The ATM Forum

Technical Committee

M4 Interface Requirements and Logical MIB: ATM Network View Version 2

AF-NM-0058.001

May, 1999

The information in this publication is believed to be accurate as of its publication date. Such information is subject to change without notice and The ATM Forum is not responsible for any errors. The ATM Forum does not assume any responsibility to update or correct any information in this publication. Notwithstanding anything to the contrary, neither The ATM Forum nor the publisher make any representation or warranty, expressed or implied, concerning the completeness, accuracy, or applicability of any information contained in this publication. No liability of any kind shall be assumed by The ATM Forum or the publisher as a result of reliance upon any information contained in this publication.

The receipt or any use of this document or its contents does not in any way create by implication or otherwise:

- Any express or implied license or right to or under any ATM Forum member company's patent, copyright, trademark or trade secret rights which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor
- Any warranty or representation that any ATM Forum member companies will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor
- Any form of relationship between any ATM Forum member companies and the recipient or user of this document.

Implementation or use of specific ATM standards or recommendations and ATM Forum specifications will be voluntary, and no company shall agree or be obliged to implement them by virtue of participation in The ATM Forum.

The ATM Forum is a non-profit international organization accelerating industry cooperation on ATM technology. The ATM Forum does not, expressly or otherwise, endorse or promote any specific products or services.

NOTE: The user's attention is called to the possibility that implementation of the ATM interoperability specification contained herein may require use of an invention covered by patent rights held by ATM Forum Member companies or others. By publication of this ATM interoperability specification, no position is taken by The ATM Forum with respect to validity of any patent claims or of any patent rights related thereto or the ability to obtain the license to use such rights. ATM Forum Member companies agree to grant licenses under the relevant patents they own on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. For additional information contact:

The ATM Forum
Worldwide Headquarters
2570 West El Camino Real, Suite 304
Mountain View, CA 94040-1313
Tel: +1-650-949-6700
Fax: +1-650-949-6705

Editor: Andrew J. Mayer, Ph.D.

Prior Editors:

af-nm-0054.000 and af-nm-0074.000:
Patrice Lamy

Table of Contents

Introduction

This document complements the M4 network-element view functional requirements and protocol independent MIB specification by providing a set of network-view managed entities which can be used to manage the ATM network both at the VP-level and at the VC-level[1]. The network-view is an aggregated view of the ATM resources, thus providing added flexibility to manage the network.

The management interface requirements and logical MIB outlined in this document are intended to guide the development of protocol-specific network management interface specifications in the ATM Forum. The purpose of defining a logical MIB is to provide a common frame of reference for the development of protocol-specific MIBs, such as those based on CMIP, SNMP, or other protocols supporting various Distributed Processing Environments (DPE), such as CORBA/IDL. The definition of protocol-specific MIBs from a common logical MIB should facilitate their potential coexistence within a public carrier's network.

The ATM Forum Management Interface Reference Architecture (shown in Figure 1-1) identified five distinct management interfaces, labeled M1 through M5. The focus of this specification is on the management interface labeled M4. The M4 interface has been defined by the ATM Forum as the interface to manage public ATM networks. This interface may be used to manage individual elements of the ATM network as well as to manage them in aggregation (e.g. subnetworks). Since requirements and managed entities to support the former have already been defined by the ATM Forum [3], the scope of this specification is on requirements and logical MIB aspects needed to support the aggregated or "network view."

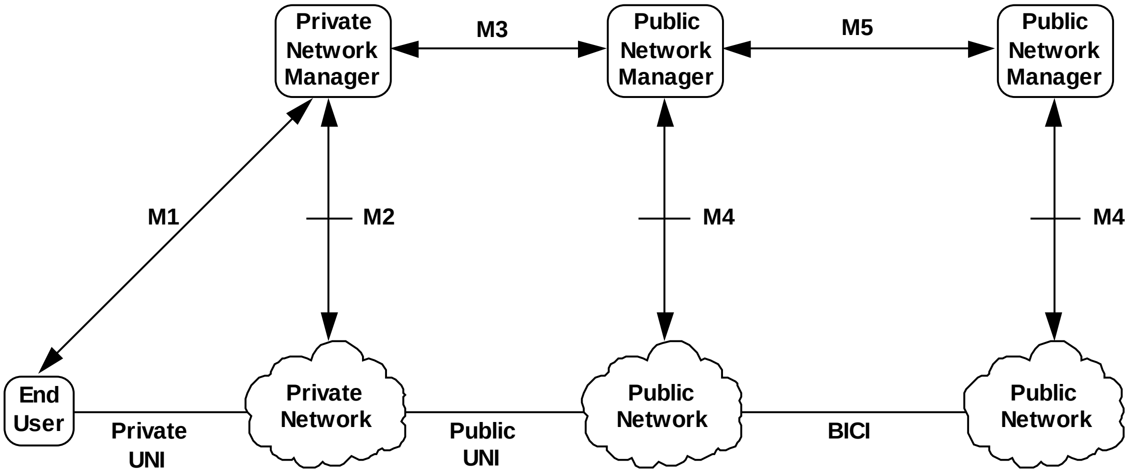


Figure 1-1 The ATM Forum Management Interface Reference Architecture

[1] Note that service management (see Appendix B for definition) is beyond the scope of this specification. This specification focuses on network management.

Section 2 of this specification provides a functional description of the ATM network based on ITU-T Recommendation I.326 [1] (ATM network functional architecture description). Section 3 provides a description of the ATM network management architecture. It shows what network management interfaces between the network managers and the managed resources can be defined. Section 4 provides network-view ATM management requirements for both the M4 network-view. Section 5 defines a logical model of the information that needs to be managed across the M4 interface in support of ATM network-level management. The control of these resources by the systems that manage them to perform the applications specified previously are described as operations. Section 6 provides outline ensembles, i.e. network management application scenarios for the most common network management configurations, both from an architectural perspective (as described in Section 3) and from a functional perspective (as described in Sections 4 and 5). The requirements and logical MIB defined in this document will be used by the ATM Forum when specifying the protocol-specific MIBs which will allow implementation of interoperable networks and management systems.

Scope

This specification provides requirements on the network-view aspects of the M4 interface needed to support ATM network management, i.e. the management of aggregates of Network Elements such as subnetworks. It also relates the M4 network-view and the M4 NE-view to build coherent management functions. The network-view is intended as an aggregate view to provide additional value.

This specification focuses on what is considered to be the initial functionality of ATM network view management. It is understood that this initial set of functions, protocol-independent managed entities, and profiles/ensembles will be enhanced in subsequent versions.

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

- Transport network configuration provisioning (including subnetwork provisioning, and link provisioning)
- Transport network connection management (including set-up/ reservation/ modification for subnetwork connection, link connection, trails, and segments)
- Network fault management (including correlation, localization, notification, for both equipment and connections, and loopback testing)
- Network performance management (including congestion monitoring, and connection and segment monitoring)
- Network accounting management
- Network security management

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

- any protection-switching, back-up functionality,
- grouping of connections,
- segment handling,
- reservation,
- accounting
- security.

This specification focuses on the interface functionality to manage the network, and does not provide requirements on the management systems themselves.

Relevant Network Management Modeling Standards

This document complements the M4 Interface Requirements and Logical MIB, ATM Network-Element View [3]. In addition, the following references are used:

IETF SNMP RFC

The main RFC of interest here is RFC1695 [4].

ITU Network Management Modeling Recommendations

This specification takes inputs from the ITU-T G.85x series of Recommendations [5,6,7], although it does not use the G.85x methodology.

- The requirements defined in Section 4 are based on Recommendation G.852-01, the enterprise viewpoint for connection management, specialized for ATM.
- The Managed Entities and their relationships in Section 5 are based on Recommendation G.853-01, the information view, specialized for ATM.
- The operations in Section 5 are based on Recommendation G.854-01, the computational interface specification, specialized for ATM.

ATM Forum CMIP Network View Version 1

This second verison of the ATM Forum Network View Requirements and Logical MIB was influenced by the CMIP implementation of Version 1 of the Network View interface found in af-nm-0073.000 [8].

Definitions

This subsection provides for any specific definition needed in this document.

Broadband Inter-Carrier Interface (BICI):
An interface between two nodes in networks of two different carriers (Public network Operators). This interface is also labeled interNNI.

Broadband Inter-Switching System Interface (BISSI):
An interface between two nodes within the network of a single Public Network Operator. This interface is also labeled intraNNI.

Element Management Layer (EML):
An abstraction of the functions provided by systems which manage each network element on an individual basis.

Element Management System (EMS):
A management system, which provides functions at the Element Management Layer, and could

also include functions at the Network Management Layer.

Layer Network: See Appendix A

Link:

An entity that defines a topological relationship (including available transport bandwidth) between two nodes in different subnetworks. Multiple links may exist between a pair of subnetworks.

Link Connection:

A link connection (e.g. at the VP-level) is a connection capable of transferring information transparently across a link. It is delineated by connection points at the boundary of the subnetwork. See Appendix A for more details.

Management Domain:

An entity used in this document to define the scope of naming.

Managed System:

An entity which is managed by one or more management systems, which can be either Element Management Systems, Subnetwork or Network Management Systems, or any other management systems.

Management System:

An entity which manages a set of managed systems, which can be either NEs, subnetworks, or other management systems.

Network Element (NE):

A system that supports at least NEFs and may also support Operation System Functions/Mediation Functions. An ATM NE may be realized as either a standalone device or geographically distributed system. It cannot be further decomposed into managed elements in the context of a given management function.

Network Element Function (NEF):

A function within an ATM entity that supports the ATM based network transport services, e.g. multiplexing, cross-connection.

Network Element Layer (NEL):

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

Network Management Layer (NML):

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

Network Management System (NMS):

An entity which implements functions at the Network Management Layer. It may also include Element Management Layer functions. A Network Management System may manage one or more other Network Management Systems.

NMS Environment:

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

Segment:

A link connection, subnetwork connection or tandem connection that involves Segment OAM Cell insertion/termination as defined in I.610.

Subnetwork:

A subnetwork is a network resource containing termination points which are available for interconnection.

Subnetwork connection (SNC):

In the context of ATM, an entity that passes ATM cells transparently, i.e. without adding any overhead. An SNC may be either a stand-alone SNC, or a concatenation of SNCs and link connections.

Subnetwork Management System (subNMS):

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

Trail:

An entity that transfers information provided by a client layer network between access points in a server layer network. The transported information can be monitored at the termination points.

User Parameter Control/Network Parameter Control (UPC/NPC):

A set of actions taken by the ATM NE to monitor and control traffic. Their main purpose is to detect violations of negotiated traffic parameters and to take appropriate action.

User Network Interface (UNI)

An interface between two nodes, one belonging a user network, the other to a Public Operator.

Relationship between nomenclatures of ITU-T Recommendations I.326 [1], G.85x [4,6,7], I.311 [9], and the M4 specifications:

M4 Network-View	**G.853-01**	**G.805**	**I.326**	**I.311**
Trail	Trail	Trail	Trail	Connection
Subnetwork Connection	Subnetwork Connection	Subnetwork Connection	Subnetwork Connection	Link
Topological Link	Link	Link	Link	---
Link Connection	Link Connection	Link Connection	Link Connection	Link
---	Tandem Connection	Tandem Connection	Tandem Connection	---
Trail Termination Point	Network Trail Termination Point	Port		Connection End Point
Subnetwork	Subnetwork	Subnetwork	Subnetwork	---
Connection Termination Point	Network Connection Termination Point	Port		Connecting Point

Note 1: italicized entries in the M4 column do not have full equivalence with the corresponding recommendation.

Note 2: ITU-T Recommendation I.610 [10] specifies a number of mechanisms for the operation, administration and maintenance of ATM networks. Some of the concepts and entities in the table above are supported by or implemented by the mechanisms of I.610 [10].

In particular:

- "trail" (G.85x, G.805, I.326) and "connection" (I.311) are related to the "end to end OAM cell flow" of I.610.

- "subnetwork connection" (G.85x, G.805, I.326) and "link" (I.311) are related to the "segment OAM cell flow" of I.610.
- "link connection" (G.85x, G.805, I.326) and "link" (I.311) are related to the "segment OAM cell flow" of I.610.
- "tandem connection" (G.85x, G.805, I.326) is related to the "segment OAM cell flow" of I.610.

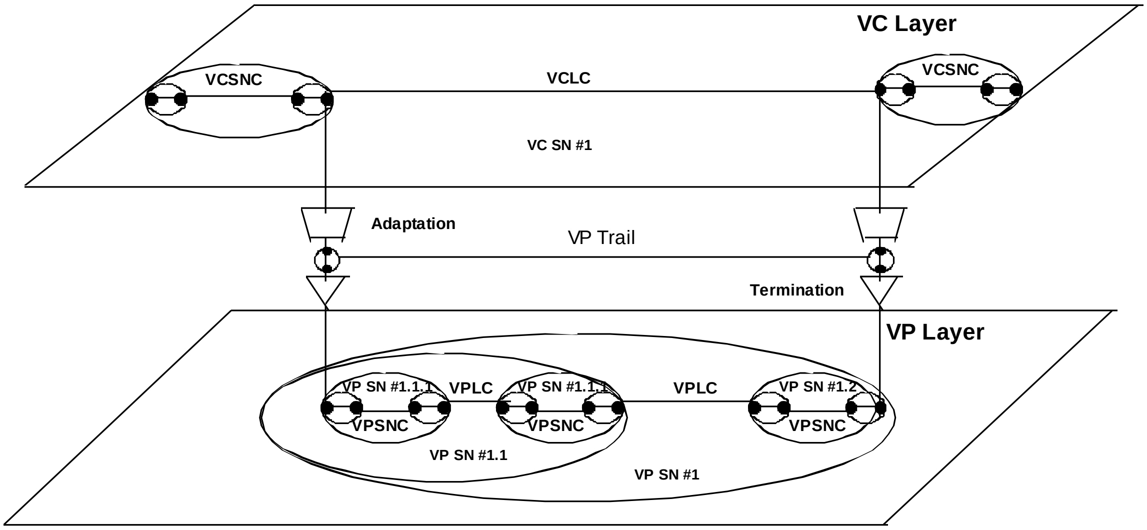
In any given network, a segment OAM cell flow only relates to one of the above concepts or entities, if that segment coincides with the topology modeled by the concept or entity.

Note 3: "Link" was used in the M4 NE-view requirements and logical MIB [3], as a link between two network elements, according to I.311 [9], which is equivalent to a "link connection" in I.326 [1]/G.805 [2] language. Here, a "link" is used as a link between two subnetworks, as defined in I.326 [1]/G.805 [2].

Transport Network Architecture Framework

The requirements and protocol-independent MIB, specified in Sections 4 and 5 respectively, have been defined to support the network architecture model defined in ITU-T Recommendation I.326 [1] and G.805 [2]. An understanding of the architectural concepts is therefore key to reading this document. Relevant excerpts from the above recommendations are provided in Appendix A.

The ATM transport network is made of *topological components* describing the network itself, and of *transport entities* describing the functions used to support the services carried by the network. The topological components are the layer networks such as VP and VC networks, subnetworks, links between subnetworks, and ports. Adaptation and termination functions assure interworking between layer networks. Subnetworks can be recursively divided into subnetworks and links. The transport entities are trail, subnetwork connections, and link connections. Figure 2-1 depicts the relationships between components and entities.



SN: SubNetwork
SNC: SubNetwork Connection
LC: Link Connection

Figure 2-1: Functional Representation of a Subnetwork Connection

The M4 Network View transport network architecture framework supports the decomposition of the network into a number of independent layer networks with a client/server relationship between adjacent layer networks. It also supports the possibility of merging the ATM VP functions and the ATM VC functions into a single ATM layer network, and generate, transport and terminate VPs and VCs as a single characteristic information, and the possibility of decomposing the ATM VP functions and the ATM VC functions into separate layer networks with a client/server relationship between them, and generate, transport and terminate ATM VCs and VPs separately. This choice is independent of the selection of a given network management physical implementation as described in the next section. In the second case, the M4 Network View supports the ability for each layer network to be described separately, and support the independent management of each layer. Last, the M4 Network View architecture supports controlling or modifying a VC layer network or a VP layer network without affecting each other from the architectural viewpoint.

Appendix A provides for a definition of these entities.

Network Management Architecture Framework

The purpose of this section is to show the logical architecture and corresponding physical configuration examples of the M4 interface, and to show the relationship between the M4 network-view and the M4 NE-view [3].

The M4 interface has been specified using two views. The Network Element view is concerned with managing individual ATM NEs. The network view is concerned with managing aggregate NEs as one or more subnetworks. This document specifies the requirements and logical MIB for the network view. The logical architecture for the M4 network view is shown in Figure 3-1.

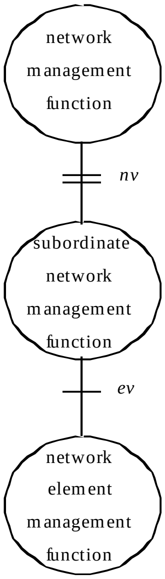

network management function
nv
subordinate network management function
ev
network element management function

Figure 3-1: M4 Network-View Logical Architecture

The figure above depicts a network management function using the network view to communicate with a subordinate network management function. For completeness, the figure also shows the NE view between the subnetwork management function and the element management function. For any particular network management application, managed entities from either the network or the network-element view can be used together. It is the purpose of the ensembles defined in Section 6 to specify the selection of the managed entities required to perform a particular application function and thus to assure interoperability.
Note that both network management functions shown above are part of the network management layer (see Appendix B).

For the purpose of operating on managed entities specified in the network view, the network management function acts in the manager role, and the subordinate network management function acts in the agent role. This manager/agent relationship does not imply the use of any specific management protocol.

The M4 logical architecture may be implemented in many ways. An implementation may use managed entities from the network view, the network element view, or both. This is illustrated in Figure 3-2.

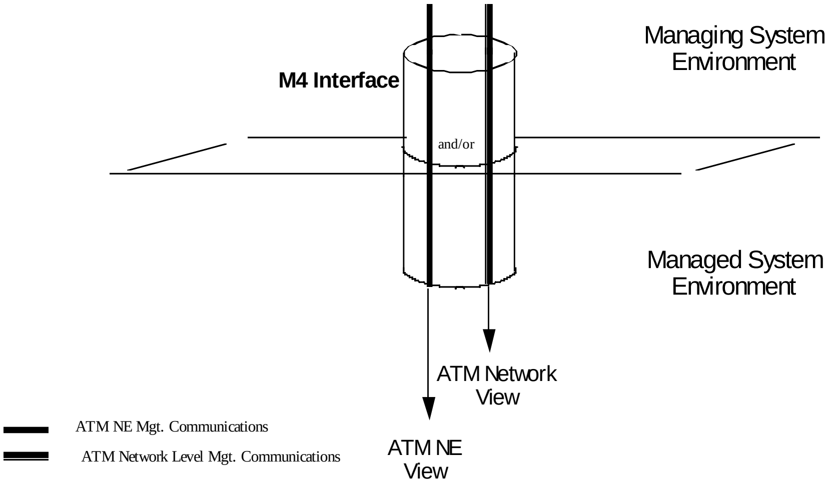

M4 Interface
and/or
Managing System Environment
Managed System Environment
ATM Network View
ATM NE View
ATM NE Mgt. Communications
ATM Network Level Mgt. Communications

Figure 3-2: Dual Views of the M4 Interface

Figure 3-3 shows examples of implementing the network view and network element view between individual management systems.

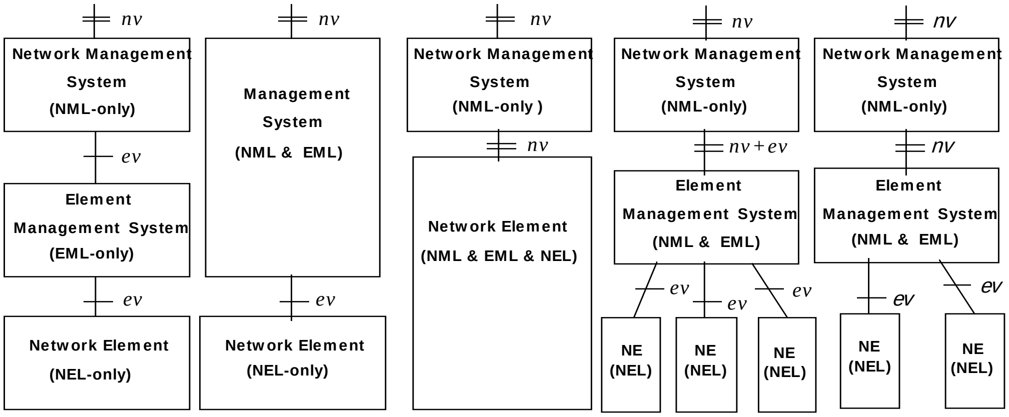

nv
Network Management System (NML-only)
ev
Element Management System (EML-only)
ev
Network Element (NEL-only)
nv
Management System (NML & EML)
ev
Network Element (NEL-only)
nv
Network Management System (NML-only)
nv
Network Element (NML & EML & NEL)
nv
Network Management System (NML-only)
nv+ev
Element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

Figure 3-3-a: hierarchical examples

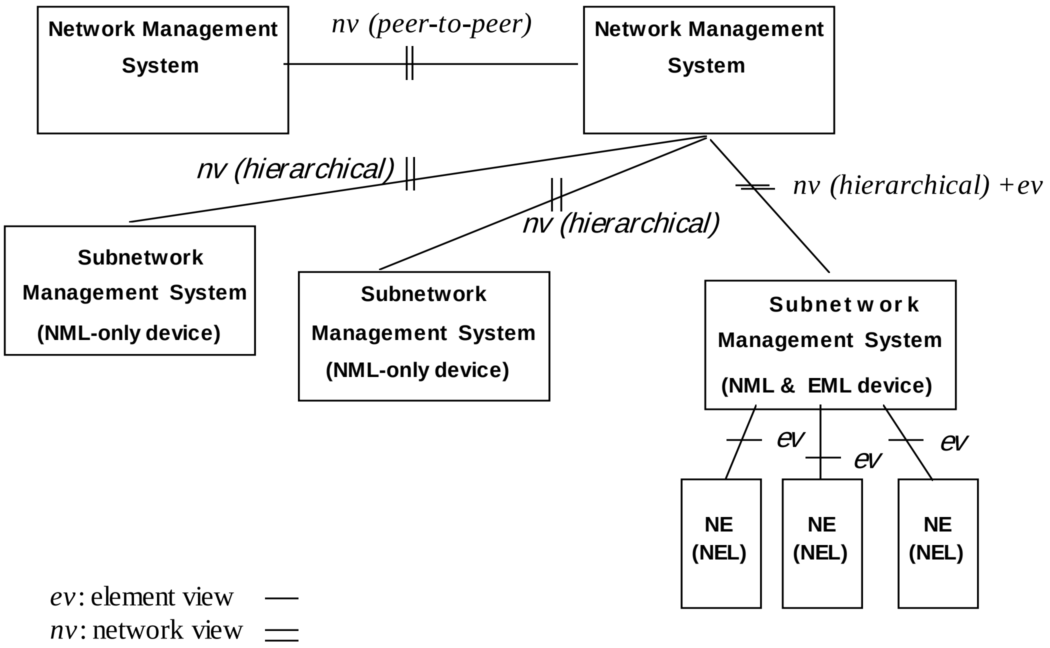

Network Management System
nv (peer-to-peer)
Network Management System
nv (hierarchical)
nv (hierarchical)
nv (hierarchical) +ev
Subnetwork Management System (NML-only device)
Subnetwork Management System (NML-only device)
Subnetwork Management System (NML & EML device)
ev
ev
ev
NE (NEL)
NE (NEL)
NE (NEL)
ev: element view
nv: network view

Figure 3-3-b: hierarchical and non-hierarchical examples

Figure 3-3: Physical Realization Examples of a Multi-layer Network Management Architecture

The ATM NE management interface requirements defined in [3] focus on EML-to-NEL and NML-to-EML interactions needed to support ATM NE management, where an ATM NE may be realized as either a stand-alone device or geographically distributed system. With respect to Figure 3-3, the requirements defined in [3] are relevant to ATM NEs supporting either NEL functions or a combination of NEL and EML functions as well as to the system(s) that manage them.

