A "sub-network" may be contained within one physical node. In the present document this partition is not used

sub-network connection: subnetwork connection is capable of transferring information transparently across a subnetwork. It is delimited by connection termination points at the boundary of the subnetwork and represents the association between these connection points. It can be a VP or a VC connection

subnetwork view: subtree of the X-interface tree. A subnetwork view belongs to a particular PNO: The PNO at the root of the subtree.

T PNO: transit PNO. This is a PNO using its own subnetwork to perform its required transit part of VP connection. It has a provider role and corresponds to a leaf in the X-interface tree, not being the **Z** side. In the "cascaded/mixed approach" case it can be both a Provider (where it acts as a Transit operator) and a Consumer (where it virtually acts as an Initiating operator)

trail: defined in ES 200 653 [9]

user: consumer of the ATM interconnection, provided by the end-to-end connection

user-to-network VPC: "transport entity" formed by the series of "connections" between "termination CPs", starting at the A User Accesspoint in the network of the A PNO and ending at an ATM Interconnection Gateway. This involves the ATM resources of more than one PNO. The User-to-network connections that are relevant to the X Interface are of the VP type

NOTE 11: Figure 2 shows how a User-to-network connection is build up by a series of other connections.

User-to-user VPC / VCC: "transport entity" formed by the series of "connections" between "termination CPs", starting at the A User Accesspoint in the network of the A PNO and ending at the Z User Accesspoint in the network of the Z PNO. This involves the ATM resources of more than one PNO. The User-to-user connections that are relevant to the X Interface can be of the VP type or the VC type

EXAMPLE 2: Figures 2 and 3 show examples of User-to-user connections.

X-interface tree: with respect to a particular VPC, X-interface relations exist between each Provider PNO and their Consumer PNO. Because each Provider has exactly one Consumer, the X-interface relations between all PNOs involved in the management of a particular VPC form a tree, the X-interface relation tree

NOTE 12: For the cascaded/mixed mode there can be several possible X-interface relation trees; the actual tree is formed at VPC setup. The root of the tree is the Initiating PNO; it uses (via X-interface controls) the PNOs (often Transit PNOs), to which it is connected in the tree via its branches. The most right leaf of the tree is the Z PNO. Figure 2 shows an example of an X-interface tree for the star mode with the A PNO being the Initiator.

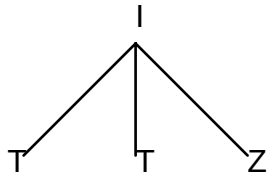


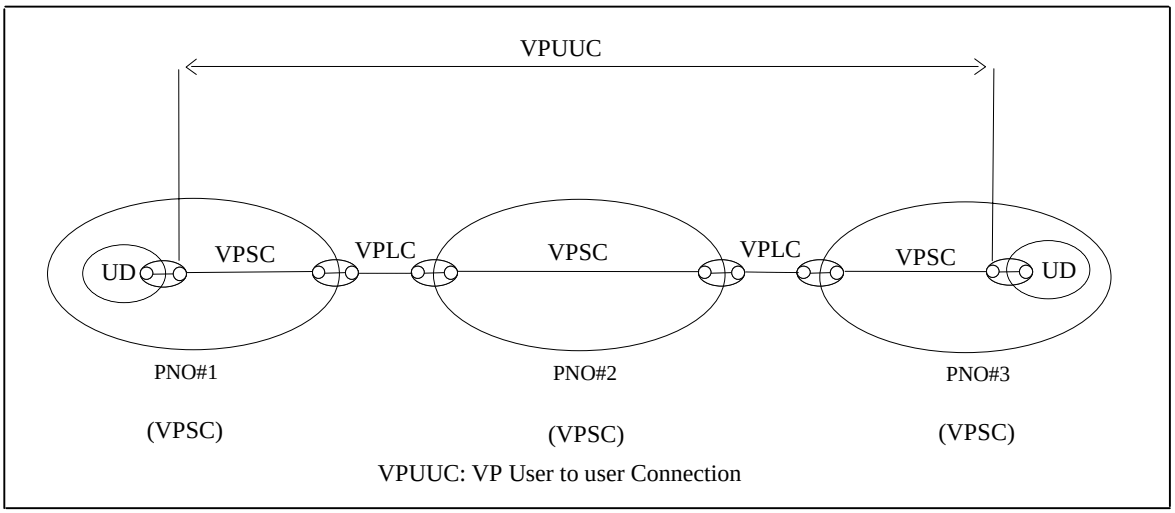
Figure 1: Example of a X-interface tree with the Initiating PNO being the A PNO

Z PNO: PNO where the Initiator ends reserving the end-to-end connection

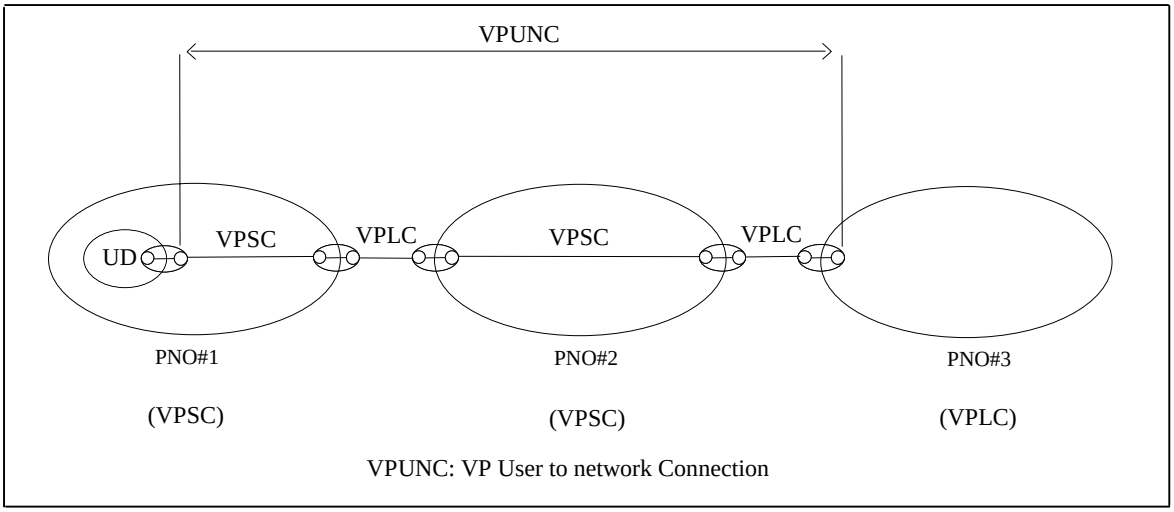
NOTE 13: If the termination point of the connection is at a User accesspoint, this is considered to be the **Z User** Accesspoint. In this case, the I PNO views the Z PNO as a PNO whose subnet is connected to the Z User.

(Note that for User-to-user Connections the VP Link connections that are part of it are not visible across the X-Interface. The Link Connections that terminate the "network side" of an end-to-end Connection are visible over the X-Interface)

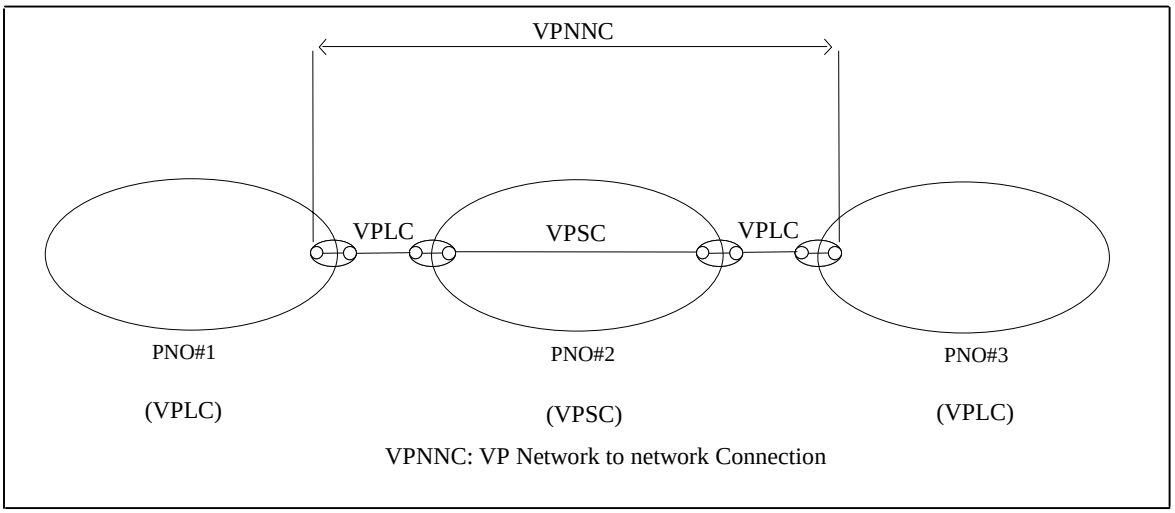
EXAMPLE 3: The following figures 2 and 3 show examples of possible end-to-end connections.



VPUUC: VP User to user Connection



VPUNC: VP User to network Connection



VPNNC: VP Network to network Connection

VPSC: VP Sub Network Connection

VPLC: VP Link Connection

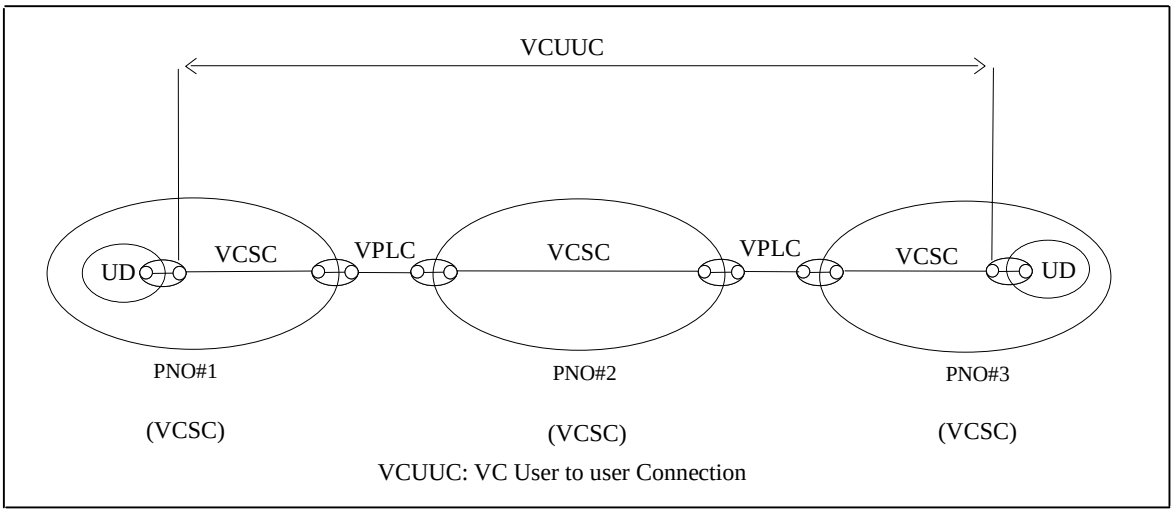
: Not visible across the X-Interface

: Visible across the X-Interface

UD: User Domain
(not visible across the X-Interface)

: Connection points

Figure 2: Examples of Functional architectures of VPCs provided by three PNOs



VCSC: VC Sub Network Connection

VPLC: VP Link Connection

: Not visible across the X-Interface

: Visible across the X-Interface

UD: User Domain
(not visible across the X-Interface)

: Connection points

Figure 3: Examples of Functional architectures of a VCC provided by three PNOs

3.2 Abbreviations

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

ASN.1	Abstract Syntax Notation One
ATM	Asynchronous Transfer Mode
ATM CP	ATM Connectivity Provider
BM	Business Management
CMIP	Common Management Information Protocol
CMISE	Common Management Information Service Element
CP	Connection Point
CTP	Connection Termination Point
EM	Element Management
GCL	Generic Classes Library
GDMO	Guidelines for the Definition of Managed Objects
IPPL	Inter PNO Physical Link
MF	Management Function
MS	Management Service
MSC	Management Service Component
NE	Network Element
NEF	Network Element Function
NGC	Network Generic Class Library
NM	Network Management
NMC	Network Management Centre
NMS	Network Management System
OS	Operations System
OSF	Operations System Function
OSS	Operations Support System
PNO	Providing Network Operator
QoS	Quality of Service
SM	Service Management
T PNO	Transit PNO
TMN	Telecommunications Management Network

TP	Termination Point
VC	Virtual Channel
VCC	Virtual Channel Connection
VCUUC	Virtual Channel User-to-user Connection
VP	Virtual Path
VPC	Virtual Path Connection
VPLC	Virtual Path Link Connection
VPNC	Virtual Path Network Connection
VPNNC	Virtual Path Network to network Connection
VPNUC	Virtual Path Network to user Connection
VPSC	Virtual Path Subnetwork Connection
VPUNC	Virtual Path User to network Connection
VPUUC	Virtual Path User to user connection

4 Management architecture

This clause describes the functional architecture for the X-interface for ATM cross connected networks. It provides an informative overview of the use of the X-interface and is therefore not normative. (Some definitions partially depend on the future acceptance of the "cascaded/mixed mode" as described in annex D. This dependence is already taken into account.)

Three concepts underpin the functional architecture for the X-interface:

- the X-interface connects two management systems, for the purpose of exchanging service level and/or network level requests with each other;
- consumer/Provider roles (also reflected by Manager / Agent);
- the future use of Star or Cascaded organizational models for communication, or a mixture of both. The choice of the organizational model will be determined by agreements between the PNOs involved in the X-interface.

These concepts are elaborated in the following subclauses.

4.1 X-interface at the Service Management level

In order to clarify the position of the X-interface within the layered management architecture outlined in ITU-T Recommendation M.3010 [7], the following definitions are adopted within the present document:

- the Network Management (NM) level is concerned with connections within the network. This means the control of topological information (subnetworks and the links between subnetworks), and subnetwork connections;
- the Service Management (SM) level is concerned with the overall connection and its associated Quality of Service (QoS).

Since PNOs can request other network operators to deliver a connection with a certain QoS, over the X-interface, without having view of and control over the topology within the individual subnetworks, this interface can be considered at the SM level. However, some functionalities described in the present document are allocated to the NM level, such as the management of topological information of the physical links between the subnetworks.

4.2 Consumer/Provider

The functional architecture for the X-interface for ATM Cross Connected networks is characterized by Consumer/Provider roles: each PNO in a VP Network connection is a Provider of a part of that connection. However, if the "cascaded/mixed mode" will be accepted, some PNOs might also have a Consumer role, since they use parts of other PNO's networks to provide their part of the connection. This is illustrated in figure 4.

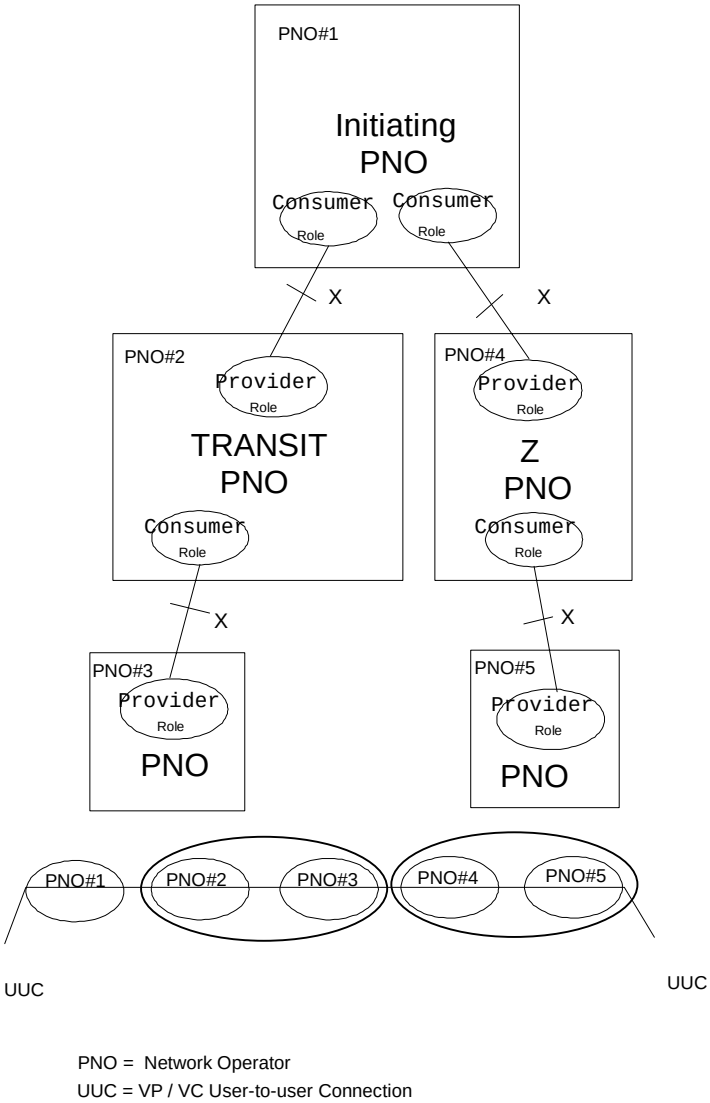
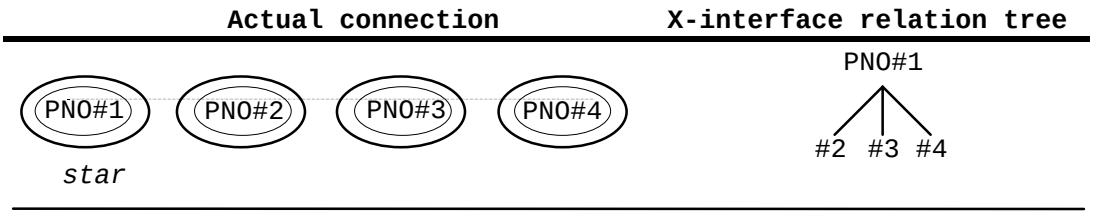


Figure 4: Consumer/provider roles over an X-interface for a specific connection

4.3 Organizational model

This subclause describes the organizational model, which will be used in establishing X-interface relationships between PNOs. It is shown in figure 5.



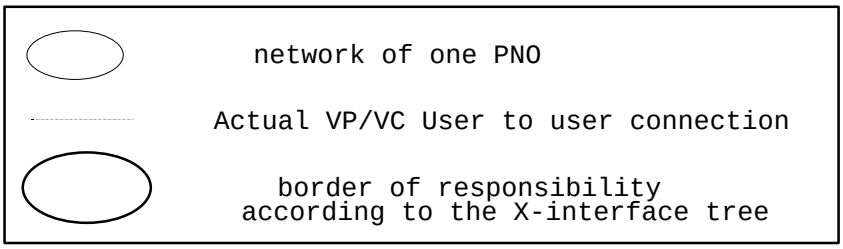


Figure 5: Organizational model

In the *Star* organization as exemplified in the figure above, PNO#1 uses the X-interface relation with all the PNOs involved. This means that PNO#1 has full responsibility for the entire connection and all the other PNOs are only responsible for their own network towards PNO#1.

The Consumer/Provider roles, introduced in subclause 4.2, are reflected by the X-interface tree: in the *Star* example above, PNO#1 has a Consumer role, whereas PNO#2, #3 and #4 have a Provider role.

4.4 Responsibility of PNOs regarding a VP connection

4.4.1 Basic VP connection responsibility rules

This subclause denotes which responsibility is required from each PNO involved in the management of a particular VP/VC connection. The following rules apply:

Rule 1 A PNO is responsible for the management of a particular VPC within its own PNO Subnetwork (if not being the ***Initiating PNO,*** who is responsible towards the connection customer).

Rule 2 Besides the responsibility in rule 1, a PNO is also responsible for the ATM Connection over the physical connection from its PNO Subnetwork to the next PNO Subnetwork on the route of a VPC (seen from the #1 network towards the #2 network).

Subclause 4.4.2 provides examples of the application of the responsibility rules with respect to the management of connection failures

4.4.2 Examples of application of the responsibility rules

The responsibility rules described in subclause 4.4.1 are illustrated by the following scenario description. The scenario starts with the existence of a particular end-to-end VPC between PNOs #1 and #3 via #2 and a particular X-interface relation tree. The scenario describes the occurrence of two cases of failures that need management action; each of them is described as one case of the scenario description. The two cases are an example of part of the life cycle of this particular VPC and take place in sequence. The scenario description is illustrated in figures 6 through 7.

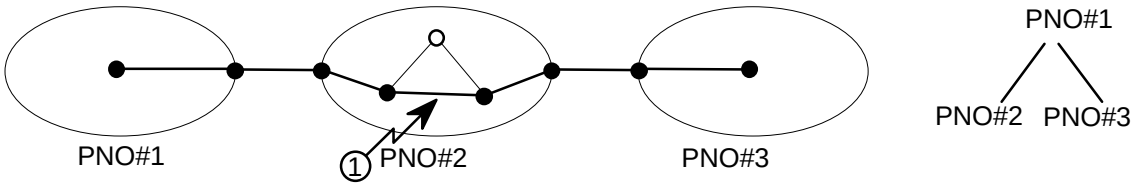


Figure 6: Case 1 of example scenario

Case 1

A fault has occurred in the VPC within the PNO subnetwork of PNO#2. PNO#2 reports to its Consumer PNO (PNO#1) about the failure and indicates that the failure is recoverable. PNO#2 starts its internal reconfiguration process. After reconfiguration has been successfully completed, PNO#2 reports this to its Consumer.

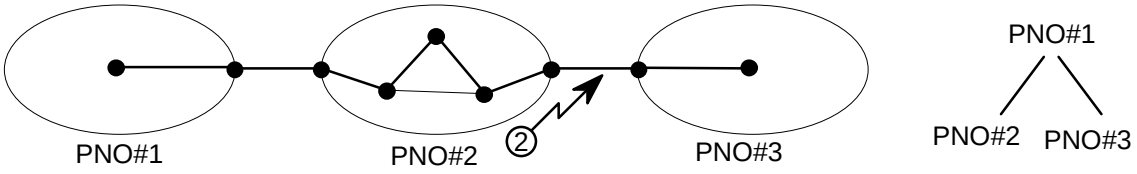


Figure 7: Case 2 of example scenario

Case 2

A fault has occurred in the inter-PNO VP link between PNO#2 and PNO#3. Since this is the outgoing inter-PNO VP link of the PNO subnetwork of PNO#2, PNO#2 is responsible for reporting this failure to its Consumer PNO (PNO#1). PNO#2 reports the failure to PNO#1 and indicates that it can recover the failure. PNO#2 performs recovery actions on the faulty inter-PNO VP link and notifies its Consumer about this and about the Id of the new physical link in the connection.

Detailed use of VP/VC protection for the management of VP/VC failures is provided in annex C.

5 Void

6 Configuration Management and Resilience Requirements

(The term 'VP/VC' is to be read as 'Virtual Path (VP) or Virtual Channel (VC) connection' in the present document).

The following requirements apply for the configuration management functionality and VP/VC resilience in association with use of the X-interface for management of ATM cross-connected networks:

1. It should be possible to set up the following types of Connections:
   - User-to-user VPC and VCC;
   - User-to-network VPC;
   - Network-to-user VPC;
   - Network-to-network VPC.
2. The network side of an end-to-end connection should terminate at an ATM Interconnection Gateway.
3. All transactions should be subject to bi-laterally agreed security processes. A minimum requirement is for 'Access Control', - Refer to annex B for further information.
4. In order to achieve unambiguous identification of entities that are manageable, it is necessary to assign unique names.
5. The X-interface is concerned with the management of the VP and VC Bearer Services.
6. The X-interface for a given PNO may assume either the provider or the consumer role, dependent on the connection.
7. Message transfer should be accomplished in near real time.

   "Real time" is taken to mean "fast enough to be able to control the process". Over the X interface, the delays are often expected to be too large to really be able to control the process. But "Near real time"Seems to be a satisfactory expression if it means "as fast as possible in order to achieve as much process control as possible".
8. Management of X-interface alarms should be undertaken in accordance with the specification defined in [5].
9. Operators are responsible for ensuring that deadlock situations across the X-interface are avoided.
10. The request for a VP/VC Connection which involves a third party PNO that is not providing any of the resources associated with the network connection should be possible. However, a PNO may choose to decline to provide resources for such a connection if no service level agreement has been established between it and the requesting PNO.
11. It should be possible to include in the VP/VC setup requests:
    - Quality of Service;
    - Interconnection Service (A PNO Originating connection, T PNO Transit Connection, Z PNO Terminating Connection);
    - Traffic description;
    - Group Identification (i.e. grouping connections according to initiator-defined criteria);
    - Connection Reference;
    - ATM VPC / VCC resilience type;

- Performance monitoring reporting.

12 It should be possible to set up the end-to-end connection via a preferred route.

13 It should be possible to verify whether the delivered path conforms to the setup request. This also provides tracing of a route of a connection.

14 Data representing the physical and logical resources comprising the subnetwork connection points will need to be maintained. An example of a physical resource is the physical link. Part of the data representing it is the physical link identifier and the allowed bandwidth. An example of a logical resource is a VP and the data representing it is the VPI.

15 Both on-request and scheduled path configuration processes (reservation, cancellation, activation, deactivation) should be supported by the X-interface. Scheduled connection state changes will be required in cases when multiple connections have to be brought into service in a predetermined sequence.

16 Schedule modification (including activation/deactivation schedules) should be supported.

17 Cell rate modification on an active connection should be supported: It should be possible to modify the data rate on an established active connection without the need to deactivate the connection.

18 Grouping of subnetwork connections and link connections in accordance with user defined criteria. end-to-end Connections may need to be grouped according to criteria other than their end points (***A/Z user access points or Interconnection Gateways)***. For example, initiators may want to refer to a mixed group of VP connections as a single unit.

19 It should be possible to set up a new connection that takes the same route as an existing connection group or connection.

20 Cancellation, activation and deactivation operations may be performed on groups of VP or VC connections.

6.1 ATM VP resilience requirements

A PNO playing the Initiator role should be able to choose between 4 different ways of protecting a User to user VP/VC connection, depending on the way the protection is performed. These are:

- Intra-Subnetwork Protection Switching;
- Intra Subnetwork Recovery;
- A-to-Z Fast Re-routing;
- VP Reconfiguration.

It should be possible to apply these four methods of VP/VC protection independently or in combination.

Detailed descriptions of these processes are provided in annex C.

7 Management services

7.1 Management services

For the Configuration Management area, the following Management Service (MS) is identified:

- VP/VC Service Provisioning - Consists of the provisioning of end-to-end VP/VC Connections.

The VP/VC Service Provisioning MS is decomposed into six Management Service Components (MSCs):

- *Establish MSC* - It allows the reservation and activation of a VPC/VCC.
- *Release MSC* - It allows the cancellation of a reservation or the deactivation of a VPC/VCC.
- *Modification MSC* - It allows the modification of the characteristics of a VPC/VCC, during reservation time, e.g. the cell rate.
- *Reconfiguration MSC* - It allows the reconfiguration of the whole or a part of a VPC/VCC, when an unrecoverable failure is detected.
- *Continuity Check MSC* - It allows a Continuity Check over a VPC/VCC.
- *Resilience MSC* - It allows the protection of the User-to-user VP/VC connection, or of a part (VPSC: VP Subnetwork connection, VCSC: VC Subnetwork connection) of it.

7.2 Management Service Components (MSC)

7.2.1 Establish MSC

Summary Description

Establishing a User-to-user VP/VC (VPUUC/VCUUC): PNO, having received a request for a VP/VC Connection, shall perform a check (Destination User Checking-request) with the **Z** PNO to ensure that the **Z** side wishes to accept the requested connection. If the response to this check (Destination User Checking-response) is positive, then another check is made with the **Z** side to see if there is sufficient cell rate available on at least one incoming link to the **Z** PNO (Check Available Cell Rate request). If the response to the check (Check Available Cell Rate -response) is positive, then possible paths will be found between the **A** and **Z** subnetworks, using the topology information. Both of these procedures are optional, in the sense that the Initiating PNO is not obliged to perform them in order to establish a VP Connection.

(A Check Available Cell Rate request may also be made with a transit PNO).

The topology information that describes the inter-PNO links has been made available by the other PNOs, either because they sent it (Topology Info Changes) or because it was read from their MIBs.

Once a path has been selected, each PNO involved in the connection is serially requested to perform a reservation (Reserve VP/VC Subnetwork Connection-request) between a specified input link and a specified adjacent subnetwork. The PNO responds (Reserve VP/VC Subnetwork Connection-response) with the result of this reservation request.

If all PNOs involved in the connection return positive results then the connection can be activated in accordance with the scheduled time specified in the reservation.

If for any reason, a given PNO Subnetwork does not activate the VP/VC Connection in accordance with the schedule, the Initiating PNO shall issue an activation request (Activate VP/VC Subnetwork Connection-request) to this PNO. This one shall then respond (Activate VP/VC Subnetwork Connection-response) with the result of the activation. However, this does not guarantee that the connection is operational.

If a reservation or activation request fails, then each PNO which has already reserved or activated the connection needs to be told to release their resource allocation for this VP/VC Connection (Release VP/VC Subnetwork Connection-request).

As a background task the topology of the network is being tracked, and any changes to either PNO Subnetworks or inter-PNO physical links are sent out to all PNOs (Topology Info Changes).

Performance Monitoring and network-resilience for VPs and VCs can be included as options in the reservation-request. (Reserve Vp Enhanced Subnetwork Connection resp. Reserve Vc Subnetwork Connection). Resilience and performance are described in the Resilience MSC and EN 300 820-3 [8] (Performance Management) respectively. The function Reserve VP Subnetwork Connection is maintained for compatibility purposes.

Establishing a Network-to-user VP (VPNUC): The end-to-end connection setup is simular to the VPUUC setup, apart from the first part of the connection. The "network side" of this end-to-end connection originates at an ATM Interconnection Gateway and is reserved with a link connection (reserve VP Link Connection-request). The responsible PNO responds (Reserve VP Link Connection-response) with the result of this reservation request. It is not possible to include the Performance and Resilience options in the this request. The rest of the end-to-end connection is reserved by using Reserve VP Subnetwork Connection requests.

Establishing a User-to-network VP (VPUNC): The end-to-end connection setup is simular to the VPUUC setup, apart from the last part of the connection. The "network side" of this end-to-end connection terminates at an ATM Interconnection Gateway and is reserved with a link connection (reserve VP Link Connection-request). The responsible PNO responds (Reserve VP Link Connection-response) with the result of this reservation request. It is not possible to include the Performance and Resilience options in the this request. Contrary to the VPUUC-case, a Destination User Checking is not possible.

The other parts of the end-to-end connection are reserved by using Reserve VP Subnetwork Connection requests.

Establishing a Network-to-network VP (VPNNC): The end-to-end connection setup is simular to the VPUUC setup, apart from the first and last part of the connection. They terminate at ATM Interconnection Gateways and are reserved with VP link connections (reserve VP Link Connection-request). The responsible PNO responds (Reserve VP Link Connection-response) with the result of this reservation request. It is not possible to include the Performance and Resilience options in the request. Contrary to the VPUUC-case, a Destination User Checking is not possible.

The remaining parts of this type of end-to-end connection are reserved by using Reserve VP Subnetwork Connection requests.

Simular to the Subnetwork Connection case, a VP Link Connection Reservation can be terminated with a Release VP Link Connection-request. It can be activated with a Activate VP Link Connection-request which has an Activate VP Link Connection-response. However, this does not guarantee that the connection is operational.

List of Functions:

- Destination User Checking *(Optional);*
- Check Available Cell Rate *(Optional);*
- Reserve VC Subnetwork Connection;
- Reserve VP Subnetwork Connection;
- Reserve Enhanced Vp Subnetwork Connection (The VP Subnetwork Connection function with Performance and Resilience options included);
- Reserve VP Link Connection;
- Activate VC Subnetwork Connection;
- Activate VP Subnetwork Connection;
- Activate VP Link Connection;
- Release VC Subnetwork Connection;
- Release VP Subnetwork Connection;
- Release VP Link Connection;
- Topology Info Changes (This is a background function that will provide network topology and status information to be used during the establishment process).

7.2.2 Release MSC

Summary Description

A VP/VC connection can be released for several reasons including expiry of the duration, a service layer request or due to reconfiguration.

There are two parts to the release of a VP/VC connection: deactivation of the VP/VC connection without ending the reservation, e.g. for a scheduled release of a connection (Deactivate VP/VC Subnetwork Connection/Deactivate VP Link Connection), and an unscheduled termination of a reservation with release of the resources (Release VP/VC Subnetwork Connection/ Release VP Link Connection).

Normally, the deactivation is made automatically in accordance with the schedule of the reservation. An explicit deactivation of the VP/VC Connection by the Initiating PNO has to be made in the following circumstances:

- failure of a PNO Subnetwork to deactivate a VP/VC Subnetwork Connection;
- rerouting of a VP/VC end-to-end connection.

The release of the reservation is made by the Initiating PNO by means of issuing a request - *Release VP/VC Subnetwork Connection-request* / *Release VP Link Connection-request* - to the other PNOs involved in the connection to release their portion of the appropriate connection and the PNOs will each respond (*Release VP/VC Subnetwor Connection-response/ Release VP Link Connection-response*) indicating the result of the requested release. This release of the reservation is required if, in setting up a End-to-end connection,a reservation or activation fails at a certain PNO.

A release request can also be received while a connection is active; it should then be released using established and agreed processes.

When a Z PNO has to release a subnetwork connection within its subnetwork the Initiating PNO will be notified (*cancel VP/VC Subnetwork Connection Notification*). The Initiating PNO then can release the connection or reconfigure it.

The same applies to a Link Connection (*cancel VP Link Connection Notification)*

If the initiating PNO does not release the connection the Z PNO can take responsibility for releasing it and send a *delete Notification* to the initiating PNO.