The ATM network management interface requirements defined in this specification focus on NML-to-NML interactions needed to support ATM subnetwork management. With respect to Figure 3-3, the requirements defined here are relevant to ATM NEs, Element Management Systems, or Network Management Systems supporting NML functions as well as to the system(s) that manage them.

To support the multiple architectures described in Figure 3-3, the ATM NE-view and the ATM network-view MIBs can be combined in multiple fashions:

NE-View Management Architecture Example:

In the NE-level Management Architecture (see Figure 3-1-1), the Network Management System (NMS) Environment (typically composed of one or more interconnected management systems) directly interfaces to the ATM NEs it manages. Note that the term "ATM NE" is used in the abstract sense, since it is possible that the actual interface terminates on a supplier-provided element management system, which manages one or more subtending NEs on an individual basis. When applied in this fashion, only the "ATM NE View" is exposed across the M4 interface.

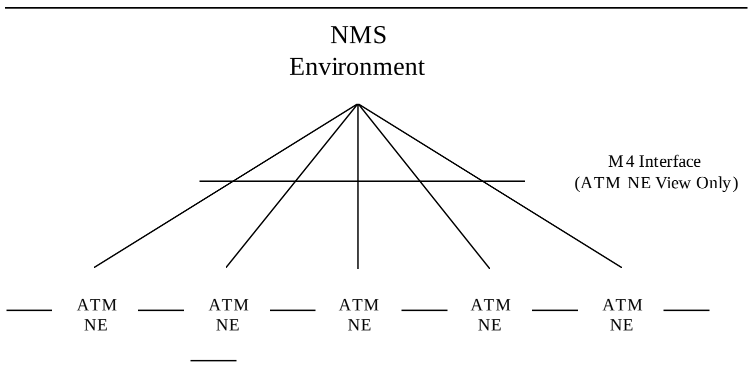


Figure 3-1-1: NE-View Management Architecture

Network-View Management Architecture Example:

In the Network-Level Management Architecture (see Figure 3-2-1), the NMS Environment interfaces to a set of subtending subnetwork Management System (SubNMS) which, in turn, interfaces to the ATM NEs within its span of control. In this architecture, the NMS Environment delegates the responsibility of managing the individual ATM NEs to the SubNMSs, and only manages the ATM subnetworks as presented by the SubNMSs. Thus the SubNMS exposes only a subnetwork (or network) view to the NMS Environment. In this architecture, the M4 Interface would apply in two locations: the first location would be between the NMS Environment and the SubNMSs, and the second location would be between the SubNMSs and the ATM NEs. Note that the M4 Interface between the NMS Environment and the SubNMSs would be required to expose an "ATM Network View" only, while the M4 Interface between the SubNMS and the ATM NEs would expose an "ATM NE View" only. The use of a Multi-Supplier subnetwork in the Network-Level Management Architecture requires alignment between the M4 ATM Network view and the M4 ATM NE view to implement the functionality requested by an NMS Environment using only the M4 ATM Network View and in turn communicate with multi-supplier ATM NEs using only the M4 ATM NE view. This requirement is imposed jointly on both the ATM Network and the ATM NE views of the M4 interface.

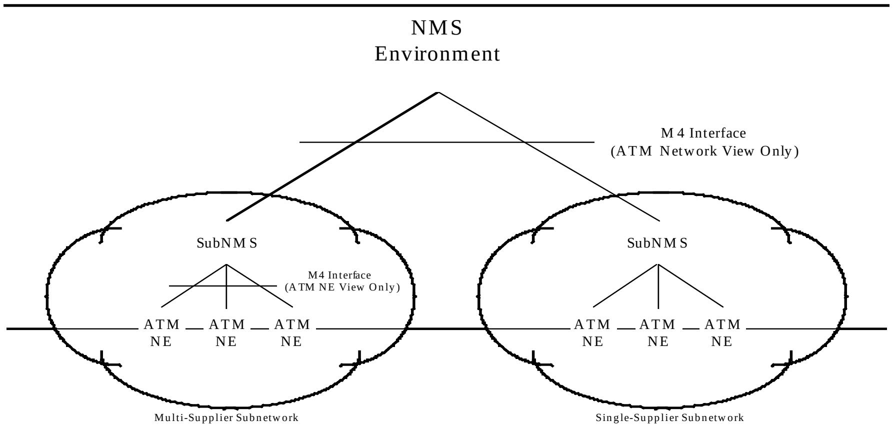


Figure 3-2-1: Example of Network-View Management Physical Configuration

NE+Network-Level Management Architecture Example:

The NE+Network-Level Management Architecture (see Figure 3-3-1) is similar to the Network-Level Management Architecture except that the M4 Interface between the NMS Environment and the SubNMSs also exposes an "ATM NE View". In this architecture, the NMS has the option to view and manage the ATM network by performing operations on the subnetwork as a whole or by performing operations on select ATM NEs. One could imagine that, for certain NMS applications, a single-entity subnetwork view would be sufficient, while for other applications a detailed view of each ATM NE comprising the subnetwork as well as their interconnections would be desirable.

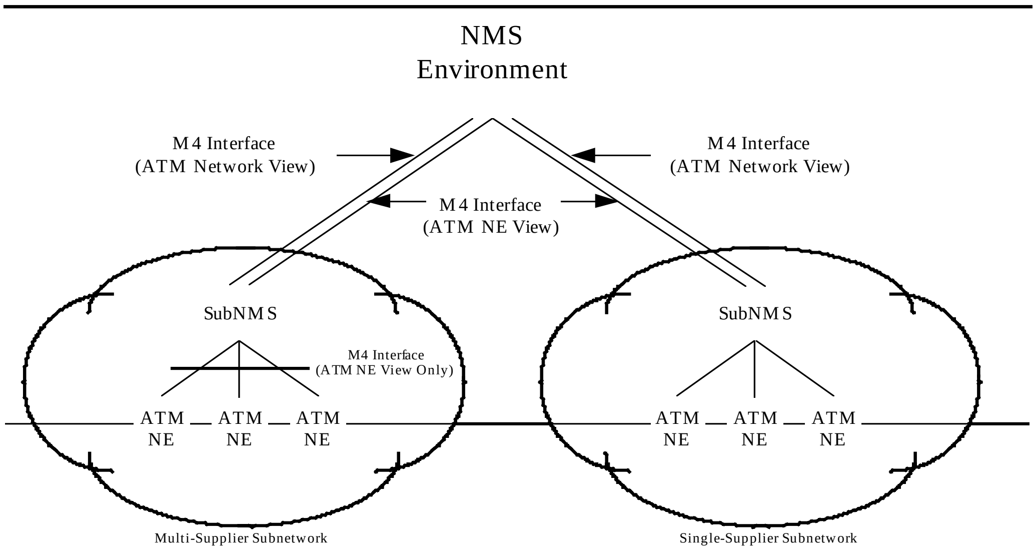


Figure 3-3-1: Example of NE+Network-Level Management Physical Configuration

Distributed Management Architecture Examples:

The management functions may be distributed across multiple subnetwork management systems. In such a case, an M4 network-view interface is needed to exchange the necessary information between the different network management systems. As a first example, a connection may span over two subnetworks, and both management systems are peers (see Figure 3-4-1). It is an evolution from the previous examples, which assumed always a hierarchical approach. To show that both peer-to-peer architectures and hierarchical architectures can be used together, in the second example of this subsection (see Figure 3-4-2), two network management systems, performing different functions (e.g. one is performing connection set-up and the other connection restoration), manage the same network resources, themselves managed by subnetwork management systems (subNMS). In this example, the two subNMSs are still peer.

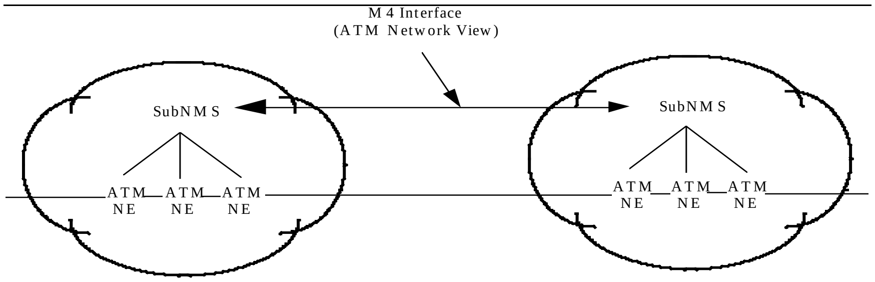


Figure 3-4-1: Example of Non-hierarchical Management Physical Configuration

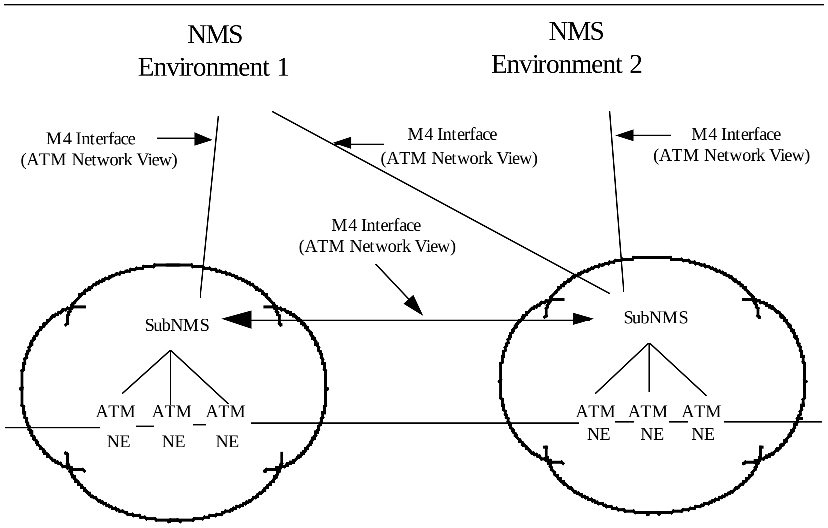


Figure 3-4-2: Example of Multi-Manager Management Physical Configuration

ATM Network Management Functions

This section provides a set of requirements to manage ATM networks. These functional requirements ask for information from the network-view, and some may require information from the NE-view . This is not an exhaustive list of functions, but it intends to be sufficient to identify an adequate set of managed entities to manage an ATM network.

The functional requirements defined in this section are intended capabilities provided by the Network Management Layer to address the different perspectives of network providers, service providers and customers, each of whom need service management and/or network management capabilities. The management systems serving these users may have peer or hierarchical relationships. Not all information needs to be provided to all users.

Each requirement in this section is prefixed by a unique identifier structured as follows:

(O/R-sect)-aa-n

- **O/R -** defines the requirement as mandatory (R) or Optional (O);
- **sect -** indicates the section number in which the requirements appears;
- **aa -** indicates the functional category of the requirement: configuration (cm), fault (fm), security (sm), performance (pm);
- **n -** provides a unique number within the section.

Transport Network Configuration Management

Transport Network Provisioning/Layered Network Provisioning

Subnetwork Provisioning

A subnetwork can be created automatically at the installation of the network or subnetwork management system which is going to manage it. Adding or removing a subnetwork is not precluded, but is not supported in this phase of the specification. In addition, subnetworks can be created by the managing system.

For each subnetwork in a layer network:

(R-4.1.1.1)-cm-1: The M4 interface shall support requests to provide information on the components of the addressed subnetwork.

(R-4.1.1.1)-cm-2: The M4 interface shall support autonomous notification of the existence or provisioning of subnetworks.

(R-4.1.1.1)-cm-3: The M4 interface shall support autonomous notifications of recent changes in the configuration of the subnetwork. This information includes relationships with other transport resources.

(R-4.1.1.1)-cm-4: The M4 interface shall support requests about the current configuration of the components of the subnetwork, which are visible over the M4 interface. This information may be necessary to understand the capabilities of the subnetwork, and for fault isolation.

(R-4.1.1.1)-cm-5: The M4 interface shall provide enough retrievable information so that a higher-level management system can derive the subnetwork topology (subnetworks or NEs, and links).

(R-4.1.1.1)-cm-6: The M4 interface shall support requests to read and write an organization-specific label for the subnetwork.

(R-4.1.1.1)-cm-7: The M4 interface shall allow a subnetwork to be created with or without termination points.

(R-4.1.1.1)-cm-8: The M4 interface shall allow a transport resource to be shared among several subnetworks.

(R-4.1.1.1)-cm-9: A subnetwork is a network resource which consists of available termination points or of groups of potential or existing termination points which are available for interconnection. The M4 network-view interface shall permit a subnetwork to be provisioned with or without an initial set of termination points. The M4 interface shall also permit the assignment and release of termination points from an existing subnetwork.

(R-4.1.1.1)-cm-11: The M4 interface shall support requests to create and delete subnetworks, except for the top subnetwork contained in the layerNetworkDomains. No subnetwork shall be deleted if it contains subnetworkConnections.

(R-4.1.1.1)-cm-12: A group of potential or existing termination points shall be represented by a Link TP (a Link End or a Logical Link TP). A Link TP may be associated with more that one subnetwork due to subnetwork partitioning.

(R-4.1.1.1)-cm-13: The M4 network view interface shall provide information identifying the Link TPs (Link Ends or Logical Link TPs) supported by a given subnetwork.

Link Provisioning

The M4 network-view supports network management requests to set-up, modify, and release subnetwork links. Following the request for a link from the management system, the network will carry out resource assignment, connection activation, modification, reconfiguration and release.

The M4 Network View model allows Links to represent unary links, composite links, and partitioned links. An ATM Link End represents the endpoint of a unary topological link. It has a one-to-one relationship with its underlying server trail. In the VP LND, a Link End represents an ATM interface associated with the underlying transport facility. In addition, interface and server trail related information may be represented in the ATM Link End. That is, the Link End may be used to represent the appropriate server trail TP information, removing the need to represent the server trail TPs across the M4 network view interface.

The ATM Logical Link TP is used to represent the endpoints of ATM Links that are composite links (formed through the aggregation of other links) or partitioned links (allocated a portion of the capacity of another link). These Logical Links are characterized by the constraining of VCI/VPI ranges, and amount of allocated bandwidth.

The example below shows a VP Link with a one-to-one relationship with a Server Trail. Note that ATM Link Ends in the VP LND may represent UNIs and NNIs. The network access profile associated with the ATM Link End describes the total bandwidth for the underlying Server Trail.

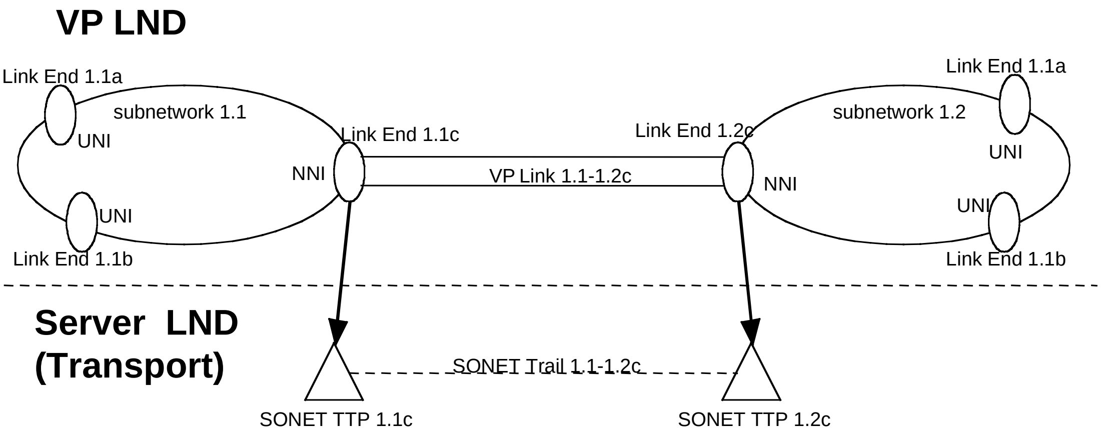


Figure 4-1 Single VP Link / Single Server Trail

The ATM Link points to two different instance of the ATM Link End. Each ATM Link End points to an entity that represents the underlying server trail termination. Information associated with the specific interface is included in the ATM Link End at the VP LND. Optionally, the ATM Link End may include server layer termination point specific information.

In the next case many VP Logical Links are supported by a single underlying server trail (and a single topological link). The ATM Link End related access profile describes the total bandwidth and entire VPI range for the trail. The trail bandwidth and VPI range are partitioned using the access profiles associated with each ATM Logical Link TP.

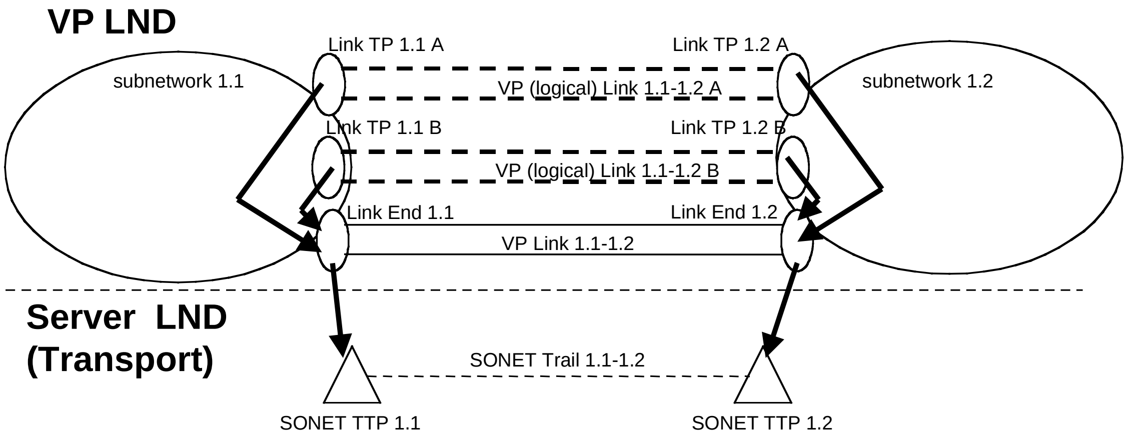


Figure 4-2 Multiple VP Links / Single Server Trail

In the following example, a single VP Link is supported by multiple underlying server trails (and multiple topological links). The VP Logical Link TP has a continuous range of VPI values, where the underlying ATM Link End related ranges are non-overlapping. Total logical link bandwidth is the sum of the topological links bandwidth.

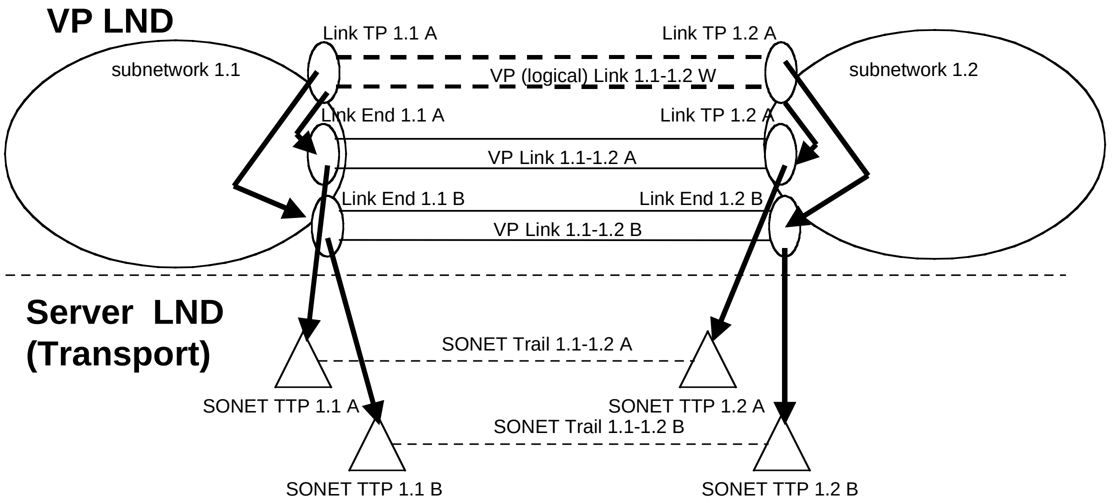


Figure 4-3 Single VP Link / Multiple Server Trails

ATM Link

The ATM Link connects two ATM Subnetworks and represents the capacity (when associated with Link End) or a portion of the capacity (when associated with Logical Link TP) of the underlying server trail. An ATM Link is terminated by two Link Termination Points (ATM Link Ends or ATM Logical Link TPs). At the VP Layer, each ATM Link End represents an ATM interface such as a UNI or NNI.

Note that multipoint links at the VC LND are for further study.

(R-4.1.1.2)-cm -1: The M4 interface shall support requests to set-up an ATM link between two VP subnetworks.
In order to establish an ATM VP link, the management system shall supply the following information:

1. the two end-points to link, specified as the identity of the supporting ATM Interface termination points, in each subnetwork
2. the link profile parameters, i.e. the bandwidth allocated to the link

(R-4.1.1.2)-cm -2: For each link that it manages, the M4 network view interface shall relate two ATM Link TPs (ATM Link Ends or ATM Logical Link TPs) with the link.

(R-4.1.1.2)-cm -3: The M4 interface shall support requests to set-up an ATM link between two VC subnetworks, where the supporting VP trail exists.
To establish an atmLink in the VC Layer Network Domain:

If the VC Link TPs (ATM Link Ends or ATM Logical Link TPs) that will support the VC Link exist, the setup link information should identify the desired VC Link TPs directly as the link endpoints.

If the VC Link is to be supported by existing VP Trail Terminations, the setup link information should identify these VP TTPs, indirectly identifying the link endpoints, along with network access profile information. The action creates the associated VC Link Ends.

Input: atmlinkTPs (ATM Link Ends or ATM Logical Link TPs) or descriptors of the endpoints: interfaceId (serverTTPId, server layer atmLinkTPId for vc Links, or atmLinkTPId), available bandwidth, and atmNetworkAccessProfilePointer or VPI range, VCI range, and maximum bandwidth

Output: Ids of new atmLink and associated atmLinkTPs (ATM Link Ends or ATM Logical Link TPs).

Errors: protocol-specific addressing errors, incorrectTerminationPoints, linkTP Connected, non-matchingRange, non-matchingBandwidth, operationFails.

Results: Creation of atmLink and, if needed, associated atmLinkTPs (ATM Link Ends or ATM Logical Link TPs), atmNetworkAccessProfiles, vpLinkConnection, vpSubnetworkConnection, vpTrail, and / or vpTTPs

(R-4.1.1.2)-cm -4: The M4 interface shall support requests to set-up an ATM link between two VC subnetworks, where the VP trail does not exist.
To establish an atmLink in the VC Layer Network Domain:
If the VC Link is to be supported by a VP connection directly between two VP subnetworks or network elements (VP Link Connection), the setup link information should indirectly identify the link endpoints using the VP Link TPs associated with the appropriate interfaces, along with network access profile information and optionally a single requested VPI value. The managed system shall then establish a VP Link Connection, creating vpTTPs or networkTTPs and the corresponding CTPs in the VP LND along with an associated atmLinkTPs (ATM Link Ends or ATM Logical Link TPs) in the VC LND.

If the VC Link will be supported by a VP connection that traverses VP subnetworks or network elements, the setup link information should identify the link endpoints indirectly using VP Link TPs (ATM Link Ends or ATM Logical Link TPs) and provide both link network access profile information along with traffic descriptor information and requested VPI endpoint values for the VP level connection. The managed system shall then establish a VP Subnetwork Connection, creating vpTTPs or networkTTPs the corresponding CTPs and atmNetworkTrafficDescriptorProfile in the VP LND along with an associated atmLinkTPs (ATM Link Ends or ATM Logical Link TPs) in the VC LND.

Input: atmlinkTPs (ATM Link Ends or ATM Logical Link TPs) or descriptors of the endpoints: interfaceId (serverTTPId, server layer atmLinkTPId for vc Links, or atmLinkTPId), available bandwidth, and atmNetworkAccessProfilePointer or VPI range, VCI range, and maximum bandwidth, and if a VP subnetwork connection is needed to support the link, traffic descriptors. Optionally, the VPI values of the endpoints of the supporting VP connection may be requested.

Output: Ids of new atmLink and associated atmLinkTPs (ATM Link Ends or ATM Logical Link TPs).

Errors: protocol-specific addressing errors, incorrectTerminationPoints, linkTP Connected, non-matchingRange, non-matchingBandwidth, operationFails.

Results: Creation of atmLink and, if needed, associated atmLinkTPs (ATM Link Ends or ATM Logical Link TPs),, atmNetworkAccessProfiles, vpLinkConnection, vpSubnetworkConnection, vpTrail, and / or vpTTPs

As shown in Appendix A, link and link connections in the client layer are supported by trails in the server layer. Note that one or more links can be configured from the underlying trail in the server network.

(R-4.1.1.2)-cm -5: The M4 interface shall support requests to modify the provisioned bandwidth of an ATM link between two subnetworks.

(R-4.1.1.2)-cm -6: The M4 interface shall support requests to release existing ATM links between component subnetworks in a ATM composite subnetwork (i.e. a subnetwork which can be decomposed in component subnetworks) and release the resources (e.g., bandwidth) assigned to the link.

(R-4.1.1.2)-cm -7 : The M4 interface shall support requests to retrieve the provisioned and the available bandwidth of a ATM link.

(R-4.1.1.2)-cm-8: The M4 interface shall support requests to configure the restoration mode of a link as: unavailable for routing and re-routing, available for routing and not re-routing; available for re-routing and not routing; or available for both routing and rerouting.

(R-4.1.1.2)-cm -9: The M4 network view interface shall support requests to modify and retrieve the weight associated with an ATM link between two subnetworks. The link weight may be set to null if undefined.

(R-4.1.1.2)-cm -10: The M4 network view interface shall support requests to modify and retrieve the customer id associated with an ATM link (private link) between two subnetworks. The customer id may be null if undefined.

(R-4.1.1.2)-cm -11: The M4 network view interface shall support requests to configure and retrieve the user label of an ATM Link. The userLabel may be used to describe additional information about the atmLink, such as a circuit identifier.

ATM Link End

(R-4.1.1.2)-cm -12: The M4 network view interface shall support requests to configure and retrieve the user label of an ATM Link End. The userLabel may be used to describe additional information about the atmLinkEnd, such as a circuit identifier.

(R-4.1.1.2)-cm -13: The M4 network view interface shall allow the administrative state of the Link End to be set {enabled, disabled}.

(R-4.1.1.2)-cm -14: The M4 network view interface shall associate an underlying server trail termination point with each Link End.

(R-4.1.1.2)-cm -15: The M4 network view interface shall support requests to configure and retrieve the interface type of an ATM Link End. The interface type may be set to: unconfigured, UNI, inter-NNI, or intra-NNI.

(O-4.1.1.2)-cm -1: The M4 network view interface should provide trail termination point specific information including but not limited to: server TTP name (user label); server TTP characteristic information type; server TTP location; and server TTP operational state.

(O-4.1.1.2)-cm –2 The M4 network view interface should provide server layer technology specific configuration information associated with the trail termination point (e.g., channalization, link coding clock source, channel number, etc.). This applies to ATM Link Ends in the VP LND.

(R-4.1.1.2)-cm-16: The M4 network view interface shall provide information describing the element and port id supporting each ATM Link End (UNIs and NNIs). This applies only to ATM Link Ends in the VP LND.

(R-4.1.1.2)-cm -17: The M4 network view interface shall support requests to configure and retrieve the loopback location identifier associated with an ATM Link End. The loopbackLocationIdentifier provides a read/write code used for OAM cell loopback purposes. Incoming OAM Loopback cells with a Loopback Location field value that matches the value of the loopbackLocationIdentifier attribute shall be looped-back over the interface. This applies only to ATM Link Ends in the VP LND.

(R-4.1.1.2)-cm -18: The M4 network view interface shall support requests to modify and retrieve a list of existing Connection Termination Points (CTPs) within the same Layer Network Domain that are supported by the ATM Link End.

(O-4.1.1.2)-cm -3: The M4 network view interface shall support requests to activate or deactivate cell scrambling on the ATM Interface that is represented by the atmLinkEnd. This applies only to ATM Link Ends in the VP LND.

(O-4.1.1.2)-cm -4: The M4 network view interface should support requests to retrieve and modify the VPI/VCI value to be used to support ILMI across the interface. This applies to only ATM Link Ends in the VP LND that have an interface type of UNI.

An external atmLinkEnd is an atmLinkEnd at the edge of the subnetwork, representing an interface (UNI or NNI), and whose atmLink is not visible to the managed system. The action provides a means to create the external atmLinkEnd, associate it with a trail termination point or include trail termination information in the atmLinkEnd, and create an associated atmNetworkAccessProfile.

(R-4.1.1.2)-cm -19: The M4 network view interface shall support requests to create an instance of the atmLinkEnd object, that represents an interface to an external network, along with the objects in the server layer network domain, if needed, that support the external atmLinkEnd. The external atmLinkEnd terminates a link that is not visible across the management interface, the link to an external network. The underlying server TTP may be identified directly (if it already exists) or indirectly, in which case the managed system may either select / create the appropriate server TTP to support the atmLinkEnd or include information about the server TTP in the atmLinkEnd. Indirect server TTP identification may indicate: interface identifier, desired total bandwidth, an atmNetworkAccessProfile identifier, or a set of descriptors providing VPI/VCI range, etc. This approach allows the possible creation of associated server layer trail terminations (or inclusion of server layer trail termination information in the atmLinkEnd) at the same time as the atmLinkEnd is created. An error condition is raised if the server trail termination point is incorrect, already used, does not have matching range or bandwidth, or if the interface is unable to provide sufficient bandwidth.

Upon creation of the atmLinkEnd, the managed system is responsible for determining and setting: availableIngressBandwidth, availableEgressBandwidth, maxAssignableIngressBandwidth, maxAssignableEgressBandwidth, atmNetworkAccessProfilePointer, and the serverTTP attributes. If a new atmNetworkAccessProfile object instance is needed, the managed system is responsible for creating it.

To create an atmLinkEnd in the VC Layer Network Domain, the information in the request may simply identify the atmLinkEndId or atmLinkTPId (ATM Link Ends or ATM Logical Link TPs) at the VP Layer as the end point. The managed system shall then create a vpTTP or networkTTP and the corresponding CTP in the VP LND along with an associated atmLinkEnd in the VC LND. This does not preclude the creation of VP Trail Terminations prior to request for creating the external atmLinkEnd. If the VP Trail Termination exists prior to the request, the requester may specify the Trail Termination as the supporting underlying server TTP(s).

Input: underlying server layer TTPs or descriptors of the endpoint: interfaceId (serverTTPId, or server layer atmLinkEndId plus connection details), user label, and atmNetworkAccessProfilePointer or VPI range, VCI range, and total bandwidth.

Output: Id of new atmLinkEnd.

Errors: incorrectTerminationPoints, serverTTP unavailable or used, non-matchingRange, non-matchingBandwidth, operationFails.

Results: Creation of atmLinkEnd and an associated atmNetworkAccessProfile, possible creation of server layer trail.

ATM Logical Link TP

(R-4.1.1.2)-cm -20: The M4 network view interface shall support requests to configure and retrieve the user label of an ATM Logical Link TP. The userLabel may be used to describe additional information about the ATM Logical Link TP, such as a circuit identifier.

(R-4.1.1.2)-cm -21: The M4 network view interface shall support requests to modify and retrieve a list of existing Connection Termination Points (CTPs) within the same Layer Network Domain that are supported by the ATM Logical Link TP.

An external atmLinkEnd is an atmLinkEnd at the edge of the subnetwork, representing an interface (UNI or NNI), and who's atmLink is not visible to the managed system. The action provides a means to create the external atmLinkEnd, associate it with a trail termination point, and create an associated atmNetworkAccessProfile.

(R-4.1.1.2)-cm -22: The M4 network view interface shall support requests to create an instance of the atmLinkEnd object, that represents an interface to an external network, along with the objects in the server layer network domain, if needed, that support the external atmLinkEnd. The external atmLinkEnd terminates a link that is not visible across the management interface, the link to an external network. The underlying server TTP may be identified directly (if it already exists) or indirectly, in which case the managed system may select or create the appropriate server TTP to support the atmLinkEnd. Indirect server TTP identification may indicate: interface identifier, desired total bandwidth, an atmNetworkAccessProfile identifier, or a set of descriptors providing VPI/VCI range, etc. This approach allows the possible creation of associated server layer trail terminations at the same time as the atmLinkEnd is created. An error condition is raised if the server trail termination point is incorrect, already used, does not have matching range or bandwidth, or if the interface is unable to provide sufficient bandwidth.

Upon creation of the atmLinkEnd, the managed system is responsible for determining and setting: availableIngressBandwidth, availableEgressBandwidth, maxAssignableIngressBandwidth, maxAssignableEgressBandwidth, atmNetworkAccessProfilePointer, and the serverTTPList attributes. If a new atmNetworkAccessProfile object instance is needed, the managed system is responsible for creating it.

To create an atmLinkEnd in the VC Layer Network Domain, the information in the request may simply identify the atmLinkEndId at the VP Layer as the end point. The managed system shall then create a vpTTP or networkTTP and the corresponding CTP in the VP LND along with an associated atmLinkEnd in the VC LND. This does not preclude the creation of VP Trail Terminations prior to request for creating the external atmLinkEnd. If the VP Trail Termination exists prior to the request, the requester may specify the Trail Termination as the supporting underlying server TTP(s).

Input: underlying server layer TTPs or descriptors of the endpoint: interfaceId (serverTTPId, or server layer atmLinkEndId plus connection details), user label, and atmNetworkAccessProfilePointer or VPI range, VCI range, and total bandwidth.

Output: Id of new atmLinkEnd.

Errors: incorrectTerminationPoints, serverTTP unavailable or used, non-matchingRange, non-matchingBandwidth, operationFails.

Results: Creation of atmLinkEnd and an associated atmNetworkAccessProfile, possible creation of server layer trail.

Subnetwork Connection Management

The M4 network-view managed entities support subnetwork management requests to set-up, reserve, release, and cancel reservation of subnetwork connections, link connections, trails, and segments.

The M4 network-view managed entities support subnetwork management requests to modify subnetwork connections, trails, and link connections.

Following the request for a network connection from the network management system, the subnetwork will carry out resource assignment, connection activation, modification, reconfiguration and release. The requirements below support that functionality.

Note that there are two implementation options for setting up permanent network connections (PVCs at VP or VC level) over the M4 interface; one is to set-up a subnetwork connection between end-points of a subnetwork, using the M4 network view, another is to build-up the connection piece-by-piece, setting-up a connection at each NE, using the M4 NE-view. This section deals with the first option because it addresses the network view. Refer to [3] for the second one. Note that it is possible to set-up a subnetwork connection, under the control of a single subnetwork manager, through a single command, creating the necessary connection termination points and the associated subnetwork connections. Depending of the network management architecture selected, either option or both may be supported.

(R-4.1.2)-cm -1 : The M4 interface shall support requests to schedule the reservation, cancellation, activation or deactivation of subnetwork connections.

(R-4.1.2)-cm -2 : The M4 interface shall support requests to retrieve the following configuration data associated with currently configured VP and VC termination points in the ATM subnetwork:

1. The ATM Interface Supporting the VP or VC
2. VPI and/or VCI Value
3. Traffic Descriptor Profile Pointer

Traffic descriptors, particularly those used to describe ABR service, can require the specification of a large number of parameters.

(CR-4.1.2) CM-3 If Constant Bit Rate service is supported, then the M4 interface should support the following parameters in the traffic descriptor profile.

1. PCR Peak Cell Rate (ingress and egress) for the CLP =0+1 traffic flow.
2. CDVT Cell Delay Variation Tolerance (ingress and egress) in relation to the PCR of the for the CLP =0+1 traffic flow.

3. CLR Maximum permissible Cell Loss Ratio (ingress and egress) for the CLP =0+1 traffic flow.

(CR-4.1.2) CM-4 If Non Real-time Variable Bit Rate service following the VBR.1 Conformance Definition is supported, then the M4 interface should support the following parameters in the traffic descriptor profile.

1. PCR Peak Cell Rate (ingress and egress) for the CLP =0+1 traffic flow.
2. SCR Sustainable Cell Rate (ingress and egress) for the CLP =0+1 traffic flow.
3. MBS Maximum Burst Size (ingress and egress) for the CLP =0+1 traffic flow.
4. CDVT Cell Delay Variation Tolerance (ingress and egress) in relation to the PCR of the for the CLP =0+1 traffic flow.
5. CLR Maximum permissible cell Loss Ratio (ingress and egress) for the CLP =0+1 traffic flow.

   Note: ITU-T Recommendation[I.371] identifies the use of a second CDVT value for SCR.

(CR-4.1.2) CM-5 If Non Real-time Variable Bit Rate service following the VBR.2 Conformance Definition is supported, then the M4 interface should support the following parameters in the traffic descriptor profile.

1. PCR Peak Cell Rate (ingress and egress) for the CLP =0+1 traffic flow.
2. SCR Sustainable Cell Rate (ingress and egress) for the CLP =0 traffic flow.
3. MBS Maximum Burst Size (ingress and egress) for the CLP =0 traffic flow.
4. CDVT Cell Delay Variation Tolerance (ingress and egress) in relation to the PCR of the for the CLP =0+1 traffic flow.
5. CLR Maximum permissible Cell Loss Ratio (ingress and egress) for the CLP =0 traffic flow

   Note: ITU-T Recommendation[I.371] identifies the use of a second CDVT value for SCR.

(CR-4.1.2) CM-6 If Non Real-time Variable Bit Rate service following the VBR.3 Conformance Definition is supported, then the M4 interface should support the following parameters in the traffic descriptor profile. Tagging option applies.

1. PCR Peak Cell Rate (ingress and egress) for the CLP =0+1 traffic flow.
2. SCR Sustainable Cell Rate (ingress and egress) for the CLP =0 traffic flow. If tagging is supported by the network, a CLP=0 cell that is not conforming to the SCR objective, but is conforming to the PCR objective, will have its CLP bit changed to 1.
3. MBS Maximum Burst Size (ingress and egress) for the CLP =0 traffic flow.
4. CDVT Cell Delay Variation Tolerance (ingress and egress) in relation to the PCR of the for the CLP =0+1 traffic flow.
5. CLR Maximum permissible Cell Loss Ratio (ingress and egress) for the CLP =0 traffic flow.

   Note: ITU-T Recommendation[I.371] identifies the use of a second CDVT value for SCR.

(CR-4.1.2) CM-7 If Real-time Variable Bit Rate service following the VBR.1 Conformance Definition is supported, then the M4 interface should support the following parameters in the traffic descriptor profile. The difference between rt-VBR compared to nrt-VBR is the QoS parameters.

1. PCR Peak Cell Rate (ingress and egress) for the CLP =0+1 traffic flow.
2. SCR Sustainable Cell Rate (ingress and egress) for the CLP =0+1 traffic flow.
3. MBS Maximum Burst Size (ingress and egress) for the CLP =0+1 traffic flow.

4. CDVT Cell Delay Variation Tolerance (ingress and egress) in relation to the PCR of the for the CLP =0+1 traffic flow.
5. CLR Maximum permissible Cell Loss Ratio (ingress and egress) for the CLP =0+1 traffic flow.

Note: ITU-T Recommendation[I.371] identifies the use of a second CDVT value for SCR.

(CR-4.1.2) CM-8 If Real-time Variable Bit Rate service following the VBR.2 Conformance Definition is supported, then the M4 interface should support the following parameters in the traffic descriptor profile. The difference between rt-VBR compared to nrt-VBR is the QoS parameters.

1. PCR Peak Cell Rate (ingress and egress) for the CLP =0+1 traffic flow.
2. SCR Sustainable Cell Rate (ingress and egress) for the CLP =0 traffic flow.
3. MBS Maximum Burst Size (ingress and egress) for the CLP =0 traffic flow.
4. CDVT Cell Delay Variation Tolerance (ingress and egress) in relation to the PCR of the for the CLP =0+1 traffic flow.
5. CLR Maximum permissible Cell Loss Ratio (ingress and egress) for the CLP =0 traffic flow.

Note: ITU-T Recommendation[I.371] identifies the use of a second CDVT value for SCR.

(CR-4.1.2) CM-9 If Real-time Variable Bit Rate service following the VBR.3 Conformance Definition is supported, then the M4 interface should support the following parameters in the traffic descriptor profile Tagging option applies.

1. PCR Peak Cell Rate (ingress and egress) for the CLP =0+1 traffic flow.
2. SCR Sustainable Cell Rate (ingress and egress) for the CLP =0 traffic flow. If tagging is supported by the network, a CLP=0 cell that is not conforming to the SCR objective, but is conforming to the PCR objective, will have its CLP bit changed to 1.
3. MBS Maximum Burst Size (ingress and egress) for the CLP =0 traffic flow.
4. CDVT Cell Delay Variation Tolerance (ingress and egress) in relation to the PCR of the for the CLP =0+1 traffic flow.
5. CLR Maximum permissible Cell Loss Ratio (ingress and egress) for the CLP =0 traffic flow.

Note: ITU-T Recommendation[I.371] identifies the use of a second CDVT value for SCR.

(CR-4.1.2) CM-10 If Unspecified Bit Rate service following the UBR.1 Conformance Definition is supported, then the M4 interface should support the following parameters in the traffic descriptor profile. UBR service does not guarantee the PCR or any QoS parameters.

1. PCR Peak Cell Rate (ingress and egress) for the CLP =0+1 traffic flow. The use of PCR for CAC, and enforcement of PCR by UPC/NPC, is network specific .
2. CDVT Cell Delay Variation Tolerance (ingress and egress) in relation to the PCR of the for the CLP =0+1 traffic flow.

(CR-4.1.2) CM-11 If Unspecified Bit Rate service following the UBR.2 Conformance Definition is supported, then the M4 interface should support the following parameters in the traffic descriptor profile. UBR service does not guarantee the PCR or any QoS parameters. Tagging option applies, the network may overwrite the CLP bit to 1 for any cell of the connection.

1. PCR Peak Cell Rate (ingress and egress) for the CLP =0+1 traffic flow. The use of PCR for CAC, and enforcement of PCR by UPC/NPC, is network specific.
2. CDVT Cell Delay Variation Tolerance (ingress and egress) in relation to the PCR of the for the CLP =0+1 traffic flow.

If the NE supports Available Bit Rate service, then the M4 interface should support the traffic service related requirements to support enhanced functions of Traffic Management 4.0, in the context of the M4 Network View interface. Specifically the following detailed requirements shall be supported and the objective **(O-4.1.2) CM-1** is desirable:

(CR-4.1.2) CM-12 If Available Bit Rate service is supported, then the M4 interface should support the following parameters in the traffic descriptor profile:

1. PCR Peak Cell Rate (ingress and egress) for the CLP =0 traffic flow.
2. MCR Minimum Cell Rate (ingress and egress) for the CLP =0 traffic flow.
3. CDVT Cell Delay Variation Tolerance (ingress and egress) in relation to the PCR of the for the CLP =0 traffic flow.
4. ICR Initial Cell Rate (ingress and egress). Rate at which a source should send initially and after an idle period. The unit is an integer number of cells/second. The value must not exceed PCR, and is usually lower.
5. TBE Transient Buffer Exposure (ingress and egress). The number of cells that the network would like to limit the source to sending during startup periods, before the first RM cell returns.
6. FRTT Fixed Round-Trip Time. The sum of the fixed and propagation delays from the source to the destination and back.
7. RDF Rate Decrease Factor (ingress and egress). Controls the rate decrease which occurs when backward RM cells with CI =1, are received. Allowed values are in {1/32768, 1/16384, 1/8192, 1/4096, 1/2048, 1/1024, 1/512, 1/ 256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1}
8. RIF Rate Increment Factor (ingress and egress). Controls the rate at which the rate increases, when a backward RM cell is received with CI =0 and NI =0. Allowed values are in {1/32768, 1/16384, 1/8192, 1/4096, 1/2048, 1/1024, 1/512, 1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1}

(O-4.1.2) CM-1 If Available Bit Rate service is supported, then it is desirable that the M4 interface supports the following parameters in the traffic descriptor profile:

1. Nrm Number RM (ingress and egress). The maximum number of data cells a source may send for each forward RM cell. Allowed values are in {2, 4, 8, 16, 32, 64, 128, 256}.
2. Trm Time RM (ingress and egress). Upper bound on the time between forward RM cells for an active source. Allowed values are computed as 100 * 2^(-k), where 0 <= k <= 7. The resulting range is from 0.78125 ms to 100 ms, with the default being 100 ms.
3. CDF Cutoff Decrease Factor (ingress and egress). Controls the rate decrease associated with lost or delayed backward RM cells). Allowed values are in {0, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1}
4. ADTF ACR Decrease Time Factor (ingress and egress). Time permitted between sending RM cells, before the rate is decreased to ICR. Range is 10 ms to 10.23 seconds, in increments of 10 ms.