List of Functions:

- Deactivate VC Subnetwork Connection;
- Deactivate VP Subnetwork Connection;
- Deactivate VP Link Connection;
- Cancel VC Subnetwork Connection (Notification);
- Cancel VP Subnetwork Connection (Notification);
- Cancel VP Link Connection (Notification);
- Release VC Subnetwork Connection;
- Release VP Subnetwork Connection;
- Release VP Link Connection.

7.2.3 Modification MSC

Summary Description

Modification is allowed during the reservation time (interval between the VP/VC Connection Reservation creation and deletion) of a VP/VC connection. Changes are possible for the scheduling and for the cell rate of a connection. The path may not be changed while altering the connection.

To perform the Modification of a VPC/VCC, the Initiating PNO requests each PNO involved in providing the End-to-end connection to perform a change reservation (Change Reservation-request), by reserving the new values of the parameters, without cancelling the old ones for its own subnetwork connection. Each PNO responds with the result of this change reservation request (Change Reservation-response). During this time, the reservation is still determined by the old parameter-values.

If any change reservation request fails, then each PNO which has already reserved the changes needs to be told to cancel the altered resource allocation for this VP/VC connection, and to keep the existing one (Cancel Change-request).

If all PNOs involved in the connection return positive results, then the modified connection can be activated by sending an Activate Change-request to each PNO, each of which will then respond with the result of the activation (Activate Change-response).

If any activate change request fails, then the reservation has to be terminated at each involved PNO by the Initiating PNO (Release VP Subnetwork Connection-request for a VpSubnetwork connection, Release VC Subnetwork Connection-request for a VcSubnetwork Connection and Release VP Link Connection-request for a VpLinkConnection).

List of Functions:

- Change Reservation;
- Cancel Change;
- Activate Change;
- Release VC Subnetwork Connection;
- Release VP Subnetwork Connection;
- Release VP Link Connection.

7.2.4 Reconfiguration MSC

Summary Description

Reconfiguration via the X-interface shall be done by the Initiating PNO (I-PNO), after an unrecoverable failure in a VP/VC subnetwork which is supporting part of a VP/VC End-to-end connection.

Reconfiguration in this case means Rerouting, which is the establishment of a replacement connection by the NMS function.

For reconfiguring a VP/VC connection, it will be necessary to find a route around the faulty subnetwork or link supporting the current VP/VC End-to-end connection.

Reconfiguration at the Intiating PNO level consists of Cancelling the Reservation and making a new Establishment using the old parameters. The reconfiguration process can occur at any time.

If the inter_PNO management of the VP/VC connection is organized using the star approach, reconfiguration of a VP/VC connection boils down to deactivating and cancelling the appropriate VP/VC subnetwork connections (Deactivate VP/VC Subnetwork Connection and Release VP/VC Subnetwork Connection) and/or VP Link Connections (Deactivate VP Link Connection and Release VP Link Connection), followed by reserving and activating new ones (Reserve Enhanced VP Subnetwork Connection, Reserve VC Subnetwork Connection and Activate VP/VC Subnetwork Connection) resp. (Reserve VP Link Connection and Activate VP Link Connection). The reconfiguration process is controlled by the I-PNO. Further details can be found in the Establish and Release MSC description.

List of Functions:

- The Establish and Release MSC functions.

7.2.5 Continuity Check MSC

Implementation of theContinuity Check is optional: the initiating PNO can only use the Continuity Check functions with PNOs that are able to support it and allow its usage.

As described in ITU-T Recommendation I.751 [3], the I PNO can delimit part of an End-to-end Connection by applying monitoring objects. Such a delimited part is called segment. A segment is typically in the domain of one PNO; however, it could be extended beyond the control of one PNO by mutual agreement.

NOTE 1: If a PNO does not support Continuity Check functionality, but the PNO's at the extremity of the segment support it, the initiating PNO can use CC functions with the PNO's at the extremity of the segment.

Summary Description

The initiating PNO of a vpConnection (vcConnection) can initiate (and stop) a Continuity Check over a vpConnection (vcConnection) segment.

The Continuity Check may involve one or more PNOs, if all PNOs involved support this by mutual agreement.

The Initiating PNO requests to all the PNOs involved in the vpConnection segment to stop the use of the F4 flow over their managed VP Subnetwork Connection (stopF4Flow-request).

(For a VC Connection, the Initiating PNO requests to all the PNOs involved in the vcConnection segment to stop the use of the F5 flow over their managed VC Subnetwork Connection) (stopF5Flow-request).

Then the Initiating PNO requests the creation of the pnoBidirectionalContinuityMonitor for the Continuity Check to the extremity PNOs of the segment (Create bidirectionalContinuityMonitor for Continuity Check-request). The addressed PNOs respond with positive or negative acknowledgement.

In case of positive response, the Initiating PNO requests the activation of the Source function to the extremity PNOs of the segment (Activate bidirectionalContinuityMonitor Source for Continuity Check-request).

NOTE 2: The Sink function is activated at the bidirectionalContinuityMonitor creation.

When both extremities Sink and Source are activated, the Continuity Check is performed.

When the Continuity Check has to be stopped, the Initiating PNO first asks for the Source deactivation to the extremity PNO of the segment (*Deactivate bidirectionalContinuityMonitor Source for Continuity Check-request*). Then it asks for the bidirectionalContinuityMonitor deletion to the extremity PNOs of the segment (*Delete bidirectionalContinuityMonitor-request*).

At the end of the Continuity Check, the Initiating PNO allows the use of the F4 flow (*allowF4Flow-request*) over their managed VP Subnetwork Connection for all the PNOs involved in the vpConnection segment.

For a vcConnection segment, the Initiating PNO allows the use of the F5 flow (*allowF5Flow-request*) over their managed VC Subnetwork Connection for all the PNOs involved in the vcConnection segment.

During the Continuity Check, if the extremity Sink detects a lossOfContinuity, the concerned PNO sends an indication to the Initiating PNO (*Report of Continuity Check*).

Operational State Changes on Bidirectional Continuity Monitor are notified to the Initiating PNO to inform it that the Continuity Check can't be performed.

List of Functions:

- Create Bidirectional Continuity Monitor for Continuity Check;
- Activate Bidirectional Continuity Monitor Source for Continuity Check;
- Deactivate Bidirectional Continuity Monitor Source for Continuity Check;
- Delete Bidirectional Continuity Monitor;
- Notification of Bidirectional Continuity Monitor Operational State change;
- Stop F4 flow;
- Allow F4 flow;
- Stop F5 flow;
- Allow F5 flow;

- Report of Continuity Check.

7.2.6 Resilience MSC

Summary Description

Implementation of the resilience MSC is optional; it can only be requested by the initiating PNO with PNOs that are able to support it. It is not valid for Vp Link Connections.

To make VP/VC connections (**working** connections) resilient, a PNO playing the initiator role will reserve its connections with a **protection** entity added. On detection of an error on the working connection there will be a change-over to the protection connection. The connection reservation is described in the Establish MSC.

The Initiator can choose among 3 different kinds of resilience mechanisms:

- Intra-Subnetwork Protection Switching;
- Intra Subnetwork Recovery;
- A-to-Z Fast re-routing.

It is possible to combine these three options in any combination.

It is presumed that the Agent will attempt to recover the working VP/VC, whether or not resilience has been requested. This recovering of the working VP or VC is considered to be *Intra Subnetwork Recovery without the use o pre-assigned resources,* and is not considered further here.

Intra-Subnetwork Protection Switching: this is performed within a Subnetwork. The working connection and the protection connection have the same end points, and the detection of an error and the following change-over to the protection traffic occur automatically, without any network-management action. The protection connection uses dedicated and pre-assigned capacity.

The network elements at the endpoints shall be able to switch to the protection connection.

Two types of pre-assignment are known from ITU-T Recommendation I.630 [6]. These are the (1:1) configuration, where the protection connection does not convey working traffic until the working connection experiences errors, and the (1+1) configuration, where the working and protection connection convey the working traffic simultaneously.

The request for this kind of resilience is performed by sending an optional parameter in the reservation request for the working VP *(Reserve Enhanced VP Subnetwork Connection*) or VC (*Reserve VC Subnetwork Connection).*

If this reservation is accepted, the response (*Reserve Enhanced VP Subnetwork Connection-response) / (Reserve VC Subnetwork Connection-response)* will also indicate if the Agent is able to provide protection switching. If it is not provided, the I PNO can decide whether to use the reservation anyway or to end it (See *Establish MSC*).

A change over to protecting traffic is reported to the I PNO. This is described in the VP/VC alarm reporting MS of EN 300 820-2 [5]: Alarm Management.

Intra Subnetwork Recovery: this is performed within a Subnetwork. The working VP/VC Subnetwork connection and the protection connection have their endpoints in common. In response to an error on the working VP Subnetwork connection the PNO with the network-error needs to trigger the recovery mechanism internally.

A recovery action may follow a failed attempt to overcome a fault by protection switching or it could be the only strategy implemented to protect a circuit.

The request for this kind of resilience mechanism is performed by sending an optional parameter in the reservation request for the working VP (*Reserve Enhanced VP Subnetwork Connection*) or VC (*Reserve VC Subnetwork Connection).*

If this reservation is accepted, the response (*Reserve Enhanced VP Subnetwork Connection-response / Reserve VC Subnetwork Connection-response)* will also indicate if the Agent is able to provide the recovery mechanism. If it is not provided, the I PNO can decide whether to use the reservation anyway or to end it. (See *Establish MSC*).

An attempt of the Agent to establish recovery will be reported to the I PNO. The result of this attempt is also reported. (VP alarm reporting MS of EN 300 820-2 [5]).

A-to-Z Fast re-routing: this provides a global (A-User to Z-User) resilience mechanism, that covers both intra-PNO and inter-PNO failures. The VP/VC is duplicated on a completely different route, notably using different inter-PNO links. The working VP/VC connection and the protection VP/VC connection only have the A user accesspoint and the Z user accesspoint in common. In a Transit network, they do not have any accesspoint in common. They convey the same traffic simultaneously. In case of an error the change-over to protecting resources is performed by the A-PNO and the Z-PNO, following a management request of the I PNO. *(Switch To Subnetwork Connection)*

Fast re-routing may follow failed attempts to overcome a fault by protection switching and/or recovery, or it could be the only strategy implemented to protect a circuit.

In general, the protection VP/VC is reserved after the working VP/VC has been reserved, by making a reservation request with two optional parameters indicating the fast-re-routing case and a reference to the working connection respectively (*Reserve Enhanced VP Subnetwork Connection / Reserve VC Subnetwork Connection).* The response (*Reserve Enhanced VP Subnetwork Connection-response / Reserve VC Subnetwork Connection-response)* will indicate if the working connection exists or if the Agent is able to provide fast re-routing. If at least one of these conditions do not apply the working connection can not be protected with Fast re-routing mechanism.

The working VP/VC shall have been reserved before, with an optional parameter in the reservation request requesting that the working VP/VC is to be protected by fast-re-routing.

If the working reservation is accepted, the response (*Reserve Enhanced VP Result-response / Reserve VC Subnetwork Connection-response*) will also indicate if the Agent is able to provide fast re-routing. If it is not provided, the I PNO can decide whether to use the reservation anyway or to end it (See *Establish MSC*).

If protection- and working connections use different T-networks they can be reserved, in those T-networks, without the fast-rerouting option. (*Reserve Enhanced VP Subnetwork Connection / Reserve VC Subnetwork Connection)*.

After having received an X interface alarm message that indicates a fault on the working connection, the I PNO can request the A and the Z PNO to change over, at their User Access Points, to the traffic arriving over the protection circuit. (*Switch To SubnetworkConnection-request*)

The A- and the Z Agents respond with the result of this request. (*Switch To SubnetworkConnection-result)*

If the request is unsuccessful the I PNO can consider reconfiguration of the overall VP/VC connection. (*Reconfiguration MSC*)

Using the function (*Switch To SubnetworkConnection*) it is possible to change back to the original working connection when it has recovered.

Figure 8 and the associated text contain an additional explanation of the processes that are associated with the different kinds of Network Resilience for a T PNO.

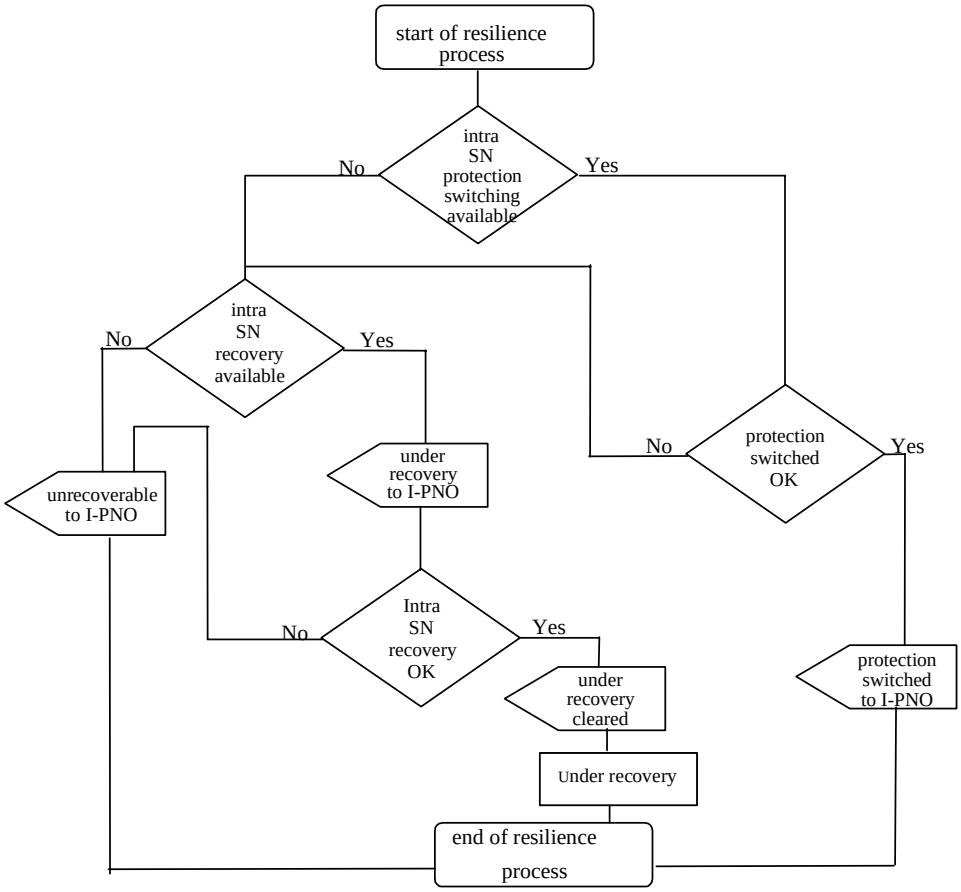


Figure 8: Resilience Process

If a fault occurs on a reserved connection in the PNO domain, different situations can occur, as represented in figure 8. The following text does not cover all possibilities, but can help to understand the diagram itself.

For the role of A/T and Z-PNO

a) The Subnetwork connection was requested with "Intra-Subnetwork Protection Switching". In This case the PNO will restore the Subnetwork connection automatically and will inform the I-PNO with a "Protection Switching Notification".

b) If a) fails and/ or Protection Switching is not requested and/ or Recovery was requested for the Subnetwork connection. In this case the recovery process at the PNO will be started as quickly as possible to restore the failed part of the Subnetwork connection and an "Under Recovery" Notification [5] will be sent to the I-PNO.

c) If b) recovers correctly a "Under Recovery, Cleared" Notification [5] is sent to the I-PNO.
If b) fails to recover, an "Unrecoverable" Notification [5] is sent to the I-PNO.

Only for the role of A and Z-PNO

If the connection has been requested to support Fast Re-routing, the following cases may happen:

a) At any point in time an action can be received from the I-PNO, requesting to switch from the working connection to the protection connection.

b) If d) happens while a recovery process is occurring, this will not be stopped; the recovery process will continue as explained in a) to c).

Only for the role of I-PNO

If, as a consequence of receiving an alarm from the A/Z-PNO, the I-PNO has requested to A- and Z-PNO Fast Re-routing, the following situations may happen (please not that the following are not Normative behaviours, but rather guidelines):

a) Both a- and Z-PNO switch successfully to the protection connection. If the PNO who experienced the fault eventually recovers, the I-PNO can, in future switch again to the original connection, using the same action on A- and Z-PNO.

b) If the A- or Z-PNO can't perform Fast Re-routing; e.g. A-PNO:

- the I-PNO can send a Fast Re-routing action to the successful PNO (switch To Snc) (in this example the Z-PNO), to return to the previous situation;
- the I-PNO can try to re-send the Fast Re-routing request to the unsuccessful PNO (switch To Snc) (in this example the A-PNO) and see if it works.

c) If nothing works and the failure is unrecoverable, the I-PNO may:

- release the whole end-to-end connection;
- use the Reconfiguration process in order to re-create a VP from scratch.

Note that IPPL failures have to be sent to all PNOs.

List of Functions:

Reserve Enhanced VP Subnetwork Connection

Reserve VC Subnetwork Connection

- Switch To Subnetwork Connection

7.3 Management Functions (MF)

For a mapping between the Management Functions and the actual CMISE functions, refer to annex D.

7.3.1 Activate Bidirectional Continuity Monitor Source for Continuity Check

This function requests to the extremity PNOs of the segment involved in a segment Continuity Check to activate the Source mechanism for OAM flow.

NOTE: The Sink mechanism is activated at bidirectionalContinuityMonitor Creation.

The message associated with this function will be sent by the Initiating PNO to the extremity PNOs of the segment. The extremity PNOs of the segment respond with the result of the activation.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function.

Request -	Bidirectional continuity monitor Id	Identity of the bidirectional continuity monitor to activate
	ControlContinuity CheckInformation	Activate Source
Response	Activation result	Result of the activation attempt

7.3.2 Activate Change

This function requests a PNO to activate a reserved connection of which the parameters have been modified.

The PNO should respond with the result of the activation.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function.

Request-	VP/VC Connection id	Identity assigned to the connection
Response-	Activate Change result	Result of the activate change attempt

7.3.3 Activate VC Subnetwork Connection

This function requests a PNO to activate a reserved VC Subnetwork connection between the specified input and output ports of its subnetwork. (Between the user and the output port for the **A** subnetwork or between the input port and the user for the **Z** subnetwork). The PNO should respond with the result of the activation.

This MF is applied only if if the connection is not activated automatically in accordance with the reserved schedule.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	VC Connection id	Identity assigned to the connection
Response-	Activation result	Result of the activation attempt

7.3.4 Activate VP Link Connection

This function requests a PNO to activate a reserved VP Link connection. The PNO should respond with the result of the activation.

This MF is applied only if if the connection is not activated automatically in accordance with the reserved schedule.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	VP Connection id	Identity assigned to the connection
Response-	Activation result	Result of the activation attempt

7.3.5 Activate VP Subnetwork Connection

This function requests a PNO to activate a reserved VP Subnetwork connection between the specified input and output ports of its subnetwork. (Between the user and the output port for the **A** subnetwork or between the input port and the user for the **Z** subnetwork). The PNO should respond with the result of the activation.

This MF is applied only if if the connection is not activated automatically in accordance with the reserved schedule.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	VP Connection id	Identity assigned to the connection
Response-	Activation result	Result of the activation attempt

7.3.6 Allow F4 flow

This function allows a PNO to use the F4 flow over a Subnetwork Connection or over an inter-PNO link.

The message associated with this function is sent by the Initiating PNO to all the PNOs involved in a segment Continuity Check when this check is ended.

It is confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request -	PNO VP Subnetwork Connection Id	Identity assigned to the concerned PNO Subnetwork Connection
	Allow test	
Response	Activation result	Result of the activation attempt

7.3.7 Allow F5 flow

This function allows a PNO to use the F5 flow over a Subnetwork Connection or over an inter-PNO link.

The message associated with this function is sent by the Initiating PNO to all the PNOs involved in a segment Continuity Check when this check is ended.

It is confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request -	PNO VC Subnetwork Connection Id	Identity assigned to the concerned PNO Subnetwork Connection
	Allow test	
Response	Activation result	Result of the activation attempt

7.3.8 Cancel Change

This function requests a PNO to cancel the previously requested changes to the parameters of a VP/VC connection.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	VP/VC Connection Id	Identity assigned to the connection
Response-	Cancel Change Result	Result of the cancel change attempt

7.3.9 Cancel VC Subnetwork Connection

This function allows a Z PNO to notify the initiating PNO that it wants the initiator to release, for a particular reason, a subnetwork connection in the Z PNO's net.

It is a non-confirmed type of operation. The following parameters are associated with the Request and Respons primitives of the function:

Request	VC Connection Id	Identity assigned to the connection
	disconnect Cause	The Z PNO's reason for his request
Response	None	

7.3.10 Cancel VP Link Connection

This function allows a PNO to notify the initiating PNO that it wants the initiator to release, for a particular reason, a Link connection in its domain.

It is a non-confirmed type of operation. The following parameters are associated with the Request and Respons primitives of the function:

Request	VP Connection Id	Identity assigned to the connection
	disconnect Cause	The PNOs' reason for his request
Response	None	

7.3.11 Cancel VP Subnetwork Connection

This function allows a Z PNO to notify the initiating PNO that it wants the initiator to release, for a particular reason, a subnetwork connection in the Z PNO's net.

It is a non-confirmed type of operation. The following parameters are associated with the Request and Respons primitives of the function:

Request	VP Connection Id	Identity assigned to the connection
	disconnect Cause	The Z PNO's reason for his request
Response	None	

7.3.12 Change Reservation

This function requests a PNO to check if it has resources available to support the altered parameters for an existing subnetwork connection which is specified by the connection identifier.

Upon reception of this request a transit PNO should check the availability of the new scheduler between the linked input and output ports of its subnetwork; if it is an A or a Z PNO the check has to be done between the input/output port of the subnetwork and the user.

The PNOs respond with the new scheduling parameters for the already established connection if the check has been successful, or with an error message if the check has shown no possiblity to change the scheduler. The connection reservation will be unchanged for either of the situations.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	VP/VC Connection Id	Identity assigned to the connection
	Forward traffic descriptor (new)	Altered peak cell rate from **A** to **Z**
	Backward traffic descriptor (new)	Altered peak cell rate from **Z** to **A**
	Schedule (new)	Changed Schedule
Response-	Change reservation result	Result of the change reservation attempt

7.3.13 Check Available Cell Rate

This function checks that the **Z** PNO has a particular cell rate available on at least one of its incoming inter-PNO links. It is only a check that the **Z** PNO has the cell rate available and not for the total amount of available cell rate.

The request can also be sent to a transit PNO. In this case the Agent-PNO will always consider itself as the Z side of the link with regard to the direction of the bandwidth in the request.

The message will include the peak cell rate required in both directions for a planned VP/VC connection. The receiving PNO (**Z** PNO or Transit-PNO in the Initiating PNO's view) will then check on which of its inter-PNO links this cell rate is available and respond to the Initiating PNO with the corresponding list of neighbouring PNOs.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	VP/VC Connection Id	Identity assigned to the connection
	Forward traffic descriptor	Peak cell rate from **A** to **Z**
	Backward traffic descriptor	Peak cell rate from **Z** to **A**
	PNO Subnetwork Id	Identity of Initiating PNO
	Schedule	List of activation times, dates and durations
Response-	Check Available Cell Rate Response	List of inter-PNO links which can accommodate the requested cell rate

7.3.14 Create Bidirectional Continuity Monitor for Continuity Check

A continuity check OAM flow is used in a vpConnection/vcConnection in order to monitor the continuity of this vpConnection/vcConnection.

The Initiating PNO creates a bidirectionalContinuityMonitor for Continuity Check at the entry edge or exit edge of the extremity PNOs of the segment.

For that creation, the Initiating PNO sends a Create bidirectionalContinuityMonitor message to the extremity PNOs of the segment.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request -	Bidirectional continuity monitor Id	Identity assigned to the created bidirectional continuity monitor
	FlowDirection	Direction in/from which the OAM flow is generated/received
Response	Creation result	Result of the create

The bidirectionalContinuityMonitor Sink mechanism represents the sink of a segment OAM flow. The bidirectionalContinuityMonitor Sink mechanism is activated at object creation.

7.3.15 Deactivate Bidirectional Continuity Monitor Source for Continuity Check

This function requests the extremity PNOs of the segment involved in a segment Continuity Check to deactivate the Source/Sink mechanism for OAM flow.

The function is used first to deactivate the Source mechanism at each extremity. The Sink mechanism will be deleted at object deletion.

The message associated with this function will be sent by the Initiating PNO to the extremity PNOs of the segment. The extremity PNOs of the segment respond with the result of the deactivation.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request -	Bidirectional continuity monitor Id	Identity of the bidirectional continuity monitor to deactivate
	ControlContinuity CheckInformation	Deactivate Source
Response	Deactivation result	Result of the deactivation attempt

7.3.16 Deactivate VC Subnetwork Connection

This function requests a PNO to deactivate the Subnetwork connection between the specified input and output ports of its subnetwork. (Between the user and the output port for the ***A*** subnetwork or between the input port and the user for the ***Z*** subnetwork). ThePNO should respond with the result of the deactivation.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	VC Connection Id	Identity assigned to the connection
Response-	Deactivation Result	Result of the deactivation attempt

7.3.17 Deactivate VP Link Connection

This function requests a PNO to deactivate the specified Link Connection. ThePNO should respond with the result of the deactivation.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	VP Connection Id	Identity assigned to the connection
Response-	Deactivation Result	Result of the deactivation attempt

7.3.18 Deactivate VP Subnetwork Connection

This function requests a PNO to deactivate the Subnetwork connection between the specified input and output ports of its subnetwork. (Between the user and the output port for the **A** subnetwork or between the input port and the user for the **Z** subnetwork). ThePNO should respond with the result of the deactivation.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	VP Connection Id	Identity assigned to the connection
Response-	Deactivation Result	Result of the deactivation attempt

7.3.19 Delete Bidirectional Continuity Monitor

The Initiating PNO deletes bidirectionalContinuityMonitor objects created for Continuity Check at the entry edge or exit edge of the extremity PNOs of the segment.

For that deletion, the Initiating PNO sends a Delete bidirectionalContinuityMonitor message to the extremity PNO of the checked segment of the connection after the end of the check.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request -	Bidirectional continuity monitor Id	Identity assigned to the deleted bidirectional continuity monitor
Response	Deletion result	Result of the delete

7.3.20 Destination User Checking

When a new VP/VC connection establish request is received from the Service Layer, it is reasonable to verify that the **Z** user is able or wants to support the requested connection before expending network resources in establishing the connection. From a PNO's point of view only a check that the **Z** PNO can accept the connection is performed.

On receiving a VP/VC connection establish request from the Service Layer, the Initiating PNO will send out a Destination User Checking-request message to the **Z** PNO, and will include the peak cell rate (in both directions), and the **A** and **Z** addresses for the requested VP Connection. The **Z** PNO will then perform the **Z** user checking and respond with the result to the Initiating PNO.

In case the **Z** user is under the **Z** PNO domain (star approach), the **Z** PNO will perform the **Z** user checking and respond with the result to the Initiating PNO.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	VP/VC Connection Id	Identity assigned to the connection
	A Address	E.164 address of the **A** user
	Z Address	E.164 address of **Z** user
	Forward Traffic Descriptor	Peak cell rate from **A** to **Z**
	Backward Traffic Descriptor	Peak cell rate from **Z** to **A**
	Schedule	List of activation times, dates and durations
	PNO Subnetwork Id	Identity of the Initiating PNO

Response-	Check Response	Result of **Z** user check
	Check Reason	Reason for failure of check
	PNO Subnetwork Id	Identity of the reporting PNO

7.3.21 Notification of Bidirectional Continuity Monitor Operational State Change

When an Operational State change occurs on a pnoBidirectionalContinuityMonitor, the Initiating PNO of the vpConnection (vcConnection) shall be informed that the Continuity Check can't be performed or shall be stopped if it is already started.

This Operational State change will be sent using the normal stateChange Notification.

7.3.22 Release VC Subnetwork Connection

This function requests a PNO to release an already confirmed reservation and to free the resources.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	VC Connection Id	Identity assigned to the connection
Response-	Release Result	Result of the cancel reservation attempt

7.3.23 Release VP Link Connection

This function requests a PNO to release an already confirmed reservation and to free the resources.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	VP Connection Id	Identity assigned to the connection
Response-	Release Result	Result of the cancel reservation attempt

7.3.24 Release VP Subnetwork Connection

This function requests a PNO to release an already confirmed reservation and to free the resources.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	VP Connection Id	Identity assigned to the connection
Response-	Release Result	Result of the cancel reservation attempt

7.3.25 Report of Continuity Check

The bidirectional continuity monitor receives a continuity check OAM flow in a vpConnection (vcConnection).

When the bidirectional continuity monitor Sink part detects a disruption on the received flow, a Report of Continuity Check Notification is sent to the Initiating PNO.

It is a non-confirmed type of operation. The following parameters are associated with the Request primitives of the function:

Request -	Bidirectional continuity monitor Id	Identity assigned to the bidirectional continuity monitor which has detected the connection disruption
	Probable Cause	Set to LOC: loss of continuity (17)

7.3.26 Reserve Enhanced VP Subnetwork Connection

This function requests the involved PNO to reserve a path across its subnetwork: between a specified input point and an adjacent subnetwork if performs the **T** subnetwork role, between the A User and an adjacent subnetwork if it performs **A** subnetwork role or between a specified input port and the Z user, if it performs the Z subnetwork role.The PNO responds with the result of the reservation.

The request for Performance can be included as an option. The response will indicate if the request for performance measurements was successful. Also, the kind of resilience can be requested as an option. In this case the response indicates whether the resilience has been provided in addition to the successful result response.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	VP Connection Id	Identity assigned to the connection
	Subnetwork Connection Id (nearend)	Identification of the nearend Subnetwork connection
	AccesPoint Id (nearend)	Identification of the nearend Accesspoint
	VP Id (nearend)	Identification of the nearend VPI
	PNO Subnetwork Id (far-end)	Identification of the adjacent PNO Subnetwork
	A Address	E.164 address of **A** user
	Z Address	E.164 address of **Z** user
	Z VPI (optional)	VPI to be used by the **Z** PNO only to allow user transparency of reconfiguration
	Forward traffic descriptor	Peak cell rate from **A** to **Z**
	Backward traffic descriptor	Peak cell rate from **Z** to **A**
	Forward QoS class	An indication of the class of QoS
	Backward QoS class	An indication of the class of QoS
	Schedule	List of activation times, dates and durations
	Performance Monitoring (optional)	Indication if Performance Monitoring has been requested
	Resilience Kind (optional)	Kind of Network Resilience
	Related Subnetwork Connection (optional)	Pointer at Subnetwork connection that is to be protected by Fast re-routing
Response-	Subnetwork Connection Id (far-end)	Identification of the far-end Subnetwork Connection
	Reservation result	Result of the reservation attempt, including the result of the request performance and for resilience

7.3.27 Reserve VC Subnetwork Connection

This function requests the involved PNO to reserve a path across its subnetwork: between a specified input point and an adjacent subnetwork if performs the **T** subnetwork role, between the A User and an adjacent subnetwork if it performs **A** subnetwork role or between a specified input port and the **Z** user, if it performs the **Z** subnetwork role. The PNO responds with the result of the reservation.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	VC Connection Id	Identity assigned to the connection
	Subnetwork Connection Id (nearend)	Identification of the nearend Subnetwork connection
	AccesPoint Id (nearend)	Identification of the nearend Accesspoint
	VP Id (nearend)	Identification of the nearend VPI
	VC Id (nearend)	Identification of the nearend VCI
	PNO Subnetwork Id (far-end)	Identification of the adjacent PNO Subnetwork
	A Address	E.164 address of **A** user
	Z Address	E.164 address of **Z** user
	Z VPI (optional)	VPI to be used by the **Z** PNO only to allow user transparency of reconfiguration
	Z VCI (optional)	VCI to be used by the **Z** PNO only to allow user transparency of reconfiguration
	Forward traffic descriptor	Peak cell rate from **A** to **Z**
	Backward traffic descriptor	Peak cell rate from **Z** to **A**
	Forward QoS class	An indication of the class of QoS
	Backward QoS class	An indication of the class of QoS
	Schedule	List of activation times, dates and durations
Response-	Subnetwork Connection Id (far-end)	Identification of the far-end Subnetwork Connection
	Reservation result	Result of the reservation attempt.

7.3.28 Reserve VP Link Connection

This function requests the involved PNO to reserve a Vp connection across an inter-Pno Link with an adjacent subnetwork. The PNO responds with the result of the reservation.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	VP Connection Id	Identity assigned to the connection
	PNO Subnetwork Id	Identification of the adjacent PNO Subnetwork
	nearEndAccesspointId	Identification of the accesspoint that is to be used for the Link Connection
	VP Id (nearend)	Identification of the VP Id
	Forward traffic descriptor	Peak cell rate from **A** to **Z**
	Backward traffic descriptor	Peak cell rate from **Z** to **A**
	Forward QoS class	An indication of the class of QoS
	Backward QoS class	An indication of the class of QoS
	Schedule	List of activation times, dates and durations
Response-	Reservation result	Result of the reservation attempt.

7.3.29 Reserve VP Subnetwork Connection

This function requests the involved PNO to reserve a path across its subnetwork: between a specified input point and an adjacent subnetwork if performs the ***T*** subnetwork role, between the A User and an adjacent subnetwork if it performs ***A*** subnetwork role or between a specified input port and the **Z** user, if it performs the ***Z*** subnetwork role. The PNO responds with the result of the reservation.

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	VP Connection Id	Identity assigned to the connection
	Subnetwork Connection Id (nearend)	Identification of the nearend Subnetwork connection
	AccesPoint Id (nearend)	Identification of the nearend Accesspoint
	VP Id (nearend)	Identification of the nearend VPI
	PNO Subnetwork Id (far-end)	Identification of the adjacent PNO Subnetwork
	A Address	E.164 address of **A** user
	Z Address	E.164 address of **Z** user
	Z VPI (optional)	VPI to be used by the **Z** PNO only to allow user transparency of reconfiguration
	Forward traffic descriptor	Peak cell rate from **A** to **Z**
	Backward traffic descriptor	Peak cell rate from **Z** to **A**
	Forward QoS class	An indication of the class of QoS
	Backward QoS class	An indication of the class of QoS
	Schedule	List of activation times, dates and durations
Response-	Subnetwork Connection Id (far-end)	Identification of the far-end Subnetwork Connection
	Reservation result	Result of the reservation attempt.