(O-4.1.2) CM-2 It is desirable that all traffic descriptor profiles should support the following parameter.

1. Profile Name A user defined name for the managed entity instance.

Connection Set-Up

Subnetwork Connection

All the requirements below apply to the M4 network-view managed entities. The subnetwork connection set-up involves a requester (e.g. the network management system), a provider (e.g. the subnetwork connection management service in a subnetwork management system), ports (here the VP or VC endpoints in the subnetwork), and the subnetwork for which the subnetwork management system is defined. The M4 network-view managed entities shall support point-to-point connections, and all types of multipoint connections. However, this does not imply that an actual implementation shall support all types of multipoint connections. As an example, in initial implementations, multicast connections are seen to have the highest priority. The protocol-specific MIBs derived from these requirements shall support profiling of the multipoint connection capability.

Point-to-point connection set-up:

(R-4.1.2.1.1)-cm -1 : The M4 interface shall support requests to set-up a VP/VC subnetwork connection within a network from any end-point to any end-point in a subnetwork. Provided with each subnetwork connection request, shall be the following information:

1. Each end-point to connect, specified as (a) or (c) for VP connection, and (b) or (d) for VC connection:
   a. the VPI value of a VP termination within a specific ATM Interface
   b. the VCI value of a VC termination within a specific VPC
   c. the identity of the supporting ATM interface termination point
   (the subnetwork connection management service selects the VPI value within the ATM Interface)
   d. the identity of the supporting VPC termination point
   (the subnetwork connection management service selects the VCI value within the VPC)
2. the identity of the VP connection
3. or the identity of the VC connection
4. Ingress and Egress Peak Cell Rate for CLP=0 and CLP=0+1 Traffic
5. Ingress and Egress Sustained Cell Rate for CLP=0 and CLP=0+1 Traffic
6. Ingress and Egress Maximum Burst Tolerance for CLP=0 and CLP=0+1 Traffic
7. Ingress and Egress Implicit CDV Tolerance for CLP=0 and CLP=0+1 Traffic
8. Ingress and Egress QOS class

(R-4.1.2.1.1)-cm -2 : For the case of point-to-point subnetwork connection establishment, the M4 interface shall supply the following information:

1. a unique connection identifier. The M4 interface shall permit the provider to inform the requester that the connection identifier is unique in the context of the provider and requester for the duration of the connection.
2. the endpoints of the connection.

(R-4.1.2.1.1)-cm -3 : The M4 interface shall provide the reason for rejection, in case a request is rejected.

(R-4.1.2.1.1)-cm -4 : The M4 interface shall specifically indicate if the request was rejected because of the VC or VP endpoints were already in use.

(R-4.1.2.1.1)-cm -5 : The M4 interface shall support requests to retrieve the list of active (PVC) subnetwork connections of a subnetwork and the state of the subnetwork connections.

(R-4.1.2.1.1)-cm -6: The M4 interface shall support requests to group subnetwork connections according to user defined criteria. Connection may be grouped according to criteria other than their source or destination [2].

(O-4.1.2.1.1)-cm -1: The M4 interface should support requests to reserve, set-up or release connection groups.

(R-4.1.2.1.1)-cm -7: The M4 interface shall support requests to set up or modify an ATM connection (VP or VC) between any two points in a network, whether they are under direct control of the Network Management System or indirect control via a subnetwork management system or an Element Management System.

(R-4.1.2.1.1)-cm -8: The M4 interface shall support requests to assign and retrieve an administrative field which provides a user identifier/label (master connection name) for each connection built across the network. This identifier/label provides an internal carrier-specific administrative handle (circuit ID) for human use. Under these conditions, the M4 interface shall allow the requester and provider to use the user identifier/label as the unique subnetwork connection identifier when communicating. The specific administration may leave the field blank or follow administration-specific rules in naming the connection.

(O-4.1.2.1.1)-cm -2: The M4 interface should support requests to assign and retrieve an administrative field which provides a master connection name for each connection end-point at the edge of the network. This field identifies the administrative name used by the adjacent carrier.

(R-4.1.2.1.1)-cm -9: The M4 interface shall be able to support an administrative field which identifies ownership of a particular connection. This ownership field can be used for administration specific use such as customer, organization , department or people names. This field is useful for associating multiple trails or connections for a particular organization.

(O-4.1.2.1.1)-cm -3 : The M4 interface should support requests to configure specific VP or VC subnetwork connections as "recoverable" or "not-recoverable."

(R-4.1.2.1.1)-cm -10: The M4 interface shall support requests to configure specific VP or VC subnetwork connections as "restorable" or "not-restorable."

(R-4.1.2.1.1)-cm -11: The M4 interface shall support requests to lock and unlock the switching of ATM cells through a point-to-point subnetwork connection. Locking a connection serves to inhibit ATM cell flow while continuing to maintain the subnetwork connection established between the two VP or VC connection termination points.

(R-4.1.2.1.1)-cm -12: The M4 interface shall support requests to individually inhibit/allow (lock/unlock) the flow of ATM cells to and from each VP or VC termination of a multipoint subnetwork connection.

[2] Example of the user criteria which may be used are: route, timing of connection establishment, timing of connection release, and usage of a particular link. The connection management service is only responsible to provide the capability to identify connections as members of a group, but it is not required to process the grouping criteria. Note also that grouping is for further study, and is not addressed in the protocol-independent MIB.

(R-4.1.2.1.1)-cm -13: The M4 interface shall support requests to retrieve the subnetwork connection route, e.g. the identifiers of the set of network elements and interfaces that an individual subnetwork connection uses across the network, or of the set of subnetwork connections and link connections making up the subnetwork connection.

(O-4.1.2.1.1)-cm -4 : The M4 interface should support requests to specify, in the set-up request, route selection criteria, such as preferred route, or physical separation of routes (e.g. end-to-end based, subnetwork-based, connection-based).

Multipoint Connection Set-up:

(R-4.1.2.1.1)-cm -14: The M4 interface shall support requests to establish multipoint VP and VC subnetwork connections from endpoints to endpoints in a subnetwork. Provided with each multipoint subnetwork connection request shall be the following information:

1. Multipoint Connection Type (multicast, merge, multicast/merge, full multipoint)
2. Primary VP or VC Termination Point
   For broadcast, merge, and multicast/merge connection types, this parameter identifies the VP or VC termination point that generates traffic to be broadcasted and/or receives traffic that has been merged. This parameter shall be set to NULL for full multipoint connection types. The same options as for the point-to-point connection shall be available.
3. Traffic Descriptors and QOS Class of the Primary Termination Point (if one exists). The same options as for the point-to-point connection shall be available.
4. Common VP or VC Termination Points
   This parameter identifies all VP or VC termination points involved in the multipoint connection except the primary VP or VC termination point. The same options as for the point-to-point connection shall be available.
5. Traffic Descriptors and QOS Class for each Common Termination Point. The same options as for the point-to-point connection shall be available.

(R-4.1.2.1.1)-cm -15: The request shall be considered as successful by the provider if at least one of the legs of a multipoint connection is established.

(R-4.1.2.1.1)-cm -16: The provider shall assure that a leg of a multipoint connection cannot be set-up if the endpoint of the potential leg is already in use by another subnetwork connection.

(R-4.1.2.1.1)-cm -17: For the case of multipoint subnetwork connection establishment, the M4 interface shall supply the following information:

1. a unique multipoint connection identifier
2. the list of the endpoints of the connection which were set-up
3. a unique identifier for each leg established
4. a list of the endpoints which were not setup, if any

Link Connection Set-Up

All the requirement below apply to the M4 network-view managed entities. The link connection set-up involves a requester (e.g. a network management system), a provider (e.g. the link connection management service in a subnetwork management system), ports (here the VP or VC endpoints in the subnetwork), and the subnetwork for which the subnetwork management system is defined. Note that a point-to-multipoint link connection cannot be set-up and that no link may be set-up within a subnetwork which cannot be decomposed.

(R-4.1.2.1.2)-cm -1 : The M4 interface shall support requests to set-up a VP/VC link connection between two subnetworks over an ATM link. Provided with each link connection request, shall be the following information:

1. Each end-point to connect, specified as (a) or (c) or (e) or (f) for VP connection, and (b) or (d) or (e) or (f) for VC connection:
   a. the VPI value of a VP termination within a specific ATM Interface, which shall be the same for each end
   b. the VCI value of a VC termination within a specific VPC, which shall be the same for both ends
   c. the identity of the supporting ATM Interface termination point with each subnetwork (the link connection management service selects the VPI value within the ATM Interface)
   d. the identity of the supporting VPC termination point within each subnetwork (the link connection management service selects the VPI value within the ATM Interface)
   e. the ATM link. In this case, the link connection service selects the VPI or VCI value within the ATM interface.
   f. the identity of the subnetworks to be connected. In this case, the link connection management service will select which ever ATM link satisfies the request
2. Ingress and Egress Peak Cell Rate for CLP=0 and CLP=0+1 Traffic
3. Ingress and Egress Sustained Cell Rate for CLP=0 and CLP=0+1 Traffic
4. Ingress and Egress Maximum Burst Tolerance for CLP=0 and CLP=0+1 Traffic
5. Ingress and Egress Implicit CDV Tolerance for CLP=0 and CLP=0+1 Traffic
6. Ingress and Egress QOS class

(R-4.1.2.1.2)-cm -2 : For the case of link connection establishment, the M4 interface shall supply the following information:

1. a unique link connection identifier
2. the endpoints of the link connection which were set-up
3. the link that contain the connection

(R-4.1.2.1.2)-cm -3 : The M4 interface shall provide the reason for rejection, in case a request is rejected.

(R-4.1.2.1.2)-cm -4 : The M4 interface shall specifically indicate if the request was rejected because of the VC or VP endpoints was already in use.

(O-4.1.2.1.2)-cm -1 : The M4 interface should support requests to reserve, set-up or release connection groups.

(R-4.1.2.1.2)-cm -5 : The M4 interface shall support requests to assign and retrieve a link connection label for each VP-level or VC-level link connection. This link connection label provides a linking mechanism to higher level service management functions. This label shall be alphanumeric with the specific naming and formatting scheme left up to the individual network operator.

(R-4.1.2.1.2)-cm -6 : The M4 interface shall support requests to assign and retrieve an external link connection label for each VP-level or VC-level link connection. This is useful when a customer link connection, either at the VP- or at the VC-level, traverses multiple carrier networks and one carrier needs to know how the next carrier identifies the connection. This label shall exist for each end of the link connection.

(R-4.1.2.1.2)-cm -7 : The M4 interface shall support requests to assign and retrieve a customer label for each VP-level or VC-level link connection. This link connection label allows the network operator to quickly know which customer(s) is affected by a change in the link connection. This customer label provides a linking mechanism to higher-level service management functions. There should be a customer for each network end-point of a connection. In most cases, the customer label will be the same for all termination points of a connection. However, there are cases where a connection connects two different customers.

(R-4.1.2.1.2)-cm -8 : The M4 interface shall support requests to lock and unlock the switching of ATM cells through a link connection. Locking an ATM connection serves to inhibit ATM cell flow while continuing to maintain the link connection established between the two VP or VC connection termination points.

(R-4.1.2.1.2)-cm -9 : The M4 network-view managed entities supporting the layered architecture described in Section 2, shall support the identification of a trail (or its corresponding managed entities) in a server layer that support a given link connection in the client layer.

Segment Set-Up

All the requirement below apply to the M4 network-view managed entities. The segment set-up involves a requester (e.g. the network management system), a provider (e.g. the segment management service in a subnetwork management system), ports (here the VP or VC endpoints in the subnetwork), and a subnetwork.

(R-4.1.2.1.3)-cm -1: The M4 interface shall support requests to configure and reconfigure active VP and VC termination points at the edges of a subnetwork as a segment end-point. This requirement may be fulfilled by using the NE-view managed entities.

(R-4.1.2.1.3)-cm -2: The M4 interface shall support requests to retrieve the data stored in the ATM subnetwork that identifies whether a particular VP or VC termination point has or has not been configured as a segment end-point. This requirement may be fulfilled by using the NE-view managed entities.

(R-4.1.2.1.3)-cm -3 : The M4 interface shall provide the reason for rejection, in case a request is rejected.

Trail Set-Up:

The M4 interface shall support requests for ATM-trails by making associations between the trail terminations. The Trail Setup request shall be decomposed into subnetwork connections and link connections by the server NMS. The creation of the components of the trail may be done as a part of trail set up or they may have been created before. When a part of the ATM-trail spans across administrative domains, the NMS federates this part of trail to the neighboring domain as a subnetwork connection request. The Trail Setup request shall contain the same information as for the subnetwork connection request.

(R-4.1.2.1.4)-cm -2: The M4 interface shall support requests to configure and reconfigure VP and VC connection termination points as ATM-trail termination points and to associate trail termination functions to them.

(R-4.1.2.1.4)-cm -3: The M4 interface shall supply the following information
after a successful trail establishment:

1. a unique trail identifier
2. the endpoints of the trail, if not assigned but the user

(R-4.1.2.1.4)-cm -4: The M4 interface shall support requests to lock and unlock the transport of ATM cells through the trail.

(O-4.1.2.1.4)-cm-1: The M4 interface shall support capability to defer the set-up of a trail.

The ATM Forum UNI Specification Version 4.0 defines an AAL Parameters Information and B-ICI Version 2.0 defines an *AAL Parameters Parameter*. This *information element* is used during the SVC set-up process to negotiate AAL-specific parameters that characterize the desired AAL processing to be performed at both ends of the SVC. While the ATM Forum UNI Specification only addresses the configuration of AAL parameters from an SVC perspective, the need to configure AAL parameters would seem to apply to some, network-terminated PVCs as well.

(R-4.1.2.1.4)-cm -5: The M4 network view interface shall support requests to configure and reconfigure the AAL Type for a given VC Trail termination point. Valid values are: AAL Type 1, Type 2(for future use), Type 3/4, and Type 5.

(R-4.1.2.1.4)-cm -6: For trail connections supporting AAL Type 1, the following information shall be configurable across the M4 network view interface:

1. The AAL type 1 subtype used by the CBR service application (e.g. 64 kbps voiceband signal transport, circuit transport);
2. The CBR service application rate (e.g. 64 kbps, 1544 kbps, 44736 kbps, n x 64 kbps)
3. The source clock frequency recovery method in use
   - Synchronous, or
   - Asynchronous - Synchronous Residual Time Stamp (SRTS), or
   - Asynchronous - Adaptive Clock.
4. Structured Data Transfer (SDT) – True or False
5. Partial Cell Payload Fill (0 < Payload Fill < 47 octets of AAL user information) (optional)
6. Forward Error Correction (FEC) Type (optional)
   - No forward error correction
   - FEC for Loss Sensitive Signal Transport (i.e., (128,124) Reed-Solomon encoded long interleaver)
   - FEC for Delay Sensitive Signal Transport (i.e., (94, 88) Reed-Solomon encoded short interleaver).
7. Cell Loss Integration Period (in milliseconds)

(R-4.1.2.1.4)-cm -7: For trail connections supporting AAL Type 3/4, the following information shall be configurable across the M4 network view interface:

1. Message Identifier (MID) field range (i.e. the lowest and highest MID field values);
2. Maximum CPCS_SDU Size (i.e. upper bound on the AAL type 3/4 CPCS_SDU size in octets where the SDU size range is $0 \leq SDU \leq 2^{16} - 1$ octets)
   - for the forward direction (i.e., the calling user to called user direction for SVC service) and
   - for the backward direction (i.e., called user to calling user direction for SVC service);
3. Mode of operation (e.g. Message or Streaming, Assured or Unassured);
4. Identify AAL type 3/4 convergence sublayer protocol options applicable to the specific service application supported by the AAL (i.e., AAL type 3/4 SSCS).

(R-4.1.2.1.4)-cm -8: For trail connections supporting AAL Type 5, the following information shall be configurable across the M4 network view interface.

1. Maximum CPCS_SDU Size (i.e., upper bound on the AAL type 5 CPCS_SDU size in octets where the SDU size range is $0 \leq SDU \leq 2^{16} - 1$ octets)
   - for the forward direction (i.e., the calling user to called user direction for SVC service) and
   - for the backward direction (i.e., called user to calling user direction for SVC service);
2. Mode of operation (e.g., Message or Streaming, Assured or Unassured);
3. Identify AAL type 5 convergence sublayer protocol options applicable to the specific service application supported by the AAL (i.e., AAL type 5 SSCS).

Connection Modification

(R-4.1.2.2)-cm -1 : The M4 interface shall support requests to modify the reservation, cancellation, activation or deactivation schedule.

Subnetwork Connection:

(R-4.1.2.2)-cm -2 : The M4 interface shall support requests to modify a VP/VC subnetwork connection within a subnetwork. Provided with each subnetwork connection modification request, shall be the following information:

1. The identity of each connection to modify, specified as (a) or (c) for VP connection, and (b) or (d) for VC connection:
   a) the VPI value of a VP termination within a specific ATM Interface for one end
   b) the VCI value of a VC termination within a specific VPC for one end
   c) the identity of the VP connection
   d) the identity of the VC connection
2. The parameters to modify:
   a) Ingress and Egress Peak Cell Rate for CLP=0 and CLP=0+1 Traffic
   b) Ingress and Egress Sustained Cell Rate for CLP=0 and CLP=0+1 Traffic
   c) Ingress and Egress Maximum Burst Tolerance for CLP=0 and CLP=0+1 Traffic
   d) Ingress and Egress Implicit CDV Tolerance for CLP=0 and CLP=0+1 Traffic
   e) Ingress and Egress QOS class

(R-4.1.2.2)-cm -3 : The M4 interface shall provide the reason for rejection, if the request is rejected.

(R-4.1.2.2)-cm -4 : The M4 interface shall support requests to add/remove VP or VC terminations/leg of a multipoint subnetwork connection.

(R-4.1.2.2)-cm -5 : The subnetwork connect management service shall consider the request to be successful if at least one of the legs is added or removed.

(R-4.1.2.2)-cm -6 : The subnetwork connection management service shall assure that a leg of a multipoint connection cannot be added if the endpoint of a potential leg is in use.

(R-4.1.2.2)-cm -7 : In case of multipoint connection modification, the M4 interface shall supply the following information:
1. a unique connection identifier
2. the list of the endpoints of the connection which were added or removed
3. an identifier for each leg which was added or removed
4. a list of the endpoints which were not added or removed, if any

Link Connection:

(R-4.1.2.2)-cm -8 : The M4 interface shall support requests to modify a VP/VC link connection. Provided with each link connection modification request, shall be the following information:
1. The identity of each link connection to modify, specified as (a) or (c) for VP connection, and (b) or (d) for VC connection:
   a) the VPI value of a VP termination within a specific ATM Interface for one end
   b) the VCI value of a VC termination within a specific VPC for one end
   c) the identity of the VP link connection
   d) the identity of the VC connection
2. The parameters to modify:
   a) Ingress and Egress Peak Cell Rate for CLP=0 and CLP=0+1 Traffic
   b) Ingress and Egress Sustained Cell Rate for CLP=0 and CLP=0+1 Traffic
   c) Ingress and Egress Maximum Burst Tolerance for CLP=0 and CLP=0+1 Traffic
   d) Ingress and Egress Implicit CDV Tolerance for CLP=0 and CLP=0+1 Traffic
   e) Ingress and Egress QOS class

(R-4.1.2.2)-cm -9 : The M4 interface shall provide the reason for rejection, in case a request is rejected.

Trail:

(R-4.1.2.2)-cm -10: The M4 interface shall support requests to modify the trail within the layer network with the same functionality as for subnetwork connections, i.e. modification of reservation, activation and deactivation schedule, bandwidth and end-points.

Connection Release

subnetwork Connections:

(R-4.1.2.3)-cm -1 : The M4 interface shall support requests to release existing subnetwork connections and release the resources (e.g., bandwidth) assigned to the individual subnetwork connection at any time. The connection shall be explicitly identified.

(R-4.1.2.3)-cm -2 : The M4 interface shall be support requests to release all existing subnetwork connections and release the resources (e.g., bandwidth) assigned to a multipoint subnetwork connection. The connection shall be explicitly identified.

(R-4.1.2.3)-cm -3 : In case of connection release, the M4 interface shall supply the identity of the connection released.

(R-4.1.2.3)-cm -4 : The M4 interface shall provide the reason for rejection, in case a release request is rejected.

Link Connections:

(R-4.1.2.3)-cm -5 : The M4 interface shall be support requests to release existing link connections and release the resources (e.g., bandwidth) assigned to the individual link connection.

(R-4.1.2.3)-cm -6 : For the case of connection release, the M4 interface shall supply the identity of the connection released.

(R-4.1.2.3)-cm -7 : The M4 interface shall provide the reason for rejection, in case a release request is rejected.

Segment:

(R-4.1.2.3)-cm -8: The M4 interface shall support requests to configure active VP and VC termination points as a non-segment end-points, and by so, release it. This requirement may be fulfilled by using the NE-view managed entities.

(R-4.1.2.3)-cm -9 : For the case segment release, the M4 interface shall give the identity of the connection released.

(R-4.1.2.3)-cm -10 : The M4 interface shall provide the reason for rejection, in case a release request is rejected.

Trails:

(R-4.1.2.3)-cm -11: The M4 interface shall support requests to release trails by breaking the associations between the trail terminations. This request shall be decomposed into subnetwork connection releases and link connections releases of those connections that were used by the trail.

(O-4.1.2.3)-cm-1: When a subnetwork connection, a link connection or a trail is released, the M4 interface should allow the underlying connectivity resources (e.g. link connection and subnetwork connection) to remain.

(O-4.1.2.3)-cm-2: If the capability to retain an underlying resource is supported, the M4 interface should support requests to tag the underlying connectivity resources as “remaining” or not.

ATM Link and Link TP PVC Trace

The linkPVCTrace Action allows a managing system to request the managed system to track down the connections that are using (or dependent on) a specific atmLink or terminate on a specific atmLinkTP (atmLinkEnd or atmLogicalLinkTP). This allows the managing system to determine the connections that are impacted by degradation or potential outage of the link. It also provides a means for the managing system to verify information.

(O-4.1.2.4)-cm-1: The M4 network view interface should support requests to perform a link PVC trace on a specific atmLink. The request should identify the atmSubnetworks on which the managed system will determine the atmSubnetworkConnections that traverse the atmLink. The request can identify multiple atmSubnetworks to scope the trace. If the request is performed on an atmLink in the VP Layer Network Domain, it can identify atmSubnetworks in both the VP and VC Layer Network Domain in order to identify the VP and VC subnetwork connections that traverse the VP Link.

The identified subnetworks and their connections must be visible to the managed system. For example, if the request is performed on an atmLink in the VP layer network domain and the request identifies an atmSubnetwork in the VC layer network domain, both the VP atmLink and the VC atmSubnetwork need to be under the purview of the managed system. In this case, the result of the action would identify the VC atmSubnetworkConnections of the identified VC atmSubnetwork that traverse the VP atmLink.

Input: List of atmSubnetworks used to scope the trace of atmSubnetworkConnections supported by the atmLink.

Output: Identifier of each atmSubnetwork along with the Ids of the atmSubnetworkConnections within each atmSubnetwork that traverse the atmLink. The results are to be ordered.

Errors: noSuchResource, if the identified atmSubnetwork does not exist.
operationFails, if atmSubnetwork or atmLink cannot be traced by the managed system.

Results: No objects created, modified, or deleted. This is a query request.

(O-4.1.2.4)-cm-2: The M4 network view interface should support requests to perform a link PVC trace on a specific atmLinkTP (atmLinkEnd or atmLogicalLinkTP). The request should identify the atmSubnetworks on which the managed system will determine the atmSubnetworkConnections that terminate at the atmLinkTP (atmLinkEnd or atmLogicalLinkTP). The request can identify multiple atmSubnetworks to scope the trace. If the request is performed on an atmLinkTP (atmLinkEnd or atmLogicalLinkTP) in the VP Layer Network Domain, it can identify atmSubnetworks in both the VP and VC Layer Network Domain in order to identify the VP and VC subnetwork connections that terminate at the VP LinkTP (atmLinkEnd or atmLogicalLinkTP).

The identified subnetworks and their connections must be visible to the managed system. For example, if the request is performed on an atmLinkTP (atmLinkEnd or atmLogicalLinkTP) in the VP layer network domain and the request identifies an atmSubnetwork in the VC layer network domain, both the VP atmLinkTP (atmLinkEnd or atmLogicalLinkTP) and the VC atmSubnetwork need to be under the purview of the managed system. In this case, the result of the action would identify the VC atmSubnetworkConnections of the identified VC atmSubnetwork that terminate the VP atmLinkTP (atmLinkEnd or atmLogicalLinkTP).

Input: List of atmSubnetworks used to scope the trace of atmSubnetworkConnections supported by the atmLinkTP (atmLinkEnd or atmLogicalLinkTP).

Output: Identifier of each atmSubnetwork along with the Ids of the atmSubnetworkConnections within each atmSubnetwork that terminate at the atmLinkTP (atmLinkEnd or atmLogicalLinkTP). The results are to be ordered.

Errors: noSuchResource, if the identified atmSubnetwork does not exist.
operationFails, if atmSubnetwork or atmLinkTP (atmLinkEnd or atmLogicalLinkTP) cannot be traced by the managed system.

Results: No objects created, modified, or deleted. This is a query request.

ATM Connection Trace

The counterpart of the linkPVCTrace Action is the connectionTrace Action. Using the connectionTrace Action on an atmSubnetworkConnection or atmTrail, the managing system may request that the managed system provide, at the lowest possible level of subnetwork partitioning, the path of the connection including virtual id values at interfaces. This action allows the managing system to verify the path for a specific connection. Also, some managed systems may choose to represent subnetwork connections at only at upper levels of subnetwork partitioning. This action allows such managed systems to reveal the connection details that may not normally be available across the interface. This information can be used in trouble resolution or to synchronize data.

(O-4.1.2.5)-cm-1: The M4 network view interface should support requests to perform a connection trace on a specific atmSubnetworkConnection or atmTrail. This request should cause the managed system to determine the path of the connection and return the path at the lowest possible level supported by the managed system. The connection trace should return the virtual id (VPI for VP LND connections or VPI/VCI for VC LND connections) for each atmLink or external interface point (atmLinkEnd or atmLogicalLinkTP) of the connection. For VC connections, the trace should be examined at both the VC level as well as the VP Level, if both LNDs are under the purview of the managed system. In cases of multipoint connection, the results should be returned ordered in a breadth first fashion

Input: No additional input beyond request.

Output: Sequence of atmLinks or external interface point (atmLinkEnd or atmLogicalLinkTP) , server TTP Id, and associated virtual id (VPI/VCI) values. In cases of VC connection traces where VP Level information is available, the VC Link should be expanded to include the VP Level (server level) trace information.

Errors: noSuchConnection, if the trace is not able to be performed.

Results: No objects created, modified, or deleted. This is a query request.

Subnetwork State Management

All the requirement below applies to the M4 network-view managed entities.

(R-4.1.3)-sm-1 : The M4 interface shall autonomously be notified of subnetwork connection operational state changes. Notifications shall indicate if the subnetwork connection is capable of performing its intended function.

(R-4.1.3)-sm-2 : The M4 interface shall be autonomously be notified of subnetwork connection protection switches, if any.

(R-4.1.3)-sm-3 : The M4 interface shall support requests to retrieve subnetwork connection availability (monitored or not-monitored, in-test, failed, dependency [another entity needed to perform its function is not active]).

(O-4.1.3)-sm-1: The M4 interface should be capable of autonomously notifying
trail operational state changes.

(R-4.1.3)-cm -5: The M4 interface shall be support requests to suspend autonomous reporting of subnetwork connections.

(R-4.1.3)-cm -6: The M4 interface shall support autonomous notifications of changes in the operational state of any subnetwork components that are visible across the M4 interface.

Transport Network Fault Management

(R-4.2)-fm-1: The M4 interface shall support the capability to record network-view alarms within a subnetwork shall be logged so that they can be retrieved by the management system.

Note also that the M4 NE-view fault reporting and filtering requirements apply here too.

Network Equipment Fault Correlation/Localization/Notification

(R-4.2.1)-fm-1 : The M4 interface shall autonomously be notified of failures detected within the ATM subnetwork, such as Termination Point failures.

Connection Testing

Loopback Test

The requirement below applies to the M4 network-view managed entities.

(R-4.2.2)-fm-1: The M4 interface shall support requests to perform an OAM cell loopback along a subnetwork connection or a segment of it, and to report back the results (i.e., pass or fail). OAM Cell Loopback tests are performed by inserting a Loopback OAM cell, with loopback location field set as specified by the management system, into the cell stream of the VPC/VCC connection or connection segment under test and verifying its return. The following information shall be supplied with each management system request to perform an OAM cell loopback test:

- The Identity of the Loopback OAM Cell Insertion Point
- The direction of the loopback

Transport Network Performance Management

Performance management provides functions to evaluate and report upon the effectiveness of the telecommunications network. Its role is to gather statistical data for the purpose of monitoring and correcting the behavior and effectiveness of the network and to aid in planning and analysis.

ATM Cell Level Protocol Monitoring

Cell Level protocol monitoring involves collecting and thresholding data counts that measure an ATM Subnetwork's ability to successfully process and deliver incoming ATM cells. Cell Level protocol monitoring is particularly concerned with protocol abnormalities detected at the Transmission Convergence Sublayer and ATM Layer of the Broadband protocol stack. Cell Level protocol monitoring also entails logging detailed information (in the ATM Subnetwork Manager) that may be retrieved and used by a management system to diagnose cell processing malfunctions.

(R-4.3.1)-pm-1: The M4 network view interface shall provide management systems the ability to retrieve current and history(15-minute) counts of the following data from each Atomic ATM VP Subnetwork (representing an NE), ATM VP Link End, and selected ATM VP and VC Network CTP within the domain managed. There will be separate counters for each entity and instance.

1. Cells Received
2. Cells Transmitted

(R-4.3.1)-pm-2: The M4 network view interface shall provide management systems the ability to retrieve current and history(15-minute) counts of the following data from each ATM VP Link End within the domain managed. There will be separate counters for each instance.

1. Discarded Cells due to Protocol Errors
   This parameter provides a count of the number of ATM cells discarded due to an unrecognizable cell header field value (e.g., unassigned VPI/VCI value, out-of-range VPI/VCI value, or invalid Payload Type Identification value).

2. Discarded Cells due to HEC Violation
   This parameter provides a count of the number of incoming ATM cells discarded due to a Header Error Check (HEC) violation.

3. Discarded Cells due to Congestion This parameter provides a count of the number of ATM cells discarded due to congestion. (CLP =0+1)

4. Number of OAM cells received and processed

(R-4.3.1)-pm-3: The M4 network view interface shall support management system requests to reset any of the performance counters to zero.

(R-4.3.1)-pm-4: The M4 network view interface shall support management system requests to define multiple sets of threshold values (i.e., threshold value packages) for the performance parameters listed above and selectively assign each set.

(R-4.3.1)-pm-5: The M4 network view interface shall provide management systems the ability to modify threshold values for the performance parameters identified above.

(R-4.3.1)-pm-6: The M4 network view interface shall support autonomous notifications (generated by the ATM Subnetwork Manager) used to report threshold crossings for thresholded parameters. Such notifications are often referred to as "threshold crossing alerts".

(O-4.3.1)-pm-1: The M4 network view interface may provide management systems with the ability to retrieve current counts of High Priority Cells discarded due to congestion.

(R-4.3.1)-pm-7: The M4 network view interface shall provide management systems the ability to retrieve history counts (thirty-two 15-minute counts) of the performance parameters.

(R-4.3.1)-pm-8: Failures, testing routines, and reconfigurations may affect the collection of performance data. When such events occur, the ATM Subnetwork Manager is expected to flag the collected data as "suspect". The M4 network view interface shall provide management systems the ability to retrieve an indication as to whether the performance counts are reliable or suspect.

(R-4.3.1)-pm-9: For each ATM VP Link TP terminating on the ATM Subnetwork, the ATM Subnetwork Manager is expected to maintain a "latest occurrence" log containing the following information for ATM cells that were discarded due to protocol errors.

1. The Abnormality Type
2. VPI/VCI Value of Discarded Cell
3. Time and Date

(R-4.3.1)-pm-10: The M4 network view interface shall support the suppression of all-zero performance monitoring counts

UPC/NPC Disagreement Monitoring

UPC and NPC algorithms are intended to police incoming cells to ensure that each access connection supported by the ATM NE is complying with pre-negotiated traffic descriptors. Based on the UNI and BICI specifications developed by the ATM Forum (see [af-uni-0011.000] and [af-bici-0013.001], respectively), non-compliant traffic may result in cell discarding or tagging. Since cells discarded due to UPC/NPC functions (a fault of the user) and cells discarded due to transmission errors and malfunctions (a fault of the network) will have the same effect on the end-to-end performance of a VPC/VCC, it is important for trouble shooting and trouble sectionalization purposes to provide network managers with the tools needed to distinguish between these two events.

(R-4.3.2)-pm-1: The M4 network view interface shall support management system requests to initiate UPC/NPC Disagreement Monitoring on a limited number of VP/VC ATM Network CTPs at any one point in time (e.g., 30 ATM links per DS3 and 90 ATM links per STS-3c).

(R-4.3.2)-pm-2: The M4 network view interface shall support management system requests to cease UPC/NPC Disagreement Monitoring that was previously activated for a VP/VC ATM Network CTPs. The identity of the VP/VC link shall be provided along with the request.

(R-4.3.2)-pm-3: The M4 network view interface shall provide management systems the ability to retrieve current (15-minute) counts of the following data from each VP/VC ATM Network CTP for which UPC/NPC Disagreement Monitoring is being performed:

1. Discarded Cells due to UPC/NPC Disagreements
2. Discarded CLP =0 Cells due to UPC/NPC Disagreements
3. Successfully Passed Cells
4. Successfully Passed CLP =0 Cells

(R-4.3.2)-pm-4: The M4 network view interface shall support management system requests to define multiple sets of threshold values (i.e., threshold value packages) for the performance parameters listed above and selectively assign each set.

(R-4.3.2)-pm-5: The M4 network view interface shall provide management systems the ability to modify threshold values for the "Discarded Cells" and the "Discarded CLP =0 Cells" performance parameters.

(R-4.3.2)-pm-6: The M4 network view interface shall support management system requests to reset any of the performance counters to zero.

(R-4.3.2)-pm-7: The M4 network view interface shall support autonomous notifications (generated by the ATM Subnetwork Manager) used to report threshold crossings for thresholded parameters. Such notifications are often referred to as "threshold crossing alerts".

(R-4.3.2)-pm-8: The M4 network view interface shall provide management systems the ability to retrieve history counts (thirty-two 15-minute counts) of the performance parameters.

(R-4.3.2)-pm-9: Failures, testing routines, and reconfigurations may affect the collection of performance data. When such events occur, the ATM Subnetwork Manager is expected to flag the collected data as "suspect". The M4 network view interface shall provide management systems the ability to retrieve an indication as to whether the performance counts are reliable or suspect.

AAL Protocol Performance Monitoring

Reference Section 2.3.4 of the M4 NE View Interface Requirements and Logical MIB[1].

(R-4.3.3)-pm-1: The M4 network view interface shall support the management messages of the AAL Type for a given VC trail termination point in the ATM Subnetwork.

AAL Type 1 Performance Monitoring

(CR-4.3.3)-pm-1: If the VC trail termination point supports AAL Type 1 protocol monitoring, the M4 network view interface shall provide management systems the ability to retrieve current (15 minute) counts of the following errors (at the CS layer) at each trail end point where receiving IWF functions (with an AAL Type 1 supporting UDT) are performed (as defined in [1]):

1. AAL Header Errors
2. Sequence Counts total violations
3. Buffer Underflows
4. Buffer Overflows
5. SDT Pointer Reframes
6. SDT Pointer Parity Check Failures

(CR-4.3.3)-pm-2: If the VC trail termination points are capable of supporting AAL Type 1 protocol monitoring, the M4 network view interface shall support management system requests to define at least one set of threshold values (i.e., threshold value packages) for the supported parameters (listed above) and selectively assign each set to one or more AAL Type 1 entities in the ATM Subnetwork.

(CR-4.3.3)-pm-3: If the VC trail termination points are capable of supporting AAL Type 1 protocol monitoring, the M4 network view interface shall provide management systems the ability to modify threshold values for the supported performance parameters (identified above).

(CR-4.3.3)-pm-4: If the VC trail termination points are capable of supporting AAL Type 1 protocol monitoring, the M4 network view interface shall support autonomous notifications (generated by the ATM Subnetwork Manager) used to report threshold crossings for the supported parameters (identified above).

(CR-4.3.3)-pm-5: If the VC trail termination points are capable of supporting AAL Type 1 protocol monitoring, the M4 network view interface shall provide the management system the ability to retrieve history counts (thirty-two 15 minute counts) of the supported performance parameters (identified above).

AAL Type 3/4 Performance Monitoring

(CR-4.3.3)-pm-6: If the VC trail termination point supports AAL Type 3/4 protocol monitoring, the M4 network view interface shall provide the management system the ability to retrieve a single, aggregate, *thresholded* current (15 minute) sum of errors count that reflects the following errors (as defined in [1]):

1. Invalid Message Identifier (MID)
2. Invalid SAR-PDU Length Indication

(CR-4.3.3)-pm-7: If the VC trail termination point supports AAL Type 3/4 protocol monitoring, the M4 network view interface shall provide the management system the ability to retrieve a *thresholded* current (15 minute) sum of errors count that reflects the following errors (as defined in [1]):

1. SAR-PDU CRC incorrect when computed
2. COM/EOM Segment with unexpected SAR Sequence Number
3. BOM/EOM Segment with unexpected MID

(CR-4.3.3)-pm-8: If the VC trail termination points are capable of supporting AAL Type 3/4 protocol monitoring, the M4 network view interface shall provide the management system the ability to retrieve separate performance counters on each trail end point that terminates the AAL Type 3/4 protocol, for each of the SAR-PDU incorrect field error types listed above.

(CR-4.3.3)-pm-9: If the VC trail termination point are capable of supporting AAL Type 3/4 protocol monitoring, the M4 network view interface shall provide the management system the ability to retrieve *thresholded* current (15 minute) counts of the number of SRI time-outs that occur on each trail end point that terminates the AAL Type 3/4 protocol.

(CR-4.3.3)-pm-10: If the VC trail termination point are capable of supporting AAL Type 3/4 protocol monitoring, the M4 network view interface shall provide the management system the ability to retrieve current (15 minute) counts of the number of aborts that occur on each connection end point that terminates the AAL Type 3/4 protocol.

(CR-4.3.3)-pm-11: If the VC trail termination point are capable of supporting AAL Type 3/4 protocol monitoring, the M4 network view interface shall provide the management system the ability to retrieve a *thresholded* current (15 minute) sum of errors count that reflects the following errors (as defined in [1]):

1. BASize Field value not valid on an incoming AAL Type 3/4 CS-PDU; i.e., < 37 octets for multi-segment messages
2. Common Part Indicator not valid (i.e., not equal to 0)
3. Alignment Field not equal to 0.

(CR-4.3.3)-pm-12: If the VC trail termination point are capable of supporting AAL Type 3/4 protocol monitoring, the M4 network view interface shall provide the management system the ability to retrieve a *thresholded* current (15 minute) sum of errors count that reflects the following errors (as defined in [1]):

1. Beginning Tag (BTag) not equal to End Tag (ETag)
2. Buffer Allocation Size (BASize) and Length fields not equal when message mode is used, or BASize < Length when streaming mode is used.
3. Actual length of CS-PDU Payload not consistent with Length field.

(CR-4.3.3)-pm-13: If the VC trail termination point are capable of supporting AAL Type 3/4 protocol monitoring, the M4 network view interface shall provide the management system the ability to retrieve separate performance counters for each connection end point that terminates the AAL Type 3/4 protocol, for each of the CS-PDU incorrect field error types (listed above).

(CR-4.3.3)-pm-14: If the VC trail termination points are capable of supporting AAL Type 3/4 protocol monitoring, the M4 network view interface shall support management system requests to define at least one set of threshold values (i.e., threshold value packages) for the *thresholded* parameters (listed above) and selectively assign each set to one or more AAL Type 3/4 entities in the ATM Subnetwork.

(CR-4.3.3)-pm-15: If the VC trail termination points are capable of supporting AAL Type 3/4 protocol monitoring, the M4 network view interface shall provide management systems the ability to modify threshold values for the supported *thresholded* performance parameters (identified above).

(CR-4.3.3)-pm-16: If the VC trail termination points are capable of supporting AAL Type 3/4 protocol monitoring, the M4 network view interface shall support autonomous notifications (generated by the ATM Subnetwork Manager) used to report threshold crossings for the supported *thresholded* parameters (identified above).

(CR-4.3.3)-pm-17: If the VC trail termination points are capable of supporting AAL Type 3/4 protocol monitoring, the M4 network view interface shall provide the management system the ability to retrieve history counts (thirty-two 15 minute counts) of the supported performance parameters (identified above).

AAL Type 5 Performance Monitoring

(CR-4.3.3)-pm-18: If the VC trail termination point supports AAL Type 5 protocol monitoring, the M4 network view interface shall provide the management system the ability to retrieve current (15 minute) sum of errors count of the following errors at each connection termination point where AAL Type 5 is terminated:(as defined in [1]):

1. Invalid CPI
2. Oversized Received SDU
3. Length Violation (a length violation results in an invalid Pad field size)

(CR-4.3.3)-pm-19: If the VC trail termination point are capable of supporting AAL Type 5 protocol monitoring, the M4 network view interface shall provide the management system the ability to retrieve current (15 minute) counts of CRC-32 violations.

(CR-4.3.3)-pm-20: If the VC trail termination point are capable of supporting AAL Type 5 protocol monitoring, the M4 network view interface shall provide the management system the ability to retrieve current (15 minute) counts of the number of reassembly timer expirations on each receiving trail end point where AAL Type 5 is terminated.

(CR-4.3.3)-pm-21: If the VC trail termination points are capable of supporting AAL Type 5 protocol monitoring, the M4 network view interface shall support management system requests to define at least one set of threshold values (i.e., threshold value packages) for the supported parameters (listed above) and selectively assign each set to one or more AAL Type 5 entities in the ATM Subnetwork.

(CR-4.3.3)-pm-22: If the VC trail termination points are capable of supporting AAL Type 5 protocol monitoring, the M4 network view interface shall provide management systems the ability to modify threshold values for the supported performance parameters (identified above).

(CR-4.3.3)-pm-23: If the VC trail termination points are capable of supporting AAL Type 5 protocol monitoring, the M4 network view interface shall support autonomous notifications (generated by the ATM Subnetwork Manager) used to report threshold crossings for the supported parameters (identified above).

(CR-4.3.3)-pm-24: If the VC trail termination points are capable of supporting AAL Type 5 protocol monitoring, the M4 network view interface shall provide the management system the ability to retrieve history counts (thirty-two 15 minute counts) of the supported performance parameters (identified above).

Performance Management of OAM Flows

The management of OAM flows focuses on two areas: those requirements related to OAM continuity checking and those requirements related to intrusive and non-intrusive Performance Management. This section identifies PM requirements for the later areas.

(CR-4.3.4)-pm-1: If the Performance Management of F4/F5 OAM flows are supported, the M4 interface shall support the following performance counters which shall be kept in relation with the ATM VC Network CTP, ATM VC Network TTP, ATM VP Network CTP, and ATM VP Network TTP at which the monitoring occurs: Lost Cells, Misinserted Cells, and User Cells.

(CR-4.3.4)-pm-2: If the Performance Management of F4/F5 OAM flows are supported and if Far-End data collection is supported, the M4 interface shall support the following performance counters in relation with the ATM VC Network CTP, ATM VC Network TTP, ATM VP Network CTP, and ATM VP Network TTP at which the monitoring occurs: Far-End Lost Cells, Far-End Misinserted Cells, and Far-End User Cells.

(CR-4.3.4)-pm-3: If the Performance Management of F4/F5 OAM flows are supported, the M4 interface shall support a history of 32 15 minute counters which shall be kept for the counters listed in the above requirements.

(CR-4.3.4)-pm-4: If the Performance Management of F4/F5 OAM flows are supported, the M4 interface shall support the ability to reset the counters listed in the above requirements.

Configuration Management of OAM Flows

The management of OAM flows focuses on two areas: those requirements related to OAM continuity checking and those requirements related to intrusive and non-intrusive Performance Management. This section identifies requirements for both areas. Presented in Section 2.1.15 of M4 NE View [1].

(CR-4.3.5)-pm-1: If the Continuity Check OAM flows are supported, the M4 interface shall support the ability to activate and de-activate sink and source mechanism for this feature separately.