7.3.30 Stop F4 flow

This function inhibits a PNO from using the F4 flow over a Subnetwork Connection or over an inter-PNO link.

The message associated with this function is sent by the Initiating PNO to all the PNOs involved in a segment Continuity Check when before this check is performed.

It is a confirmed type of operation. The following parameters are associated with the Request and response primitives of the function:

Request -	PNO VP Subnetwork Connection Id	Identity assigned to the concerned PNO Subnetwork Connection
	Inhibit test	
Response	Deactivation result	Result of the deactivation attempt

7.3.31 Stop F5 flow

This function inhibits a PNO from using the F5 flow over a Subnetwork Connection or over an inter-PNO link.

The message associated with this function is sent by the Initiating PNO to all the PNOs involved in a segment Continuity Check when before this check is performed.

It is a confirmed type of operation. The following parameters are associated with the Request and response primitives of the function:

Request -	PNO VC Subnetwork Connection Id	Identity assigned to the concerned PNO Subnetwork Connection
	Inhibit test	
Response	Deactivation result	Result of the deactivation attempt

7.3.32 Switch To Subnetwork Connection

This function requests the involved (A and Z) PNO's to change over, at the endpoint that represents the user-connection, to the traffic arriving over an alternative VP/VC Subnetwork connection. The VP/VC Subnetwork connection that is addressed shall contain a pointer to the alternative connection.

The A- and the Z Agents respond with the result of this request (Switch To SubnetworkConnection-result).

Request:	VP/VC Connection Id	Identity assigned to the addressed VP/VC Subnetwork connection
Response:	Switch To SubnetworkConnection Result	Result of the Change-over

7.3.33 Topology Info Changes

In order that each PNO is able to determine appropriate routes for VP/VC connections (both when establishing or reconfiguring VP/VC connections) it is essential that each PNO keeps an up to date map of the topology of the network. Using the co-operative management approach, it is only necessary for the PNOs to know the topology of the inter-PNO connections, not the topology of each PNO's own network. These inter-PNO connections may be international links or they may be links between multiple PNOs located within a single country.

To be able to keep the local inter-PNO topology map up to date, each PNO shall send an indication of any changes that it makes to its inter-PNO transmission links to all the other PNOs. A similar message can be sent by a new PNO when joining the network. The details that should be sent shall include the installed capacity of the transmission link, the identity of the PNOs at each end and the status of the transmission links.

When two PNOs contract to connect together a new transmission link, an agreement is required as to which one of the PNOs will send out the topology change and assign an identity to the link.

This function has two elements that require a different set of parameters, one for handling the addition or removal of PNOs and one for handling the addition, modification and removal of inter PNO links.

For PNO addition/removal

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	PNO Subnetwork Id	Identity assigned to the PNO
	Message Status	Addition or removal of PNO
Response-	PNO Subnetwork Id	Identity of PNO sending confirmation

For Inter PNO link addition/modification/removal

It is a confirmed type of operation. The following parameters are associated with the Request and Response primitives of the function:

Request-	ATM Path Cell Rate	Capacity of the Inter-PNO link
	PNO Subnetwork Id owning the link	Identity of the PNO owning the link
	Interconnected PNO Subnetwork Id	Identity of the interconected PNO
	Message Type	Add, modify or remove inter-PNO link
	Link Status	Link active or faulty
	Inter PNO ATM Path QoS	QoS of physical link
	Inter PNO ATM Path Id	Identity of the link
Response-	PNO Subnetwork Id	Identity of PNO sending confirmation
	Inter PNO ATM Path Id	Identity of the link

8 Management information model

8.1 Introduction

NOTE 1: The visibility across the X-interface of object classes and their attributes, as well as access control information are a subject for agreements between Operators.

NOTE 2: The present model is restricted to the VP Bearer Service.

The information model described in this subclause is inherited from the ones contained in the ES 200 653 [9] and from ITU-T Recommendation I.751 [3].

For the specialization of the X Managed Object Classes the concept of Profile, as defined in the ES 200 653 [9], is used. According to this definition, to profile a managed object is to add additional normative text which restricts conditionality (e.g. specifies that a conditional package is or is not present) and adds behaviour to it.

Using this concept, the description of the Managed Object Classes is done in two steps (see figure 12):

1. A profile of the object classes of ES 200 653 [9] and ITU-T Recommendation I.751 [3] adapted for the X is defined;

2. The X Managed Object Classes are inherited from the profile defined in step 1.

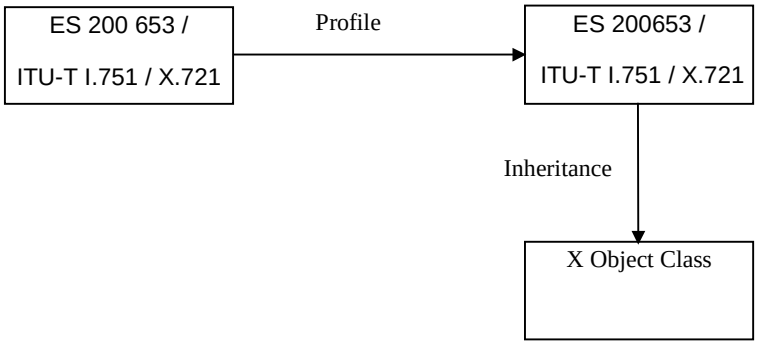
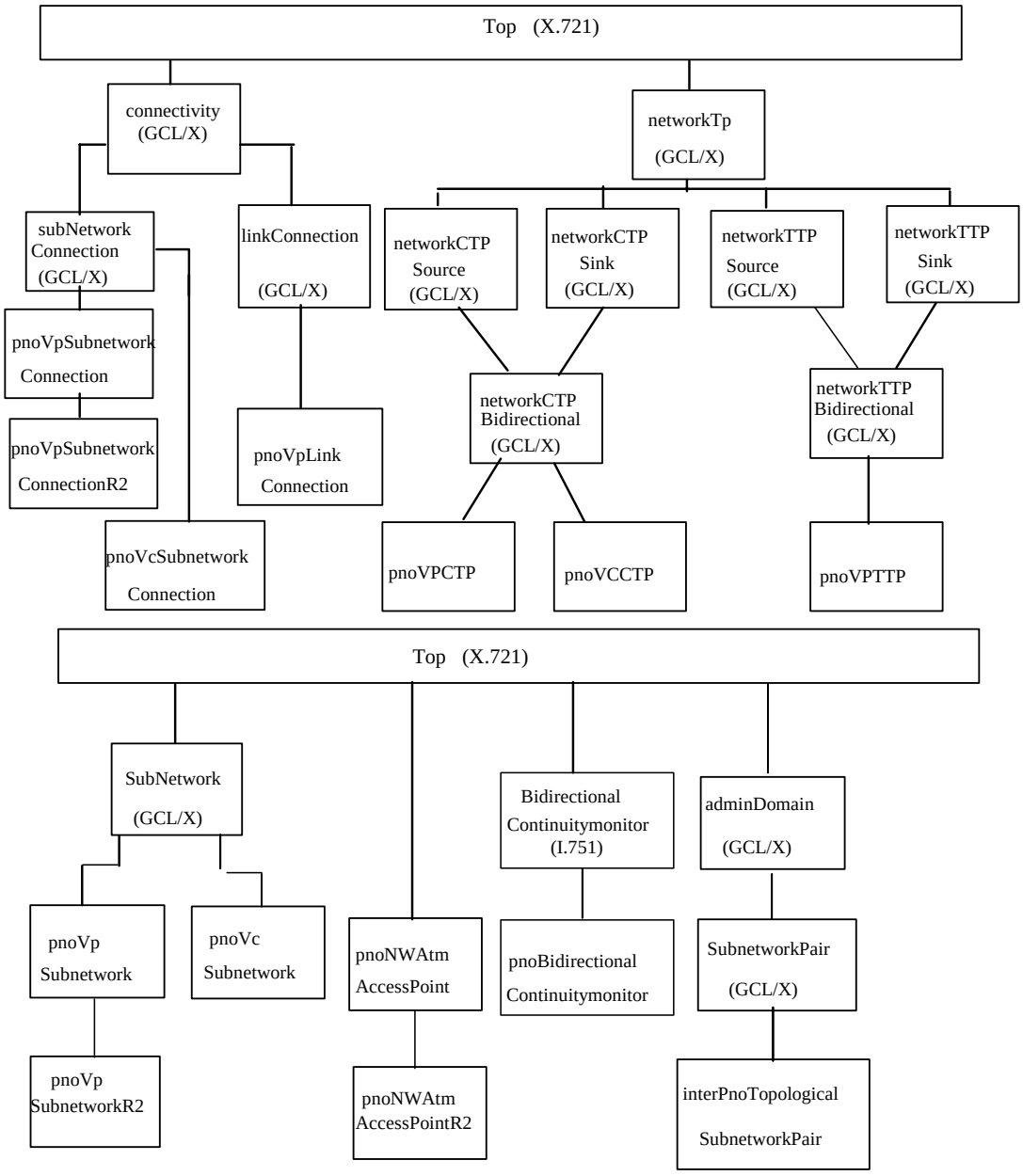


Figure 9: Methodology for the X Object Classes specialization

8.2 Inheritance tree



X.721: ITU-T Recommendation X.721 [2] - Definition of mnagement Information.
GCL: ES 200 653 [3] - Network level Generic Classes Library.
GCL/X: Profiled object classes from ES 200 653 [3].

NOTE: For reason of simplicity, only the objects defined in this specification are shown in the Inheritance tree.

Figure 10: Inheritance tree

8.3 Entity Relationship Diagram

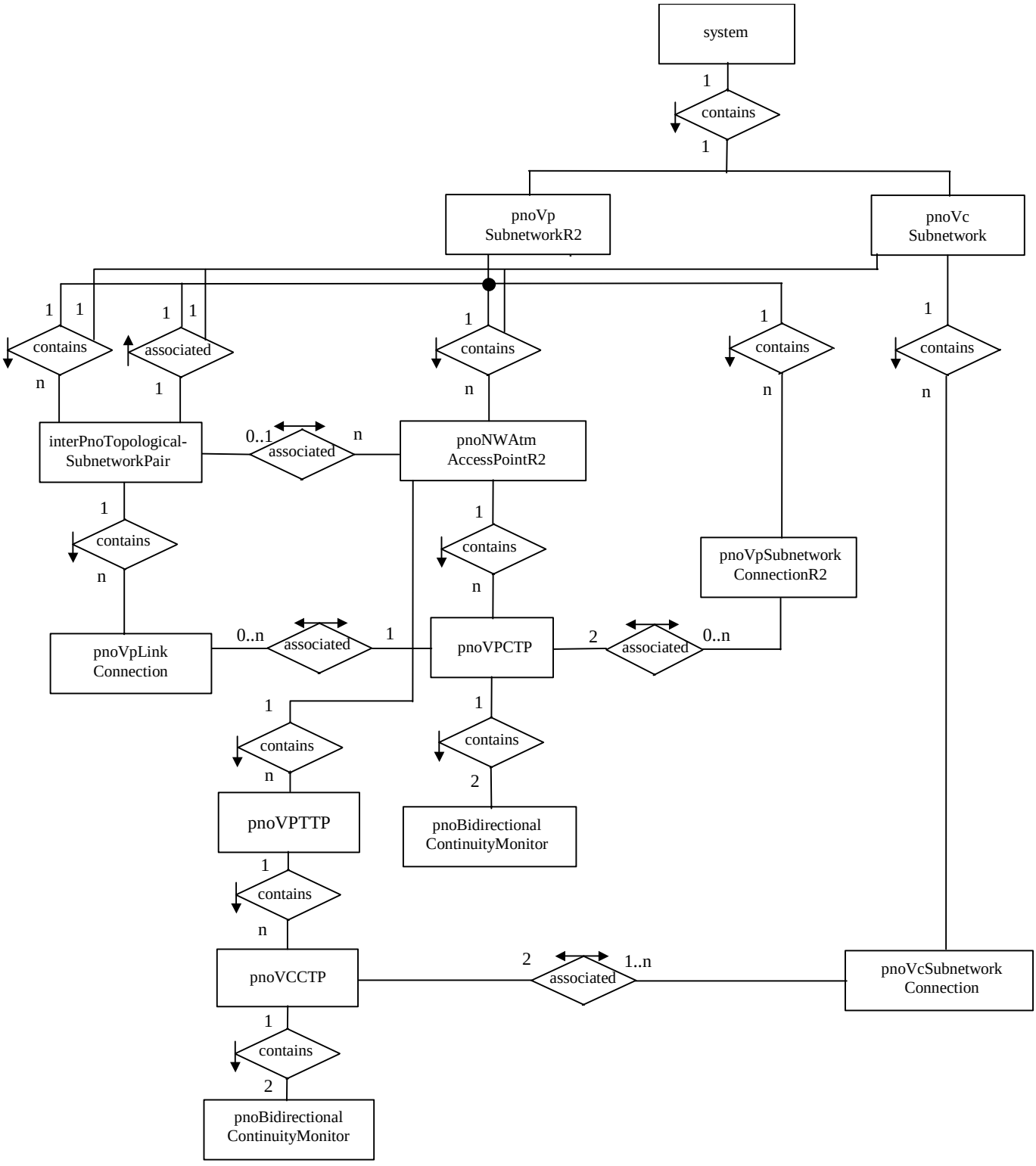


Figure 11: Entity Relationship Diagram (superclasses not shown)

8.4 ES 200 653, ITU-T Recommendations I.751 and X.721 Object Classes adapted for XIF

Subclause 8.4 specifies which conditional packages are actually used in the superclasses of the X Interface model. It is not a formal GDMO description.

8.4.1 Connectivity

PROFILE NOTE:

The attribute *signalid* will not be used in the X Managed Object Class. Its value should be set to NULL.

The conditional packages:

- assignmentStatePackage;
- availabilityStatusPackage;
- lifecycleStatePackage;
- alarmSeverityAssignmentPointerPackage;
- supportedByPackage;
- userLabelPackage; and
- qualityOfConnectivityServicePackage;

are not required for the X Managed Object Classes.

The conditional packages:

- createDeleteNotificationsPackage;
- attributeValueChangeNotificationPackage;
- stateChangeNotificationPackage;
- administrativeStatePackage;
- operationalStatePackage;
- tmnCommunicationAlarmInformationPackage; and
- zEndNWTPListPackage;

are mandatory for the X Managed Object Classes.

```
connectivity MANAGED OBJECT CLASS.

DERIVED FROM "ITU-T Recommendation X.721 [10]/ISO/IEC 10165-2 [11]: 92":top;
CHARACTERIZED BY
connectivityPackage PACKAGE
BEHAVIOUR
connectivityBehaviour BEHAVIOUR
DEFINED AS
"See ES 200 653 [9]"
;;
ATTRIBUTES
signalid GET,
mode GET,
aEndNWTPList GET,
"ITU-T Recommendation M.3100 [1]:92":directionality GET;

CONDITIONAL PACKAGES
"ITU-T Recommendation M.3100 [1]:92":createDeleteNotificationsPackage PRESENT IF "See
ES 200 653 [9]",
"ITU-T Recommendation M.3100 [1]:92":attributeValueChangeNotificationPackage PRESENT IF "See
ES 200 653 [9]",
```

```
"ITU-T Recommendation M.3100 [1]:92":stateChangeNotificationPackage PRESENT IF "See ES 200 653 [9]",
administrativeStatePackage PRESENT IF "See ES 200 653 [9]",
assignmentStatePackage PRESENT IF "See ES 200 653 [9]",
"ITU-T Recommendation X.721 [10]": availabilityStatusPackage PRESENT IF "See ES 200 653 [9]",
lifecycleStatePackage PRESENT IF "See ES 200 653 [9]",
"ITU-T Recommendation M.3100 [1]:92":operationalStatePackage PRESENT IF "See ES 200 653 [9]",
"ITU-T Recommendation M.3100 [1]:92":tmnCommunicationsAlarmInformationPackage PRESENT IF "See
ES 200 653 [9]",
"ITU-T Recommendation M.3100 [1]:92":alarmSeverityAssignmentPointerPackage PRESENT IF "See
ES 200 653 [9]",
supportedByPackage PRESENT IF "See ES 200 653 [9]",
"ITU-T Recommendation M.3100 [1]:92":userLabelPackage PRESENT IF "See ES 200 653 [9]",
qualityOfConnectivityServicePackage PRESENT IF "See ES 200 653 [9]",
zEndNWTPListPackage PRESENT IF "See ES 200 653 [9]";
;;
REGISTERED AS {es200653MObjectClass 6};
```

8.4.2 Link Connection

PROFILE NOTE:

The conditional packages:

- serverTrailPackage;
- CompositePointerPackage;
- layerTrailPackage;
- zEndNWTPListPackage (Inherited from OC "connectivity").

are not required for the X Managed Object Class

```
linkConnection MANAGED OBJECT CLASS
DERIVED FROM connectivity;
CHARACTERIZED BY
linkConnectionPackage PACKAGE
BEHAVIOUR
linkConnectionBehaviour BEHAVIOUR
DEFINED AS "
„SeeES 200 653"
;;
ATTRIBUTES
"ITU-T Recommendation M.3100[1]92":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 {es 200653MObjectClass 13};
```

8.4.3 Subnetwork Connection

PROFILE NOTE:

The conditional packages:

- compositePointerPackage;
- componentPointerPackage;
- userLabelPackage;
- durationSchedulingPackage;
- dailyBasisSchedulingPackage;
- weeklyBasisSchedulingPackage;

- monthlyBasisSchedulingPackage;
- occasionalSchedulingPackage;

are not required for the X Managed Object Classes.

```
subNetworkConnection MANAGED OBJECT CLASS
DERIVED FROM connectivity;
CHARACTERIZED BY
subNetworkConnectionPackage PACKAGE
BEHAVIOUR
subNetworkConnectionBehaviour BEHAVIOUR
DEFINED AS
"See ES 200 653 [9]"
;;
ATTRIBUTES
subnetworkConnectionId GET;
CONDITIONAL PACKAGES
compositePointerPackage PRESENT IF "See ES 200 653 [9]",
componentPointerPackage PRESENT IF "See ES 200 653 [9]",
"ITU-T Recommendation M.3100 [1]:92":userLabelPackage PRESENT IF "See ES 200 653 [9]",
durationSchedulingPackage PRESENT IF "See ES 200 653 [9]",
dailyBasisSchedulingPackage PRESENT IF "See ES 200 653 [9]",
weeklyBasisSchedulingPackage PRESENT IF "See ES 200 653 [9]",
monthlyBasisSchedulingPackage PRESENT IF "See ES 200 653 [9]",
occasionalSchedulingPackage PRESENT IF "See ES 200 653 [9]";
REGISTERED AS {es200653MObjectClass 24};
```

8.4.4 Network TP

PROFILE NOTE:

The attribute *signalid* will not be used in the X Managed Object Class. Its value should be set to NULL.

The attribute "mode" is currently not used but will be kept. It's value will always be pointToPoint(0).

The conditional packages connectivityPointerPackage, neAssignmentPackage, tmnCommunicationsAlarmInformationPackage, sncPointerPackage, networkTPPointerPackage, userLabelPackage, assignmentStatePackage, availabilityStatusPackage, lifecycleStatePackage, supportedByPackage, attributeValueChangeNotificationPackage, administrativeStatePackage, operationalStatePackage and stateChangeNotificationPackage are not required for the X Managed Object Class.

```
networkTP MANAGED OBJECT CLASS
DERIVED FROM "ITU-T Recommendation X.721 [10]/ISO/IEC 10165-2 [11]: 92":top;
CHARACTERIZED BY
"ITU-T Recommendation M.3100 [1]:92":createDeleteNotificationsPackage,
networkTPPackage PACKAGE
BEHAVIOUR
networkTPBehaviour BEHAVIOUR
DEFINED AS
"See ES 200 653 [9]"
;;
ATTRIBUTES
signalid GET,
mode GET;
CONDITIONAL PACKAGES
connectivityPointerPackage PRESENT IF "See ES 200 653 [9]",
neAssignmentPackage PRESENT IF "See ES 200 653 [9]",
"ITU-T Recommendation M.3100 [1]:92":tmnCommunicationsAlarmInformationPackage PRESENT IF "See
ES 200 653 [9]",
sncPointerPackage PRESENT IF "See ES 200 653 [9]",
networkTPPointerPackage PRESENT IF "See ES 200 653 [9]",
"ITU-T Recommendation M.3100 [1]:92":attributeValueChangeNotificationPackage PRESENT IF "See
ES 200 653 [9]",
"ITU-T Recommendation M.3100 [1]:92":userLabelPackage PRESENT IF "See ES 200 653 [9]",
administrativeStatePackage PRESENT IF "See ES 200 653 [9]",
assignmentStatePackage PRESENT IF "See ES 200 653 [9]",
"ITU-T Recommendation X.721 [10]":availabilityStatusPackage PRESENT IF "See ES 200 653 [9]",
lifecycleStatePackage PRESENT IF "See ES 200 653 [9]",
"ITU-T Recommendation M.3100 [1]:92":operationalStatePackage PRESENT IF "See ES 200 653 [9]",
"ITU-T Recommendation M.3100 [1]:92":stateChangeNotificationPackage PRESENT IF "See ES 200 653 [9]",
supportedByPackage PRESENT IF "See ES 200 653 [9]";
;;
REGISTERED AS {es200653MObjectClass 18};
```

8.4.5 Network CTP Sink

PROFILE NOTE:

The conditional packages channelNumberPackage, ctpInstancePackage, networkCTPPackage and serverTTPPointerPackage, are not required for the X Managed Object Class.

```
networkCTPSink MANAGED OBJECT CLASS
DERIVED FROM networkTP;
CHARACTERIZED BY
networkCTPSinkPackage PACKAGE
BEHAVIOUR
networkCTPSinkBehaviour BEHAVIOUR
DEFINED AS
"See ES 200 653 [9]"
;;
CONDITIONAL PACKAGES
"ITU-T Recommendation M.3100 [1]:92":channelNumberPackage PRESENT IF "See ES 200 653 [9]",
"ITU-T Recommendation M.3100 [1]:92":ctpInstancePackage PRESENT IF "See ES 200 653 [9]",
networkCTPPackage PRESENT IF "See ES 200 653 [9]",
serverTTPPointerPackage PRESENT IF "See ES 200 653 [9]";
;;
REGISTERED AS {es200653MObjectClass 15};
```

8.4.6 Network CTP Source

PROFILE NOTE:

The conditional packages channelNumberPackage, ctpInstancePackage, networkCTPPackage and serverTTPPointerPackage, are not required for the X Managed Object Class.

```
networkCTPSource MANAGED OBJECT CLASS
DERIVED FROM networkTP;
CHARACTERIZED BY
networkCTPSourcePackage PACKAGE
BEHAVIOUR
networkCTPSourceBehaviour BEHAVIOUR
DEFINED AS
"See ES 200 653 [9]"
;;
CONDITIONAL PACKAGES
"ITU-T Recommendation M.3100 [1]:92":channelNumberPackage PRESENT IF "See ES 200 653 [9]",
"ITU-T Recommendation M.3100 [1]:92":ctpInstancePackage PRESENT IF "See ES 200 653 [9]",
networkCTPPackage PRESENT IF "See ES 200 653 [9]",
serverTTPPointerPackage PRESENT IF "See ES 200 653 [9]";
;;
REGISTERED AS {es200653MObjectClass 16};
```

8.4.7 Network TTP sink

PROFILE NOTE:

The conditional packages supportableClientListPackage, ttpInstancePackage, clientCTPListPackage are not required for the X Managed Object Class.

```
networkTTPSink MANAGED OBJECT CLASS
DERIVED FROM networkTP;
CHARACTERIZED BY
networkTTPSinkPackage PACKAGE
BEHAVIOUR
networkTTPSinkBehaviour BEHAVIOUR
DEFINED AS
"See ES 200 653 [9]"
;;
CONDITIONAL PACKAGES
"Recommendation M.3100 [1]: 1992":supportableClientListPackage
PRESENT IF "an instance supports it",
"Recommendation M.3100 [1]: 1992":ttpInstancePackage
PRESENT IF "an instance supports it",
clientCTPListPackage
PRESENT IF "an instance supports it";
REGISTERED AS {es200653MObjectClass 20};
```

8.4.8 Network TTP source

PROFILE NOTE:

The conditional packages supportableClientListPackage, clientCTPListPackage are not required for the X Managed Object Class.

The conditional package, ttpInstancePackage is required for the X Managed Object Class.

```
networkTTPSource MANAGED OBJECT CLASS
DERIVED FROM networkTP;
CHARACTERIZED BY
networkTTPSourcePackage PACKAGE
BEHAVIOUR
networkTTPSourceBehaviour BEHAVIOUR
DEFINED AS
"See ES 200 653 [9]"
;;
CONDITIONAL PACKAGES
"ITU-T Recommendation M.3100 [1]: 1992":supportableClientListPackage
PRESENT IF "an instance supports it",
"Recommendation M.3100 [1]: 1992":ttpInstancePackage
PRESENT IF "an instance supports it",
clientCTPListPackage
PRESENT IF "an instance supports it";
REGISTERED AS {es200653MObjectClass 21};
```

8.4.9 Network CTP Bi-directional

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

8.4.10 Network TTP bi-directional

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

8.4.11 Admin Domain

PROFILE NOTE:

The conditional packages adminDomainIdPackage, systemTitlePackage and userLabelPackage, are not required for the X Managed Object Class.

```
adminDomain MANAGED OBJECT CLASS
DERIVED FROM "ITU-T Recommendation X.721 [10]/ISO/IEC 10165-2 [11]: 92":top;
CHARACTERIZED BY
adminDomainPackage PACKAGE
BEHAVIOUR
adminDomainBehaviour BEHAVIOUR
DEFINED AS
"See ES 200 653 [9]"
;;
;
CONDITIONAL PACKAGES
adminDomainIdPackage PRESENT IF "See ES 200 653 [9]",
systemTitlePackage PRESENT IF "See ES 200 653 [9]",
"ITU-T Recommendation M.3100 [1]:92":userLabelPackage PRESENT IF "See ES 200 653 [9]";
;;
REGISTERED AS {es200653MObjectClass 2};
```

8.4.12 Subnetwork Pair

PROFILE NOTE:

The attributes *trailList* and *signalid* will not be used in the X-interface Managed Object Class. The attribute *trailList* will always empty. The attribute *signalid* will be set to NULL.

```
subnetworkPair MANAGED OBJECT CLASS
DERIVED FROM adminDomain;
CHARACTERIZED BY
subnetworkPairPackage PACKAGE
BEHAVIOUR
networkTPBehaviour BEHAVIOUR
DEFINED AS
"See ES 200 653 [9]"
;;
ATTRIBUTES
aEndPoint GET,
zEndPoint GET,
trailList GET,
subNetworkPairId GET,
signalid GET;
;;
REGISTERED AS {es200653MObjectClass 25};
```

8.4.13 Sub-Network

PROFILE NOTE:

The conditional packages attributeValueChangeNotificationPackage, signalidPackage, userLabelPackage, assignmentStatePackage, availabilityStatusPackage, lifecycleStatePackage, supportedByPackage, containedNWCTPListPackage, containedNWTTPListPackage, containedLinkListPackage, containedSubNetworkListPackage, containedInSubNetworkListPackage and linkPointerListPackage are not required for the X Managed Object Class.

The conditional packages stateChangeNotificationPackage,, subNetworkIdPackage, administrativeStatePackage and operationalStatePackage, are mandatory for the X Managed Class.

```
subNetwork MANAGED OBJECT CLASS
DERIVED FROM "ITU-T Recommendation X.721 [10]/ISO/IEC 10165-2 [11]: 92":top;;
CHARACTERIZED BY
"ITU-T Recommendation M.3100 [1]:92":createDeleteNotificationsPackage,
subNetworkPackage PACKAGE
BEHAVIOUR
subNetworkBehaviour BEHAVIOUR
DEFINED AS
"See ES 200 653 [9]"
;;
CONDITIONAL PACKAGES
"ITU-T Recommendation M.3100 [1]:92":stateChangeNotificationPackage PRESENT IF "See ES 200 653 [9]",
"ITU-T Recommendation M.3100 [1]:92":attributeValueChangeNotificationPackage PRESENT IF "See
ES 200 653 [9]",
signalidPackage PRESENT IF "See ES 200 653 [9]",
"ITU-T Recommendation M.3100 [1]:92":userLabelPackage PRESENT IF "See ES 200 653 [9]",
subNetworkIdPackage PRESENT IF "See ES 200 653 [9]",
administrativeStatePackage PRESENT IF "See ES 200 653 [9]",
assignmentStatePackage PRESENT IF "See ES 200 653 [9]",
"ITU-T Recommendation X.721 [10]":availabilityStatusPackage PRESENT IF "See ES 200 653 [9]",
lifecycleStatePackage PRESENT IF "See ES 200 653 [9]",
"ITU-T Recommendation M.3100 [1]:92":operationalStatePackage PRESENT IF "See ES 200 653 [9]",
supportedByPackage PRESENT IF "See ES 200 653 [9]",
containedNWCTPListPackage PRESENT IF "See ES 200 653 [9]",
containedNWTTPListPackage PRESENT IF "See ES 200 653 [9]",
containedLinkListPackage PRESENT IF "See ES 200 653 [9]",
containedSubNetworkListPackage PRESENT IF "See ES 200 653 [9]",
containedInSubNetworkListPackage PRESENT IF "See ES 200 653 [9]",
linkPointerListPackage PRESENT IF "See ES 200 653 [9]";
REGISTERED AS {es200653MObjectClass 23};
```

8.4.14 bidirectionalContinuityMonitor

PROFILE NOTE:

The conditional package flowDirectionPackage is mandatory in the X Managed Object Class.

The managing system shall request the creation of the bidirectionalContinuityMonitor object instance (or its subclasses) and the activation of the Continuity Check function at the same time in the CREATE request by setting the sinkCCMechanismActive attribute to TRUE.

```
bidirectionalContinuityMonitor MANAGED OBJECT CLASS
    DERIVED FROM "Rec. X.721 [10] |ISO/IEC-10165-2": top;
    CHARACTERIZED BY
    "ITU-T Recommendation M.3100 [1]:92":tmnCommunicationsAlarmInformationPackage,
    "ITU-T Recommendation M.3100 [1]:92":stateChangeNotificationPackage,
    bidirectionalContinuityMonitorPackage PACKAGE
BEHAVIOUR bidirectionalContinuityMonitorBeh;
ATTRIBUTES
    continuityMonitorId         GET,
sinkCCMechanismActive           GET,
sourceCCMechanismActive     GET,
"ITU-T Recommendation X.721 [10] | ISO/IEC-10165-2": operationalState   GET;
ACTIONS
controlCC;;;
CONDITIONAL PACKAGES
flowDirectionPackage
PRESENT IF "the monitor object instance is contained in CTP";
REGISTERED AS {i751ObjectClass 7};
bidirectionalContinuityMonitorBeh BEHAVIOUR
DEFINED AS "See I.751 [3]";;
```

8.4.15 system

PROFILE NOTE:

The conditional packages administrativeStatePackage, supportedFeaturesPackage are not required

for the X managed object classes. Attribute operationalState always has the value "enabled".

Attribute usageState always has the value "active".

Attribute systemTitle always has the value NULL.

```
system  MANAGED OBJECT CLASS
DERIVED FROM    top;
CHARACTERIZED BY
systemPackage   PACKAGE
ATTRIBUTES
systemId    GET,
systemTitle GET,
operationalState    GET,
usageState  GET;;;
CONDITIONAL PACKAGES
administrativeStatePackage  PACKAGE
ATTRIBUTES
administrativeState GET-REPLACE;
REGISTERED AS   {smi2Package 14};   PRESENT IF "an instance supports it.",
supportedFeaturesPackage    PACKAGE
ATTRIBUTES
supportedFeatures   GET-REPLACE ADD-REMOVE;
REGISTERED AS   {smi2Package 15}; PRESENT IF "an instance supports it.";
REGISTERED AS   {smi2MObjectClass 13};
```

8.5 ATM X VP/VC Object Classes

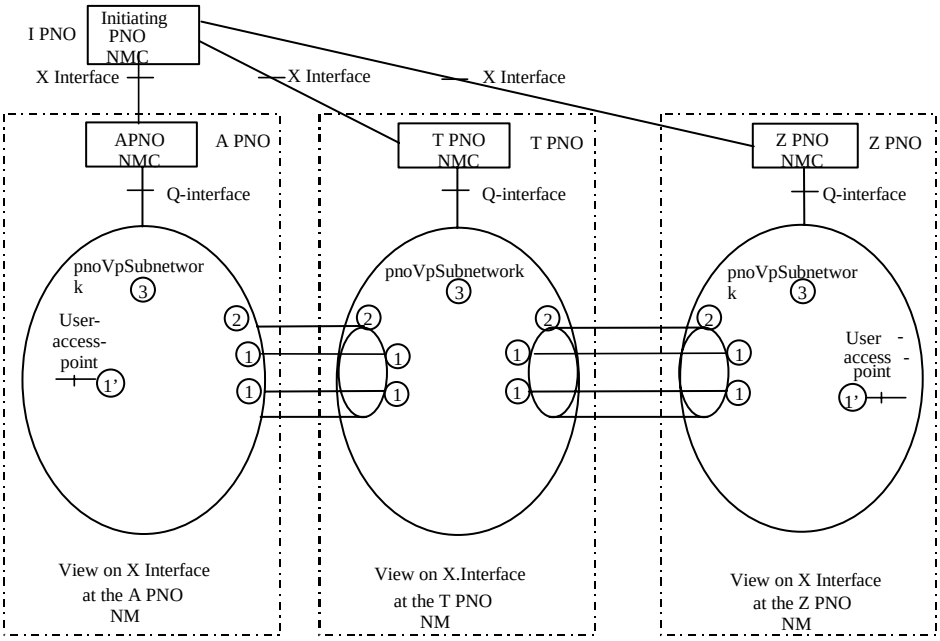
8.5.1 Introduction

The Information Model described represents the view of the ***Initiating*** PNO on the Transit and ***Z*** PNOs.