(CR-4.3.5)-pm-2: If the Continuity Check OAM flows are supported, the M4 interface shall provide the status (on/off) of the sink and source mechanisms separately.

(CR-4.3.5)-pm-3: If the Continuity Check OAM flows are supported, the M4 interface shall provide the operational status (enabled/disabled) related to the functioning of the feature as intended.

(CR-4.3.5)-pm-4: If the Performance Management of F4/F5 OAM flows are supported, the M4 interface shall allow a mechanism to configure this feature in intrusive or non-intrusive modes.

(CR-4.3.5)-pm-5: If the Performance Management of F4/F5 OAM flows are supported and if the intrusive mode is used, the M4 interface shall support the ability to activate and de-activate the sink mechanism and the source mechanism for this feature separately.

(CR-4.3.5)-pm-6: If the Performance Management of F4/F5 OAM flows are supported and if the intrusive mode is used, the M4 interface shall support the ability to activate and de-activate only the sink mechanism for this feature.

(CR-4.3.5)-pm-7: If the Performance Management of F4/F5 OAM flows are supported, the M4 interface shall provide the status (on/off) of the sink and source mechanisms separately.

(CR-4.3.5)-pm-8: If the Performance Management of F4/F5 OAM flows are supported and if the non-intrusive mode is used, the M4 interface shall allow the ability to configure the monitored flow as Segment or End-to-End.

(CR-4.3.5)-pm-9: If the Performance Management of F4/F5 OAM flows are supported and if the non-intrusive mode is used, the M4 interface shall provide the information related to the monitored flow as Segment or End-to-End.

(CR-4.3.5)-pm-10: If the Performance Management of F4/F5 OAM flows are supported and if the intrusive mode is used, the M4 interface shall provide the ability to configure the monitoring block size.

(CR-4.3.5)-pm-11: If the Performance Management of F4/F5 OAM flows are supported and if Far-End PM data collection is supported, the M4 interface shall allow their activation and de-activation.

(CR-4.3.5)-pm-12: If the Performance Management of F4/F5 OAM flows are supported and if Far-End PM data collection is supported, the M4 interface shall provide the information regarding its status (active or not).

(CR-4.3.5)-pm-13: If the Performance Management of F4/F5 OAM flows are supported and if backward PM data reporting is supported, the M4 interface shall allow their activation and de-activation.

(CR-4.3.5)-pm-14: If the Performance Management of F4/F5 OAM flows are supported and if backward PM data reporting is supported, the M4 interface shall provide the information regarding its status (active or not).

(CR-4.3.5)-pm-15: If the Performance Management of F4/F5 OAM flows are supported, the M4 interface shall provide the operational status (enabled/disabled) related to the functioning of the feature as intended.

(CR-4.3.5)-pm-16: If the Performance Management of F4/F5 OAM flows are supported, the M4 interface shall provide the operational status (enabled/disabled) related to the functioning of the feature as intended.

Network Accounting Management

FFS

Network Security Management

(O-4.5)-sm-1: The M4 interface should subject to authentication and access control all transactions between the management system and the subnetwork management system, and may record them in a security audit trail for subsequent security-related processing.

Protocol Independent MIB

Introduction

This section provides a description of protocol-independent Managed Entities for the M4 network-view. These managed entities support only a subset of the requirements specified in Section 4. The following are not supported at this time:

- any protection-switching, back-up functionality,
- grouping of connections,
- segment handling,
- reservation,

.

Note that, in addition, security and accounting are not covered in this document, neither at the requirement level, nor at the managed entity level.

The protocol-independent managed entities represent the information needed to manage the network resources, their states, and their state transitions. This data may be manipulated by different operations, which are defined along with the managed entities. Examples of how the operations can be used are described in Section 6.

Managed Entity List:

The following managed entities are specified for the M4 network-view:

network
vcLayerNetworkDomain
vcLinkConnection
vcLinkEnd
vcLogicalLinkTP
vcNetworkAccessProfile
vcRoutingProfile
vcSubnetwork
vcSubnetworkConnection
vcNetworkCTP
vcNetworkTTP
vcTopologicalLink

vcTrail
vpLayerNetworkDomain
vpLinkConnection
vpLinkEnd
vpLogicalLinkTP
vpNetworkAccessProfile
vpRoutingProfile
vpSubnetwork
vpSubnetworkConnection
vpNetworkCTP
vpNetworkTTP
vpTopologicalLink

vpTrail

The following managed entities needed in the M4 network-view are already specified in the NE-view (af-nm-0020.001):

aal1Profile
aal3/4Profile
aal5Profile
alarmRecord
alarmSeverityAssignmentProfile
atmCellProtocolMonitoringCurrentData (contained in ATM VP Link End)
atmCellProtocolMonitoringHistoryData (contained in ATM VP Link End)
atmCellProtocolMonitoringLogRecord
atmTrafficLoadCurrentData (contained in ATM VP Subnetwork, ATM VP Link End, ATM VP Network CTP, and ATM VC Network CTP)
atmTrafficLoadHistoryData (contained in ATM VP Subnetwork, ATM VP Link End, ATM VP Network CTP, and ATM VC Network CTP)
CESServiceProfile

congestionDiscardCurrentData (contained in ATM VP/VC Subnetwork, ATM VP Link End)
congestionDiscardHistoryData (contained in ATM VP/VC Subnetwork, ATM VP Link End)
eventForwardingDiscriminator
latestOccurrenceLog (contained in Network)
log
tcAdaptorProtocolMonitoringCurrentData (contained in ATM VP Link End)
tcAdaptorProtocolMonitoringHistoryData (contained in ATM VP Link End)
thresholdData (contained in Network)
trafficDescriptor

Managed Entity Description Format:

Each Managed Entity starts with a brief overall description.

The description is followed by a set of attributes, and a set of relationships. The relationships shall also be retrievable, can be created, modified, and deleted.

Next comes a set of notifications that the Managed Entity can emit.

Last comes a set of Operations. The operation description consists of input parameters to the operation, output parameters, error conditions, and a behaviour description. For all Managed Entities, the input parameters are such that:

- The Managed Entity ID is not specified as an input parameter, since the identification is already accomplished by the protocol used (e.g. SNMP or CMIP). If needed by a new protocol, it will then be added.
- Selection criteria may be used, as supported by the protocol capabilities selected.

Operations on attributes are implied in this logical MIB by the read/write or read-only qualifiers associated with each attribute.

Relationships for the Managed Entities are described in a protocol-independent way. The specification of the method to represent the relationship is left to the protocol-specific MIB. In the protocol-specific MIBs, they may be represented by containment, by attributes (e.g. supportedByObjectList, affectedByObjectList, pointers), by object references, or any other method. Operations on relationships are defined without reference to any protocol-specific implementation.

The current list of relationships is reflecting relationships between the transport entities introduced in this document. Relationships with support managed entities, such as log, or with existing NE-view managed entities, will be added in the next issue.

Relationship description semi-formal textual conventions

The following conventions have been used to the textual representation of ER-diagrams. The text is not expressed in a formal language but follows certain stylistic conventions. The following conventions have been used for the textual representation of the ER-diagrams.

1/ Comments are introduced after "--"

2/ The format used is as follows:

<subject> <RELATION> <cardinality> <object>

with <subject> and <object> being managed entity types.

Most common cardinality options are:

(n..*) n or more
(0..1) zero or one

(n) exactly n

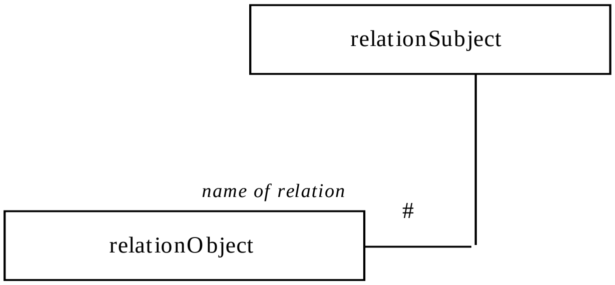
(n..m) n to m

3/ Relationships are described as if they were asymmetric; with this a subject-object distinction is made

4/ The containment relationship is denoted by "GROUPS," "IS_MADE_OF" so as not to imply the use of containment as defined in OSI management.

5/ When the object or subject is not an actual managed entity; but a generalized term such as "anyone of these managed entities," this is indicated by using brackets.

Relationship description semi-formal graphical conventions:



With:

#: cardinality:
zero or more: ●
zero or one: ○
exactly n: n

When M4 managed entities are implemented as CMIP or SNMP managed objects, there will not necessarily be a one-to-one correspondence between the managed entity and the object. In the protocol-specific MIBs, the relationships between managed entities may be represented by a managed object or a pointer. Operations on the relationships are defined without reference to any protocol-specific implementation. If an entity is not implemented as a managed object, then the operations specified on that entity may be implemented in another way.

Managed Entity Descriptions:

network

The Network managed entity groups all the managed entities visible over the M4 interface. The managed entities grouped under Network may span several transport layers (e.g. the VP and VC layers).

This managed entity is automatically created when the network is initialized. It is not created or deleted by the managing system.

Attributes

Network ID: This read-only attribute provides a unique name for the managed entity instance.

Notifications

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

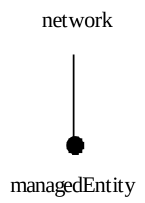
Relationships:

With managed resources: The network managed entity is made of a set of transport and other managed entities (e.g. log).

Semi-formal representation:

IS_MADE_OF (0..*) [managed entities]

Graphical representation:



network Query Operations:

Operation: query network For Contained Managed Entities

INPUT PARAMETERS:

none (see Section 5.1.2)

OUTPUT PARAMETERS:
containedManagedEntities: sequence of managedEntitiesId
ERROR CONDITIONS:
protocol-specific

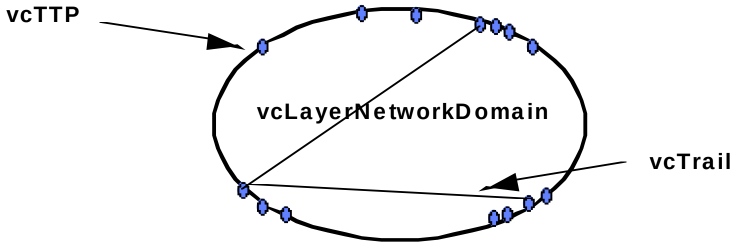
BEHAVIOUR

The network managed entity plays the role of container for the existing Managed Entities. This operation allows the requester (client) to query the contained Managed Entities. It does not affect the relationship. The reply is a sequence of Managed Entities belonging to the network.

vcLayerNetworkDomain

The layer network domain is defined to support the requirement for independent layer management, here the VC layer.

An ATM layer is concerned with the generation and transfer of characteristic information, i.e. ATM cells. The layer network domain managed entity represents the part of the ATM layer which is available to a managing system through the M4 interface. It contains only managed entities from a single ATM layer, here the VC layer. Note that it is assumed that a LayerNetworkDomain contains one and only one subnetwork, which can be further decomposed.



There may be several layer network domains within a single network.

It is assumed that the layer network domain is created automatically at the installation of the superior network managed entity. The automatic creation of instances of this managed entity shall be reported over the M4 interface to the managing system.

The managing system may subsequently create and delete other instances of the layer network domain provided there are no dependent entities.

Attributes

Signal Identification: This read-only attribute represents the characteristic information of the layer network domain. Here, it is fixed to VC.

User Label: This read/write attribute allows a manager to represent additional information about the layer network domain

Notifications

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this managed entity.

Attribute Value Change: This notification is used to report changes of the user label.

Relationships:

Note that the layer network domain acts as a container for entities which provide access to a transport layer such as trails and network trail termination points. These are contained in layer network domain rather than a subnetwork because they can not be partitioned, as is possible for a subnetwork.

With vcTTP: A vcLayerNetworkDomain is delimited by zero or more vcTTPs.

With vcTrail: A vcLayerNetworkDomain groups zero or more vcTrails.

With vcSubnetwork: A vcLayerNetworkDomain is partitioned into zero or more vcSubnetworks.

With vcTrailRequest: A vcLayerNetworkDomain can have zero or more vctrailRequests modifying the trails it groups.

Semi-Formal representation:

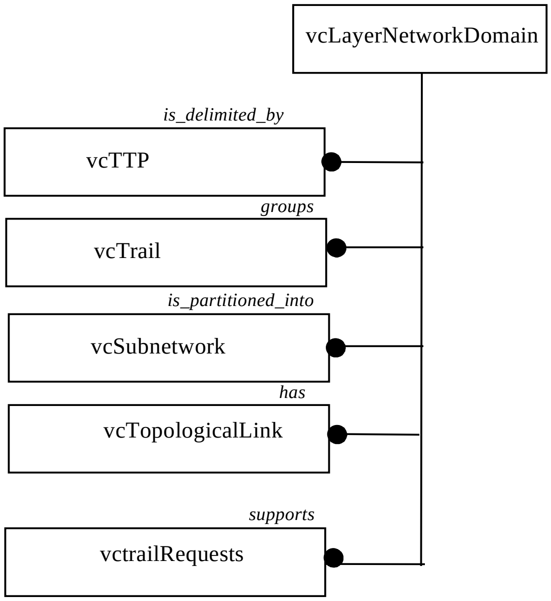
IS_DELIMITED_BY (0..*) vcTTP

GROUPS (0..*) vcTrail

IS_PARTITIONED_INTO (0..*) vcSubnetwork

SUPPORTS (0..*) vcTrailRequests

Graphical representation:



vcLayerNetworkDomain Query Operations:

Operation: query vcLayerNetworkDomain For Delimiting vcTTPs

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
delimitingVcTTPs : set of vcTTPIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each vcTTP is delimiting a vcLayerNetworkDomain. This operation allows the requester (client) to query the delimiting Managed Entities. It does not affect the relationship. It matches for all the delimiting vcTTPs. The reply is a set of vcTTPIds.

Operation: query vcLayerNetworkDomain For existing vcTrails

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
existingVcTrail: set of vcTrailId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

The vcLayerNetworkDomain plays a role of container for the existing vcTrails. This operation allows the requester (client) to query the contained vcTrails. It does not affect the relationship. It matches for all the contained vcTrails. The reply is a set of vcTrailIds belonging to the vcLayerNetworkDomain.

Operation: query vcLayerNetworkDomain for component vcSubnetwork

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
componentVcSubnetwork: vcSubnetworkId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR:
A vcLayerNetworkDomain contains one vcSubnetwork. This query is an downward query for the contained vcSubnetwork. It does not affect the relationship. It matches for the contained vcSubnetwork that is a direct component of the vcLayerNetworkDomain. The reply is a vcSubnetworkId.

vcLayerNetworkDomain: vcTrail set-up operations

Operation: set up vcTrail set-up

INPUT PARAMETERS:
vcTTPa : choice of vcNetworkCTPId, or Descriptor
vcTTPz : choice of vcNetworkCTPId, or Descriptor
AdministrativeState (optional)
retainedResource (optional)

Descriptor:
interfaceId (Choice of server TTPId)
vpi (optional)
vci (optional)
trafficDescriptors (optional)
qos (optional)

OUTPUT PARAMETERS:
newVcTrail : vcTrailId
vcTTPa: vcTTPId
vcTTPz: vcTTPId
ERROR CONDITIONS:
protocol-specific addressing errors
incorrectTerminationPoints : vcTTPId
reflectedTTPDisabled : vcTTPId
reflectedTTPLocked: vcTTPId
vcTrailTerminationPointConnected : vcTTPId
non-matchingDescriptors: set of vcTTPId
operationFails
BEHAVIOUR:

This operation allows the requester (user) to set-up a point-to-point Trail between two non-connected TTPs of the addressed Layer Network Domain, identified directly, or identified by an interface within which the Layer Network Domain selects a point. In the latter case, a set of optional descriptors may be provided (vpi, vci, traffic descriptors, qos). An error condition will be raised if the termination points are incorrect (i.e. do not belong to the Layer Network Domain), if the two TTPs are already used, if they do not have matching traffic descriptors, or if the Layer Network Domain is unable to provide sufficient bandwidth (operations failure). The result of the operation is:

- the creation of two TTPs and a Trail associated to both of them in the case of the Layer Network Domain creating the vcTTPs
- the creation of a Trail associated to two existing TTPs in the case of TTPs already existing in the Layer Network Domain.

Operation: setup vcTrailRequest

INPUT PARAMETERS:
none (see Section 5.1.2)
requestActionInfo: setup, modify, release, addTps, removeTps
requestCommittedTime (optional)
relatedTrails: zero or one vcTrailId (optional)
OUTPUT PARAMETERS:
vcTrailrequest: vcTrailRequestId
ERROR CONDITIONS:
protocol-specific errors, and
invalidTrailId: trailId
invalidTime
operationFails
BEHAVIOUR

This operation allows the requester (client) to setup an vcTrailRequest.. The reply is a vcTrailRequestId.

Operation: addTps To Multipoint Trail:

INPUT PARAMETERS:
see Section 5.1.2, and
trail : trailId
ttpZ : choice of ttpId, or Descriptor
AdministrativeState (optional)

Descriptor:
interfaceId (Choice of networkCTPId, or server TTPId)
vpi (optional)
vci (optional)
trafficDescriptors (optional)
qos (optional)

OUTPUT PARAMETERS:
ttpZ: ttpId

ERROR CONDITIONS:
protocol-specific addressing errors
incorrecttrail : trailId
incorrectTerminationPoints : ttpId
reflectedTPDisabled : ttpId
reflectedTPLocked: ttpId
trailTerminationPointConnected : ttpId
non-matchingDescriptors: set of ttpId
operationFails

BEHAVIOUR:

This operation allows the requester (user) to add a point to a multipoint trail. The TTP to connect are identified directly or indirectly. In the latter case, a set of optional descriptors may be provided (vpi, vci, traffic descriptors, qos). An error condition will be raised if the termination points are incorrect (e.g. do not belong to the LND), if the TTP is already used, or if the LND is unable to provide sufficient bandwidth (operations failure). The result of the operation is:

- the association of a TTP with a trail,
- possibly, the creation of a TTPTP.

Operation: release vcTrail

INPUT PARAMETERS:
see Section 5.1.2, and
Choice of VpTTP : set of vcTTPIds

OUTPUT PARAMETERS:
released vcTrail: vcTrailId
released vcTTPs: set of vcTTPId

ERROR CONDITIONS:

protocol-specific addressing errors
incorrectTerminationPoints : vcTTPId
vcTrailTerminationPointNotConnected : vcTTPId
incorrectTrails: vcTrailId

BEHAVIOUR:

This operation allows the requester (user) to release a point-to-point Trail between two connected TTPs of the addressed Layer Network Domain, identified directly, identified by one of the connected TTPs, or identified by the pair of the TTPs. An error condition will be raised if the termination points are incorrect (i.e. do not belong to the Layer Network Domain), if the two TTPs are not connected, or if the Trail is not connected. The result of the operation is:

- the deletion of two TTPs and the Trail associated to both of them.

vcLayerNetworkDomain: link Management Operations:

Operation: Make External vcLinkEnd

INPUT PARAMETERS:

linkTPendpoint: choice of vpTTPId, vcLinkTPDetails
linkTPprofile: choice of atmNetworkAccessProfileId, or ProfileDetails

vcLinkTPDetails:

interfaceId (vpLinkEnd or vpLogicalLinkTP)
ingressTrafficDescriptor
egressTrafficDescriptor
requestedVPI Optional
userLabel Optional

ProfileDetails:

vpiOrVciRange,
maxNumActiveVPCAllowed,
totalEgressBandwidth,
totalIngressBandwidth,
maxNumActiveVCCAllowed

OUTPUT PARAMETERS:

newLinkTP : linkEndId

ERROR CONDITIONS:

protocol-specific addressing errors
serverTTP unavailable or used
incorrectTerminationPoints : vpTTPId
linkTerminationPointConnected : linkEndId
non-matchingDetails(profile): set of linkEndId, interfaceId, or atmProfileId
operationFails

BEHAVIOUR:
This operation creates an instance of the atmLinkEnd object, that represents an interface to an external network, along with the objects in the server layer network domain, if needed, that support the external atmLinkEnd. The external atmLinkEnd terminates a link that is not visible across the management interface, the link to an external network. The underlying server TTP may be identified directly (if it already exists) or indirectly, in which case the managed system may select or create the appropriate server TTP to support the atmLinkEnd. Indirect server TTP identification may indicate: interface identifier, desired total bandwidth, an atmNetworkAccessProfile identifier, or a set of descriptors providing VPI/VCI range, etc. This approach allows the possible creation of associated server layer trail terminations at the same time as the atmLinkEnd is created. An error condition is raised if the server trail termination point is incorrect, already used, does not have matching range or bandwidth, or if the interface is unable to provide sufficient bandwidth.

Upon creation of the atmLinkEnd, the managed system is responsible for determining and setting: availableIngressBandwidth, availableEgressBandwidth, maxAssignableIngressBandwidth, maxAssignableEgressBandwidth, atmNetworkAccessProfilePointer, and the serverTTPList attributes. If a new atmNetworkAccessProfile object instance is needed, the managed system is responsible for creating it.

To create an atmLinkEnd in the VC Layer Network Domain, the information in the request may simply identify the atmLinkEndId at the VP Layer as the end point. The managed system shall then create a vpTTP or networkTTP and the corresponding CTP in the VP LND along with an associated atmLinkEnd in the VC LND. This does not preclude the creation of VP Trail Terminations prior to request for creating the external atmLinkEnd. If the VP Trail Termination exists prior to the request, the requester may specify the Trail Termination as the supporting underlying server TTP(s).

Operation: Remove External vcLinkEnd
INPUT PARAMETERS:
linkTPId : vcLinkEndId
removeServerTTPIndicator: Optional, Boolean

OUTPUT PARAMETERS:
removedLinkTP : linkEndId

ERROR CONDITIONS:
protocol-specific addressing errors
incorrectTerminationPoints : linkEndId
alreadyConnected: existing connections
operationFails

BEHAVIOUR:
This ACTION removes an instance of the atmLinkEnd object, that represents an interface to an external network. For VC Level external atmLinkEnds, this action optionally removes the objects in the server layer network domain that support the external atmLinkEnd. The external atmLinkEnd terminates a link that is not visible across the management interface, the link to an external network. An error condition is raised if the atmLinkEnd is incorrect or supports any connections.

Upon removal of the atmLinkEnd, the managed system is responsible for determining if the associated atmNetworkAccessProfile object instance is shared by another atmLinkEnd or atmLink. If the atmNetworkAccessProfile is not shared, the managed system is responsible for deleting it. Also, the managed system shall remove the reference to the atmLinkEnd from the supportedLinkTPList attribute of all subnetworks that supported the removed external atmLinkEnd.

To remove an external atmLinkEnd in the VC Layer Network Domain, the remove link TP information may indicate that the supporting vpTTP(s) or networkTTP(s) in the VP Layer Network Domain should be deleted along with the atmLinkEnd: If the supporting server layer (VP Layer) TTP is not used by another VC Layer atmLinkEnd, the managed system shall then remove the vpTTP or networkTTP and the corresponding CTP in the VP LND along with the atmLinkEnd. When removing the VP Layer atmNetworkCTP, the managed system shall determine if the associated atmNetworkTrafficDescriptorProfile is shared. If this atmNetworkTrafficDescriptorProfile is not shared, the managed system is responsible for deleting it.

Input: atmLinkEndId. If removing a VC Layer atmLinkEnd: removeServerTTPIndicator

Output: Confirmation or Error Conditions.

Errors: protocol-specific addressing errors, noSuchTpInstance, alreadyConnected, operationFails.

Results: Removal of atmLinkEnd and an associated atmNetworkAccessProfile. Possible removal of VP TTP, VP CTP, and atmNetworkTrafficDescriptorProfile for VC Layer atmLinkEnd removal.

Operation: Setup vcTopologicalLink

linkTPa : choice of vcLinkTPId, or vcLinkDetails
linkTPz : choice of vcLinkTPId, or vcLinkDetails

vcLinkTPId: vcLinkEndId or vcLogicalLinkTPId

vcLinkDetail:

ingressBandwitdth
egressBandwidth
interfaceId
choice of atmNetworkAccessProfileID or profileDetails
fromTrafficDescriptor Optional
toTrafficDescriptor Optional
fromVPI Optional
toVPI Optional

ProfileDetails:

vpiOrVciRange,
maxNumActiveVPCAllowed,
maxEgressBandwidth,
maxIngressBandwidth,
maxNumActiveVCCAllowed

OUTPUT PARAMETERS:

newLink : linkId
linkTPa: vcLinkTPId
linkTPz: vcLinkTPId

ERROR CONDITIONS:

protocol-specific addressing errors
incorrectTerminationPoints : vcLinkTPId
linkTerminationPointConnected : vcLinkTPId
non-matchingDetails(profile or link): set of vcLinkTPId, interfaceId, or atmProfileId
operationFails

BEHAVIOUR:
This operation sets up a point-to-point link at the VC Layer Network Domain between two VC subnetworks. This VC Link may directly connect two VP subnetworks or network elements supported by a VP link connection, or it may be supported by a VP connection that traverses VP subnetworks or network elements. The managing system may request a specific VPI endpoint values for the supporting VP connection. The linkTPs are identified either directly or indirectly with an interface identifier, available bandwidth, and an atmNetworkAccessProfile identifier, or a set of descriptors providing VPI/VCI range, etc. In cases where the VC Link is not supported directly by a VP link connection and the underlying VP connections are not previously established, the action needs to specify the traffic descriptors for the underlying VP trail and supporting subnetwork connections. This approach allows the creation of the VC Link TPs (including possible creation of associated VP layer Trail Terminations and subnetwork connections) at the same time as the atmLink setup if needed. An error condition is raised if the link termination points are incorrect, already used, do not have matching range or bandwidth, or if the interface is unable to provide sufficient bandwidth.

To setup an atmLink in the VC Layer Network Domain:

If the VC Link TPs that will support the VC Link exist, the setup link information should identify the desired VC Link TPs directly as the link endpoints.

If the VC Link is to be supported by existing VP Trail Terminations, the setup link information should identify these VP TTPs, indirectly identifying the link endpoints, along with network access profile information. The action creates the associated VC Link TPs.

If the VC Link is to be supported by a VP connection directly between two VP subnetworks or network elements (VP Link Connection), the setup link information should indirectly identify the link endpoints using the VP Link TPs associated with the appropriate interfaces, along with network access profile information and optionally a single requested VPI value. The managed system shall then establish a VP Link Connection, creating vpTTPs or networkTTPs and the corresponding CTPs in the VP LND along with an associated atmLinkEnds in the VC LND.

If the VC Link will be supported by a VP connection that traverses VP subnetworks or network elements, the setup link information should identify the link endpoints indirectly using VP Link TPs and provide both link network access profile information along with traffic descriptor information and requested VPI endpoint values for the VP level connection. The managed system shall then establish a VP Subnetwork Connection, creating vpTTPs or networkTTPs the corresponding CTPs and atmNetworkTrafficDescriptorProfile in the VP LND along with an associated atmLinkEnds in the VC LND.

Input: atmlinkTPs (atmLinkEnds or atmLogicalLinkTPs) or descriptors of the endpoints: interfaceId (serverTTPId, server layer atmLinkTPId for vc Links, atmLinkEndId or atmLogicalLinkTPId), available bandwidth, and atmNetworkAccessProfilePointer or VPI range, VCI range, and maximum bandwidth, and if a VP subnetwork connection is needed to support the link, traffic descriptors. Optionally, the VPI values of the endpoints of the supporting VP connection may be requested.

Operation: release vcTopologicalLink

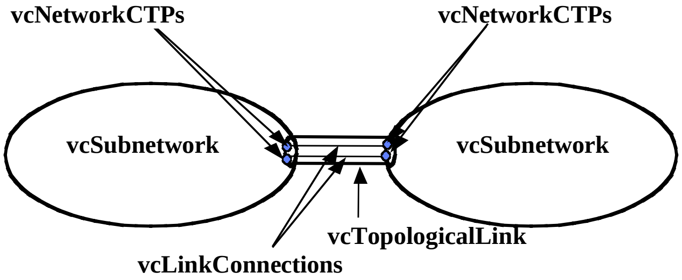
INPUT PARAMETERS:
see Section 5.1.2, and
Choice of vcLinkTPId; vcLinkEndId; linkId
OUTPUT PARAMETERS:
released vcTopologicalLink: linkId
released vcLinkTermination: set of vcLinkEnds or vcLogicalLinkTPs
released vpTTPs: set of vpTTPId (only if not a logical link)
ERROR CONDITIONS:
protocol-specific addressing errors
incorrectTerminationPoints : vcLinkTPId, or vcLinkEndId
incorrectLink: vcLinkId
BEHAVIOUR:

This operation allows the requester (user) to release a point-to-point Link between two connected LinkEnds or LogicalLinkTPs of the addressed Layer Network Domain, identified directly, identified by one of the connected endpoints. An error condition will be raised if the termination points are incorrect (i.e. do not belong to the Layer Network Domain), or if the Link is not connected. The result of the operation is:

- the deletion of two link endpoints, and the link associated to both of them. The underlying server trail is also deleted in cases where the link is not a logical link.

vcLinkConnection

This managed entity represents a I.326 link connection, derived from the G.805 definition, i.e. "a transport entity which transfers information between "ports" across a link." This entity is explicitly created by a network management function. A linkConnection cannot be created between a composite subnetwork and one of its component subnetwork. Only point-to-point linkConnections are supported.



Attributes

vcLinkConnection ID: This read-only attribute provides a unique name for the managed entity instance within the management domain.

Signal Identification: This read-only attribute, set at creation, describes the signal that is transferred across the link. Here, it is fixed to VC.

Directionality: This attribute is always set to "bidirectional."

User Label: This read/write attribute provides an arbitrary label corresponding to the connection which is established.

availability Status: This read-only attribute identifies whether or not the managed entity is capable of performing its normal functions (Failed or no unavailability condition existing).

Administrative State: This read/write attribute is used to lock and unlock cell flow through the link connection.

retainedResource: This read/write attribute indicates if the managed entity instance needs to be retained when component of a composite connection (linkConnection, subnetworkConnection), or when supporting a linkConnection (trail)

Notifications

Attribute Value Change: This notification is used to report changes of the user label.

State Change: This notification is used to report changes to the State attributes of this managed entity. The notification shall identify the state attribute that changed, its old value, and its new value.

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this managed entity.

Relationships:

With vcTopologicalLinks: A topological link is a group of link connections sharing the same extremities. This relationship involves one and only one instance of the link managed entity, and zero or more instances of the linkConnection Managed Entity.

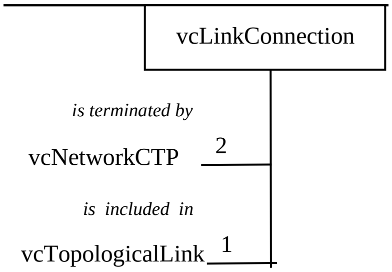
With vcNetworkCTPs: A vcLinkConnection has two vcNetworkCTPs, one on each subnetwork that it is linking.

semi-formal representation:

IS_INCLUDED_IN (1) vcTopologicalLink

IS_TERMINATED_BY (2) vcNetworkCTP

graphical representation:



vcLinkConnection Query Operations:

Operation: query vcLinkConnection For Containing vcTopologicalLink

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
containingVcTopologicalLinks: vcTopologicalLinkId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

The vcTopologicalLink plays a role of container for the existing vcLinkConnections. This operation allows the requester (client) to query the containing Managed Entities. It does not affect the relationship. It matches for the containing vcTopologicalLink. The reply is the vcTopologicalLinkId.

Operation: query vcLinkConnection For terminating vcNetworkCTPs

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatingNetworkCTPs: set of networkCTPIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

A vcLinkConnection is terminated on two vcNetworkCTPs. This operation allows the requester (client) to query the terminating Managed Entities. It does not affect the relationship. It matches for the two terminating vcNetworkCTPs. The reply is a set of two vcNetworkCTPIds.

vcLinkEnd

This managed entity is used to represent the termination of a topologicalLink at the VC-layer and is associated with a single vpNetworkTTP. The vcLink managed entity is used to store VC link-level configuration data. This managed entity is created by the management system.

Attributes

VcLinkEnd ID: This read-only attribute provides a unique name for the managed entity instance in the VC LND.

Administrative State: This settable attribute allows for the configuration of the administrative state of the vcLinkEnd.

Availability Status: This read-only attribute describes the operational status (working, degraded, not-working) of the vcLinkEnd.

Egress Maximum Assignable Bandwidth: This read only attribute identifies the maximum amount of bandwidth assignable on the link in the Egress direction (outbound or away from the ATM NE).

Ingress Maximum Assignable Bandwidth: This read only attribute identifies the maximum amount of bandwidth assignable on the link in the Ingress direction (inbound or towards the ATM NE).

Egress available Bandwidth: This read-only attribute identifies the amount of bandwidth left on the link in the Egress direction (outbound or away from the ATM NE).

Ingress available Bandwidth: This read-only attribute identifies the amount of bandwidth left on the link in the Ingress direction (inbound or towards the ATM NE).

User Label: This string may be used to describe additional information about the atmLinkEnd, such as a circuit identifier.

Link TP Type: Describes the interface type that the atmLinkEnd supports: UNI, inter-NNI, intra-NNI, or unconfigured.

Notifications

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this managed entity.

Attribute Value Change: This notification is used to report changes to the attribute changes of this managed entity. The notification shall identify the attribute that changed, its old value, and its new value.

Relationships

With vcTopologicalLink: Each vcTopologicalLink may be terminated by an instance of the vcLinkEnd managed entity.

With vcLogicalLinkTP: Each vcLogicalLinkTP of a partitioned or composite vcTopologicalLink may be supported by one or more instances of the vcLinkEnd managed entity.

With vcSubnetwork: One vcLinkEnd managed entity is associated with one or more vcSubnetworks.

With serverTTP: Each vcLinkEnd is supported by one instance of a TTP managed entity in the serverLayer (vpNetworkTTPBidirectional for a vcLinkEnd or where the NE view is supported along with the network-view, a vpBidirectionalTTP for a vcLinkEnd).

With vcNetworkAccessProfile: Each vcLinkEnd may use one atmNetworkAccessProfile.

With vcNetworkCTP: Existing Connection Termination Points: A list of CTPs within the same layer network domain that are supported by the atmLinkEnd. This attribute provides the association between the atmLinkEnd (and underlying server trail) and the same layer network domain CTPs supported at the atmLinkEnd. That is, a VC atmLinkEnd identifies the VC CTPs supported at the interface point.

semi-formal representation:

TERMINATES (0..1) vcTopologicalLink

SUPPORTS (0..*) vcLogicalLinkTP

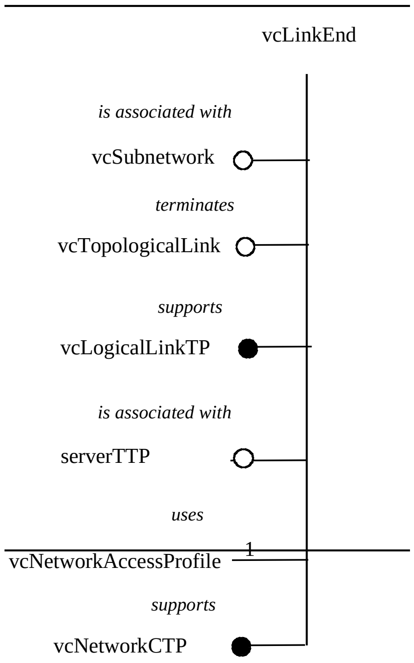
IS_ASSOCIATED_WITH (1..*) vcSubnetwork

IS_ASSOCIATED_WITH (0..1) serverTTP

USES (1) vcNetworkAccessProfile

SUPPORTS (0..*) vcNetworkCTPs

graphical representation:



vcLinkEnd: vcLinkEnd Query Operations

Operation: query vcLinkEnd For Terminated vcTopologicalLink

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatedLinkEnds: set of LinkEndIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR:

Each topologicalLink is terminated by two LinkEnds. This operation allows the requester (client) to query the terminated topologicalLink. It does not affect the relationship. It matches for terminated topologicalLink. The reply is an topologicalLinkId.

Operation: query vcLinkEndTP For Delineated vcSubnetwork

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
delineatedSubnetworks: subnetworkIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each LinkEnd delineates one or more subnetwork. This operation allows the requester (client) to query the delineated subnetwork. It does not affect the relationship. It matches for the delineated subnetwork. The reply is a subnetworkIds.

Operation*:* query vcLinkEnd for associated vpTTP

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
associatedVcTTP: vcTTPId

ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each vcLinkEnd is associated with a vpTTP. This operation allows the requester (client) to query for the associated managed entity. It does not affect the relationship. It matches for the associated vpTTP. The reply is the associated vpTTP.

Operation: associate vcLinkEnd with supporting vpTTP

INPUT PARAMETERS:
none (see Section 5.1.2)
associatedVpTTP
OUTPUT PARAMETERS:
associatedVpTTP: vpTTPId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each vcLinkEnd at the VC-level is supported by a vpTTP managed entity, which provides transport for it. This operation allows the requester (client) to associate the vcLinkEnd with a single vpTTP. This operation may take place at the creation of the vcTopologicalLink or of the vcLinkEnd. The reply is the associated vpTTP.

vcLinkEnd: Trace Operations

Operation: vcLinkEnd PVC Trace

INPUT PARAMETERS:
List of vcSubnetworks
- used to scope the trace of vcSubnetworkConnections terminated on the LinkEnd
OUTPUT PARAMETERS:
Identifier of each atmSubnetwork along with the Ids of the atmSubnetworkConnections within each atmSubnetwork that terminate on the LinkEnd
{ vcSubnetworkId { vcSubnetworkConnectionId }* }*
ERROR CONDITIONS:
identified atmSubnetwork does not exist.

atmSubnetwork or atmLinkTP cannot be traced by the managed system.

BEHAVIOUR:

This operation requests that the managed system perform a link PVC trace on a specific vcLinkEnd, identifying vcSubnetworkConnections that terminate on the vcLinkEnd. The operation should identify the atmSubnetworks on which the managed system will determine the vcSubnetworkConnections that terminate on the vcLinkEnd. The request can identify multiple vcSubnetworks to scope the trace. The identified subnetworks and their connections must be visible to the managed system.

vcLogicalLinkTP

This managed entity is used to represent the termination of a topologicalLink that is a partitioned or composite link at the VC-layer. The vcLogicalLinkTP managed entity is used to store any link-level configuration data. This managed entity is created by the management system.

Attributes

vcLogicalLinkTP ID: This read-only attribute provides a unique name for the managed entity instance in the VC LND.

Egress Maximum Assignable Bandwidth: This read only attribute identifies the maximum amount of bandwidth assignable on the logical link in the Egress direction (outbound or away from the ATM NE).

Ingress Maximum Assignable Bandwidth: This read only attribute identifies the maximum amount of bandwidth assignable on the logical link in the Ingress direction (inbound or towards the ATM NE).

Egress available Bandwidth: This read-only attribute identifies the amount of bandwidth left on the logical link in the Egress direction (outbound or away from the ATM NE).

Ingress available Bandwidth: This read-only attribute identifies the amount of bandwidth left on the logical link in the Ingress direction (inbound or towards the ATM NE).

VCI Range: This read/write parameter identifies the range of VCI values that may be used over the logical link.

User Label: This string may be used to describe additional information about the atmLogicalLinkTP, such as a circuit identifier.

Notifications

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this managed entity.

Attribute Value Change: This notification is used to report changes to the attribute changes of this managed entity. The notification shall identify the attribute that changed, its old value, and its new value.

Relationships

With vcTopologicalLink: Each vcTopologicalLink may be terminated by an instance of the vcLogicalLinkTP managed entity.

With vcLinkEnd: Each vcLogicalLinkTP of a partitioned or composite vcTopologicalLink may be supported by one or more instances of the vcLinkEnd managed entity.

With vcSubnetwork: One vcLogicalLinkTP managed entity is associated with one or more vcSubnetwork.

With vcNetworkAccessProfile: Each vcLogicalLinkTP may use one atmNetworkAccessProfile. Note that vcNetworkAccessProfile may restrict for partitioned links and extend for composite links, the characteristics described in the vcNetworkAccessProfile associated with the vcLinkEnd.

With vcNetworkCTP: Existing Connection Termination Points: A list of CTPs within the same layer network domain that are supported by the atmLogicalLinkTP. This attribute provides the association between the atmLogicalLinkTP and the same layer network domain CTPs supported at the atmLogicalLinkTP. That is, a VC atmLogicalLinkTP identifies the VC CTPs supported at the interface point.

semi-formal representation:

TERMINATES (0..1) vcTopologicalLink

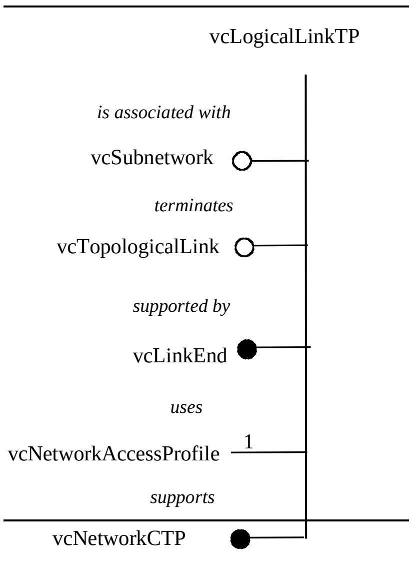
IS SUPPORTED BY (1..*) vcLinkEnd

IS_ASSOCIATED_WITH (1..*) vcSubnetwork

USES (1) vcNetworkAccessProfile

SUPPORTS (0..*) vcNetworkCTPs

graphical representation:



vcLogicalLinkTP: vcLogicalLinkTP Query Operations

Operation: query vcLogicalLinkTP For Terminated vcTopologicalLink

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatedTopologicalLinks: set of topologicalLinkIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR:

Each topologicalLink may be terminated by two LogicalLinkTPs. This operation allows the requester (client) to query the terminated topologicalLink. It does not affect the relationship. It matches for terminated topologicalLink. The reply is an topologicalLinkId.

Operation: query vcLogicalLinkTP For Delineated vcSubnetwork

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
delineatedSubnetworks: subnetworkIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each logicalLinkTP delineates a subnetwork. This operation allows the requester (client) to query the delineated subnetwork. It does not affect the relationship. It matches for the delineated subnetwork. The reply is a subnetworkIds.

Operation*:* query vcLogicalLinkTP For associated vcLinkEnds

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
associatedVcLinkEnds: vcLinkEndIds

ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each vcLogicalLinkTP is associated with one or more vcLinkEnds. This operation allows the requester (client) to query for the associated managed entity. It does not affect the relationship. It matches for the associated vcLinkEnd. The reply is the associated vcLinkEnd.

Operation: associate vcLogicalLinkTP with supporting vcLinkEnd

INPUT PARAMETERS:
none (see Section 5.1.2)
associatedVcLinkEnd

OUTPUT PARAMETERS:
associatedVcLinkEnd: vcLinkEndId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each vcLogicalLinkTP at the VC-level is supported by one or more vcLinkEnd managed entity, which represents server transport for it. This operation allows the requester (client) to associate the vcLogicalLinkTP with one or more vcLinkEnds. This operation may take place at the creation of the vcTopologicalLink or of the vcLogicalLinkTP. The reply is the associated vcLinkEnd.

vcLogicalLinkTP: Trace Operations

Operation: vcLogicalLinkTP PVC Trace

INPUT PARAMETERS:
List of vcSubnetworks
- used to scope the trace of vcSubnetworkConnections terminated on the LogicalLinkTP
OUTPUT PARAMETERS:
Identifier of each atmSubnetwork along with the Ids of the atmSubnetworkConnections within each atmSubnetwork that terminate on the Logical Link TP
{ vcSubnetworkId { vcSubnetworkConnectionId }* }*
ERROR CONDITIONS:
identified atmSubnetwork does not exist.
atmSubnetwork or atmLogicalLinkTP cannot be traced by the managed system.
BEHAVIOUR:

This operation requests that the managed system perform a link PVC trace on a specific vcLogicalLinkTP, identifying vcSubnetworkConnections that terminate on the vcLogicalLinkTP. The operation should identify the atmSubnetworks on which the managed system will determine the vcSubnetworkConnections that terminate on the vcLogicalLink. The request can identify multiple vcSubnetworks to scope the trace. The identified subnetworks and their connections must be visible to the managed system.

vcNetworkAccessProfile

The vcNetworkAccessProfile managed entity contains information that describe the maximum ingress and egress bandwidth, along with the VCI values that are applies to the vcLink, vcLinkEnd, or the vcLogicalLinkTP instances that point to it.

This managed entity is created by the managing system.

Attributes

vcNetworkAccessProfile ID: This read-only attribute provides a unique name for the managed entity instance.

total Egress Bandwidth: This read/write attribute identifies the total aggregate egress bandwidth for a link or a linkTP (linkEnd or logicalLinkTP).

total Ingress Bandwidth: This read/write attribute identifies the total aggregate ingress bandwidth for a link or a linkTP (linkEnd or LogicalLinkTP).

maximum Number of Active Connection Allowed: This read/write attribute identifies the maximum number of concurrently active VP (for a vpLayerNetworkDomain) or VC (for a vcLayerNetworkDomain) connections that a link or a linkTP (linkEnd or LogicalLinkTP) may support.