The ***Initiating*** PNO takes the role of Manager for the establishment and control of a User-to-user VP Connection. The created instances of the Objects are only managed by the ***Initiating*** PNO.

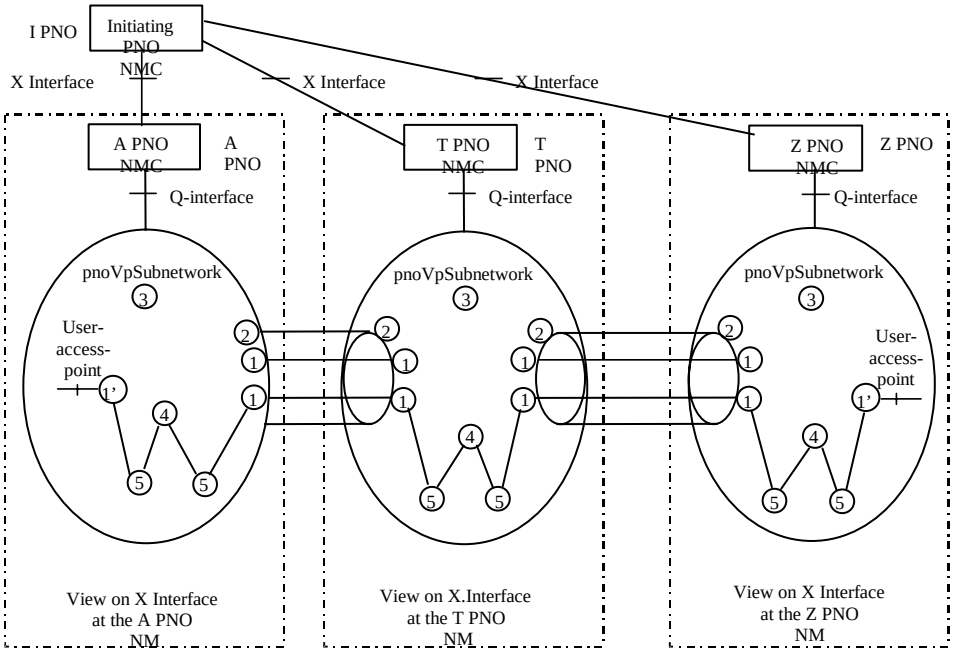
The following figures illustrate the management view from the ***Initiating*** PNO. Figure 12 represents the management view on the topological objects. Figure 13 represents the view after a User-to-user VP Connection has been setup. Figure 14 represents the view after a Vp User-to-network Connection has been setup and figure 15 the view after a Vc User-to-user connection has been setup.

In these examples the Initiating PNO and the A PNO are not the same.



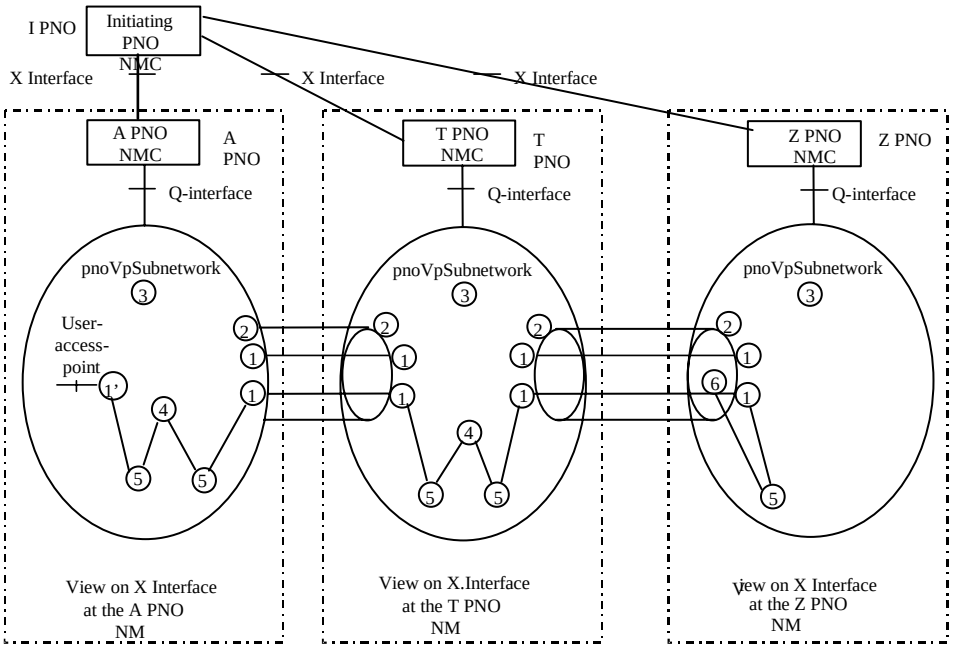
Legend:
1': pnoNW AtmAccessPoint for user access (no association with (2)).
1: pnoNW AtmAccessPoint.
2: InterPno TopologicalSubnetworkPair.
3: pno VpSubnetwork.
NM: Network Management.

Figure 12: Topological Management View on the X-Interface



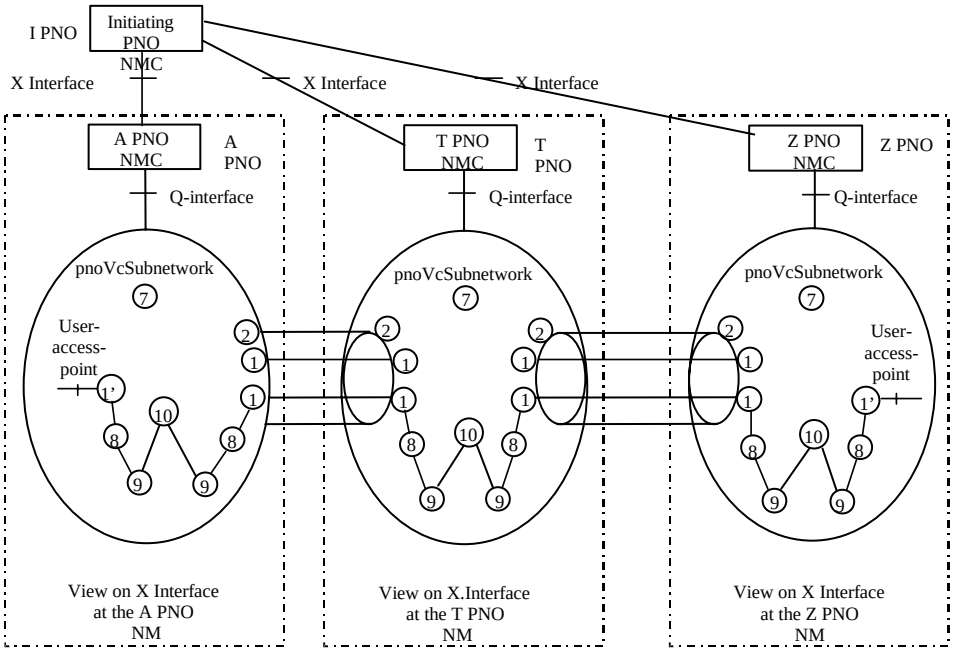
Legend:
1': pnoNW AtmAccessPoint for user access (no association with (2)).
1: pnoNW AtmAccessPoint.
2: interPnoTopologicalSubnetworkPair.
4: pno VpSubnetworkConnection.
5: pno VPCTP (VP Connection Termination Point.

Figure 13: Management View after a User-to-user VP Connection has been setup



Legend:
1': pnoNW AtmAccessPoint for user access (no association with (2)).
1: pnoNW AtmAccessPoint.
2: interPnoTopologicalSubnetworkPair.
4: pnoVpSubnetworkConnection.
5: pnoVPCTP (VP connection Termination Point).

Figure 14: Management View after a User-to-network VP Connection has been setup



Legend:
1': pnoNW AtmAccessPoint for user access (no association with (2)).
1: pnoNW AtmAccessPoint.
2: interPno TopologicalSubnetworkPair.
8: pno VpTTP (VP Trail Termination Point)
9: pnoVCCTP (VC Connection Termination Point).

Figure 15: Management View after a User-to-user VC Connection has been setup

For describing the topology in the Information Model, the A and Z Subnetworks and Access Points correspond to the name of the PNOs taken in the alphabetical order, according to what is specified in the ITU-T Recommendation M.1400 [2]. In the context of a VP/VC Connection, this notation is used independently for naming the Origin (A) and Destination (Z).

The VP Connection Id that is used for identifying uniquely the end-to-end VP Connection, is composed in the Information Model of two attributes:

- InitiatingPnoSubnetwork - Which identifies the Initiating Subnetwork;
- InitiatingVpConnectionId - Which is an identifier of the VP Connection allocated by the Initiating Subnetwork.

The same applies to a VP link Connection Id (InitiatingPnoSubnetwork, InitiatingVpConnectionId) and a VC Connection Id (InitiatingPnoSubnetwork, InitiatingVcConnectionId)

8.5.2 Connection Fragment

PNO VC Subnetwork Connection

```
pnoVcSubnetworkConnection MANAGED OBJECT CLASS
DERIVED FROM "ES 200 653 [9] ":subNetworkConnection;
CHARACTERIZED BY
pnoVcSubnetworkConnectionPackage PACKAGE
BEHAVIOUR
pnoVcSubnetworkConnectionBehaviour BEHAVIOUR
DEFINED AS
"It represents a VC connection across a PNO subnetwork. The Initiating PNO only views this
connection as a whole, with no details regarding the identification of VCCs which compose this
connection inside the PNO domain.
In order to guarantee a unique name for a pnoVcSubnetworkConnection object created as a result of
the Reserve Request, the value for its naming Attribute (SubnetworkConnectionId attribute, inherited
from the SubnetworkConnection MOC) shall be set to the concatenation of the values of the
initiatingPnoId and initiatingVcConnectionId.
The attributes initiatingPnoSubnetworkId and initiatingVcConnectionId are used by the Initiating PNO
to assign an identity to the VC Connection.
```

```
The procedure Change Reservation is performed by setting the value of the attribute
changeReservationInformation to the new value.
The procedure Cancel Change is done by replacing the value of this attribute with the default value,
which is null.
The operationalState has two possible values: Disabled or Enabled. Disabled means that a failure has
occurred in the PNO VC Subnetwork Connection. Enabled means that the VC Subnetwork Connection is in
service.
The administrativeState has two possible values: Locked and unlocked. When an instance of this
object is created, the administrativeState shall be set to locked.
A stateChangeNotification (old value = locked, new value = unlocked) will be sent by each T/Z PNO to
the initiating PNO at the beginning of an activation slot.
A stateChangeNotification (old value = unlocked, new value = locked) will be sent to the initiating
PNO at the end of an activation slot.
The procedure Activate VC Subnetwork Connection is performed by setting the administrative state to
unlocked. The procedure Deactivate VC Subnetwork Connection is performed by setting the
administrative state to locked. The activation procedure is only used if the connection is not
unlocked automatically according to the schedule, and it can also be used to allow testing outside
of the scheduled timeslots. Whenever the operationalState has the value Disabled, the activation
procedures will not succeed, and the value of the administrativeState remains locked. If the
activation procedures are used outside of the scheduled timeslots, it may be refused, for whatever
reason, in which case the value of the administrativeState also remains locked.
When an instance of pnoVcSubnetworkConnection is created this either leads to the creation of one or
two instances of pnoVCCTP, or to the usage of instances of pnoVCCTP that are already used by other
pnoVcSubnetworkConnections. In the last case there may not be overlap with the time schedule of
these Connections.
The aEndNWTPList and the zEndNWTPList point to the pnoVCCTP instances associated with the
pnoVcSubnetworkConnection.
The notifications issued by this object are sent to the Initiating PNO.
The object Creation Notification should be sent after the Succesful Reserve Result.
The notification cancelVcNetworkConnectionNotification is sent by the Z PNO to the initiating PNO.
Resilience functionality is included in the Conditional Packages kindOfResiliencePackage,
relatedSncPackage and
SwitchToSncPackage."
;;
ATTRIBUTES
initiatingPnoSubnetworkId GET,
initiatingVcConnectionId GET,
forwardQoSClass GET,
backwardQoSClass GET,
vcSchedulers GET,
changeReservationInformation REPLACE-WITH-DEFAULT
                    DEFAULT VALUE                   ASN1XatmModule.defaultChangeReservationInfo
                    GET-REPLACE;
ACTIONS
activateChange;;;
CONDITIONAL PACKAGES
cancelVcNetworkConnectionNotificationPkg
PRESENT IF "it is a Z PNO Subnetwork for the corresponding VcConnection",
vcTestStatePackage
PRESENT IF "the T/Z PNO wants to know when he can make an internal CC",
kindOfResiliencePackage
PRESENT IF "The Subnetworkconnection is protected by some kind of resilience mechanism.",
relatedSncPackage
PRESENT IF "If there shall be a reference to an other Subnetworkconnection.",
switchToSncPackage
PRESENT IF "If it is an A- or Z connection, and the PNO wants to offer the possibility to change-
over to the traffic of an other connection that has the same End Point.";
REGISTERED AS {xatmObjectClass 7};
PNO VP Link Connection
pnoVpLinkConnection MANAGED OBJECT CLASS
DERIVED FROM "ES 200 653 [9]": linkConnection;
CHARACTERIZED BY
pnoVpLinkConnectionPackage PACKAGE
BEHAVIOUR
pnoVpLinkConnectionBehaviour BEHAVIOUR
DEFINED AS
"It represents a VP connection across an Inter Pno Link. The Initiating PNO only views this
connection as a whole, with no details regarding the identification of trails which serve this
connection.
The attributes initiatingPnoSubnetworkId and initiatingVpConnectionId are used by the Initiating PNO
to assign an identity to the VP Connection.
In order to guarantee a unique name for a pnoVpLinkConnection object created as a result of the
Reserve Request, the value for its naming Attribute (connectionId attribute, inherited from the
linkConnection MOC) shall be set to the concatenation of the values of the initiatingPnoId and
initiatingVpConnectionId in the ASN.1 syntax of the reservePnoVpLinkConnection Action received from
the Manager.
The procedure Change Reservation is performed by setting the value of the attribute
changeReservationInformation to the new value.
```

```
The procedure Cancel Change is done by replacing the value of this attribute with the default value,
which is null.
The operationalState has two possible values: Disabled or Enabled. Disabled means that a failure has
occurred in the PNO VP Link Connection. Enabled means that the VP Link Connection is in service.
The administrativeState has two possible values: Locked and unlocked. When an instance of this
object is created, the administrativeState shall be set to locked.
A stateChangeNotification (old value = locked, new value = unlocked) will be sent by the Agent PNO
to the initiating PNO at the beginning of an activation slot.
A stateChangeNotification (old value = unlocked, new value = locked) will be sent to the initiating
PNO at the end of an activation slot.
The procedure Activate VP Link Connection is performed by setting the administrative state to
unlocked. The procedure Deactivate VP Link Connection is performed by setting the administrative
state to locked. The activation procedure is only used if the connection is not unlocked
automatically according to the schedule, and it can also be used to allow testing outside of the
scheduled timeslots. Whenever the operationalState has the value Disabled, the activation procedures
will not succeed, and the value of the administrativeState remains locked. If the activation
procedures are used outside of the scheduled timeslots, it may be refused, for whatever reason, in
which case the value of the administrativeState also remains locked.
When an instance of pnoVpLinkConnection is created, this either leads to the creation of one
instance of pnoVPCTP, or to the usage of an instance of pnoVPCTP that is already used by other
pnoVpLinkConnections. In the last case there may not be overlap with the time schedule of these
Connections.
The aEndNWTPList points to the pnoVPCTP instance associated with the pnoVpLinkConnection.
The notifications issued by this object are sent to the Initiating PNO.
The object Creation Notification should be sent after the Succesful Reserve Result.
The notification cancelVpLinkConnectionNotification is sent to the initiating PNO"
;;
ATTRIBUTES
initiatingPnoSubnetworkId GET,
initiatingVpConnectionId GET,
forwardQoSClass GET,
backwardQoSClass GET,
vpSchedulers GET,
changeReservationInformation REPLACE-WITH-DEFAULT
                    DEFAULT VALUE                   ASN1XatmModule.defaultChangeReservationInfo
                    GET-REPLACE;
ACTIONS
activateChange;;;
CONDITIONAL PACKAGES
cancelVpLinkConnectionNotificationPkg
PRESENT IF "if a PNO wants to use it";
REGISTERED AS {xatmObjectClass 8};
PNO VP Subnetwork Connection
pnoVpSubnetworkConnection MANAGED OBJECT CLASS
DERIVED FROM "ES 200 653 [9] ":subNetworkConnection;
CHARACTERIZED BY
pnoVpSubnetworkConnectionPackage PACKAGE
BEHAVIOUR
pnoVpSubnetworkConnectionBehaviour BEHAVIOUR
DEFINED AS
"It represents a VP connection across a PNO subnetwork. The Initiating PNO only views this
connection as a whole, with no details regarding the identification of VPCs which compose this
connection inside the PNO domain.
The attributes initiatingPnoSubnetworkId and initiatingVpConnectionId are used by the Initiating PNO
to assign an identity to the VP Connection.
The procedure Change Reservation is performed by setting the value of the attribute
changeReservationInformation to the new value.
The procedure Cancel Change is done by replacing the value of this attribute with the default value,
which is null.
The operationalState has two possible values: Disabled or Enabled. Disabled means that a failure has
occurred in the PNO VP Subnetwork Connection. Enabled means that the VP Subnetwork Connection is in
service.
The administrativeState has two possible values: Locked and unlocked. When an instance of this
object is created, the administrativeState shall be set to locked.
A stateChangeNotification (old value = locked, new value = unlocked) will be sent by each T/Z PNO to
the initiating PNO at the beginning of an activation slot.
A stateChangeNotification (old value = unlocked, new value = locked) will be sent to the initiating
PNO at the end of an activation slot.
The procedure Activate VP Subnetwork Connection is performed by setting the administrative state to
unlocked. The procedure Deactivate VP Subnetwork Connection is performed by setting the
administrative state to locked. The activation procedure is only used if the connection is not
unlocked automatically according to the schedule, and it can also be used to allow testing outside
of the scheduled timeslots. Whenever the operationalState has the value Disabled, the activation
procedures will not succeed, and the value of the administrativeState remains locked. If the
activation procedures are used outside of the scheduled timeslots, it may be refused, for whatever
reason, in which case the value of the administrativeState also remains locked.
When an instance of pnoVpSubnetworkConnection is created this either leads to the creation of one or
two instances of pnoVPCTP, or to the usage of instances of pnoVPCTP that are already used by other
```

```
pnoVpSubnetworkConnections. In the last case there may not be overlap with the time schedule of
these Connections.
The aEndNWTPList and the zEndNWTPList point to the pnoVPCTP instances associated with the
pnoVpSubnetworkConnection.
The notifications issued by this object are sent to the Initiating PNO.
The object Creation Notification should be sent after the Succesful Reserve Result.
The notification cancelVpNetworkConnectionNotification is sent by the Z PNO to the initiating PNO"
;;
ATTRIBUTES
initiatingPnoSubnetworkId GET,
initiatingVpConnectionId GET,
forwardQoSClass GET,
backwardQoSClass GET,
vpSchedulers GET,
changeReservationInformation REPLACE-WITH-DEFAULT
                    DEFAULT VALUE                   ASN1XatmModule.defaultChangeReservationInfo
                    GET-REPLACE;
ACTIONS
activateChange;;;
CONDITIONAL PACKAGES
cancelVpNetworkConnectionNotificationPkg
PRESENT IF "it is a Z PNO Subnetwork for the corresponding VpConnection",
vpTestStatePackage
PRESENT IF "the T/Z PNO wants to know when he can make an internal CC";
REGISTERED AS {xatmObjectClass 1};
PNO VP Subnetwork Connection R2
pnoVpSubnetworkConnectionR2 MANAGED OBJECT CLASS
DERIVED FROM pnoVpSubnetworkConnection;
CHARACTERIZED BY
pnoVpSubnetworkConnectionR2Package PACKAGE
BEHAVIOUR
pnoVpSubnetworkConnectionR2Behaviour BEHAVIOUR
DEFINED AS
"This managed object represents a VP connection across a PNO subnetwork with resilience
functionality included. This object class should be used instead of pnoVpSubnetworkConnecion.
In order to guarantee a unique name for a pnoVpSubnetworkConnectionR2 object created as a result of
the Reserve Request, the value for its naming Attribute (SubnetworkConnectionId attribute, inherited
from the SubnetworkConnection MOC) shall be set to the concatenation of the values of the
initiatingPnoId and initiatingVpConnectionId."
;;;;
CONDITIONAL PACKAGES
kindOfResiliencePackage
PRESENT IF "The Subnetworkconnection is protected by some kind of resilience mechanism.",
relatedSncPackage
PRESENT IF "If there shall be a reference to an other Subnetworkconnection.",
switchToSncPackage
PRESENT IF "If it is an A- or Z connection, and the PNO wants to offer the possibility to change-
over to the traffic of an other connection that has the same End Point.";
REGISTERED AS {xatmObjectClass 9};
```

8.5.3 Connection Termination Point Fragment

Pno VC CTP

```
pnoVCCTP MANAGED OBJECT CLASS
DERIVED FROM "ES 200 653 [9] ":networkCTPBidirectional;
CHARACTERIZED BY
pnoVCCTPPackage PACKAGE
BEHAVIOUR
pnoVCCTPBehaviour BEHAVIOUR
DEFINED AS
" This managed object class represents the endpoint of a PNO VC Subnetwork Connection.
Two instances of the pnoVCCTP object are assigned to a pnoVcSubnetworkConnection instance.
The assignment of the VCI associated with this connection is done by the NearEnd NMS.
Contention for a particular VCI may be a problem i.e. the attempt to simultaneously allocate the
same VCI by two adjacent PNOs on the same Virtual Path going over a physical link. To avoid this, a
scheme should be adopted whereby one PNO starts selecting VCIs from bottom end of the VCI range, and
the other starts from the top. This will be effected by agreement between PNOs for each particular
Virtual Path. The Create/Delete Notifications are not used."
;;
ATTRIBUTES
"ITU-T Recommendation I.751 [3] (1996)":vcCTPId GET;
;;
REGISTERED AS {xatmObjectClass 10};
```

Pno VPCTP

```
pnoVPCTP MANAGED OBJECT CLASS
```

```
DERIVED FROM "ES 200 653 [9] ":networkCTPBidirectional;
CHARACTERIZED BY
pnoVPCTPPackage PACKAGE
BEHAVIOUR
pnoVPCTPBehaviour BEHAVIOUR
DEFINED AS
" This managed object class represents the endpoint of a PNO VP Subnetwork Connection.
Two instances of the pnoVPCTP object are assigned to a pnoVpSubnetworkConnection instance.
The assignment of the VPI associated with this connection is done by the NearEnd NMS.
Contention for a particular VPI may be a problem i.e. the attempt to simultaneously allocate the
same VPI by two adjacent PNOs on the same physical link. To avoid this, a scheme should be adopted
whereby one PNO starts selecting VPIs from bottom end of the VPI range, and the other starts from
the top. This will be effected by agreement between PNOs for each particular physical link. The
Create/Delete Notifications are not used.
"
;;
ATTRIBUTES
"ITU-T Recommendation I.751 [3] (1996)":vpCTPId GET;
;;
REGISTERED AS {xatmObjectClass 2};
```

Pno VP TTP

```
pnoVPTTP MANAGED OBJECT CLASS
DERIVED FROM "ES 200 653 [9] ":networkTTPBidirectional;
CHARACTERIZED BY
pnoVPTTPackage PACKAGE
BEHAVIOUR
pnoVPTTPBehaviour BEHAVIOUR
DEFINED AS
"This managed object class gives the VPI-value of a Virtual Path that is, or can be used as, a -
server- for a -client VC- over an inter-PNO link.
The Virtual Path that provides this capability needs not necessarily to be visible over the X-
interface.
The value of naming attribute tTPId, inherited from OC networkTTPSource should be the same as the
VPI-value of the Virtual Path."
;;;;
REGISTERED AS {xatmObjectClass 11};
```

8.5.4 Topology Fragment

Inter PNO Topological Subnetwork Pair

```
interPnoTopologicalSubnetworkPair MANAGED OBJECT CLASS
DERIVED FROM "ES 200 653 [9] ":subNetworkPair;
CHARACTERIZED BY
"ITU-T Recommendation M.3100 [1] (1995)":createDeleteNotificationsPackage,
"ITU-T Recommendation M.3100 [1] (1995)":stateChangeNotificationPackage,
"ITU-T Recommendation M.3100 [1] (1995)":attributeValueChangeNotificationPackage,
interPnoTopologicalSubnetworkPairPackage PACKAGE
BEHAVIOUR interPnoTopologicalSubnetworkPairDefinition BEHAVIOUR
DEFINED AS
"An interPnoTopologicalSubnetworkPair represents a bundle of physical links between two PNOs at the
cell level. It gives information about the maximum capacity of these physical links. This is used by
the path searching algorithm.
The interPnoTopologicalSubnetworkPair is bidirectional.
The inherited attributes aEndPoint and zEndPoint contain the two subnetwork object instances
associated to the SubnetworkPair. As it was stressed in subclause 9.5.1 the aEnd and zEnd correspond
to the PNOs taken in the alphabetical order.
The trailList attribute inherited from subNetworkPair will always be empty.
The Notifications from this object shall be broadcasted to every PNOs in the Network.
The name type of the attribute subNetworkPairId, inherited from subnetworkPair, is a pString.
Any change in the attribute listOfAtmAccessPointPairResources shall be reported.
OperationalState Disabled means that a failure has occurred in the complete bundle of Inter-PNO
Physical Links. Enabled means that this bundle is in service."
;;
ATTRIBUTES
"Rec. X.721 [10] | ISO/IEC 10165-2 [11] ": operationalState GET,
listOfAtmAccessPointPairResources GET;
;;
REGISTERED AS {xatmObjectClass 3};
```

PNO NW ATM Access Point

```
pnoNWAtmAccessPoint MANAGED OBJECT CLASS
DERIVED FROM "Rec. X.721 [10] | ISO/IEC 10165-2 [11] ": top;
CHARACTERIZED BY
"ITU-T Recommendation M.3100 [1] (1995)":tmnCommunicationsAlarmInformationPackage,
"ITU-T Recommendation M.3100 [1] (1995)":stateChangeNotificationPackage,
```

```
"ITU-T Recommendation M.3100 [1] (1995)":attributeValueChangeNotificationPackage,
pnoNWAtmAccessPointPackage PACKAGE
BEHAVIOUR pnoNWAtmAccessPointDefinition BEHAVIOUR
DEFINED AS
"A pnoNWAtmAccessPoint object represents either an endpoint of a physical link at the cell level
between two PNOs or a User Network Interface (UNI).
For UNI's the value of attribute associatedSubnetworkPairId is always ´UNI´.
When a failure is detected on this Access Point or on the associated Physical Link a failure
notification indication is given across the Xcoop.
The Notifications from this object shall be sent to every PNO Subnetwork.
Changes in attribute maxNumVPIBitsSupported are reported by an attributeValueChangeNotification."
;;
ATTRIBUTES
pnoNWAccessPointId GET,
associatedSubNetworkPairId GET,
"Rec. X.721 [10] | ISO/IEC 10165-2 [11] ": operationalState GET,
"ITU-T Recommendation I.751 [3] (1996)":maxNumVPIBitsSupported GET;
;;
REGISTERED AS {xatmObjectClass 5};
```

PNO NW ATM Access Point R2

```
pnoNWAtmAccessPointR2 MANAGED OBJECT CLASS
DERIVED FROM pnoNWAtmAccessPoint;
CHARACTERIZED BY
pnoNWAtmAccessPointR2Package PACKAGE
BEHAVIOUR pnoNWAtmAccessPointR2Definition BEHAVIOUR
DEFINED AS
"This object class is to be used instead of pnoNWAtmAccessPoint.
Changes in attribute maxNumVCIBitsSupported are reported by an attributeValueChangeNotification."
;;
ATTRIBUTES
"ITU-T Recommendation I.751 [3] (1996)":maxNumVCIBitsSupported GET;
;;
REGISTERED AS {xatmObjectClass 12};
```

PNO VC Subnetwork

```
pnoVcSubnetwork MANAGED OBJECT CLASS
DERIVED FROM "ES 200 653 [9] ":subNetwork;
CHARACTERIZED BY
pnoVcSubnetworkPackage PACKAGE
BEHAVIOUR pnoVcSubnetworkDefinition BEHAVIOUR
DEFINED AS
"A pnoVcSubnetwork Object represents the complete Subnetwork of a certain Operator, from a
topological point of view, at the VC layer.
A PNO VC Subnetwork offers external interfaces to other PNO VC Subnetworks through PNO NW ATM Access
Points.
From a Connectivity point of view, pnoVcSubnetworks are crossed by Subnetwork Connections.
The pnoVcSubnetwork manages the establishment (reservation) and release of Subnetwork Connections.
So pnoVcSubnetworkConnection object instances are created when connections are requested.
The establishment of a subnetwork connection is performed by the action
reservePnoVcSubnetworkConnection. In the case of a positive result the SuccessfulReserve information
has to provide the far-endVPCTPId, the far-end VCCTPId, the far-endAPId and the far-
endAssociatedAPId if it comes from an A- or a Transit Subnetwork. In the case of a successful
reservation in a Z Subnetwork the first part of the CHOICE in SuccessfulReserve may be returned,
instead of the zAddress. In this case, the far-endassociatedAPId could be filled with a ´NULL´
pString.
The Actions inherited from subNetwork:
addToSubNetworkConnection,
deleteFromSubNetworkConnection,
setupSubNetworkConnection,
releaseSubNetworkConnection
are not used.
The Notifications of this object shall be broadcasted to every PNO participating in the X Interface
system.
The Action reservePnoVcSubnetworkConnection is performed by the Initiating PNO with the Transit and
Z PNOs.
The Action giveAvailableLinks can be performed by the initiating PNO with the Transit Pno's and the
Z PNO.
The Action checkUser is performed by the initiating PNO with the Z PNO.
The operationalState has two possible values: Disabled or Enabled. Disabled means that a failure has
occurred in the PNO VC Subnetwork. Enabled means that the Subnetwork is in service.
."
;;
ACTIONS
giveAvailableLinks,
checkUser,
reservePnoVcSubnetworkConnection,
releasePnoVcSubnetworkConnection;
```

```
;
REGISTERED AS {xatmObjectClass 13};
```

PNO VP Subnetwork

```
pnoVpSubnetwork MANAGED OBJECT CLASS
DERIVED FROM "ES 200 653 [9] ":subNetwork;
CHARACTERIZED BY
pnoVpSubnetworkPackage PACKAGE
BEHAVIOUR pnoVpSubnetworkDefinition BEHAVIOUR
DEFINED AS
"A pnoVpSubnetwork Object represents the complete Subnetwork of a certain Operator, from a
topological point of view, at the VP layer.
A PNO VP Subnetwork offers external interfaces to other PNO VP Subnetworks through PNO NW ATM Access
Points.
From a Connectivity point of view, pnoVpSubnetworks are crossed by Subnetwork Connections.
The pnoVpSubnetwork manages the establishment (reservation) and release of Subnetwork Connections
and of Link Connections. So pnoVpSubnetworkConnection and pnoVpLinkConnection object instances are
created when connections are requested.
The establishment of a subnetwork connection is performed by the action
reservePnoVpSubnetworkConnection. In the case of a positive result the SuccessfulReserve information
has to provide the far-endVPCTPId, the far-endAPId and the far-endAssociatedAPId if it comes from a
Transit Subnetwork. In the case of a successful reservation in a Z Subnetwork the first part of the
CHOICE in SuccessfulReserve may be returned, instead of the zAddress. In this case the far-
endassociatedAPId could be filled with a ´NULL´ pString.
The Actions inherited from subNetwork:
addToSubNetworkConnection,
deleteFromSubNetworkConnection,
setupSubNetworkConnection,
releaseSubNetworkConnection
are not used.
The Notifications of this object shall be broadcast to every PNO participating in the X Interface
system.
The Action reservePnoVpSubnetworkConnection is performed by the Initiating PNO with the Transit and
Z PNOs.
The Action giveAvailableLinks can be performed by the initiating PNO with the Transit Pno's and the
Z PNO.
The Action checkUser is performed by the initiating PNO with the Z PNO.
The operationalState has two possible values: Disabled or Enabled. Disabled means that a failure has
occurred in the PNO VP Subnetwork. Enabled means that the Subnetwork is in service.
."
;;
ACTIONS
giveAvailableLinks,
checkUser,
reservePnoVpSubnetworkConnection,
releasePnoVpSubnetworkConnection;
;;
REGISTERED AS {xatmObjectClass 4};
```

PNO VP SubnetworkR2

```
pnoVpSubnetworkR2 MANAGED OBJECT CLASS
DERIVED FROM pnoVpSubnetwork;
CHARACTERIZED BY
pnoVpSubnetworkR2Package PACKAGE
BEHAVIOUR pnoVpSubnetworkR2Definition BEHAVIOUR
DEFINED AS
"This object class is to be used instead of PnoVpSubnetwork.
Next to the establishment (reservation) and release of Subnetwork Connections it manages the same
for Link Connections. So pnoVpSubnetworkConnection and / or pnoVpLinkConnection object instances are
created when connections are requested.
The establishment of a Link Connection is performed by the action reservePnoVpLinkConnection. In
case of a positive result the SuccesfulReserveVpLink Result provides the AccesspointId of the link
and the VPCTPId of the Link connection."
;;
ACTIONS
reservePnoVpLinkConnection,
releasePnoVpLinkConnection;
;;
CONDITIONAL PACKAGES
vpPerformAndResiliencePackage
PRESENT IF "Performance Monitoring and/or Network resilience is supported. If applications from the
initiating PNO
and the A/T/Z-PNO support this package the included action will be used for reserving a
vpConnection, instead of the action reservePnoVpSubnetworkConnection.";
REGISTERED AS {xatmObjectClass 14};
```

8.5.5 Continuity Check Fragment

```
pnoBidirectionalContinuityMonitor MANAGED OBJECT CLASS
DERIVED FROM "ITU-T Recommendation I.751 [3] (1996)":bidirectionalContinuityMonitor;
REGISTERED AS {xatmObjectClass 6};
```

8.6 Package Definitions

8.6.1 Attribute Value Change Notification

This package is defined in ITU-T Recommendation M.3100 [1].

8.6.2 cancelVcNetworkConnectionNotification

```
cancelVcNetworkConnectionNotificationPkg PACKAGE
BEHAVIOUR
cancelVcNetworkConnectionNotificationPkgBehaviour BEHAVIOUR
DEFINED AS
"This package allows the Z PNO to notify the Initiating PNO to release the VC Network Connection."
;;
NOTIFICATIONS
cancelVcNetworkConnectionNotification;
REGISTERED AS {xatmPackage 3};
```

8.6.3 cancelVpLinkConnectionNotification

```
cancelVpLinkConnectionNotificationPkg PACKAGE
BEHAVIOUR
cancelVpLinkConnectionNotificationPkgBehaviour BEHAVIOUR
DEFINED AS
"This package allows a PNO to notify the Initiating PNO to release the VP Link Connection."
;;
NOTIFICATIONS
cancelVpNetworkConnectionNotification;
REGISTERED AS {xatmPackage 4};
```

8.6.4 cancelVpNetworkConnectionNotification

```
cancelVpNetworkConnectionNotificationPkg PACKAGE
BEHAVIOUR
cancelVpNetworkConnectionNotificationPkgBehaviour BEHAVIOUR
DEFINED AS
"This package allows the Z PNO to notify the Initiating PNO to release the VP Network Connection."
;;
NOTIFICATIONS
cancelVpNetworkConnectionNotification;
REGISTERED AS {xatmPackage 1};
```

8.6.5 Create Delete Notification

This package is defined in ITU-T Recommendation M.3100 [1].

8.6.6 flowDirection

This package is defined in ITU-T Recommendation I.751 [3].

8.6.7 KindOfResilience

```
kindOfResiliencePackage PACKAGE
BEHAVIOUR
kindOfResiliencePackageBehaviour BEHAVIOUR
DEFINED AS
"This package indicates that the concerned Subnetworkconnection instance (the working connection)
has been provided with a resilience mechanism by assigning a protection entity to it. On detection
of an error on the working entity there will be a change-over to the traffic from the protection
entity.
Three kinds of protection can be provided:
```

```
-   Protection switching: Where, within the Subnetwork, the working connection and the protection
entity have the same end points, and the detection of an error and the following change-over to the
protection traffic occur automatically, without any network-management action.
Within protection switching the next cases are identified:
1+1 configuration: where the protection entity and the working connection convey the same traffic
simultaneously.
1:1 configuration: where the protection entity does not convey the working connection's traffic
until the change over has occurred.
Recovery with Pre-assigned Resources: Within the Subnetwork, the working connection and the
protection entity (connection) have the same end points. It is presumed that protecting resources of
some kind will be pre-assigned by the Agent. When the change-over to protected traffic has to occur,
the Agent's network management needs to take action to establish and activate the protection
connection.
Fast Rerouting: This includes more than one Subnetwork. The protection connection and the working
connection only have the A user accesspoint and the Z user accesspoint in common. In a T network
they do not have any accesspoint in common. They convey the same traffic simultaneously, and the
change-over to protection resources is performed by management.
The information held in the kindOfResilience attribute emanates from the pnoVpSubnetwork
(pnoVcSubnetwork) managed object as a result of the behaviour of the pnoVpSubnetwork
(pnoVcSubnetwork) managed object."
;;
ATTRIBUTES
KindOfResilience    GET;
REGISTERED AS {xatmPackage 5};
```

8.6.8 relatedSnc

```
relatedSncPackage PACKAGE
ATTRIBUTES
relatedSNC  DEFAULT VALUE ASN1XatmModule.emptyString GET;
REGISTERED AS {xatmPackage 6};
```

8.6.9 State Change Notification

This package is defined in ITU-T Recommendation M.3100 [1].

8.6.10 switchToSnc

```
switchToSncPackage PACKAGE
BEHAVIOUR
switchToSncPackageBehaviour BEHAVIOUR
DEFINED AS
"At the end-point of the instance of Subnetworkconnection that contains this package the possibility
is offered to change-over to the traffic of an other connection. Both connections should refer to
each other.
The information held in the relatedSNC attribute emanates from the pnoVpSubnetwork (pnoVcSubnetwork)
managed object as a result of the behaviour of the pnoVpSubnetwork (pnoVcSubnetwork) managed
object."
;;
ACTIONS
switchToSnc;
REGISTERED AS {xatmPackage 7};
```

8.6.11 TMN Communication Alarm Information

This package is defined in ITU-T Recommendation M.3100 [1].

8.6.12 vcTestState

```
vcTestStatePackage PACKAGE
BEHAVIOUR
vcTestStatePkgBehaviour BEHAVIOUR
DEFINED AS
"This package gives an indication if it is allowed or not to perform tests on the PNO VC
Connection."
;;
ATTRIBUTES
vcTestState GET-REPLACE;
REGISTERED AS {xatmPackage 8};
```

8.6.13 vpPerformAndResilience

```
vpPerformAndResiliencePackage PACKAGE
```

```
BEHAVIOUR
vpPerformAndResilienceBehaviour  BEHAVIOUR
DEFINED AS
"The establishment of a Subnetwork connection is performed by the action
reservePnoEnhancedVpSubnetworkConnection. In case of a positive result the
ReservePnoEnhancedVPResult information has to provide the farEndVPCTPId, the farEndAPId and the
farEndAssociatedAPId if it comes from a Transit Subnetwork. In the case of a successful reservation
in a Z Subnetwork the first part of the CHOICE in SuccessfulPerformResilience may be returned,
instead of the zAddress. In this case the far-endassociatedAPId could be filled with a 'NULL'
pString.
The Action reservePnoEnhancedVpSubnetworkConnection is performed by the Initiating PNO with the
Transit(s) and Z PNOs."
;;
  ACTIONS
reservePnoEnhancedVpSubnetworkConnection;
REGISTERED AS {xatmPackage 9};
```

8.6.14 vpTestState

```
vpTestStatePackage PACKAGE
BEHAVIOUR
vpTestStatePkgBehaviour BEHAVIOUR
DEFINED AS
"This package gives an indication if it is allowed or not to perform tests on the PNO VP
Connection."
;;
ATTRIBUTES
vpTestState GET-REPLACE;
REGISTERED AS {xatmPackage 2};
```

8.7 Attribute Definitions

8.7.1 Associated SubNetwork Pair Id

```
associatedSubNetworkPairId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   ASN1XatmModule.NameType;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
    associatedSubnetworkPairIdBehaviour BEHAVIOUR
DEFINED AS "This attribute contains a pointer to the SubNetworkPair to whom the PNO NW ATM Access
Point belongs to.";;
REGISTERED AS {xatmAttribute 1};
```

8.7.2 Backward QoS Class

```
backwardQoSClass ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   ASN1XatmModule.VpQoSClass;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
    backwardQoSClassBehaviour BEHAVIOUR
DEFINED AS "This attribute contains an indication of the QoS class in the backward direction of the
VPC/VCC. The class corresponds to specified values of delay, error rate and protection level
values";;
REGISTERED AS {xatmAttribute 2};
```

8.7.3 Change Reservation Information

```
changeReservationInformation ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   ASN1XatmModule.ChangeReservationInfo;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
    changeReservationInformationBehaviour BEHAVIOUR
DEFINED AS "This attribute contains the new parameters for modification of the PNO VP/VC Subnetwork
Connection or the pnoVpLinkConnection. These parameters are kept while the modification is not
activated by the Initiating PNO using the activateChange action.
Before the change is actually performed by this action, the new parameters are stored by means of
REPLACE operation on this attribute.
The cancel of the modification is done through the operation REPLACE-BY-DEFAULT on this
attribute.";;
REGISTERED AS {xatmAttribute 3};
```

8.7.4 continuityMonitorId

Defined in ITU-T Recommendation I.751 [3]

8.7.5 Forward QoS Class

```
forwardQoSClass ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   ASN1XatmModule.VpQoSClass;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
    forwardQoSClassBehaviour BEHAVIOUR
DEFINED AS "This attribute contains an indication of the QoS class in the forward direction of the
VPC/VCC. The class corresponds to specified values of delay, error rate and protection level
values";;
REGISTERED AS {xatmAttribute 4};
```

8.7.6 Initiating Pno Subnetwork Id

```
initiatingPnoSubnetworkId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   ASN1XatmModule.InitiatingPnoSubnetworkId;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
    initiatingPnoSubnetworkIdBehaviour BEHAVIOUR
DEFINED AS "This attribute identifies the Initiating PNO Subnetwork, and together with the
initiatingVpConnectionId (VcConnectionId), identifies the end-to-end VP Connection (VC Connection).
The use of pString for this attribute is preferred"
;;
REGISTERED AS {xatmAttribute 5};
```

8.7.7 Initiating Vc Connection Id

```
initiatingVcConnectionId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   ASN1XatmModule.InitiatingVcConnectionId;
    MATCHES FOR EQUALITY, SUBSTRINGS;
    BEHAVIOUR
    initiatingVcConnectionIdBehaviour BEHAVIOUR
DEFINED AS "This attribute contains the identifier that the Initiating PNO Subnetwork assigns to the
VC Connection, and together with the initiatingPnoSubnetworkId, identifies the end-to-end VC
Connection."
;;
REGISTERED AS {xatmAttribute 11};
```

8.7.8 Initiating Vp Connection Id

```
initiatingVpConnectionId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   ASN1XatmModule.InitiatingVpConnectionId;
    MATCHES FOR EQUALITY, SUBSTRINGS;
    BEHAVIOUR
    initiatingVpConnectionIdBehaviour BEHAVIOUR
DEFINED AS "This attribute contains the identifier that the Initiating PNO Subnetwork assigns to the
VP Connection, and together with the initiatingPnoSubnetworkId, identifies the end-to-end VP
Connection."
;;
REGISTERED AS {xatmAttribute 6};
```

8.7.9 kindOfResilience

```
kindOfResilience ATTRIBUTE
WITH ATTRIBUTE SYNTAX  ASN1XatmModule.ResilienceKind;
BEHAVIOUR
kindOfResilienceBehaviour BEHAVIOUR
DEFINED AS
"This attribute indicates the kind(s) of resilience-mechanism(s) that are used on the working
connection.
The meaning of the different values is as follows:
    . protSwitch1plus1,     --Protection Switching 1+1 (T)
    . protSwitch1to1,       --Protection Switching 1:1 (T)
    . recovPreAssRes,   --Recovery with Pre-Assigned Resources (T)
    . fastReRouting"        --Fast Re-routing  (T)
;;
REGISTERED AS {xatmAttribute 12};
```

8.7.10 List of ATM Access Point Pair Resources

```
listOfAtmAccessPointPairResources ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   ASN1XatmModule.ListOfAtmAccessPointPairResources;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
    listOfAtmAccessPointPairResourcesBehaviour BEHAVIOUR
DEFINED AS "This attribute contains the list of the NW ATM Access Points contained within the Inter
PNO Topological Subnetwork Pair, with their associated Cell Rate and QoS Class.
The aPnoAtmAccessPointId contains the Id of the access point that belongs to the network in
aEndPoint. The same applies to zPnoAtmAccessPointId and zEndPoint.
The maxAtoZ bandwith refers to the direction from the aPnoAtmAccessPoint to the zPnoAtmAccessPoint.
The maxZtoA bandwith refers to the other direction";;
REGISTERED AS {xatmAttribute 7};
```

8.7.11 Max Num VCI Bits Supported

It is defined in the ITU-T Recommendation I.751 [3].

8.7.12 Max Num VPI Bits Supported

It is defined in the ITU-T Recommendation I.751 [3].

8.7.13 Operational State

It is defined in the ITU-T Recommendation X.721 [10].

8.7.14 PNO NW Access Point Id

```
pnoNWAccessPointId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   ASN1XatmModule.NameType;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
    pnoNWAccessPointIdBehaviour BEHAVIOUR
DEFINED AS "This attribute is used for naming pnoNWAtmAccessPoint objects.";;
REGISTERED AS {xatmAttribute 8};
```

8.7.15 relatedSNC

```
relatedSNC ATTRIBUTE
WITH ATTRIBUTE SYNTAX  ASN1XatmModule.InitiatingVpConnectionId;
MATCHES FOR EQUALITY, SUBSTRINGS;
BEHAVIOUR
relatedSncBehaviour BEHAVIOUR
DEFINED AS
"This attribute points at an other Subnetworkconnection, which is either a protection connection of
the fast re-routing kind, or a working connection."
;;
REGISTERED AS {xatmAttribute 13};
```

8.7.16 sinkCCMecanismActive

Defined in ITU-T Recommendation I.751 [3].

8.7.17 sourceCCMechanismActive

Defined in ITU-T Recommendation I.751 [3]

8.7.18 Trail Termination Point Id

This package is defined in ITU-T Recommendation M.3100 [1].

8.7.19 VCCTP Id

It is defined in the ITU-T Recommendation I.751 [3].

8.7.20 VC Schedulers

```
vcSchedulers ATTRIBUTE
WITH ATTRIBUTE SYNTAX   ASN1XatmModule.VcSchedulers;
MATCHES FOR EQUALITY;
BEHAVIOUR;
vcSchedulersBehaviour BEHAVIOUR.
DEFINED AS "This attribute includes a scheduler specifying a number of time intervals on 24 hour
time-of-day clock, pertaining to selected days of the week for each month. The traffic descriptor is
associated with this scheduler allowing the change of the bandwith on a schedule basis.";;
REGISTERED AS {xatmAttribute 14};
```

8.7.21 vcTestState

```
vcTestState ATTRIBUTE
WITH ATTRIBUTE SYNTAX   ASN1XatmModule.VcTestState;
MATCHES FOR EQUALITY;
BEHAVIOUR
vcTestStateBehaviour BEHAVIOUR
DEFINED AS "This attribute gives an indication if it is allowed or not to perform tests on the PNO
VC Connection."
;;
REGISTERED AS {xatmAttribute 15};
```

8.7.22 VPCTP Id

It is defined in the ITU-T Recommendation I.751 [3].

8.7.23 VP Schedulers

```
vpSchedulers ATTRIBUTE
    WITH ATTRIBUTE SYNTAX   ASN1XatmModule.VpSchedulers;
    MATCHES FOR EQUALITY;
    BEHAVIOUR
    vpSchedulersBehaviour BEHAVIOUR
DEFINED AS "This attribute includes a scheduler specifying a number of time intervals on 24 hour
time-of-day clock, pertaining to selected days of the week for each month. The traffic descriptor is
associated with this scheduler allowing the change of the bandwith on a schedule basis.";;
REGISTERED AS {xatmAttribute 9};
```

8.7.24 vpTestState

```
vpTestState ATTRIBUTE
WITH ATTRIBUTE SYNTAX   ASN1XatmModule.VpTestState;
MATCHES FOR EQUALITY;
BEHAVIOUR
vpTestStateBehaviour BEHAVIOUR
DEFINED AS "This attribute gives an indication if it is allowed or not to perform tests on the PNO
VP Connection."
;;
REGISTERED AS {xatmAttribute 10};
```

8.8 Attribute Group Definitions

No attribute groups were identified.

8.9 Parameter Definitions

No parameters were identified.

8.10 Notification Definitions

8.10.1 Cancel VC Network Connection Notification

```
cancelVcNetworkConnectionNotification NOTIFICATION
BEHAVIOUR
cancelVcNetworkConnectionNotificationBehaviour BEHAVIOUR
DEFINED AS "This notification is issued by the Z PNO to ask the initiating PNO to release the VC
Network Connection."
;;
WITH INFORMATION SYNTAX ASN1XatmModule.CancelVcNetworkConnectionNotification;
REGISTERED AS {xatmNotification 2};
```

8.10.2 Cancel VP Link Connection Notification

```
cancelVpLinkConnectionNotification NOTIFICATION
BEHAVIOUR
cancelVpLinkConnectionNotificationBehaviour BEHAVIOUR
DEFINED AS "This notification is issued by a PNO to ask the initiating PNO to release the VP Link
Connection."
;;
WITH INFORMATION SYNTAX ASN1XatmModule.CancelVpNetworkConnectionNotification;
REGISTERED AS {xatmNotification 3};
```

8.10.3 Cancel VP Network Connection Notification

```
cancelVpNetworkConnectionNotification NOTIFICATION
BEHAVIOUR
cancelVpNetworkConnectionNotificationBehaviour BEHAVIOUR
DEFINED AS "This notification is issued by the Z PNO to ask the initiating PNO to release the VP
Network Connection."
;;
WITH INFORMATION SYNTAX ASN1XatmModule.CancelVpNetworkConnectionNotification;
REGISTERED AS {xatmNotification 1};
```

8.11 Action Definitions

8.11.1 Activate Change

```
activateChange ACTION
    BEHAVIOUR
activateChangeBehaviour BEHAVIOUR
DEFINED AS "This action is requested by the Initiating PNO when it wants to activate a change
reservation already confirmed by all the PNO Subnetworks in the path.
The activateChange action consists of changing the parameters of the current activation, in
accordance with the previous changeReservation procedure.
This action entails to replace the vpSchedulers attribute with the proper component of the
changeReservationInformation attribute. Then, the changeReservationInformation attribute will be
replaced with its default value. All these operations are internal to the concerned PNO, so these
changes of the attribute values should be notified by using the attributeValueChangeNotification."
;;
    MODE CONFIRMED;
    WITH REPLY SYNTAX   ASN1XatmModule.ActivateChangeResult;
REGISTERED AS{xatmAction 1};
```

8.11.2 Check User

```
checkUser ACTION
    BEHAVIOUR
checkUserBehaviour BEHAVIOUR
DEFINED AS "The objective of this action is to check if the Z User is willing to and is able to
accept the proposed VP or VC Connection.
The meaning of the different check user causes is as follows:
bandwidthNotAvailable: the destination user is not able to support the band width(s) required in the
scheduler.
userNotAvailable: it includes the following cases:
non-existent user;
user availability status with the value notAvailable;
user operational state with the value disabled;
user administrative state with the value locked.
All of this cases are internal to each PNO (not visible from the X-Interface).
```

```
userNotCompatible: it includes the following case:
with regard to a list of origin users which the user at the destination does not want to have a
connection with. "
;;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX     ASN1XatmModule.CheckUserInformation;
    WITH REPLY SYNTAX       ASN1XatmModule.CheckUserResult;
REGISTERED AS{xatmAction 2};
```

8.11.3 controlCC

Defined in ITU-T Recommendation I.751 [3].

8.11.4 Give Available Links

```
giveAvailableLinks ACTION
    BEHAVIOUR
giveAvailableLinksBehaviour BEHAVIOUR
DEFINED AS "The result of this Action is a list of PNOs that are adjacent to the PNO that receives
this request and have sufficient cell rate available on their links with the receiving PNO to
support the proposed VP- or VC Connection."
;;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX ASN1XatmModule.GiveAvailableLinksInformation;
    WITH REPLY SYNTAX       ASN1XatmModule.GiveAvailableLinksResult;
REGISTERED AS{xatmAction 3};
```

8.11.5 Release PNO VC Subnetwork Connection

```
releasePnoVcSubnetworkConnection ACTION
    BEHAVIOUR
releasePnoVcSubnetworkConnectionBehaviour BEHAVIOUR
DEFINED AS "This action is performed by the Initiating PNO requesting the clearing down of the VC
subnetwork connection. This will delete the pnoVcSubnetworkConnection object instance that makes up
the connection. The related pnoVCCTP object instance(s) that are not in use by other VC subnetwork
connections are deleted too."
;;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX ASN1XatmModule.ReleaseSubNetworkConnectionInformation;
    WITH REPLY SYNTAX           ASN1XatmModule.ReleaseSubNetworkConnectionResult;
REGISTERED AS{xatmAction 6};
```

8.11.6 Release PNO VP Link Connection

```
releasePnoVpLinkConnection ACTION
    BEHAVIOUR
releasePnoVpLinkConnectionBehaviour BEHAVIOUR
DEFINED AS "This action is performed by the Initiating PNO requesting the clearing down of the VP
link connection. This will delete the pnoVpLink Connection object instance that makes up the
connection. The related pnoVPCTP object instance that is not in use by other VP link connections are
deleted too."
;;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX     ASN1XatmModule.ReleaseSubNetworkConnectionInformation;
    WITH REPLY SYNTAX       ASN1XatmModule.ReleaseSubNetworkConnectionResult;
REGISTERED AS{xatmAction 7};
```

8.11.7 Release PNO VP Subnetwork Connection

```
releasePnoVpSubnetworkConnection ACTION
    BEHAVIOUR
releasePnoVpSubnetworkConnectionBehaviour BEHAVIOUR
DEFINED AS "This action is performed by the Initiating PNO requesting the clearing down of the VP
subnetwork connection. This will delete the pnoVpSubnetworkConnection object instance that makes up
the connection. The related pnoVPCTP object instance(s) that are not in use by other VP subnetwork
connections are deleted too."
;;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX ASN1XatmModule.ReleaseSubNetworkConnectionInformation;
    WITH REPLY SYNTAX           ASN1XatmModule.ReleaseSubNetworkConnectionResult;
REGISTERED AS{xatmAction 4};
```

8.11.8 Reserve PNO Enhanced VpSubnetworkConnection

reservePnoEnhancedVpSubnetworkConnection ACTION
BEHAVIOUR
reservePnoEnhancedVpSubnetworkConnection-Beh BEHAVIOUR
DEFINED AS
"This action requests the involved PNO to reserve a path across its subnetwork: between a specified input point and an adjacent subnetwork if performs the T subnetwork role, between the A User and an adjacent subnetwork if it performs A subnetwork role or between a specified input port and the Z user, if it performs the Z subnetwork role. It has the same behaviour as Action reservePnoVpSubnetworkConnection, but with the additional capability to request for performance monitoring and network resilience. The acceptance or rejection of a reservation is independent of the request for performance monitoring and resilience.
The response to a rejected reservation is identical to the response for Action reservePnoVpSubnetworkConnection.
The response to an accepted reservation has the result for performance monitoring and resilience added.
If performance monitoring is requested and the reservation is accepted, the successful response will also indicate if performance monitoring is accepted (pmOK (2)), or rejected (pmNotPossible (0)).
If performance monitoring is accepted one or two other objects (i.e. pnoLCBidirectionalPerformanceMonitor & pnoSNCBidirectionalPerformanceMonitor)are created, together with the creation of a pnoVpSubnetworkConnection object.
The request for resilience can be included as an option. It is possible to ask for one type of resilience, or for a combination of types.
If resilience is requested and the reservation is accepted, the successful response will also indicate if and what types of resilience are accepted:
Protection Switching Available: protection switching capability is available for this VP connection (T) or not available (F).
Recovery Resources Assigned: whether (T) or not (F) the resources for recovery have been assigned.
fastReroutingResult: a fast re-routing is possible(T) for this VP connection, or not possible (F).
If a working Subnetwork connection has been assigned a fast re-routing protection Subnetwork connection, its attribute relatedSNC should refer to this protection connection and vice versa.
In future this action should be used instead of the reservePnoVpSubnetworkConnection action if the initiating and A / T / Z-PNO have both implemented it. The 'old' action reservePnoVpSubnetworkConnection is only implemented for backward compatibility reasons. In order to guarantee a unique name for a pnoVpSubnetworkConnection object created as a result of the Reserve Request, the value for its naming Attribute (SubnetworkConnectionId attribute inherited from the SubnetworkConnection MOC) shall be set to the concatenation of the values of the initiatingPnoId and initiatingVpConnectionId in the ASN.1 syntax of the reservePnoEnhancedVpSubnetworkConnection Action received from the Manager.";;
MODE CONFIRMED;
WITH INFORMATION SYNTAX ASN1XatmModule.ReservePnoEnhancedVp;
WITH REPLY SYNTAX ASN1XatmModule.ReservePnoEnhancedVpResult;
REGISTERED AS {xatmAction 8};

8.11.9 Reserve Pno Vc Subnetwork Connection

reservePnoVcSubnetworkConnection ACTION
BEHAVIOUR
reservePnoVcSubnetworkConnectionBehaviour BEHAVIOUR
DEFINED AS "This action requests the involved PNO to reserve a virtual channel across its subnetwork:between a specified input point and an adjacent subnetwork if performs the T subnetwork role, between the A User and an adjacent subnetwork if it performs A subnetwork role or between a specified input port and the Z user, if it performs the Z subnetwork role.
The result of this action is the acceptance or reject of the connection reservation request (regarding the start time, the stop time and eventually the periodicity requested).
If the connection reservation is rejected, the reason is returned. The meanings of the different causes for rejection are as follows:
nearEndQosNotAvailable(1): With the *nearEndPnoSubnetworkId* the Agent will find the corresponding *interPnoTopologicalSubnetworkPair* Object Instance. Within this instance, the Agent will look for the element containing the *nearEndApId* in its *listOfAtmAccessPointPairResources* attribute.
Once the Agent has found the proper element in the list, it shall check whether the forwardQoSClass and BackwardQoSClass fields in the *reserveVcInformation* can be provided according to the *atmPathQoS* field in the *listOfAtmAccessPointPairResources* attribute by the virtual path *nearEndVPCTPId*, terminating at the *nearEndAPId*.
scheduleNotAvailable(2): the agent can NOT provide the *vcConnection* with the specified *vcScheduler*.
As described in the previous cause, once the Agent has found the proper element in the *listOfAtmAccessPointPairResources* attribute, it shall check the following:
At the nearEnd side (CASE 1):
Whether the requested AtoZTrafficDescriptor of each slot of the VcScheduler fits in the current available incoming bandwidth of virtual path *nearEndVPCTPId* terminating at *nearEndAPId*. Note that the current available incoming bandwidth is internal to the agent (not visible from the X Interface) and shall be obtained by means of the max. incoming bandwidth and the previous reservations performed on this nearEndAPId.

Whether the requested ZtoATrafficDescriptor of each slot of the VcScheduler fits in the current available outgoing bandwidth of virtual path *nearEndVPCTPId* terminating at *nearEndAPId*. Note that the current available outgoing bandwidth is internal to the agent (not visible from the XInterface) and shall be obtained by means of the max. outgoing bandwidth and the previous reservations performed on this nearEndAPId.
If the virtual path *nearEndVPCTPId*, terminating at *nearEndAPId* does not fulfil the two above conditions, the Reservation Action Response will contain an unsuccessful response with the scheduleNotAvailable reserve cause.
At the far End side (case 2: TRANSIT and A), the conditions to be checked are:
The requested AtoZTrafficDescriptor of each slot of the VcScheduler fits in the current available outgoing bandwidth of some virtual path terminating at the *farEndAPId* which is being checked. Note that the current available outgoing bandwidth is internal to the agent (not visible from the X Interface) and shall be obtained by means of the max. outgoing bandwidth and the previous reservations performed on this farEndAPId which is being checked.
The requested ZtoATrafficDescriptor of each slot of the VcScheduler fits in the current available incoming bandwidth of some virtual path terminating at the FarEndAPId which is being checked. Note that the current available incoming bandwidth is internal to the agent (not visible from the Xcoop interface) and shall be obtained by means of the max. incoming bandwidth and the previous reservations performed on this farEndAPId which is being checked.
If there is no element (link) in the ListOfAtmAccessPointPairResources attribute where a virtual path exists or internally can be created to fulfil the two above conditions, the Reservation Action Response will contain an unsuccessful response with the ScheduleNotAvailable reserve cause.
At the far End side (case 3: DESTINATION Z net), the conditions to be checked are:
The requested AtoZTrafficDescriptor of each slot of the VcScheduler fits in the current available outgoing bandwidth of some virtual path terminating at the UNI resource. Note that the current available outgoing bandwidth on this resource is internal to the agent (not visible from the Xcoop interface) and can be obtained by means of the max. outgoing bandwidth on this UNI resource on the Q3 level and the previous reservations performed on this UNI associated to the destinationAddress.
The requested ZtoATrafficDescriptor of each slot of the VcScheduler fits in the current available incoming bandwidth of some virtual path terminating at the UNI. Note that the current available incoming bandwidth is internal to the agent (not visible from the X Interface) and can be obtained by means of the max. incoming bandwidth on this UNI resource on the Q3 level and the previous reservations performed on this UNI associated to the Destination Address.
If at the UNI, associated to the Destination Address, no virtual path exists or can internally be created to fulfil the two above conditions, the Reservation Action Response will contain a negative response with the ScheduleNotAvailable reserve cause.
. **zVpiBusy(4):** the zVPi of the Vp that terminates at nearEndAPId and which is used to carry the Vc (see ReserveInformation ASN.1 definition) is already used during the specified slot times.
. **nearEndVpiOutOfRange(5):** This happens when the maximum range of subnetwork connections that can be used to on the pnoNWAccessPoint Object Instance (designed by nearEndAPId) have previously been allocated.
. **zVpiOutOfRange (6):** This happens when the maximum range of subnetwork connections on the pnoNWAccessPoint Object Instance at USER side have previously been allocated.
. **nearEndSNUnknown (7):** the nearEndPnoSubnetworkId (see ReserveInformation ASN.1 definition) is unknown.
. **farEndSNUnknown (8):** the farEndPnoSubnetworkId (see ReserveInformation ASN.1 definition) is unknown.
. **userNotAvailable (9):** it includes the following cases:
none-existent user,
user availability status with the value *notAvailable*,
user operational state with the value *disabled*,
user administrative state with the value *locked*.
. **userNotCompatible(10):** it includes the following case:
with regard to a list of Origin users which the destination user does not want to have a connection with.
. **nearEndAPisUnknown(11):** the nearEndAPId (see ReserveInformation ASN.1 definition) is unknown.
. **modeNotAvailable(12):** the mode (see ReserveInformation ASN.1 definition) is not pointToPoint(0).
. **initiatingPnoSNUnknown(13):** the PNO indicated by initiatingPnoSubnetworkId (see ReserveInformation ASN.1 definition) is unknown.
. **farEndQosNotAvailable(14):**
At the far End side (case 1: TRANSIT and A), the conditions to be checked are:
Whether the forwardQoSClass and BackwardQoSClass fields in the ReserveInformation can be provided by some virtual path terminating at the pnoNWAtmAccess Point Object Instance identified in the element being checked according to the atmPathQoS field in the ListOfAtmAccessPointPairResources attribute.
If there are no elements in the ListOfAtmAccessPointPairResources attribute that fulfil the above condition, then the Agent will respond with an unsuccessful response with the farEndQoSNotAvailable reserve cause.
At the far End side (case 2: DESTINATION Z net), the conditions to be checked are:
Whether the forwardQoSClass and BackwardQoSClass fields in the ReserveInformation can be provided by some virtual path terminating at the UNI resource according to its Quality of Service at the Q3 network level.
. **refused(15):** Used when an agent PNO refuses the reservation for whatever reason
. **nearEndVciBusy(16):** the nearEndVCCTPId (see ReserveInformation ASN.1 definition) is already used by an another vcConnection during the specified slot times.
. **zVciBusy(17):** the zVCi (see ReserveInformation ASN.1 definition) is already used during the specified slot times.

. **nearEndVciOutOfRange(18):** This happens when the maximum range of Vc subnetwork connections on the pnoNWAccessPoint Object Instance (designed by nearEndAPId) have previously been allocated.
. **zVciOutOfRange(19):** This happens when the maximum range of subnetwork connections on the pnoNWAccessPoint Object Instance at
USER side have previously been allocated.
In case of a Z PNO it is possible to define the zVPi and zVCi within the reserveInformation. If this requested zVPi or
zVCi cannot be provided, the reserveCause zVpiBusy or zVciBusy is also to be responded.
If the connection reservation is accepted, a pnoVcSubnetworkConnection object is created."
;;
MODE CONFIRMED;
WITH INFORMATION SYNTAX ASN1XatmModule.ReserveVcInformation;
WITH REPLY SYNTAX ASN1XatmModule.ReserveVcResult.
REGISTERED AS {xatmAction 9}.

8.11.10 Reserve PNO VP Link Connection

reservePnoVpLinkConnection ACTION
BEHAVIOUR
reservePnoVpLinkConnectionBehaviour BEHAVIOUR
DEFINED AS "This action reqests a PNO to make a VP Link Connection reservation across an 'Inter-Pno link' with an adjacent PNO.
It is requested by the Initiating PNO. The result of this action is the acceptance or reject of the connection reservation request (regarding the start time, the stop time and eventually the periodicity requested).
If the *adjacentPnoSubnetworkId* (see ASN.1 definition) is specified in the request, the Agent will look if there is a corresponding *interPnoTopologicalSubnetworkPair* Object Instance and if this instance contains a suitable link. As seen from the Agent here the direction 'aToZ' in the request is in the direction of the adjacent subnetwork.
If the SEQUENCE *'nearEndAPId, nearEndVPCTPId, nearEndPnoSubnetworkId'* (see ASN.1 definition) is specified in the request, the link has already been chosen by the I PNO and the Agent will check if it can provide the connection. As seen from the Agent, here the direction '*aToZ*' in the request is in the direction of the Agents' own subnetwork.
If the connection reservation is rejected, the reason is returned. The meanings of the different causes for rejection are as follows:
linkQosNotAvailable(1): The requested *aToZQoSC*lass and *zToAQoSClass* in the reserveLinkInformation can NOT be provided by the Link (*atmPathQoS* field in the *listOfAtmAccessPointPairResources* attribute).
linkScheduleNotAvailable(2): the agent can NOT provide the requested *vpLinkConnection* with the specified *vpScheduler*.
The Agent shall check the following:
(If the SEQUENCE 'nearEndAPId, nearEndVPCTPId, nearEndPnoSubnetworkId' is specified in the request):
('incoming' and 'outgoing' are with regard to the Agents' subnetwork)
Whether the requested AtoZTrafficDescriptor of each slot of the VpScheduler fits in the current available incoming bandwidth. Note that the current available incoming bandwidth is internal to the agent (not visible from the X Interface) and shall be obtained by means of the max. incoming bandwidth and the previous reservations performed on the nearEndAPId.
Whether the requested ZtoATrafficDescriptor of each slot of the VpScheduler fits in the current available outgoing bandwidth. Note that the current available outgoing bandwidth is internal to the agent (not visible from the XInterface) and shall be obtained by means of the max. outgoing bandwidth and the previous reservations performed on this nearEndAPId.
If the nearEndAPId does not fulfil the two above conditions, the Reservation Action Response will contain an unsuccessful response with the scheduleNotAvailable reserve cause.
(If *adjacentPnoSubnetworkId* is specified in the request):
('incoming' and 'outgoing' are with regard to the Agents' subnetwork)
The requested AtoZTrafficDescriptor of each slot of the VpScheduler fits in the current available outgoing bandwidth for the link which is being checked. Note that the current available outgoing bandwidth is internal to the agent (not visible from the X Interface) and shall be obtained by means of the max. outgoing bandwidth and the previous reservations performed on this link which is being checked.
The requested ZtoATrafficDescriptor of each slot of the VpScheduler fits in the current available incoming bandwidth for the link which is being checked. Note that the current available incoming bandwidth is internal to the agent (not visible from the Xcoop interface) and shall be obtained by means of the max. incoming bandwidth and the previous reservations performed on this farrEndAPId which is being checked.
If there is no element (link) in the ListOfAtmAccessPointPairResources attribute to fulfil the two above conditions, the Reservation Action Response will contain an unsuccessful response with the ScheduleNotAvailable reserve cause.
. **nearEndVpiBusy(3):** requested nearEndVPCTPId in the reserveLinkInformation (see ASN.1 definition) is already used by an another vpConnection during the specified slot times.
. **nearEndVpiOutOfRange(4):** requested nearEndVPCTPId in the reserveLinkInformation (see ASN.1 definition) is outside of the VPI-range as determined by the involved pnoNWAccessPoint Object Instance.
. **nearEndSNUnknown(5):** nearEndPnoSubnetworkId (see ReserveLinkInformation ASN.1 definition) is unknown.
. **nearEndAPisUnknown(6):** nearEndAPId (see ReserveLinkInformation ASN.1 definition) is unknown.
. **modeNotAvailable(7):** mode (see ReserveLinkInformation ASN.1 definition) is not pointToPoint(0).

. **initiatingPnoSNUnknown(8):** PNO indicated by initiatingPnoSubnetworkId (see ReserveLinkInformation ASN.1 definition) is unknown.
. **refused(9):** used when an agent PNO refuses the reservation for whatever reason
If the connection reservation is accepted, a pnoVpLinkConnection object is created."
;;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX ASN1XatmModule.ReserveVpLinkInformation;
    WITH REPLY SYNTAX ASN1XatmModule.ReserveVpLinkResult;
REGISTERED AS {xatmAction 10};

8.11.11 Reserve PNO VP Subnetwork Connection

reservePnoVpSubnetworkConnection ACTION
BEHAVIOUR
reservePnoVpSubnetworkConnectionBehaviour BEHAVIOUR
DEFINED AS "This action requests the involved PNO to reserve a path across its subnetwork: between a specified input point and an adjacent subnetwork if performs the T subnetwork role, between the A User and an adjacent subnetwork if it performs A subnetwork role or between a specified input port and the Z user, if it performs the Z subnetwork role. The result of this action is the acceptance or reject of the connection reservation request (regarding the start time, the stop time and eventually the periodicity requested).
If the connection reservation is rejected, the reason is returned. The meanings of the different causes for rejection are as follows:
. nearEndQosNotAvailable(1): With the *nearEndPnoSubnetworkId* the Agent will find the corresponding *interPnoTopologicalSubnetworkPair* Object Instance. Within this instance, the Agent will look for the element containing the nearEndApId in its *listOfAtmAccessPointPairResources* attribute.
Once the Agent has found the proper element in the list, it shall check whether the forwardQoSClass and BackwardQoSClass fields in the *reserveInformation* can be provided by the *nearEndAPId* according to the *atmPathQoS* field in the *listOfAtmAccessPointPairResources* attribute.
. scheduleNotAvailable(2): the agent can NOT provide the vpConnection with the specified vpScheduler.
As described in the previous cause, once the Agent has found the proper element in the *listOfAtmAccessPointPairResources* attribute, it shall check the following:
At the nearEnd side (CASE 1):
Whether the requested AtoZTrafficDescriptor of each slot of the VpScheduler fits in the current available incoming bandwidth. Note that the current available incoming bandwidth is internal to the agent (not visible from the X Interface) and shall be obtained by means of the max. incoming bandwidth and the previous reservations performed on this nearEndAPId.
Whether the requested ZtoATrafficDescriptor of each slot of the VpScheduler fits in the current available outgoing bandwidth. Note that the current available outgoing bandwidth is internal to the agent (not visible from the XInterface) and shall be obtained by means of the max. outgoing bandwidth and the previous reservations performed on this nearEndAPId.
If the nearEndAPId does not fulfil the two above conditions, the Reservation Action Response will contain an unsuccessful response with the scheduleNotAvailable reserve cause.
At the far End side (case 2: TRANSIT and A), the conditions to be checked are:
The requested AtoZTrafficDescriptor of each slot of the VpScheduler fits in the current available outgoing bandwidth on the FarEndAPId which is being checked. Note that the current available outgoing bandwidth is internal to the agent (not visible from the X Interface) and shall be obtained by means of the max. outgoing bandwidth and the previous reservations performed on this farEndAPId which is being checked.
The requested ZtoATrafficDescriptor of each slot of the VpScheduler fits in the current available incoming bandwidth on the FarEndAPId which is being checked. Note that the current available incoming bandwidth is internal to the agent (not visible from the Xcoop interface) and shall be obtained by means of the max. incoming bandwidth and the previous reservations performed on this farrEndAPId which is being checked.
If there is no element (link) in the ListOfAtmAccessPointPairResources attribute to fulfil the two above conditions, the Reservation Action Response will contain an unsuccessful response with the ScheduleNotAvailable reserve cause.
At the far End side (case 3: DESTINATION Z net), the conditions to be checked are:
The requested AtoZTrafficDescriptor of each slot of the VpScheduler fits in the current available outgoing bandwidth on the UNI resource. Note that the current available outgoing bandwidth on this resource is internal to the agent (not visible from the Xcoop interface) and can be obtained by means of the max. outgoing bandwidth on this UNI resource on the Q3 level and the previous reservations performed on this UNI associated to the destinationAddress.
The requested ZtoATrafficDescriptor of each slot of the VpScheduler fits in the current available incoming bandwidth on the UNI. Note that the current available incoming bandwidth is internal to the agent (not visible from the X Interface) and can be obtained by means of the max. incoming bandwidth on this UNI resource on the Q3 level and the previous reservations performed on this UNI associated to the Destination Address.
If the UNI associated to the Destination Address does not fulfil the two above conditions, the Reservation Action Response will contain a negative response with the ScheduleNotAvailable reserve cause.
. **nearEndVpiBusy(3):** the nearEndVPCTPId (see ReserveInformation ASN.1 definition) is already used by an another vpConnection during the specified slot times.
. **zVpiBusy(4):** the zVPi (see ReserveInformation ASN.1 definition) is already used during the specified slot times.
. **nearEndVpiOutOfRange(5):** This happens when the maximum range of subnetwork connections on the pnoNWAccessPoint Object Instance (designed by nearEndAPId) have previously been allocated.

```
. zVpiOutOfRange(6): This happens when the maximum range of subnetwork connections on the
pnoNWAccessPoint Object Instance at
 USER side have previously been allocated.
. nearEndSNUnknown(7): the nearEndPnoSubnetworkId (see ReserveInformation ASN.1 definition) is
unknown.
. farEndSNUnknown(8): the farEndPnoSubnetworkId (see ReserveInformation ASN.1 definition) is
unknown.
. userNotAvailable(9): it includes the following cases:
none-existant user,
user availability status with the value notAvailable,
user operational state with the value disabled,
user administrative state with the value locked.
. userNotCompatible(10): it includes the following case:
with regard to a list of Origin users which the destination user does not want to have any
connection with.
. nearEndAPisUnknown(11): the nearEndAPId (see ReserveInformation ASN.1 definition) is unknown.
. modeNotAvailable(12): the mode (see ReserveInformation ASN.1 definition) is not pointToPoint(0).
. initiatingPnoSNUnknown(13): the PNO indicated by initiatingPnoSubnetworkId (see ReserveInformation
ASN.1 definition) is unknown.
. farEndQosNotAvailable(14):
At the far End side (case 1: TRANSIT and A), the conditions to be checked are:
Whether the forwardQoSClass and BackwardQoSClass fields in the ReserveInformation can be provided by
the pnoNWAtmAccess Point Object Instance identified in the element being checked according to the
atmPathQoS field in the ListOfAtmAccessPointPairResources attribute.
If there are no elements in the ListOfAtmAccessPointPairResources attribute that fulfil the above
condition, then the Agent will respond with an unsuccessful response with the farEndQoSNotAvailable
reserve cause.
At the far End side (case 2: DESTINATION Z net), the conditions to be checked are:
Whether the forwardQoSClass and BackwardQoSClass fields in the ReserveInformation can be provided by
the UNI resource according to its Quality of Service at the Q3 network level.
. refused(15): Used when an agent PNO refuses the reservation for whatever reason
In case of a Z PNO it is possible to define the zVPi within the reserveInformation. If this
requested zVPi can not be provided, the reserveCause zVpiBusy is also to be responded.
If the connection reservation is accepted, a pnoVpSubnetworkConnection object is created."
;;
    MODE CONFIRMED;
    WITH INFORMATION SYNTAX ASN1XatmModule.ReserveInformation;
    WITH REPLY SYNTAX               ASN1XatmModule.ReserveResult;
REGISTERED AS {xatmAction 5};
```

8.11.12 switchToSnc

```
switchToSnc ACTION
BEHAVIOUR
switchToSncBehaviour BEHAVIOUR
DEFINED AS
"This action requests the Agent involved to change over, at the user end point, to the traffic
arriving from an alternative route which is referred to in the attribute related SNC.
The responses are:
. switch_OK (0), when the change-over was successful.
. switch_NotOK (1), when the change over failed."
;;
MODE CONFIRMED;
WITH INFORMATION SYNTAX     ASN1XatmModule.SwitchToSncInformation;
WITH REPLY SYNTAX           ASN1XatmModule.SwitchToSncResult;
REGISTERED AS {xatmAction 11};
```

8.12 Name Binding Definitions

8.12.1 interPnoTopologicalSubnetworkPair-pnoVcSubnetwork

```
interPnoTopologicalSubnetworkPair-pnoVcSubnetwork NAME BINDING
    SUBORDINATE OBJECT CLASS interPnoTopologicalSubnetworkPair;
    NAMED BY
    SUPERIOR OBJECT CLASS pnoVcSubnetwork;
    WITH ATTRIBUTE "ES 200 653 [9] ": subNetworkPairId;
BEHAVIOUR
interPnoTopologicalSubnetworkPair-pnoVcSubnetworkBehaviour BEHAVIOUR
DEFINED AS
"The interPnoTopologicalSubnetworkPair object is not created or deleted by system management
protocol. An instance of the object is created when a bundle of physical links between one PNO and
another, comes into existence.
The object is deleted when the underlying resource ceases to exist."
;;
REGISTERED AS {xatmNameBinding 6};
```

8.12.2 interPnoTopologicalSubnetworkPair-pnoVpSubnetwork

```
interPnoTopologicalSubnetworkPair-pnoVpSubnetwork NAME BINDING
    SUBORDINATE OBJECT CLASS interPnoTopologicalSubnetworkPair;
    NAMED BY
    SUPERIOR OBJECT CLASS pnoVpSubnetwork;
    WITH ATTRIBUTE "ES 200 653 [9] ": subNetworkPairId;
BEHAVIOUR
interPnoTopologicalSubnetworkPair-pnoVpSubnetworkBehaviour BEHAVIOUR
DEFINED AS
"The interPnoTopologicalSubnetworkPair object is not created or deleted by system management
protocol. An instance of the object is created when a bundle of physical links between one PNO and
another, comes into existence.
The object is deleted when the underlying resource ceases to exist."
;;
REGISTERED AS {xatmNameBinding 1};
```

8.12.3 interPnoTopologicalSubnetworkPair-pnoVpSubnetworkR2

```
interPnoTopologicalSubnetworkPair-pnoVpSubnetworkR2 NAME BINDING
    SUBORDINATE OBJECT CLASS interPnoTopologicalSubnetworkPair;
    NAMED BY
    SUPERIOR OBJECT CLASS pnoVpSubnetworkR2;
    WITH ATTRIBUTE "ES 200 653 [9] ": subNetworkPairId;
BEHAVIOUR
interPnoTopologicalSubnetworkPair-pnoVpSubnetworkR2Behaviour BEHAVIOUR
DEFINED AS
" (This namebinding should be used instead of interPnoTopologicalSubnetworkPair-pnoVpSubnetwork
which is maintained for compatibility purposes). The interPnoTopologicalSubnetworkPair object is not
created or deleted by system management protocol. An instance of the object is created when a bundle
of physical links between one PNO and another, comes into existence.
The object is deleted when the underlying resource ceases to exist. "
;;
REGISTERED AS {xatmNameBinding 7};
```

8.12.4 pnoBidirectionalContinuityMonitor-pnoVCCTP

```
pnoBidirectionalContinuityMonitor-pnoVCCTP NAME BINDING
SUBORDINATE OBJECT CLASS pnoBidirectionalContinuityMonitor;
NAMED BY SUPERIOR OBJECT CLASS pnoVCCTP;
WITH ATTRIBUTE  "ITU-T Recommendation I.751 [3] (1996)":continuityMonitorId;
CREATE
WITH-REFERENCE-OBJECT, WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE
DELETES-CONTAINED-OBJECTS;
REGISTERED AS {xatmNameBinding 8};
```

8.12.5 pnoBidirectionalContinuityMonitor-pnoVPCTP

```
pnoBidirectionalContinuityMonitor-pnoVPCTP NAME BINDING
SUBORDINATE OBJECT CLASS pnoBidirectionalContinuityMonitor;
NAMED BY SUPERIOR OBJECT CLASS pnoVPCTP;
WITH ATTRIBUTE  "ITU-T Recommendation I.751 [3] (1996)":continuityMonitorId;
CREATE
WITH-REFERENCE-OBJECT, WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE
DELETES-CONTAINED-OBJECTS;
REGISTERED AS {xatmNameBinding 5};
```

8.12.6 pnoNWAtmAccessPoint-pnoVpSubnetwork

```
pnoNWAtmAccessPoint-pnoVpSubnetwork NAME BINDING
    SUBORDINATE OBJECT CLASS pnoNWAtmAccessPoint;
    NAMED BY
    SUPERIOR OBJECT CLASS pnoVpSubnetwork;
    WITH ATTRIBUTE pnoNWAccessPointId;
BEHAVIOUR
pnoNWAtmAccessPoint-pnoVpSubnetworkBehaviour BEHAVIOUR
DEFINED AS
"The pnoNWAtmAccessPoint object is not created or deleted by system management protocol. An instance
of the object is created when a single physical link, the end point of which this object represents,
comes into existence. This may be on the creation, or at any time during the existence of the
interPnoTopologicalSubnetworkPair object instance associated with a particular pnoNWAtmAccessPoint
object instance.
```

```
A pnoNWAtmAccessPoint object instance may be created representing the external connection to an End
User in a Z PNO, in which case there will be no associated subnetworkPairId, as such an object would
not be related to an inter pno physical link.
The object is deleted when the related physical link ceases to exist, or in the case of a Z PNO,
when the related connection to a User ceases to exist.
The creation and deletion of pnoNWAtmAccessPoint object instances are only notified to other PNOs,
if these instances are associated with interPnoTopologicalSubnetworkPair instance, via the
attributeValueChangeNotification, reporting on changes to the listOfAtmAccessPointPairResources
attribute.
No reports for the creation and deletion of pnoNWAtmAccessPoint object instances serving connections
to users in Z PNOs will be made."
;;
REGISTERED AS {xatmNameBinding 2};
```

8.12.7 PnoNWAtmAccessPointR2-pnoVcSubnetwork

```
pnoNWAtmAccessPointR2-pnoVcSubnetwork NAME BINDING
    SUBORDINATE OBJECT CLASS pnoNWAtmAccessPointR2;
    NAMED BY
    SUPERIOR OBJECT CLASS pnoVcSubnetwork;
    WITH ATTRIBUTE pnoNWAccessPointId;
BEHAVIOUR
poNWAtmAccessPointR2-pnoVcSubnetworkBehaviour BEHAVIOUR
DEFINED AS
"The pnoNWAtmAccessPointR2 object is not created or deleted by system management protocol. An
instance of the object is created when a single physical link, the end point of which this object
represents, comes into existence. This may be on the creation, or at any time during the existence
of the interPnoTopologicalSubnetworkPair object instance associated with a particular
pnoNWAtmAccessPointR2 object instance.
A pnoNWAtmAccessPointR2 object instance may be created representing the external connection to an
End User in a Z PNO, in which case there will be no associated subnetworkPairId, as such an object
would not be related to an inter pno physical link.
The object is deleted when the related physical link ceases to exist, or in the case of a Z PNO,
when the related connection to a User ceases to exist.
The creation and deletion of pnoNWAtmAccessPointR2 object instances are only notified to other PNOs,
if these instances are associated with interPnoTopologicalSubnetworkPair instance, via the
attributeValueChangeNotification, reporting on changes to the listOfAtmAccessPointPairResources
attribute.
No reports for the creation and deletion of pnoNWAtmAccessPointR2 object instances serving
connections to users in Z PNOs will be made."
;;
REGISTERED AS {xatmNameBinding 9};
```

8.12.8 pnoNWAtmAccessPointR2-pnoVpSubnetworkR2

```
pnoNWAtmAccessPointR2-pnoVpSubnetworkR2 NAME BINDING
    SUBORDINATE OBJECT CLASS pnoNWAtmAccessPointR2;
    NAMED BY
    SUPERIOR OBJECT CLASS pnoVpSubnetworkR2;
    WITH ATTRIBUTE pnoNWAccessPointId;
BEHAVIOUR
poNWAtmAccessPointR2-pnoVpSubnetworkR2Behaviour BEHAVIOUR
DEFINED AS
"(This namebinding should be used instead of pnoNWAtmAccessPoint-pnoVpSubnetwork,which is maintained
for compatibility reasons).
The pnoNWAtmAccessPointR2 object is not created or deleted by system management protocol. An
instance of the object is created when a single physical link, the end point of which this object
represents, comes into existence. This may be on the creation, or at any time during the existence
of the interPnoTopologicalSubnetworkPair object instance associated with a particular
pnoNWAtmAccessPointR2 object instance.
A pnoNWAtmAccessPointR2bject instance may be created representing the external connection to an End
User in a Z PNO, in which case there will be no associated subnetworkPairId, as such an object would
not be related to an inter pno physical link.
The object is deleted when the related physical link ceases to exist, or in the case of a Z PNO,
when the related connection to a User ceases to exist.
The creation and deletion of pnoNWAtmAccessPointR2 object instances are only notified to other PNOs,
if these instances are associated with interPnoTopologicalSubnetworkPair instance, via the
attributeValueChangeNotification, reporting on changes to the listOfAtmAccessPointPairResources
attribute.
No reports for the creation and deletion of pnoNWAtmAccessPointR2 object instances serving
connections to users in Z PNOs will be made."
;;
REGISTERED AS {xatmNameBinding 10};
```

8.12.9 pnoVCCTP-pnoVPTTP

```
pnoVCCTP-pnoVPTTP NAME BINDING
```

```
    SUBORDINATE OBJECT CLASS pnoVCCTP;
    NAMED BY
    SUPERIOR OBJECT CLASS pnoVPTTP;
    WITH ATTRIBUTE "ITU-T Recommendation I.751 [3] (1996)": vcCTPId;
BEHAVIOUR
pnoVCCTP-pnoVPTTP-Beh BEHAVIOUR
DEFINED AS
"A pnoVCCTP object instance is created by the reservePnoVcSubnetworkConnection action, when a VCI is
allocated on a virtual path going over a link between two PNOs. The object instance is deleted by
the action releasePnoVcSubnetworkConnection, when it is not used by other
pnoVcSubnetworkConnections."
;;
REGISTERED AS {xatmNameBinding 11};
```

8.12.10 pnoVcSubnetwork-system

Defined by Subnetwork–system in ES 200 653 [9]

8.12.11 pnoVcSubnetworkConnection-pnoVcSubnetwork

```
pnoVcSubnetworkConnection-pnoVcSubnetwork NAME BINDING
    SUBORDINATE OBJECT CLASS pnoVcSubnetworkConnection;
    NAMED BY
    SUPERIOR OBJECT CLASS pnoVcSubnetwork;
    WITH ATTRIBUTE "ES 200 653 [9] ":subNetworkConnectionId;
BEHAVIOUR
pnoVcSubnetworkConnection-pnoVcSubnetworkBehaviour BEHAVIOUR -- Ar PnoV. in pnoV.. geändert
DEFINED AS
"A pnoVcSubnetworkConnection object instance is created by the reservePnoVcSubnetworkConnection
action, when a VCI is allocated on a physical link between two PNOs. The object instance is deleted
by the action releasePnoVcSubnetworkConnection."
;;
REGISTERED AS {xatmNameBinding 12};
```

8.12.12 pnoVPCTP-pnoNWAtmAccessPoint

```
pnoVPCTP-pnoNWAtmAccessPoint NAME BINDING
    SUBORDINATE OBJECT CLASS pnoVPCTP;
    NAMED BY
    SUPERIOR OBJECT CLASS pnoNWAtmAccessPoint;
    WITH ATTRIBUTE "ITU-T Recommendation I.751 [3] (1996)": vpCTPId;
BEHAVIOUR
pnoVPCTP-pnoNWAtmAccessPointBehaviour BEHAVIOUR
DEFINED AS
"A pnoVPCTP object instance is created by the reservePnoVpSubnetworkConnection action, when a VPI is
allocated on a physical link between two PNOs. The object instance is deleted by the action
releasePnoVpSubnetworkConnection, when it is not used by other pnoVpSubnetworkConnections."
;;
REGISTERED AS {xatmNameBinding 3};
```

8.12.13 pnoVPCTP-pnoNWAtmAccessPointR2

```
pnoVPCTP-pnoNWAtmAccessPointR2 NAME BINDING
    SUBORDINATE OBJECT CLASS pnoVPCTP;
    NAMED BY
    SUPERIOR OBJECT CLASS pnoNWAtmAccessPointR2;
    WITH ATTRIBUTE "ITU-T Recommendation I.751 [3] (1996)": vpCTPId;
BEHAVIOUR
pnoVPCTP-pnoNWAtmAccessPointR2Behaviour BEHAVIOUR
DEFINED AS
"(This namebinding should be used instead of pnoVPCTP-pnoNWAtmAccessPoint, which is maintained for
compatibility reasons).
A pnoVPCTP object instance is created by the reservePnoVpSubnetworkConnection action, when a VPI is
allocated on a physical link between two PNOs. The object instance is deleted by the action
releasePnoVpSubnetworkConnection, when it is not used by other pnoVpSubnetworkConnections."
;;
REGISTERED AS {xatmNameBinding 13};
```

8.12.14 PnoVpLinkConnection - interPnoTopologicalSubnetworkPair

```
pnoVpLinkConnection-interPnoTopologicalSubnetworkPair NAME BINDING
    SUBORDINATE OBJECT CLASS pnoVpLinkConnection;
    NAMED BY
    SUPERIOR OBJECT CLASS interPnoTopologicalSubnetworkPair;
    WITH ATTRIBUTE "ITU-T Recommendation M.3100 [1] (1995)":connectionId;
```

```
BEHAVIOUR
pnoVpLinkConnection-interPnoTopologicalSubnetworkPair-Beh BEHAVIOUR
DEFINED AS
"A pnoVpLinkConnection object instance is created by the reservePnoVpLinkConnection action, when a
VPI is allocated on a physical link between two PNOs. The object instance is deleted by the action
releasePnoVpLinkConnection."
;;
REGISTERED AS {xatmNameBinding 14};
```

8.12.15 pnoVpSubnetworkConnection-pnoVpSubnetwork

```
pnoVpSubnetworkConnection-pnoVpSubnetwork NAME BINDING
    SUBORDINATE OBJECT CLASS pnoVpSubnetworkConnection;
    NAMED BY
    SUPERIOR OBJECT CLASS pnoVpSubnetwork;
    WITH ATTRIBUTE "ES 200 653 [9] ":subNetworkConnectionId;
BEHAVIOUR
pnoVpSubnetworkConnection-pnoVpSubnetworkBehaviour BEHAVIOUR
DEFINED AS
"A pnoVpSubnetworkConnection object instance is created by the reservePnoVpSubnetworkConnection
action, when a VPI is allocated on a physical link between two PNOs. The object instance is deleted
by the action releasePnoVpSubnetworkConnection."
;;
REGISTERED AS {xatmNameBinding 4};
```

8.12.16 pnoVpSubnetworkConnectionR2-pnoVpSubnetworkR2

```
pnoVpSubnetworkConnectionR2-pnoVpSubnetworkR2 NAME BINDING
    SUBORDINATE OBJECT CLASS pnoVpSubnetworkConnectionR2;
    NAMED BY
    SUPERIOR OBJECT CLASS pnoVpSubnetworkR2;
    WITH ATTRIBUTE "ES 200 653 [9] ":subNetworkConnectionId;
BEHAVIOUR
pnoVpSubnetworkConnectionR2-pnoVpSubnetworkR2Behaviour BEHAVIOUR
DEFINED AS
"(This namebinding should be used instead of pnoVpSubnetworkConnection-pnoVpSubnetwork which is
maintained for compatibility purposes). A pnoVpSubnetworkConnectionR2object instance is created by
the reservePnoEnhancedVpSubnetworkConnection action, when a VPI is allocated on a physical link
between two PNOs. The object instance is deleted by the action releasePnoVpSubnetworkConnection."
;;
REGISTERED AS {xatmNameBinding 15};
```

8.12.17 pnoVpSubnetworkR2-system

Defined by Subnetwork–system in ES 200 653 [9].

8.12.18 pnoVPTTP-pnoNWAtmAccessPointR2

```
pnoVPTTP-pnoNWAtmAccessPointR2 NAME BINDING
    SUBORDINATE OBJECT CLASS pnoVPTTP;
    NAMED BY
    SUPERIOR OBJECT CLASS pnoNWAtmAccessPointR2;
    WITH ATTRIBUTE "ITU-T Recommendation M.3100 [1] (1995)": tTPId;
BEHAVIOUR
pnoVPTTP-pnoNWAtmAccessPointR2Behaviour BEHAVIOUR
DEFINED AS
"A pnoVPTTP object instance is created by:
The reservePnoVpLinkConnection action, when the initiating PNO wants the VP link connection to serve
as a trail for User-to-user VCCs who are also to be reserved over the X-Interface.
The reservePnoVcSubnetworkConnection Action when an internal VpLinkconnection is created to carry
the VCC over the physical link between two subnetworks.
The object instance is deleted by the action releasePnoVcSubnetworkConnection or
releasePnoVpLinkConnection when there are no other VCs left that use the VP trail."
;;
REGISTERED AS {xatmNameBinding 16};
```

8.12.19 Subnetwork - system

This namebinding is defined in ES 200 653 [9].

8.13 ASN.1 Module

```
ASN1XatmModule {ccitt(0) identified-organization (4) etsi (0) en300820-1 (820) informationModel (0)
asn1Module (2) asn1TypesModule (0)}
DEFINITIONS IMPLICIT TAGS::=
BEGIN
IMPORTS
NameType
    FROM ASN1DefinedTypesModule {ccitt (0) recommendation (0) m (13) gnm (3100)
informationModel (0) asn1Modules (2) asn1DefinedTypeModule (0)}
    VpiValue,
    VciValue,
    PeakCellRate,
    SustainableCellRate,
    CDVTolerance,
    MaxBurstSize
    FROM AtmMIBMod {itu-t(0) recommendation(0) i(9) atmm(751) informationModel(0) asn1Module(2)
atm(0)}
    ReleaseSubNetworkConnectionResult,
    Mode
    FROM ES 200 653 [9] {ccitt (0) identified-organization (4) etsi (0) ets (653) informationModel
(0) asn1Module (2) es200653 (0)}
    StopTime, Time24
    FROM Attribute-ASN1Module {joint-iso-ccitt ms(9) smi(3) part2(2) asn1Module(2) 1}
    ObjectInstance FROM CMIP-1 {joint-iso-ccitt  ms(9) cmip(1) modules(0) protocol(3)};
xatmInfoModel OBJECT IDENTIFIER::= {ccitt(0) identified-organization(4) etsi(0) en300820-1(820)
informationModel(0)}
xatmSpecificExtension OBJECT IDENTIFIER::= {xatmInfoModel standardSpecificExtension(0)}
xatmObjectClass OBJECT IDENTIFIER::= {xatmInfoModel managedObjectClass(3)}
xatmPackage OBJECT IDENTIFIER::= {xatmInfoModel package(4)}
xatmNameBinding OBJECT IDENTIFIER::= {xatmInfoModel nameBinding(6)}
xatmAttribute OBJECT IDENTIFIER::= {xatmInfoModel attribute(7)}
xatmAction OBJECT IDENTIFIER::= {xatmInfoModel action(9)}
xatmNotification OBJECT IDENTIFIER::= {xatmInfoModel notification(10)}
ActivateChangeResult::= Result
AtmPathQoS::= INTEGER (0..99)
BidirectionalTrafficDescriptor:: = SEQUENCE {
                    aToZ        TrafficDescriptor,
                    zToA        TrafficDescriptor}
CancelVcNetworkConnectionNotification::= SEQUENCE {
                    initiatingPnoSubnetworkId       InitiatingPnoSubnetworkId,
                    initiatingVcConnectionId        InitiatingVcConnectionId,
                    disconnectCause                 DisconnectCause}
CancelVpNetworkConnectionNotification::= SEQUENCE {
                    initiatingPnoSubnetworkId       InitiatingPnoSubnetworkId,
                    initiatingVpConnectionId        InitiatingVpConnectionId,
                    disconnectCause                 DisconnectCause}
ChangeReservationInfo::= VpSchedulers
CheckUserInformation::= SEQUENCE {
                    initiatingPnoSubnetworkId       InitiatingPnoSubnetworkId,
                    initiatingVpConnectionId        InitiatingVpConnectionId,
                    vpSchedulers                        VpSchedulers,
                    aAddress            E164Address,        -- address of the A side
                    zAddress            E164Address}    -- address of the Z side
CheckUserResult::=CHOICE {
                checkingOk          NULL,
                checkingNotok       UserCause}
DailySchedule::= SEQUENCE OF DaySlot
DaySlot::= SEQUENCE {
                slotBegin           Time24,
                slotEnd             Time24,
                bandwidth       BidirectionalTrafficDescriptor}
defaultChangeReservationInfo    ChangeReservationInfo::= {
            startTime           continual: NULL,
                stopTime        continual: NULL,
                scheduleMechanism   durationScheduling: {
                                aToZ    nothing: NULL,
                                zToA    nothing: NULL}}
Digit::= INTEGER (0..9)
DisconnectCause ::= ENUMERATED {normal(0), unrecoverableFailure(1)}
emptyString   InitiatingVpConnectionId::= pString:  "SPACE"
E164Address::= SEQUENCE SIZE (1.. 15) OF Digit
FastReroutingResult::= ENUMERATED {
                    fastReroutingReserved (0),
                    fastReroutingNotAvailable(1),
                    noRelatedSnc (2)}
GiveAvailableLinksInformation::= SEQUENCE {
```

```
                InitiatingPnoSubnetworkId   InitiatingPnoSubnetworkId,
                itiatingVpConnectionId      initiatingVpConnectionId,
                aPnoId          [0]         NameType OPTIONAL,
                vpSchedulers                    VpSchedulers,
                aToZQoSClass                VpQoSClass, -- A to Z means from the nearend to the far-
end
                zToAQoSClass                VpQoSClass} -- Z to A means from the far-end to the
nearend
GiveAvailableLinksResult::= CHOICE {
                null            NULL,
                listOfSubnetworks   SET OF  NameType}
InitiatingPnoSubnetworkId::=    NameType
InitiatingVcConnectionId    ::= NameType
InitiatingVpConnectionId    ::= NameType
ListOfAtmAccessPointPairResources::= SET OF SEQUENCE {
                    aPnoAtmAccessPointId            NameType,
                    zPnoAtmAccessPointId            NameType,
                    maxAtoZBandwith             INTEGER,
                    maxZtoABandwith             INTEGER,
                    atmPathQoS                  AtmPathQoS}
MonthDay::= INTEGER (1..31)
MonthlySchedule::= SEQUENCE OF MonthSlot
MonthSlot::= SEQUENCE {
                slotBegin       TimeMonth,
                slotEnd         TimeMonth,
                bandwidth       BidirectionalTrafficDescriptor}
OccasionalSchedule::= SEQUENCE OF OccasionalSlot
OccasionalSlot::= SEQUENCE {
                slotBegin           StartTime,
                slotEnd             StopTime,
                bandwidth       BidirectionalTrafficDescriptor}
PerformanceMonitoring::= ENUMERATED{pmNotPossible (0), pmNotRequested (1), pmOK (2)}
ReleaseSubNetworkConnectionInformation::= CHOICE {
                        snc ObjectInstance,
                        SEQUENCE {
                    initiatingPnoSubnetworkId   InitiatingPnoSubnetworkId,
                            initiatingVpConnectionId        InitiatingVpConnectionId}}
ReserveCause::= ENUMERATED {    insufficientCellRate(0),    nearEndQosNotAvailable(1),
                    scheduleNotAvailable(2),    nearEndVpiBusy(3), zVpiBusy(4),
                    nearEndVpiOutOfRange(5),    zVpiOutOfRange(6),
                    nearEndSNUnknown(7),        farEndSNUnknown(8),
                    userNotAvailable(9),        userNotCompatible(10),
                    nearEndAPisUnknown (11),    modeNotAvailable (12),
                    initiatingPnoSNUnknown (13), farEndQosNotAvailable(14),
                    refused (15)}
ReserveVcCause::= ENUMERATED {  insufficientCellRate(0),     nearEndQosNotAvailable(1),
                    scheduleNotAvailable(2),     zVpiBusy(4),
                    nearEndVpiOutOfRange(5),    zVpiOutOfRange(6),
                    nearEndSNUnknown(7),        farEndSNUnknown(8),
                userNotAvailable(9),        userNotCompatible(10),
                    nearEndAPisUnknown (11),    modeNotAvailable (12),
                    initiatingPnoSNUnknown (13), farEndQosNotAvailable(14),
                    refused (15),   nearEndVciBusy(16),
                    zVciBusy(17),   nearEndVciOutOfRange(18),
                    zVciOutOfRange(19)}
ResilienceResult::= SEQUENCE {
            protectSwitchAvailable      BOOLEAN         OPTIONAL,
            recoveryResourcesAssigned  [0]  BOOLEAN     OPTIONAL,
            fastReroutingResult  [1]    FastReroutingResult     OPTIONAL}
ReserveInformation::= SEQUENCE {
            initiatingPnoSubnetworkId       InitiatingPnoSubnetworkId,
            initiatingVpConnectionId            InitiatingVpConnectionId,
            aPnoId                  [1]  NameType OPTIONAL,
            CHOICE {
                aAddress                [0]  E164Address,
                SEQUENCE {
                    nearEndAPId         NameType,
                    nearEndVPCTPId      VpiValue,
                nearEndPnoSubnetworkId  NameType}},
        CHOICE {
                farendPnoSubnetworkId           NameType,
                SEQUENCE {
    aAddress            E164Address,
                zAddress            E164Address}},
            zVPi                    [0]  VpiValue OPTIONAL,
            aToZQoSClass                VpQoSClass,
            zToAQoSClass                VpQoSClass,
            configurationType           Mode,
```

```
            vpSchedulers                VpSchedulers}
ReserveVcInformation::= SEQUENCE {
            initiatingPnoSubnetworkId       InitiatingPnoSubnetworkId,
            initiatingVcConnectionId            InitiatingVcConnectionId,
            aPnoId              [1]     NameType OPTIONAL,
            CHOICE {
    aAddress            [0]     E164Address,
                SEQUENCE {
                    nearEndAPId         NameType,
                    nearEndVPCTPId      VpiValue,
                    nearEndVCCTPid      VciValue,
                    nearEndPnoSubnetworkId  NameType}},
            CHOICE {
                farendPnoSubnetworkId       NameType,
                SEQUENCE {
aAddress        E164Address,
                    zAddress        E164Address}},
            zVPi                    [0]  VpiValue OPTIONAL,
            aToZQoSClass                VcQoSClass,
            zToAQoSClass                VcQoSClass,
            configurationType           Mode,
            vcSchedulers                VpSchedulers,
            relatedSNC              InitiatingVcConnectionId OPTIONAL,
            resilienceKind              ResilienceKind OPTIONAL,
            performanceMonitorRequested [2]     BOOLEAN}
ReservePnoEnhancedVp::= SEQUENCE {
        initiatingPnoSubnetworkId       InitiatingPnoSubnetworkId,
            initiatingVpConnectionId            InitiatingVpConnectionId,
            aPnoId               [1]    NameType OPTIONAL,
            CHOICE {
                aAddress             [0]    E164Address,
            SEQUENCE {
                    nearEndAPId         NameType,
                    nearEndVPCTPId      VpiValue,
                    nearEndPnoSubnetworkId  NameType}},
            CHOICE {
                farendPnoSubnetworkId       NameType,
                SEQUENCE {  aAddress        E164Address,
                        zAddress        E164Address}},
            zVPi           [0]  VpiValue OPTIONAL,
            aToZQoSClass                VpQoSClass,
            zToAQoSClass                VpQoSClass,
            configurationType           Mode,
            vpSchedulers                VpSchedulers,
            relatedSNC              InitiatingVpConnectionId OPTIONAL,
            resilienceKind              ResilienceKind OPTIONAL,
            performanceMonitorRequested    [2]  BOOLEAN}
ReserveVpLinkCause::=ENUMERATED {
linkQosNotAvailable(1),
                linkScheduleNotAvailable(2),
                nearEndVpiBusy(3),
                nearEndVpiOutOfRange(4),
                nearEndSNUnknown(5),
                nearEndAPisUnknown (6),
                modeNotAvailable (7),
initiatingPnoSNUnknown (8),
                refused (9)}
ReserveVpLinkInformation::= SEQUENCE {
    initiatingPnoSubnetworkId   InitiatingPnoSubnetworkId,
    initiatingVpConnectionId        InitiatingVpConnectionId,
    aPnoId           [1]    NameType OPTIONAL,
    CHOICE {
    adjacentPnoSubnetworkId     NameType,
        SEQUENCE {
            nearEndAPId         NameType,
            nearEndVPCTPId      VpiValue,
            nearEndPnoSubnetworkId  NameType}},
aToZQoSClass            VpQoSClass,
    zToAQoSClass            VpQoSClass,
    configurationType       Mode,
    vpSchedulers            VpSchedulers}
ReservePnoEnhancedVpResult::= CHOICE {
        unsuccessfulResult  [0] ReserveCause,
        successfulResult    [1] SuccessfulPerformResilience}

ReserveResult::= CHOICE {
        unsuccessfulResult  [0] ReserveCause,
        successfulResult        [1] SuccessfulReserve}
```

```
ReserveVcResult::= CHOICE {
        unsuccessfulResult  [0] ReserveVcCause,
        successfulResult        [1] SuccessfulVcReserve}
ReserveVpLinkResult::= CHOICE {
        unsuccessfulResult  [0] ReserveVpLinkCause,
        sucessfulResult     [1] SuccessfulReserveVpLinkResult}
ResilienceKind::= SEQUENCE
        {
        protSwitch1plus1    BOOLEAN,    --Protection Switching 1+1 (T)
        protSwitch1to1      BOOLEAN,    --Protection Switching 1:1 (T)
        recovPreAssRes      BOOLEAN,    --Recovery with Pre-Assigned Resources (T)
        fastReRouting       BOOLEAN --Fast Re-routing  (T)
        }
Result::= ENUMERATED {successful(0), unsuccessful(1)}
StartTime::= StopTime
SubnetworkConnectionId::=NameType
SuccessfulPerformResilience::= SEQUENCE {
            performanceMonitoring   PerformanceMonitoring,
            resilienceResult     [1]    ResilienceResult   OPTIONAL,
                farEndInfo      CHOICE {
                farEndId SEQUENCE {
                            far-endVPCTPId      VpiValue,
                        far-endAPId     NameType,
                            far-endassociatedAPId   NameType},
                    zAddress    [0]     E164Address}}
SuccessfulReserve::= CHOICE {
            SEQUENCE {
                far-endVPCTPId      VpiValue,
                far-endAPId     NameType,
                far-endassociatedAPId   NameType},
            zAddress       [0]   E164Address}
SuccessfulReserveVpLinkResult::= SEQUENCE {
                    adjacentVPCTPId VpiValue,
                    adjacentAPId        NameType
                    }
SuccessfulVcReserve::= SEQUENCE {
            performanceMonitoring       PerformanceMonitoring,
                resilienceResult           [1]  ResilienceResult    OPTIONAL,
                farEndInfo   CHOICE {
                        farEndId    SEQUENCE {
                                far-endVPCTPId      VpiValue,
                                far-endVCCTPid      VciValue,
                                far-endAPId     NameType,
                                far-endassociatedAPId   NameType},
                        zAddress   [0]  E164Address}}
SwitchToSncInformation::= NULL
SwitchToSncResult::= ENUMERATED {switch-OK(0), switch-NotOK(1)}
TimeMonth::= SEQUENCE {
            monthDay     MonthDay,
            time        Time24}
TimeWeek::= SEQUENCE {
            weekDay  WeekDay,
            time        Time24}
- This Typedefinition of TrafficDescriptor does not align with
- "Descriptor" as defined in I.751 [3].
- In future versions of the X-interface standard ETSI shall
- consider migration to the Type that is used in I.751 [3].
TrafficDescriptor::= CHOICE {
        nothing NULL,
        trafficContract SEQUENCE {
                peakCellRate            PeakCellRate,
                cDVTolerancePCR     CDVTolerance,
                sustainableCellRate     [0] SustainableCellRate OPTIONAL,
                cDVToleranceSCR     [1] CDVTolerance        OPTIONAL,
                maxBurstSize            MaxBurstSize        OPTIONAL}}
UserCause   ::= ENUMERATED {bandwithNotAvailable(0), userNotAvailable(1), userNotCompatible(2)}
VpQoSClass  ::= INTEGER (0..99)
VcQoSClass  ::= INTEGER (0..99)
VcTestState::= ENUMERATED{inhibitTest(0), allowTest(1)}
VpTestState::= ENUMERATED{inhibitTest(0), allowTest(1)}
VpSchedulers    ::= SEQUENCE {
            startTime       StopTime,
        stopTime        StopTime,
            scheduleMechanism    CHOICE {
                durationScheduling       [0]    BidirectionalTrafficDescriptor,
        dailyScheduling          [1]    DailySchedule,
                weeklyScheduling        WeeklySchedule,
```

```
        monthlyScheduling           MonthlySchedule,
                occasionalScheduling         [2]    OccasionalSchedule}}
VcSchedulers::= VpSchedulers
WeekDay::= ENUMERATED {
                sunday (0),
                monday (1),
                tuesday (2),
                wednesday (3),
                thursday (4),
                friday (5),
                saturday (6)}
WeeklySchedule::= SEQUENCE OF WeekSlot
WeekSlot::= SEQUENCE {
        slotBegin   TimeWeek,
        slotEnd TimeWeek,
        bandwidth   BidirectionalTrafficDescriptor}
END
```

Annex A (informative): About the influence of timers on the X-interface communication

Over the X-interface, the response time to a reserve-request can be very long, due to the fact that some agents might need a long time to carry out the CMISE-indication. It is obvious that a manager cannot wait forever for the result of his request.

This annex describes how a manager can cope with this by using a timer.

As neither CMIP nor the rest of the OSI-stack have timers defined that specify the maximum responsetime, the application (the Manager or the Agent) will have to specify it.

Most applications will automatically end an association after a relatively short time.

To watch long response-times, the Manager can apply an additional internal timer (internal = "not visible on the X-interface").

So, the following cases can be considered. (a, b, c, d):

a) The response-time is relatively short ("real time"), no communications-error:

   - this is the normal case that is covered by the present model.

b) The response-time is usually relatively short, but a communication-error causes the association (connection) to end, during the time that the manager is waiting for the result of his request.

   - The Agent cannot use its CMISE response to the original CMISE indication because, due to the lost association, the invocation of the response is now meaningless. The Manager should wait (for a relatively short time) and then there will be two possibilities:

   1) The reservePnoVpSubnetworkConnection has a successfulResult: The Agent uses the object-creation Notification. The Agent is the invoker of a new association that is used for transmitting the Notification.

   2) The reservePnoVpSubnetworkConnection has an unsuccessfulResult: There is no Notification for the agent to send. The manager, however, is aware of the situation (the association has ended), so he can send the same reserveRequest again in order to get the unsuccessfulResult as a response.

      - (Or, first the Manager can do a GET on pnoVpSubnetworkConnection. If it does not see the Instance it can send the same reserveRequest again).

Case b) is also covered by the present model.

c) The response-time is long, no communication-error. This also addresses General Requirements G.7 and G.5:

   1) The internal "Manager-timer" watches the response-time. If the time is passed, the manager can act as if no request has been made at all and bring back the system to the "pre-request" state. To notify the Agent that it gave up waiting the Manager has to send a "releasePnoVpSubnetworkConnection" Action.

d) The response-time is long and a communications-error causes the association to end (or the manager ends it to reduce costs). There are two possibilities:

   1) The reservePnoVpSubnetworkConnection has a successfulResult within the waiting-time of the manager: The agent uses the object-creation Notification. The Agent is the invoker of a new association that is used for transmitting the Notification.

   2) The reservePnoVpSubnetworkConnection has an unsuccessfulResult: There is no Notification for the agent to send. The manager waits until the time has passed (remember that the waiting has to end some time), it brings back his system in the "pre-request" state and, to be certain, sends a "releasePnoVpSubnetworkConnection" Action to the Agent.

Annex B (informative): Security Aspects

With regard to Access Control as seen from a particular Agent's view the next table should be considered.

Table B.1

Objectclass	Access to the instances in accordance with the GDMO definition of the OC for:	Notifications are sent to:
PnoVcSubnetwork	All other PNOs.	All other PNOs
PnoVpSubnetwork	All other PNOs.	All other PNOs
PnoVpSubnetworkR2	All other PNOs.	All other PNOs.
PnoVcSubnetworkConnection (note 1)	Only the PNO that requested the creation of the instances involved. (Initiating PNO)	Only the PNO that requested the creation of the instances involved. (Initiating PNO)
PnoVpLinkConnection (note 1)	Only the PNO that requested the creation of the instances involved. (Initiating PNO)	Only the PNO that requested the creation of the instances involved. (Initiating PNO)
PnoVpSubnetworkConnection (note 1)	Only the PNO that requested the creation of the instances involved. (Initiating PNO)	Only the PNO that requested the creation of the instances involved. (Initiating PNO)
PnoVpSubnetworkConnectionR2 (note 1)	Only the PNO that requested the creation of the instances involved. (Initiating PNO)	Only the PNO that requested the creation of the instances involved. (Initiating PNO)
PnoNWAtmAccessPoint (note 2)	All other PNOs.	All other PNOs.
PnoNWAtmAccessPointR2 (note 2)	All other PNOs.	All other PNOs
InterPnoTopologicalSubnetworkPair	All other PNOs.	All other PNOs
PnoVCCTP	Only those PNOs that requested a pnoVcSubnetworkConnection that points to the pnoVCCTP.	Only those PNOs that requested a pnoVcSubnetworkConnection that points to the pnoVCCTP.
PnoVPCTP	Only those PNOs that requested a pnoVpSubnetworkConnection /LinkConnection that points to the pnoVPCTP.	Only those PNOs that requested a pnoVpSubnetworkConnection/Link Connection that points to the pnoVPCTP.
pnoVPTTP	The PNOs that use it as a "Server-TTP" for VC connections.	The PNOs that use it as a "Server-TTP" for VC connections.
NOTE 1: A Manager, trying to read all Instances of pnoVpSubnetworkConnection shall only get the Instances in which he is the initiating PNO. Whether his attempt to read all instances should be registered is a matter for "Security Management". NOTE 2: Z nets: After a successful reserve-request has been made in its subnet, an Agent-PNO is free to decide if it wants to create an instance of pnoNWAtmAccessPoint for the user-side of the connection.		

In general, the usage of a "closed user group" is suggested; only PNOs that are in the closed user group are allowed to set up an association with the appropriate PNO.

How this Access Control is implemented is not within the scope of the present document.

Annex C (informative): ATM VP/VC Resilience

A PNO playing the Initiator role should be able to choose between 4 different ways of protecting a User to user VP/VC connection, depending on the way the protection is performed. These are:

- Intra-Subnetwork Protection Switching;
- Intra Subnetwork Recovery;
- A-to-Z Fast Re-routing;
- VP/VC Reconfiguration.

It should be possible to apply these four methods of VP/VC protection independently or in combination.

The detailed explanation of each of the four definitions listed above is achieved with the support of a figure representing each different situation. The topology map that will be used as a basis for the examples is provided in figure C.1, in which no connections are depicted.

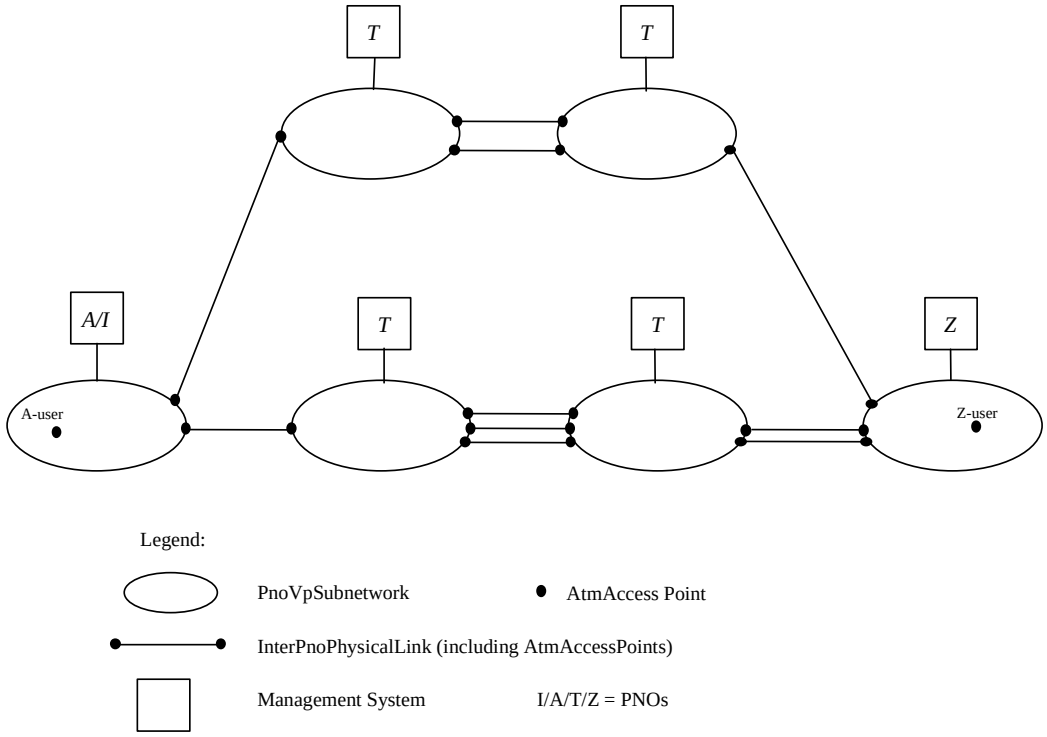


Figure C.1: General Topology map with User to user and VP/VC Subnetwork connections omitted

ITU-T Recommendation I.630 [6] requires that a connection is protected by one or more other connections. Therefore two types of connections, 'working' and 'protection' - were defined. The working connection is assigned to transport the traffic, whereas the protection connection is assigned to transport the traffic when protection is active. For Intra-Subnetwork Protection Switching, the configurations 1+1 and 1:1 were considered [6]. The (1:1) concept is also applicable to 'Fast Re-routing'. The use of simultaneous transmission of traffic on working and protection VP/VC connections and the availability of rapid switching to the protection connection in the event of a failure to allow Fast Re-routing is described later in this subclause.

Intra-Subnetwork Protection Switching

Protection switching is an event which may be activated by a PNO (within its own Subnetwork) supporting a VP/VC Subnetwork connection in response to an alarm generated by a network entity. The requirement is for the network elements to be able to select (i.e. switch to) dedicated and pre-assigned - capacity, which is protecting primary resources involved in the provision of the VP/VC, on receipt of the alarm.

The ATM Protection Switching can be a 1+1 or a 1:1 configuration. The (1+1) configuration conveys the protected traffic simultaneously on the working and protection connection. By contrast, the protection connection does not simultaneously convey protected traffic in the (1:1) configuration, but relies on traffic being rapidly switched onto it if the working connection fails.

Further, a requirement for protection switching is that it can be activated without the involvement of a manual network management function. Accordingly, it is a function that shall be automatic and very rapid.

Intra-Subnetwork Protection Switching does not include Protection Switching of IPPL.

The I-PNO will choose the Intra-Subnetwork Protection Switching option when it is sufficient to protect a connection against faults/performance degradations that occur within the Subnetwork of an A/T or Z-PNO.

The PNO experiencing the fault performs Protection Switching on its Subnetwork connection, keeping the endpoints unchanged, as represented in the figure C.2.

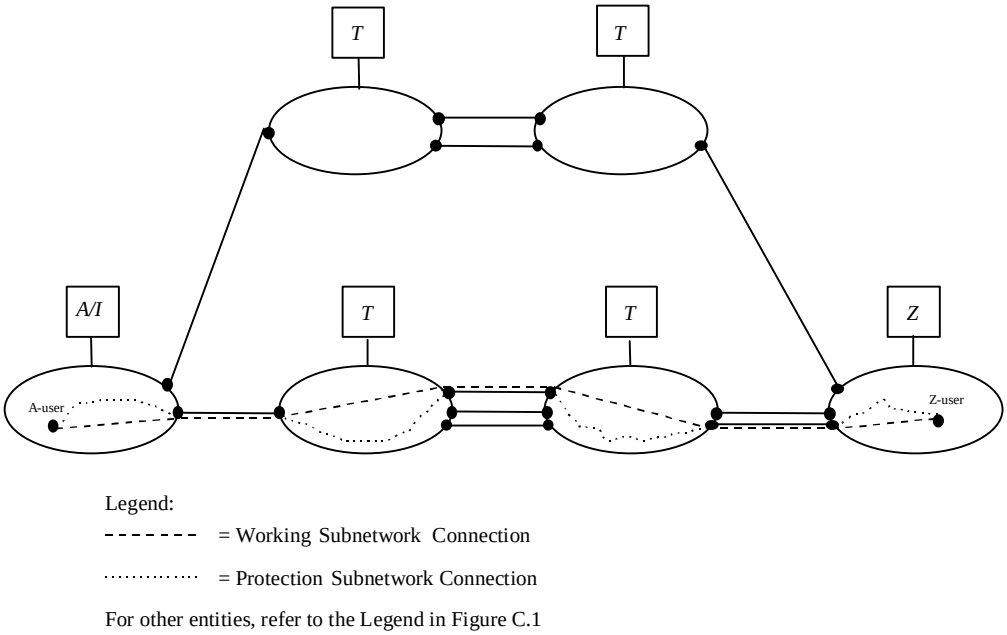


Figure C.2: Intra Subnetwork protection Switching

The PNO experiencing the fault will have to notify the I-PNO that he has successfully performed the protection switching on his part of the connection.

Intra-Subnetwork Recovery

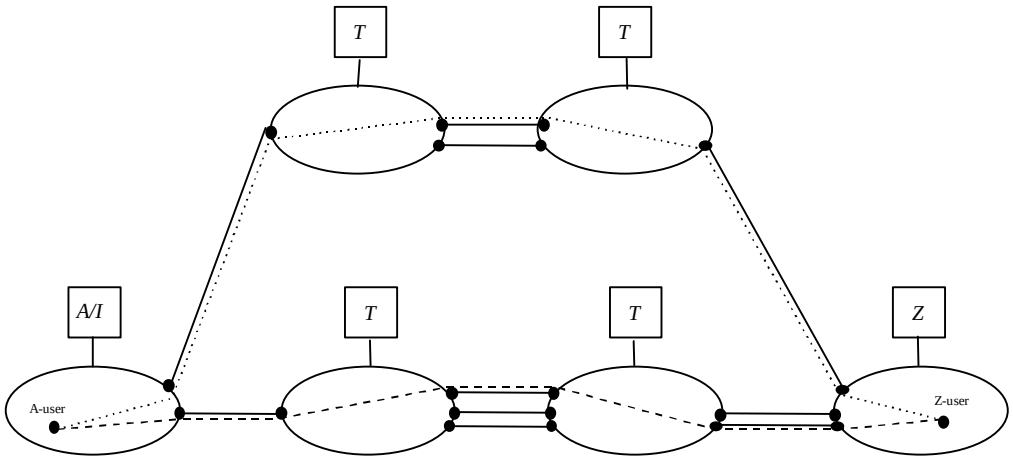
Recovery means a change-over to spare or back-up resources resulting from an internal network management action of an A, T or Z-PNO, which successfully restores an existing VP or VC to an operational state in a short time (e.g. seconds or minutes). A recovery action may follow a failed attempt to overcome a network resource failure by protection switching, or recovery action could be the only strategy implemented to protect a connection.

In the latter case, when a PNO experiences a network resource failure, it should start the recovery process and immediately send out the proper notification to the I-PNO.

The out-of-service time affecting the connection depends on each individual case and on how the recovery processes are implemented. In all cases, the user will experience service failure.

A-to-Z Fast Re-routing

For highly guaranteed connections the I-PNO might require a global (A User to Z User) resilience mechanism, that protects against both intra-PNO and inter-PNO failures. For this requirement, the End-to-end connection needs to be duplicated on a completely different route, especially on different inter-PNO links and needs to be continuously active during the lifetime of the protected connection, as depicted in figure C.3.



Legend:

= Working Connection

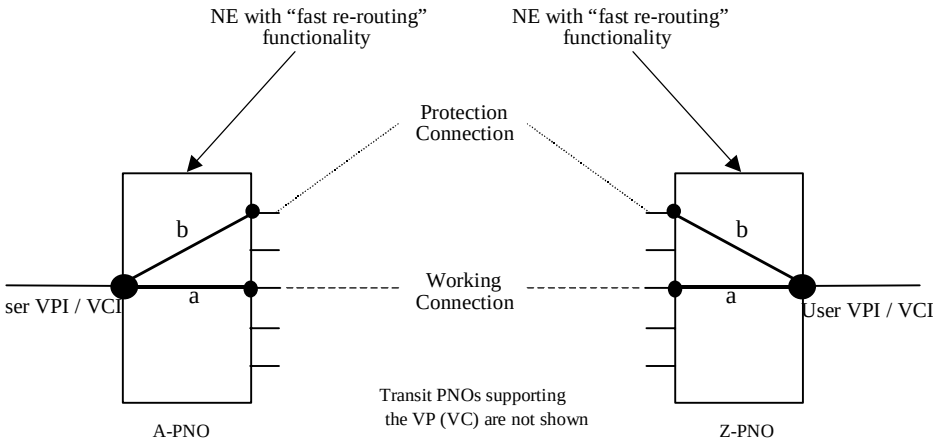
= Protection Connection

For other entities, refer to the Legend in Figure C.1

Figure C.3: A to Z Fast Re-routing

If a failure occurs, the traffic will need to be routed, as soon as possible, from the working to the protection connection. This mechanism is called 'Fast Re-routing'. Some cell loss associated with the failure is to be expected however.

Figure C.4 depicts the basic mechanism where the traffic from the A-user connection is simultaneously routed over working connection 'a' and protection connection 'b' from the designated network element inside the A-PNO's Subnetwork. Incoming traffic at the Z-PNO's Subnetwork should be taken from the 'a' connection but may be switched to the 'b' connection on receipt of the appropriate notification from the I-PNO.



Legend:

NE:	Network Element		
VP	Virtual Path	VC	Virtual Channel
VPI	Virtual Path Identifier	VCI	Virtual Channel Identifier

Figure C.4: Fast Re-routing at A- or Z-Subnetwork

The processes to invoke Fast Re-routing are as follows.

If a PNO detects a failure in its domain it will send a notification to the I-PNO. The I-PNO will request the A-PNO and the Z-PNO to take the traffic arriving from the alternative route (i.e. to switch from working connection (a) to protection connection (b))

VP Reconfiguration

Here, the I-PNO does not reserve an alternative path in advance, nor asks for special protected connections.

After receiving the notification from a PNO that a failure has occurred in its Subnetwork, the I-PNO has the choice to wait for the notification that the failure has been cleared or to start to set-up an alternative route for the affected connection, using the reservation-procedures.

Annex D (informative): Mapping between Management Functions, CMISE Services and ObjectClasses

Table D.1

FUNCTIONS	CMISE SERVICES	OBJECT CLASSES / Operations
Activate Bidir. Cont. Mon. Source for Continuity Check	M-ACTION	PnoBidirectionalContinuityMonitor/ controlContinuity Check Action
Activate Change	M-ACTION	PnoVpSubnetworkConnection(+R2), pnoVcSubnetworkConnection, pnoVpLinkConnection / activateChange Action
Activate VC Subnetwork Connection	M-SET	PnoVcSubnetworkConnection / administrativeState Attribute
Activate VP Link Connection	M-SET	PnoVpLinkConnection / administrativeState Attribute
Activate VP Subnetwork Connection	M-SET	PnoVpSubnetworkConnection(+R2) / administrativeState Attribute
Allow F4 flow	M-SET	PnoVpSubnetworkConnection(+R2) / vpTestState Attribute
Allow F5 flow	M-SET	PnoVcSubnetworkConnection / vcTestState Attribute
Cancel Change	M-SET	PnoVpSubnetworkConnection(+R2), PnoVcSubnetworkConnection, pnoVpLinkConnection / changeReservationInformation Attribute
Cancel VC Subnetwork Connection	M-EVENT-REPORT	PnoVcSubNetworkConnection / CancelVcNetworkConnection Notification
Cancel VP Link Connection	M-EVENT-REPORT	PnoVpLinkConnection / CancelVpLinkConnection Notification
Cancel VP Subnetwork Connection	M-EVENT-REPORT	PnoVpSubNetworkConnection(+R2) / CancelVpNetworkConnection Notification
Change Reservation	M-SET	PnoVpSubnetworkConnection(+R2), PnoVcSubnetworkConnection, PnoVpLinkConnection / changeReservationInformation Attribute
Check Available Cell Rate	M-ACTION	PnoVpSubnetwork(+R2), PnoVcSubnetwork / giveAvailableLinks Action
Create Bidirectional Continuity Monitor for Continuity Check	M-CREATE	PnoBidirectionalContinuityMonitor / Create
Deactivate Bidir. Cont. Mon. Source for Continuity Check	M-ACTION	PnoBidirectionalContinuityMonitor / controlContinuity Check Action
Deactivate VC Subnetwork Connection	M-SET	PnoVcSubnetworkConnection / administrativeState Attribute
Deactivate VP Link Connection	M-SET	PnoVpLinkConnection / administrativeState Attribute
Deactivate VP Subnetwork Connection	M-SET	PnoVpSubnetworkConnection(+R2) / administrativeState Attribute
Delete Bidirectional Continuity Monitor	M-DELETE	PnoBidirectionalContinuityMonitor / Delete
Destination User Checking	M-ACTION	PnoVpSubnetwork(+R2), pnoVcSubnetwork / checkUser Action
Notification of Bidirectional Continuity Monitor Operational State change	M-EVENT-REPORT	PnoBidirectionalContinuityMonitor / stateChangeNotification Notification
Release VC Subnetwork Connection	M-ACTION	PnoVcSubnetwork / ReleasePnoVcSubnetworkConnection Action

FUNCTIONS	CMISE SERVICES	OBJECT CLASSES / Operations
Release VP Link Connection	M-ACTION	PnoVpSubnetworkR2 / ReleasePnoVpLinkConnection Action
Release VP Subnetwork Connection	M-ACTION	PnoVpSubnetwork(+R2) / ReleasePnoVpSubnetworkConnection Action
Report of Continuity Check	M-EVENT-REPORT	PnoBidirectionalContinuityMonitor / tmnCommunicationAlarmInformation Notification
Reroute PNO Subnetwork Connection		Not used
Reserve Enhanced Vp Subnetwork Connection	M-ACTION	PnoVpSubnetworkR2 / reservePnoEnhancedVpSubnetworkCon--nection Action
Reserve VC Subnetwork Connection	M-ACTION	PnoVcSubnetwork / reservePnoVcSubnetworkConnection Action
Reserve VP Link Connection	M-ACTION	PnoVpSubnetworkR2 / reservePnoVpLinkConnection Action
Reserve VP Subnetwork Connection	M-ACTION	PnoVpSubnetwork / reservePnoVpSubnetworkConnection Action
Stop F4 flow	M-SET	PnoVpSubnetworkConnection / vpTestState Attribute
Stop F5 flow	M-SET	PnoVcSubnetworkConnection / vcTestState Attribute
Switch To Subnetwork Connection	M-ACTION	PnoVpSubnetworkConnectionR2, PnoVcSubnetworkConnection, / switchToSnc Action
Topology Info Changes	M-EVENT-REPORT	InterPnoTopologicalSubnetwork--Pair / createDeleteNotification / attributeValueChangeNotification
VC Subnetwork Connection Activation Notification (the administrativeState is automatically changed by the agent PNO)	M-EVENT-REPORT	PnoVcSubnetworkConnection / stateChangeNotification
VP Link Connection Activation Notification (the administrativeState is automatically changed by the agent PNO)	M-EVENT-REPORT	PnoVpLinkConnection/ stateChangeNotification
VP Subnetwork Connection Activation Notification (the administrativeState is automatically changed by the agent PNO)	M-EVENT-REPORT	PnoVpSubnetworkConnection(+R2) / stateChangeNotification

Annex E (informative): Mapping parameters VpQosClass and VcQosClass to ATM QoS

In the VP/VC reserve request a certain QoS is included. This parameter is represented by an INTEGER value (0..99). Since ATM QoS is described by various performance parameters (ITU-T Recommendation I.356 [14]), a QoS table should be agreed between the PNOs that use the X-Interface. This table should map integer values to specific combinations of values of the several QoS performance parameters. (Also, see [8]).

For a T subnetwork, the requested values of the QoS performance parameters should refer to the part of the connection that goes across the Agent's subnetwork and the "far-end" link.

For an A subnetwork, the requested values of the QoS performance parameters should refer to the part of the connection that originates at the A user and goes across the Agent's subnetwork and the "far-end" link.

For a Z subnetwork, the requested values of the QoS performance parameters should refer to the part of the connection that goes across the Agent's subnetwork and terminates at the Z user.

If possible, the requested values of the QoS performance parameters could be split into a subnetwork connection (SNC) part and a link connection (LC) part.

Figure E1 shows an example of a table, with no values filled in. It is possible that not all performance parameters have to be used.

QoS Parameter	**CLR**		**CTD**		**CMR**		**CDV**		**CER**		**SECBR**	
INT	SNC	LC	SNC	LC	SNC	LC	SNC	LC	SNC	LC	SNC	LC
1												
2												
3												
etc												

Figure E.1: A table to map integer values to QoS performance parameter values

Annex F (informative): Future Organizational Models

This clause describes different organizational models, which might be used in future for establishing X-interface relationships between PNOs. These models should be flexible enough to accommodate many different combinations of interconnected PNOs. However, the choice of which model to use will be determined by agreements between the participating PNOs. The implications that the future organizational models will have on the X-interface model need to be further studied.

Since there are two extreme organizational models and a mixture of those extremes, there can be (at least) three different organizations for a particular VP connection (if there are more than three PNOs involved). These three organizations are shown in figure F.1:

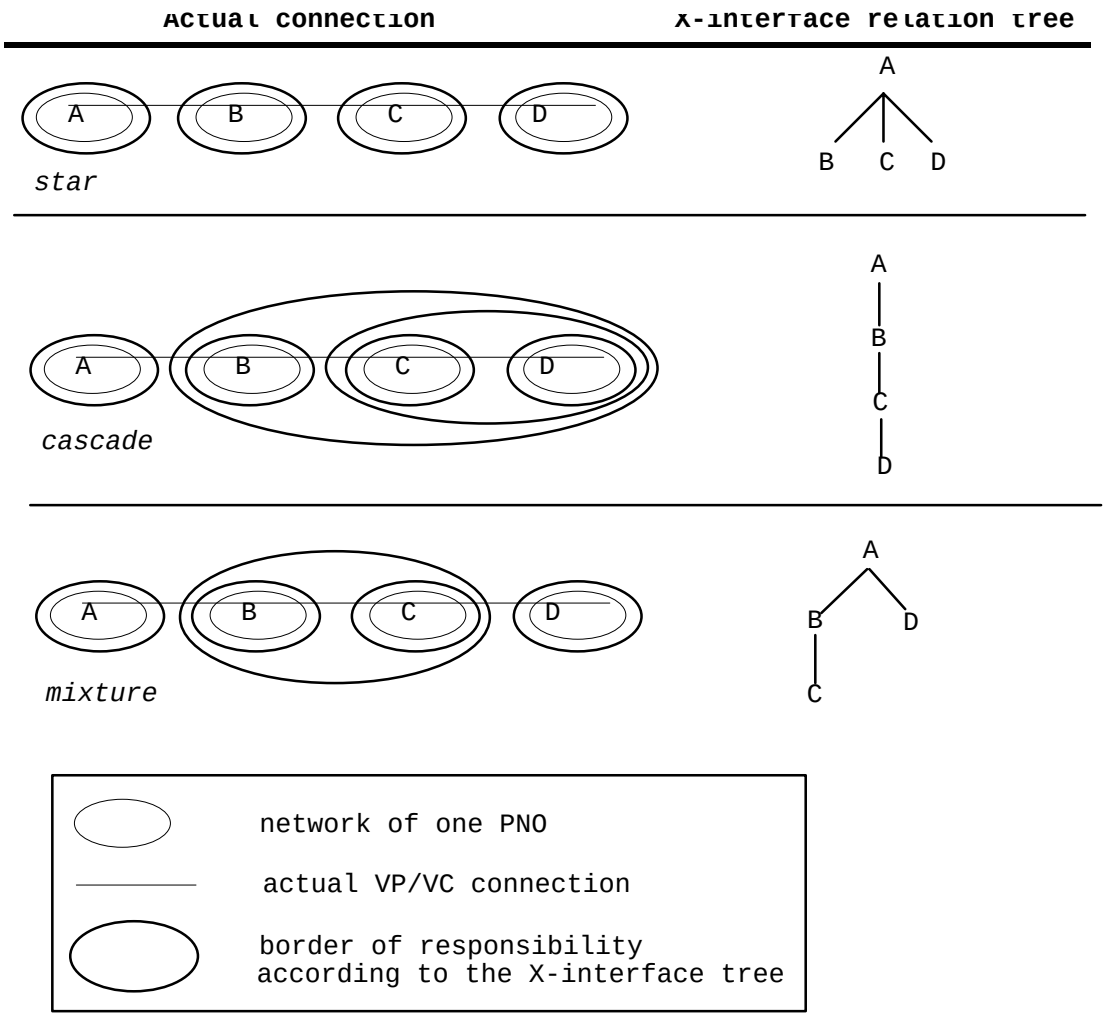


Figure F.1: Organizational models

In the *Star* organization as exemplified in the figure above, PNO A uses the X-interface relation with all the PNOs involved. This means that PNO A has full responsibility for the entire connection and all the other PNOs are only responsible for their own network towards PNO A.

In the *Cascade* organization in this example, PNO A delegates the responsibility for the VP connection outside its network to B; on its turn, B delegates the responsibility for the VP connection outside its network to C, etc. This means that PNO A has responsibility for the entire connection; B, however, is **towards PNO A** not only responsible for its own network but also for this particular VP connection throughout the networks of PNO C and D; on its turn, PNO C is **towards PNO B** not only responsible for its own network but also for this particular VP connection throughout the networks of PNO D, etc.

In the "*Mixture"* organization in this particular example (other examples are also possible), PNO A uses an X-interface relationship with both PNO B and PNO D (i.e. a star organization). However, PNO B involves PNO C to reach PNO D. Therefore PNO B is responsible towards PNO A for both its network and the connection throughout the network of PNO C (i.e. cascade organization).

The Consumer/Provider roles, introduced in subclause 4.2, are reflected by the X-interface tree: in the Star example above, PNO A has a Consumer role, whereas PNO B, C and D have a Provider role. In the Cascade example above, PNO A has a Consumer role, PNO B has a Provider role towards PNO A,and a Consumer role towards PNO C, etc. In the Mixture example PNO A has a Consumer role, whereas PNO B and D have a Provider role towards PNO A; PNO B also has a Consumer role towards PNO C; PNO C has Provider role towards PNO B. Concluding: if a PNOx is directly above a PNOy in the X-interface tree, then PNOx has a Consumer role, and PNOy has a Provider role.

F.1 Responsibility of PNOs regarding a VP/VC connection

This clause denotes which responsibility is required from each PNO involved a particular VP/VC connection. The following rules apply:

- **Rule 1:** A PNO is responsible for the management of a particular VPC/VCC within its own PNO Subnetwork and the part of this VPC/VCC throughout all the PNO Subnetworks below it in the X-interface tree for this connection (i.e. it is responsible for its subnetwork view). It is responsible for this part of the connection towards its parent in the X-interface tree (if not being the Initiating PNO, which is responsible towards the connection customer).

- **Rule 2:** Besides the responsibility in rule 1, a PNO is also responsible for the ATM Connection over the physical connection from its PNO Subnetwork to the next PNO Subnetwork on the route of a VP/VCC (seen from the A towards the Z).

- **Rule 3:** A PNO can delegate part of its management tasks outside its own PNO subnetwork to one or more other PNOs (these PNOs have a Provider role). This delegation consists of maintaining a requested Quality of Service for this particular connection and reporting about it (via the X-interface to its parent in the X interface tree).

F.1.1 Examples of application of the responsibility rules

The responsibility rules described in clause F.1 are illustrated by the following scenario description. The scenario starts with the existence of a particular VPC/VCC between PNOs A and C via B and a particular X-interface relation tree. The scenario describes the occurrence of 4 failures that need management action; each of them is described in one step of the scenario description. The 4 steps are an example of part of the life cycle of this particular VPC/VCC and take place in sequence. The scenario description is illustrated in figures F.2 through F.5.

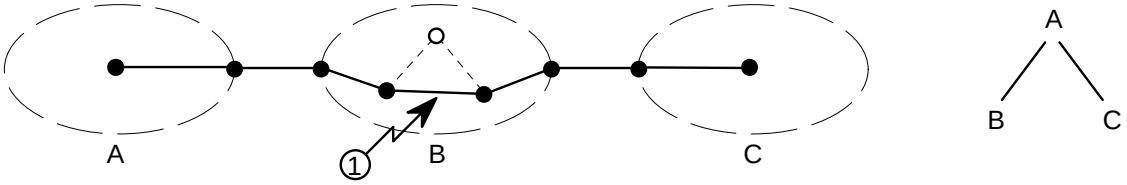


Figure F.2: Step 1 of example scenario

Step 1: A fault has occurred in the VPSC / VCSC within the PNO subnetwork of PNO B. PNO B reports to its Consumer PNO (PNO A) about the failure and indicates that the failure is recoverable. PNO B starts the reconfiguration process and ensures that alarms are suppressed. After reconfiguration has been successfully completed, PNO B reports this to its Consumer.

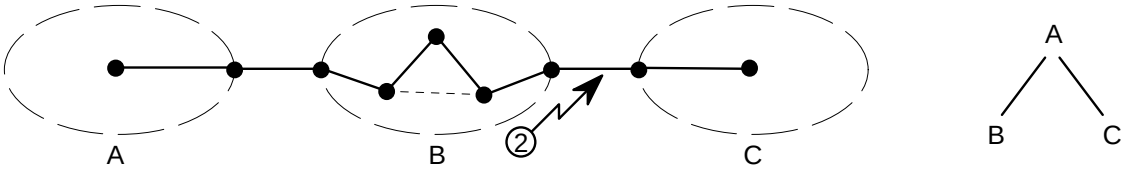


Figure F.3: Step 2 of example scenario

Step 2: A fault has occurred in the inter-PNO VP / VCLC between PNO B and PNO C. Since this is the outgoing inter-PNO VP / VCLC of the PNO subnetwork of PNO B, PNO B is responsible for reporting this failure to its Consumer PNO (PNO A). PNO B reports the failure to PNO A and indicates that it can recover the failure. PNO B performs recovery on the faulty inter-PNO VP/VC LC and notifies its Consumer about this and about the Id of the new physical link in the connection and takes action to suppress associated alarms.

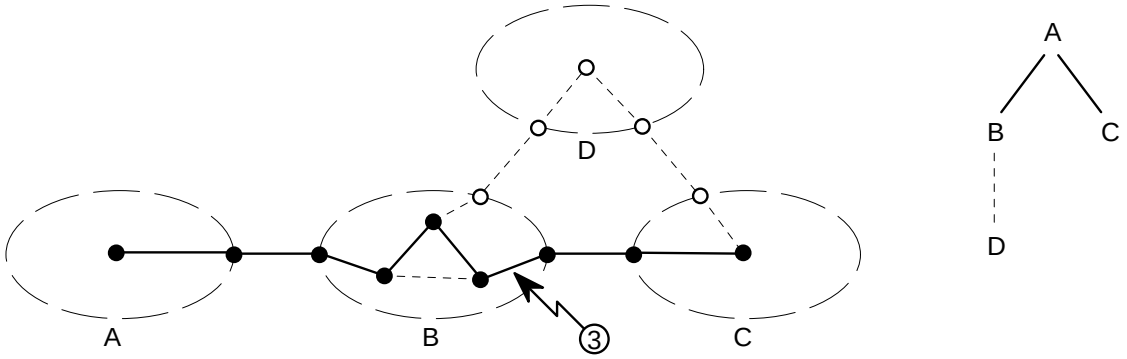


Figure F.4: Step 3 of example scenario

Step 3: A fault has occurred in the VPSC / VCSC within the PNO subnetwork of PNO B. PNO B reports this to PNO A. Let us assume that protection switching and reconfiguration of the VPSC / VCSC within the PNO subnetwork of PNO B are not possible. PNO B, however, believes that it can reroute the VPSC / VCSC via another PNO and indicates this to PNO A. PNO B starts a reconfiguration process, in which it reroutes the VPSC / VCSC via PNO D, and notifies its Consumer about this (for suppressing alarms). PNO B reserves a VPSC / VCSC within the PNO subnetwork of PNO D. After successful reservation, PNO D reports back to PNO B (its Consumer) the Id of the physical link between PNO D and PNO C. After reconfiguration has been completed, PNO B reports successful reconfiguration to its Consumer (PNO A) and also informs him about the Id of the new physical link between PNO D and PNO C (the physical link for which PNO B is held responsible). PNO A asks PNO C to cancel the VPSC / VCSC within the PNO subnetwork of PNO C without releasing the Z user and asks PNO C to reserve a new PNO VPSC / VCSC according to the new situation. Due to step three in the scenario the X-interface relation turns from *star* to a mixture of *star* and *cascaded*.

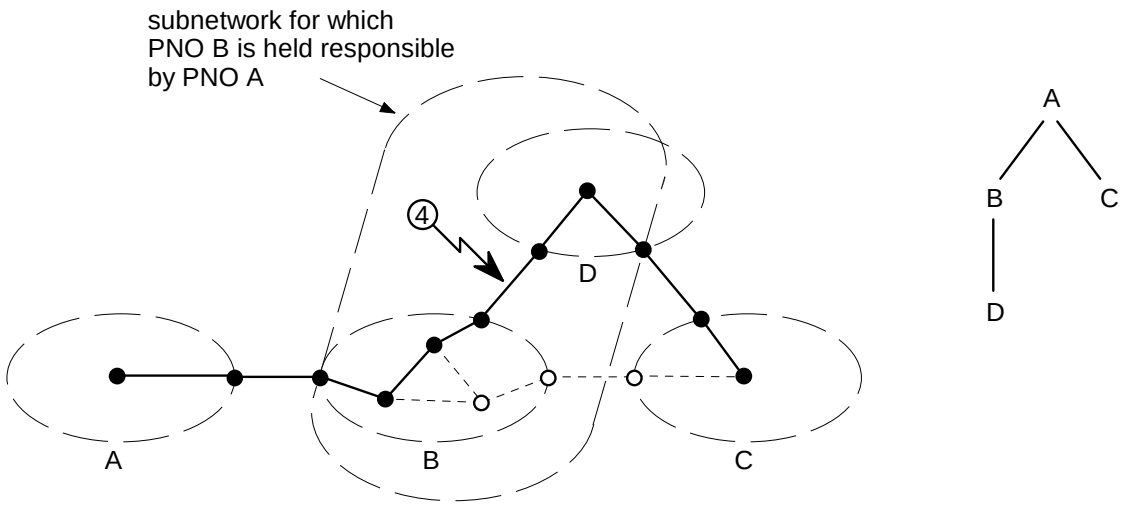


Figure F.5: Step 4 of example scenario

Step 4: A fault has occurred in the physical link between PNO B and PNO D. Because this physical link is now part of the subnetwork of PNO B, PNO B reports the failure to its Consumer (PNO A) as if it were a VPSC / VCSC failure (instead of a link failure). PNO B indicates that the failure is recoverable. PNO B performs protection switching on the inter-PNO VP/VC LC between PNO B and PNO D and notifies its Consumer about this. Associatd alarms should then be suppressed.

F.2 Scenarios

This clause contains scenarios that apply if the cascaded or mixed mode is to be used.

For all scenarios, the following VP/VC connections are possible between PNO A (Initiating) and PNO E (Destination): A-B-C-D-E, A-B-X-Y-D-E and A-B-X-Y-Z-E.

The following cases may be distinguished:

- case 1 (normal): A-B-C-D-E;
- case 2 (resulting from malfunctioning C): A-B-X-Y-D-E;
- case 3 (resulting from malfunctioning D): A-B-X-Y-Z-E.

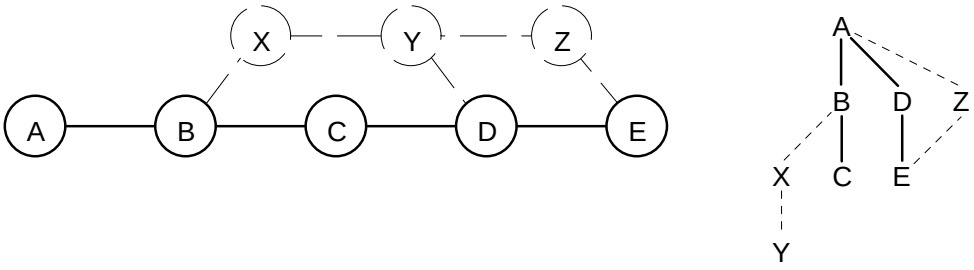
The following X-interface relations apply (of course other X-interface relations may exist, but they are not used in these particular cases; between brackets, the X-interface relations which are no longer used are mentioned):

- case 1: A-B, B-C, A-D and D-E;
- case 2: A-B, (B-C), B-X, X-Y, A-D and D-E;
- case 3: A-B, (B-C), B-X, X-Y, (A-D), (D-E), A-Z, Z-E.

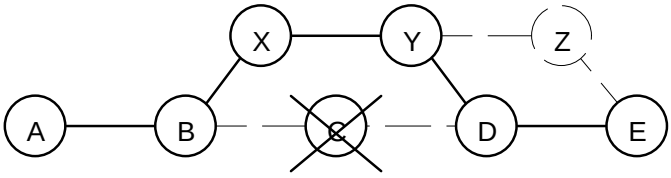
For these cases, the following scenarios are elaborated:

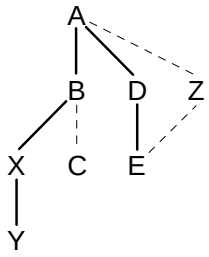
- scenario 1: PNO A establishes a VP/VC connection with PNO E;
- scenario 2: in the end situation of scenario 1, something goes wrong with the connections around PNO C (i.e. going from case 1 to case 2);
- scenario 3: in the end situation of scenario 1, something goes wrong with the connections around PNO D (i.e. going from case 1 to case 3);
- scenario 4: in the end situation of scenario 1, the connection parameters are modified, without changing the route.

Case 1:

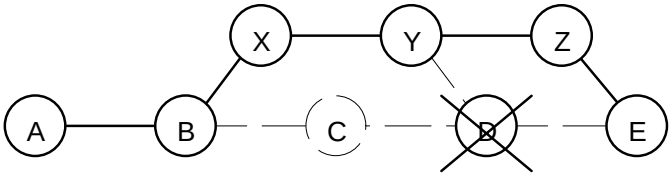


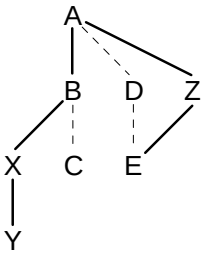
Case 2:



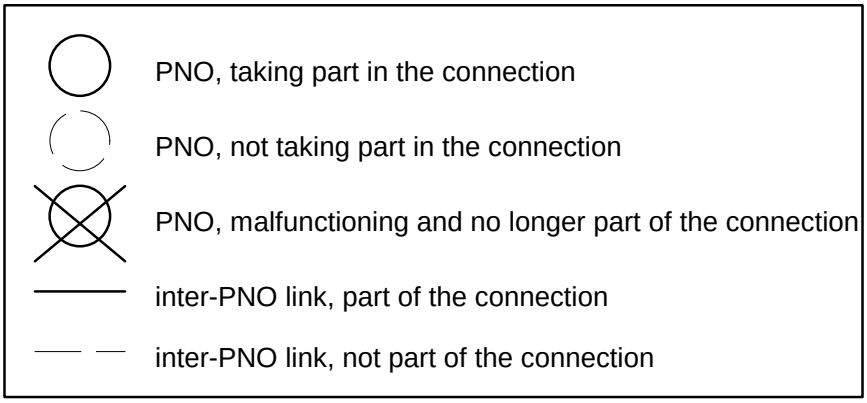


Case 3:

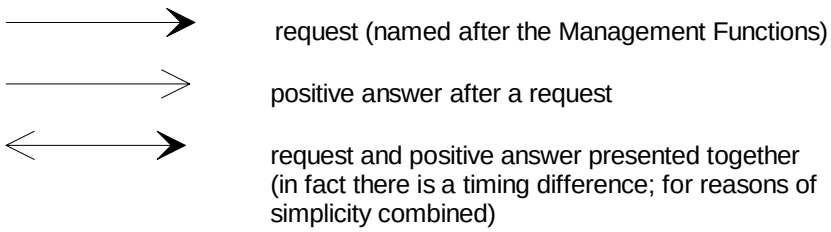




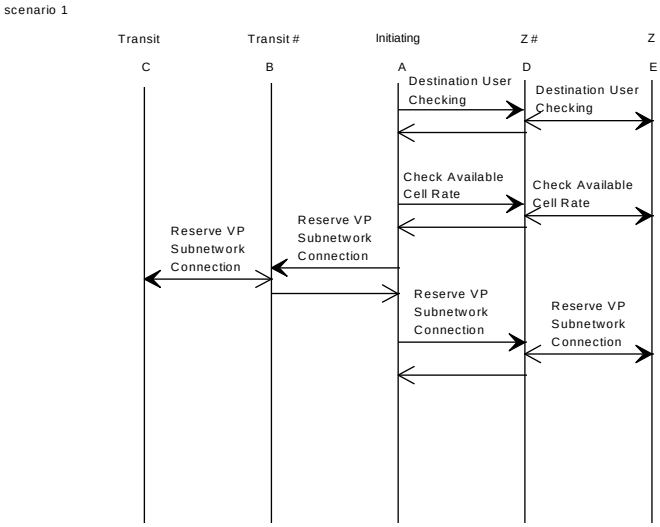
(connection maps) (X-interface relation trees)



In the next figures (flow charts), the following legend is applicable:

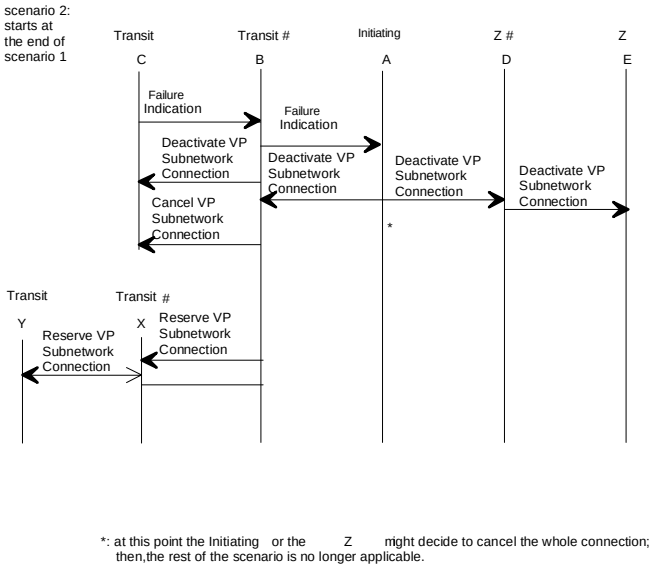


Scenario 1: PNO A establishes a VP/VC connection with PNO E.



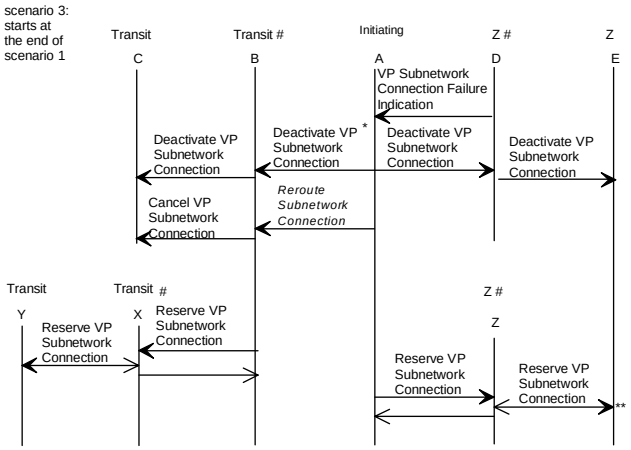
NOTE: Transit # and Z # act respectivly as Transit and Z from the view point of the Initiating PNO, even though they use other Transit and Z PNOs for the establisment of the connection.

Scenario 2: in the end situation of scenario 1, something goes wrong with the connections around PNO C (i.e. going from case 1 to case 2).



NOTE: At this point the Initiating PNO or the Z PNOmight decide to cancel the whole connection, in which case the rest of the scenario is no longer applicable.

Scenario 3: in the end situation of scenario 1, something goes wrong with the connections around PNO D (i.e. going from case 1 to case 3).



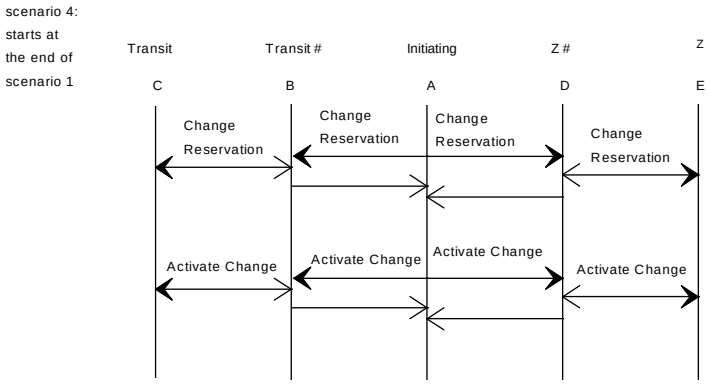
*: at this point the Initiating or the Z might decide to cancel the whole connection; then,the rest of the scenario is no longer applicable.

** since E receives a request for an already allocated VP identifier, E will recognise this as a request to reconnect to the existing connection to its destination user

NOTE 1: At this point the Initiating or the Z PNO might decide to cancel the whole connection, in which case the rest of the scenario is no longer applicable.

NOTE 2: Since PNO E receives a request for an already allocated VP identifier, PNO E will recognize this as a request to reconnect to the existing connection to its destination user.

Scenario 4: in the end situation of scenario 1, the connection parameters are modified, without changing the route.



Annex G (informative): Traffic Descriptor reference problems

Problems in the Abstract Syntax Notation One (ASN.1) part of the present document.

The "Schedule" datatypes (Daily-, Weekly-, Monthly-, Occasional-Schedule) that are in VpSchedulers and defined in the present document are also defined in ES 200 653 [9]. The reason that they are not imported from ES 200 653 [9] is that in ES 200 653 [9] TrafficDescriptor cannot be found.

The situation can be illustrated like this.

Typedefinitions in the old situation: (In the earlier ETSI X-interface versions: EN 300 820-1 [13]).

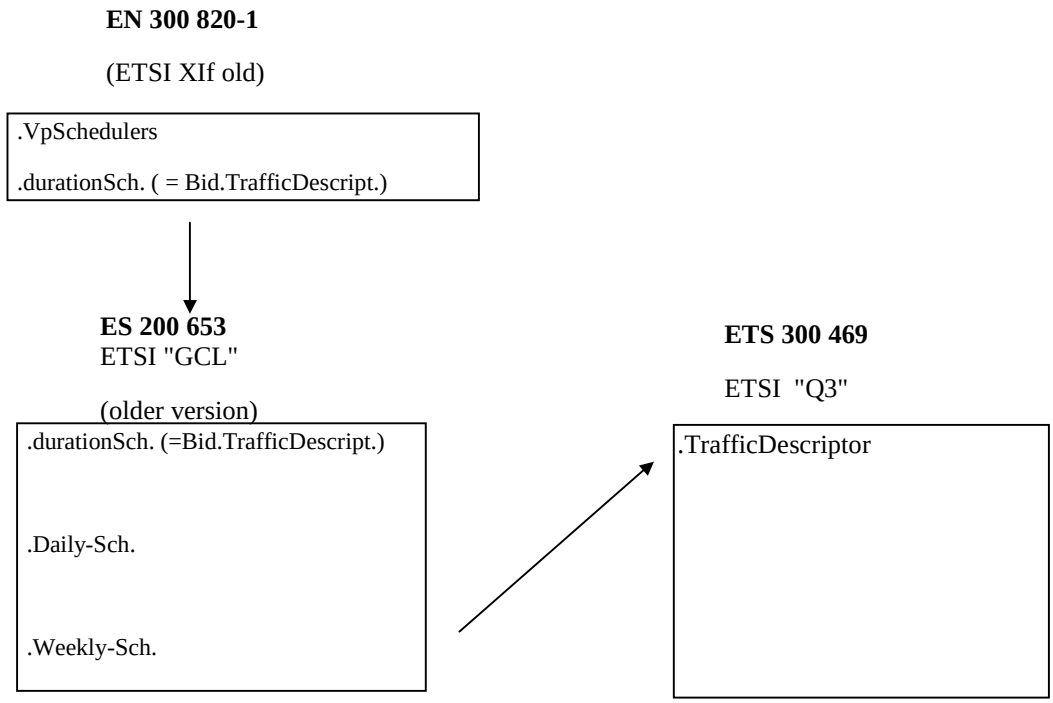


Figure G.1: trafficDescriptor reference during the development of the X-interface

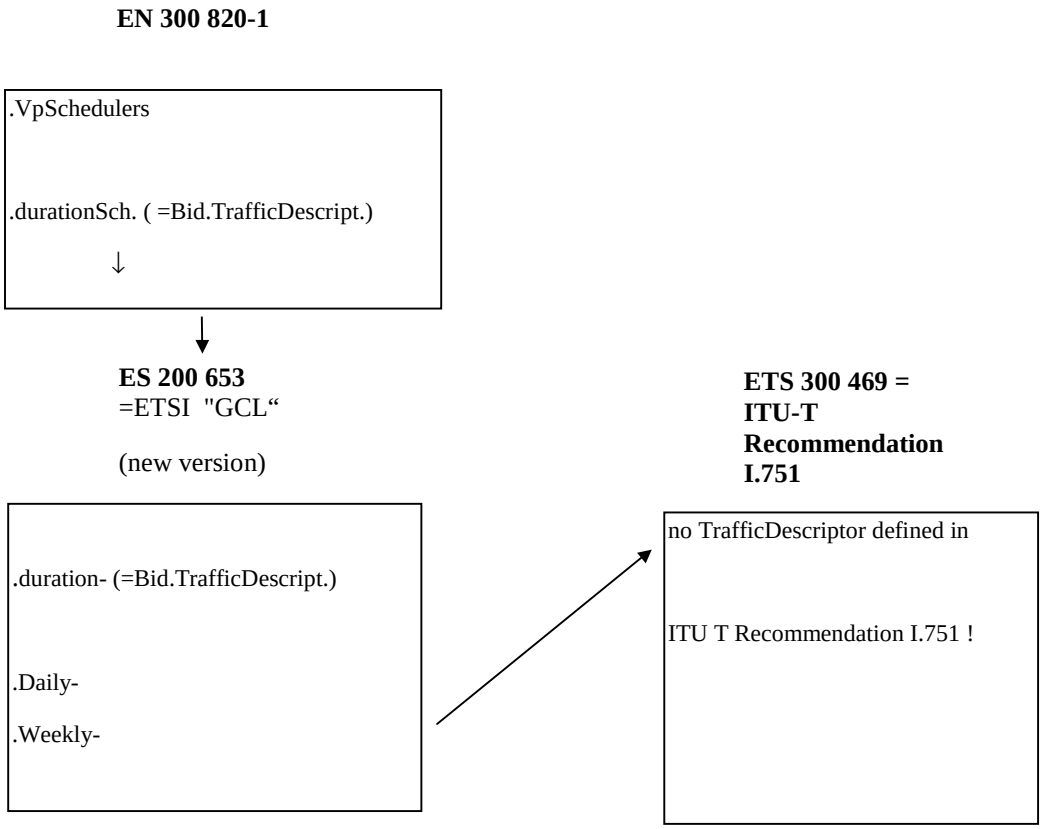


Figure G.2: trafficDescriptor reference problems

The problem is the endorsement of ETS 300 469 [12] with ITU Recommendation I.751 [3]: ETS 300 469 [12] now is a copy of ITU Recommendation I.751 [3]. There is no Type *TrafficDescriptor* defined in ITU Recommendation I.751 [3], so ES 200 653 [9] imports a non-existing Datatype.

To cope with this the Typedefinition *TrafficDescriptor* is included in the present document and all "ScheduleTypes" (and their "subtypes") that are also in ES 200 653 [9] are defined again (copied) in the present document to prevent "circular definitions".

It is recommended that the typedefinition of *TrafficDescriptor* (the one that is defined in the present document) will be included in ES 200 653 [9] in order to be able to remove the typedefinition of the „ScheduleTypes" from the present document.

NOTE: There is a typedefinition *BandwidthScheduling* in ES 200 653 [9] that is the same as *VpSchedulers*, only *monthlySchedule* and *occasionalSchedule* are interchanged.

So later, it is also possible to remove the typedefinition of VpSchedulers from the present document.

Bibliography

The following material, though not specifically referenced in the body of the present document (or not publicly available), gives supporting information.

- ITU-T Recommendation I.610: "B-ISDN operation and maintenance principles and functions".
- ITU-T Recommendation Q.822: "Stage 2 and stage 3 description for the Q3 interface-performance monitoring".
- ITU-T Recommendation Q.821: "Stage 2 and stage 3 description for the Q3 interface-alarm surveillance".
- ITU-T Recommendation X.720: "Information technology - Open Systems Interconnection - Structure of management information: Management information model".
- ITU-T Recommendation X.208: "Specification of Abstract Syntax Notation One".
- ITU-T Recommendation Q.811: "Q3 Lower layers Protocols".
- ITU-T Recommendation Q.812: "Q3 Higher layers Protocols".
- ITU-T Recommendation G.774: "Transmission and multiplexing SDH Information Model for the Network Element view".
- ETSI EN 300 371: "Transmission and Multiplexing (TM); Plesiochronous Digital Hierarchy (PDH) information model for the Network Element (NE) view".
- ITU-T Recommendation M.3020: "TMN Interface Specification Methodology".
- ITU-T Recommendation I.371: "Traffic control and congestion control in B-ISDN".
- ITU-T Recommendation I.357: "B-ISDN semi-permanent connection availability".

History

Document history		
V1.1.1	July 1998	Publication
V1.1.3	February 2000	Public Enquiry PE 200023: 2000-02-09 to 2000-06-09
V1.1.4	September 2000	Vote V 20001124: 2000-09-25 to 2000-11-24