VPI or VCI ID Range: This read/write attribute describes the virtual ID range (VCIs in the vcLayerNetworkDomain and/or VPIs in the vpLayerNetworkDomain) that may be used for Connections associated with a link or linkTP

Notifications

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

Relationships:

None originating from this managed entity.

Operations:

None beyond the setting and querying of the read/write attributes.

vcRoutingProfile

This managed entity provides a set of routing constraints that can be applied to a new connection or trail during setup. A routing profile may be created automatically based on the routing description in the setup operation. The management system may also create profiles directly. Each vcSubnetworkConnection or vcTrail may point to a vcRoutingProfile. Connections should not be established (or re-established) if the routing criteria cannot be met. If maxHops is specified, the connection should not be established (or re-established) if the maximum number of hops is exceeded

The maxHops attribute is the maximum number of hops between nodes that the new connection may traverse. This attribute may be set to NULL to indicate that the maxHops criteria does not apply.

The routeDescriptionList attribute is a list of objects (such as Links, Subnetworks, existing connections) and their use in routing (exclude, mandatory, preferred, same route, diverse route).

The connection types that the routing profile supports are indicated in the connectionTypesSupported attribute. For all types of multipoint connections at least the sameRoute criteria may be applied. All of the criteria may be applied to point-to-point connections.

Managed entities (such as vcSubnetwork, vcLink, or managedElement, etc) may be referenced by the routeDescriptionList as being excluded, mandatory, or preferred. If a managed entity is described as mandatory it must be used in setting up a new connection. An attempt must be made during setup to include a managed entity described as preferred. An excluded managed entity must not be used in a connection.

Connection objects (such as vcTrail, vcSubnetworkConnection, etc) may be referenced by the routeDescriptionList as same route or diverse route. A new connection being created should follow the same route as a sameRoute referenced managed entity. A new connection must follow a different route than a referenced managed entity referred to as diverseRoute.

The routing information in setup operations may be either explicitly stated in the operation or the operation can point to an existing instance of the vcRoutingProfile managed entity.

Attributes

vcRoutingProfile ID: This read-only attribute provides a unique name for the managed entity instance within the management domain.

connectionTypeSupported: This read/write attribute represents the type of connection supported (e.g. point-to-point, full multipoint,...).

routeDescriptionList: This read/write attribute is a list of objects (such as Links, Subnetworks, existing connections) and their use in routing (exclude, mandatory, preferred, same route, diverse route).

maxHops: This read/write attribute is the maximum number of hops between nodes that the new connection may traverse. This attribute may be set to NULL to indicate that the maxHops criteria does not apply

Notifications

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this managed entity.

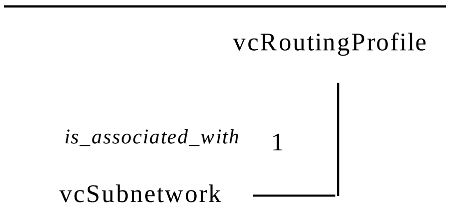
Relationships:

With vcSubnetwork: A vcRoutingProfile is associated with a given subnetwork. It can apply to any subnetworkConnection within that network.

Semi-formal representation:

IS_ASSOCIATED_WITH (1) vcSubnetwork

Graphical representation:



Additional Subnetwork Operation:

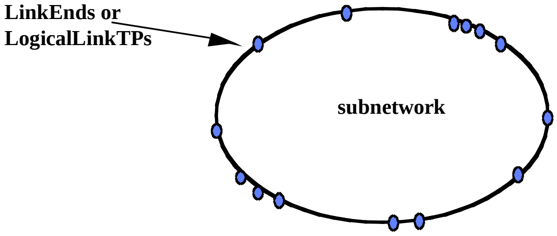
Operation: setup vcRoutingProfile

INPUT PARAMETERS:
none (see Section 5.1.2)
routeDescriptionList: list of subnetworkConnectionId, LinkId, SubnetworkId (optional)
connectionTypeSupported: broadcast, merge, composite, multipoint, pt-to-pt
maxHops: integerorNULL (optiona)
OUTPUT PARAMETERS:
vcRoutingProfile: vcRoutingProfileId
ERROR CONDITIONS:
protocol-specific errors, and
invalidSubnetworkId: subnetworkId
operationFails
BEHAVIOUR

This operation allows the requester (client) to setup a vcRoutingProfile. The reply is a vcTrailRequestId.

vcSubnetwork

A subnetwork (according to G.805) is a topological component used for carrying characteristic information. An ATM subnetwork carries ATM cells. Subnetwork are delineated by Link Ends and Logical Link TPs. Note that a subnetwork may be empty. Subnetworks are used for making subnetwork connections. This Managed Entity is specialized per layer, here the VC layer.



Attributes

Subnetwork ID: This read-only attribute provides a unique name for the managed entity instance within the management domain.

Signal Identification: This attribute represents the specific format that the resource carries.
It is fixed here to the vcLayer.

user Label: This read/write attribute identifies the managing organization.

availability Status: This read-only attribute identifies whether or not the managed entity is capable of performing its normal functions (Failed, degraded, or no unavailability condition existing).

Supported by Object List: The supportedByObjectList points to managed elements that support the subnetwork. (specific information about these elements is available through the M4 NE view)

Notifications

Attribute Value Change: This notification is used to report changes of the user label.

Managed Entity Creation: This notification is used to report the creation of an instance of this Managed Entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this Managed Entity.

Relationships:

With vcSubnetworkConnection: A subnetwork contains zero or more subnetwork connections. Note that the verb "contain" does not imply here a containment relationship in the OSI management sense.

With vcSubnetworks: A subnetwork may be partitioned into one or more subnetworks.

With vcTopologicalLink: A composite vcSubnetwork contains vcTopologicalLinks between its component subnetworks.

With vcLogicalLinkTP: a subnetwork is delineated by zero or more vcLogicalLinkTPs.

With vcLinkEnd: a subnetwork is delineated by zero or more vcLinkEnds.

Semi-Formal representation:

GROUPS (0..*) vcSubnetworkConnection

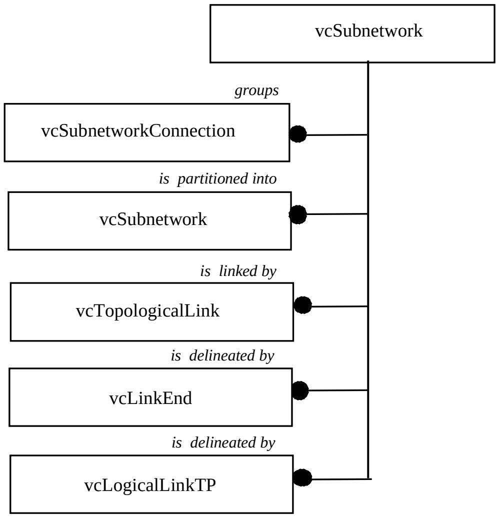
IS_PARTITIONED_INTO (0..*) vcSubnetwork

IS_LINKED_BY (0..*) vcTopologicalLink

IS_DELINEATED_BY (0..*) vcLinkEnd

IS_DELINEATED_BY (0..*) vcLogicalLinkTP

Graphical representation:



Subnetwork Query Operations

Operation: query vcSubnetwork For existing vcSubnetworkConnections

INPUT PARAMETERS:

none (see Section 5.1.2)

OUTPUT PARAMETERS:

existingVcSubnetworkConnection: set of vcSubnetworkConnectionId

ERROR CONDITIONS:

protocol-specific

BEHAVIOUR

The subnetwork plays a role of container for the existing subnetworkConnections. This operation allows the requester (client) to query the contained subnetworkConnections. It does not affect the relationship. It matches for all the contained subnetworkConnections. The reply is a set of subnetworkConnectionIds belonging to the subnetwork.

Operation: query vcSubnetwork for component vcSubnetworks

INPUT PARAMETERS:

none (see Section 5.1.2)

OUTPUT PARAMETERS:

componentVcSubnetworks : set of vcSubnetworkIds

ERROR CONDITIONS:

protocol-specific, and
unpartitionedSubnetwork : SubnetworkId

BEHAVIOUR:

A subnetwork may be partitioned in lower-level subnetworks. This query is an downward query the corresponding direct component subnetworks for component subnetworks. There may be more than one component subnetworks. The addresses subnetwork plays a role of container for the existing subnetworkConnections. This operation allows the requester (client) to query the component subnetworks. It does not affect the relationship. The query raises an error condition if the subnetwork is at the lowest-level of partitioning. It matches for all the component Subnetworks. The reply is a set of component subnetworks.

Operation: query vcSubnetwork for vcTopologicalLinks between its component vcSubnetworks

INPUT PARAMETERS:

set of component vcSubnetworkIds (optional)

OUTPUT PARAMETERS:

containedVcTopologicalLinks : set of vcTopologicalLinkIds

ERROR CONDITIONS:

protocol-specific, and
unpartitionedSubnetwork : SubnetworkId

BEHAVIOUR

Each topologicalLink connects two component subnetworks. This operation allows the requester (client) to query the composite subnetwork for the contained topologicalLinks. It does not affect the relationship. It matches for all the contained topologicalLinks. The reply is a set of topologicalLinkIds.

Operation: query vcSubnetwork For Delineating vcLinkEnds and vcLogicalLinkTPs

INPUT PARAMETERS:

none (see Section 5.1.2)

OUTPUT PARAMETERS:

delineatingVcTopologicalLinkTPs : set of vcLinkEndIds and vcLogicalLinkTPIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

This operation allows the requester (client) to query the Subnetwork Managed Entity for the link termination points that delineate it. It does not affect the relationship. It matches for all the delineating vcLinkEnds and vcLogicalLinkTPs. The reply is a set of vcLinkEndIds and vcLogicalLinkIds.

Subnetwork: SubnetworkConnection management operations

Operation: set up vcSubnetworkConnection

INPUT PARAMETERS:
see Section 5.1.2, and
networkCTPa : choice of networkCTPId, or Descriptor
networkCTPz : choice of networkCTPId, or Descriptor
administrativeState (optional)
retainedResource (optional)

Descriptor:
interfaceId (Choice of networkCTPId, or server TTPId)
vpi (optional)
vci (optional)
trafficDescriptors (optional)
qos (optional)
OUTPUT PARAMETERS:
newSNC : SNCId
networkCTPa: networkCTPId
networkCTPz: networkCTPId
ERROR CONDITIONS:
protocol-specific addressing errors
incorrectTerminationPoints : networkCTPId
reflectedTPDisabled : networkCTPId
reflectedTPLocked: networkCTPId
networkCTPConnected : networkCTPId
non-matchingDescriptors: set of networkCTPId
operationFails
BEHAVIOUR:

This operation allows the requester (user) to set-up a point-to-point connection between two non-connected networkCTPs of the addressed subnetwork. The networkCTPs to connect are identified directly or indirectly. In the latter case, a set of optional descriptors may be provided (vpi, vci, traffic descriptors, qos). An error condition will be raised if the termination points are incorrect (e.g. do not belong to the subnetwork), if the two networkCTPs are already used, if they do not have matching traffic descriptors, or if the subnetwork is unable to provide sufficient bandwidth (operations failure). The result of the operation is:

- the creation of two networkCTPs and a subnetworkConnection associated to both of them in the case of the subnetwork creating the networkCTPs
- the creation of a subnetworkConnection associated to two existing networkCTPs in the case of networkCTPs already existing in the subnetwork.

Operation: modify vcSubnetworkConnection

INPUT PARAMETERS:

see Section 5.1.2, and
networkCTPa : choice of networkCTPId, or Descriptor
networkCTPz : choice of networkCTPId, or Descriptor

Descriptor:

vpi (optional)
vci (optional)
trafficDescriptors (optional)
qos (optional)

OUTPUT PARAMETERS:

SNC : SNCId

ERROR CONDITIONS:

protocol-specific addressing errors
incorrectTerminationPoints : networkCTPId
reflectedTPDisabled : networkCTPId
reflectedTPLocked: networkCTPId
non-matchingDescriptors: set of networkCTPId
operationFails

BEHAVIOUR:

This operation allows the requester (user) to modify a connection between networkCTPs of the addressed subnetwork. The networkCTPs to modify are identified directly. A set of optional descriptors may be provided (vpi, vci, traffic descriptors, qos). An error condition will be raised if the termination points are incorrect (e.g. do not belong to the subnetwork), or if the subnetwork is unable to provide sufficient bandwidth (operations failure).

Operation: addTps To SubnetworkConnection:

INPUT PARAMETERS:

see Section 5.1.2, and
subnetworkConnection : subnetworkConnectionId
networkCTPz : choice of networkCTPId, or Descriptor
AdministrativeState (optional)

Descriptor:

interfaceId (Choice of networkCTPId, or server TTPId)
vpi (optional)

vci (optional)
trafficDescriptors (optional)
qos (optional)
OUTPUT PARAMETERS:
networkCTPz: networkCTPId
ERROR CONDITIONS:
protocol-specific addressing errors
incorrectSubnetworkConnection : subnetworkConnectionId
incorrectTerminationPoints : networkCTPId
reflectedTPDisabled : networkCTPId
reflectedTPLocked: networkCTPId
networkCTPConnected : networkCTPId
non-matchingDescriptors: set of networkCTPId
operationFails
BEHAVIOUR:

This operation allows the requester (user) to add a point to a multipoint connection. The networkCTP to connect are identified directly or indirectly. In the latter case, a set of optional descriptors may be provided (vpi, vci, traffic descriptors, qos). An error condition will be raised if the termination points are incorrect (e.g. do not belong to the subnetwork), if the networkCTP is already used, or if the subnetwork is unable to provide sufficient bandwidth (operations failure). The result of the operation is:

- the association of a networkCTP with a subnetworkConnection,
- possibly, the creation of a networkCTP.

Operation*:* release vcSubnetworkConnection

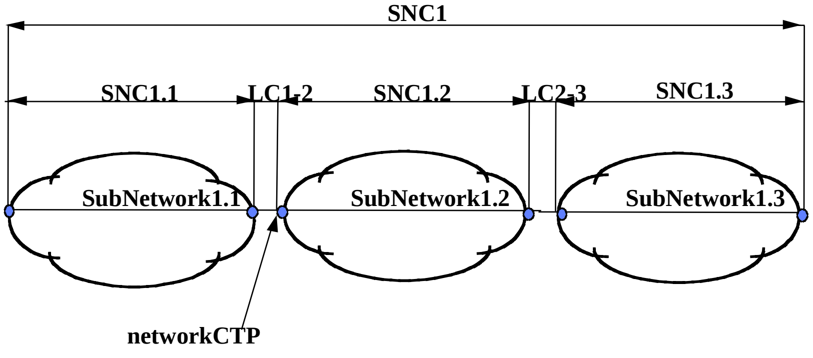
INPUT PARAMETERS:
none (see Section 5.1.2)
Choice of networkCTP: set of networkCTPId
OUTPUT PARAMETERS:
vcSubnetworkConnectionID: vcSubnetworkConnectionId
released networkCTP: set of networkCTPId
ERROR CONDITIONS:
protocol-specific addressing errors
incorrectvcNetworkCTPs : networkCTPId
networkCTPNotConnected : networkCTPId
incorrectsubnetworkConnection: subnetworkConnectionId
BEHAVIOUR

This operation allows for the release of an vcSubnetworkConnection between two connected networkCTPs of the same subnetworks, the subnetworkConnection or the networkCTPs involved being identified directly. It matches for the subnetworkConnection or the networkCTP. The reply is the released subnetworkConnectionID and the released networkCTPs.

vcSubnetworkConnection

This managed entity represents a G.805 subnetwork connection (SNC), i.e. "a transport entity which transfers information across a subnetwork. It is formed by the association of "ports" at the boundary of the subnetwork." This entity is explicitly created by a network management function.

A subnetwork connection in a composite subnetwork consists of a series of subnetworkConnections and vcLinkConnections. A subnetworkConnection cannot be created between a composite subnetwork and one of its component subnetwork. The figure below shows this relationship (in this Figure, subnetwork connection SNC1 is decomposed as follows: SNC1=SNC1.1+LC1-2+SNC1.2+LC2-3+SCN1.3).



SNC: subnetworkConnection
LC: linkConnection

Attributes

vcSubnetworkConnection ID: This read-only attribute provides a unique name for the managed entity instance within the management domain.

Directionality: This attribute is always set to "bidirectional."

availability Status: This read-only attribute identifies whether or not the managed entity is capable of performing its normal functions (Failed or no unavailability condition existing).

Administrative State: This read/write attribute is used to lock and unlock cell flow through the subnetwork connection.

User Label: This read/write attribute identifies the customer to which the service is delivered.

restorableIndicator: This read/write attribute is used to configure the connection as restorable or non-restorable.

retainedResource: This read/write attribute indicates if the managed entity instance needs to be retained when component of a composite connection (linkConnection, subnetworkConnection), or when supporting a linkConnection (trail) *provisionType:* This read/write attribute indicates whether the route for the associated subnetworkConnection is specified by the administrator (manual) or determined by the system (automatic) that may include managing and managed entities of the subnetwork

Notifications

State Change: This notification is used to report changes to the State attributes of this managed entity. The notification shall identify the state attribute that changed, its old value, and its new value.

Attribute Value Change: This notification is used to report changes of the user label.

Managed Entity Creation: This notification is used to report the creation of an instance of this Managed Entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this Managed Entity.

Relationships:

With networkCTPs: A subnetwork connection has at least two networkCTPs.

With subnetworkConnections and vcLinkConnections: A composite subnetworkConnection is made of multiple linkConnections (at least one) and inner subnetworkConnections (at least two).

With Quality Of Transport Descriptors: The traffic descriptors may be grouped in a separate entity common to multiple managed entities. This relationship is not formally represented here, since this separate entity has not been defined explicitly here.

With routingProfiles: A subnetwork connection may be constrained by a routingProfile.

Semi-formal representation:

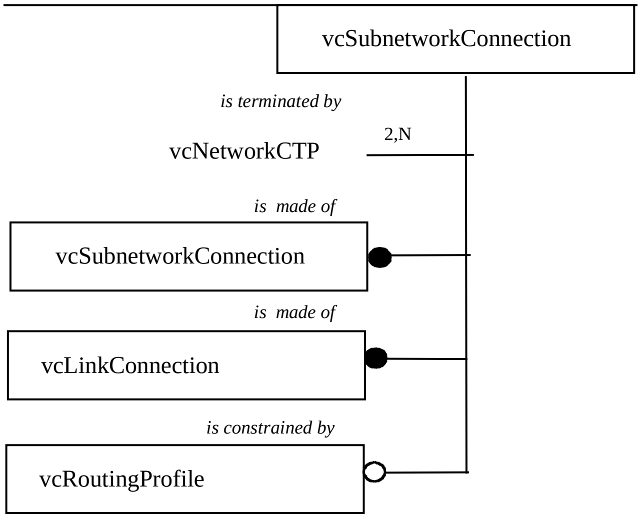
IS_TERMINATED_BY (2..*) networkCTPs

IS_MADE_OF (0..*) linkConnection -- composite case: at least one

IS_MADE_OF (0..*) subnetworkConnection -- composite case: at least two

IS_CONSTRAINED_BY (0..1) routingProfile

Graphical representation:



subnetworkConnection: subnetworkConnection query operations

Operation: query subnetworkConnection For terminating networkCTPs

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
Terminating networkCTPs: set of networkCTPIds
ERROR CONDITIONS:
protocol-specific

BEHAVIOUR:

A point-to-point subnetworkConnection is terminated on two networkCTPs. This operation allows the requester (client) to query the terminating Managed Entities. It does not affect the relationship. It matches for the two terminating networkCTPs. The reply is a set of networkCTPIds.

Operation: query vcSubnetworkConnection For Component vcSubnetworkConnections

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
componentSubnetworkConnections: set of subnetworkConnectionIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR:

A subnetworkConnection in a partitioned subnetwork can be decomposed into link connections and subnetwork connections. This query allows the requester (client) to query the component subnetwork connections (linkConnection can be derived indirectly, once the component subnetwork connections are known). It does not affect the relationship. It matches for all subnetworkConnections. The reply is a set of subnetworkConnectionIds.

vcSubnetworkConnection: Trace Operations

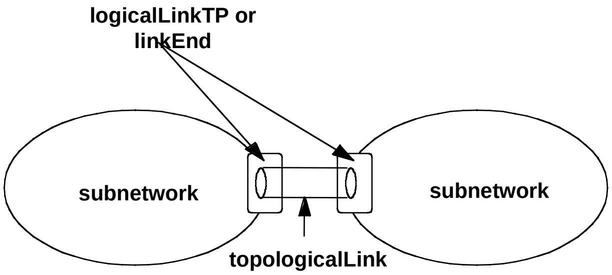
Operation: vcSubnetworkConnection Connection Trace

INPUT PARAMETERS:
no additional input parameters
OUTPUT PARAMETERS:
Sequence of atmLinks or external interface points (linkTP), and associated virtual id values. In cases of VC connection traces where VP Level information is available, the VC Link should be expanded to include the VP Level (server level) trace information.
{ (vcLinkId | vcLinkTPId) virtualId { (vpLinkId | vpLinkTPId) virtualId }* }*
ERROR CONDITIONS:
unable to perform trace on connection
BEHAVIOUR:

This operations determines the path of the vcSubnetworkConnection and returns the path at the lowest possible level supported by the managed system. For example, at the lowest level of subnetwork partitioning. The connection trace returns the virtual id (VPI for VP LND connections or VPI/VCI for VC LND connections) for each atmLink or external interface point (linkTP) of the connection. For VC connections, the trace should be examined at both the VC level as well as the VP Level, if both LNDs are under the purview of the managed system. In cases of multipoint connection, the results should be returned in a breadth first fashion.

vcTopologicalLink

A topological link is a link between two subnetworks. A unary link, terminated by linkEnds, is a topological link that corresponds directly with an underlying server trail. A topological link terminated by logicalLinkTPs represents either a composite link (a group of unary links) or partitioned link (a portion of a unary link). There can be multiple topological links between subnetworks. A topological link cannot be created between a composite subnetwork and one of its component subnetworks. This entity may be explicitly created by the network management system.



Attributes

vcTopologicalLink ID: This read-only attribute provides a unique name for the managed entity instance within the management domain.

Signal Identification: This read-only attribute, set at creation, describes the signal that is transferred across the link. Here, it is fixed to VC.

Directionality: This attribute is always set to "bidirectional."

Operational State: This read-only attribute identifies whether or not this instance of the link managed entity is capable of performing its normal function (i.e., transport ATM cells).

Provisioned Bandwidth: This read/write attribute identifies the maximum amount of bandwidth configured for the link.

Available Bandwidth: This read-only attribute identifies the amount of bandwidth left on the link.

RestorationMode: This read/write attribute is used to configure the restoration mode of a link as: unavailable for routing and re-routing, available for routing and not re-routing; available for re-routing and not routing; or available for both routing and rerouting.

Customer Identification: This string identifies the customer who may use a private link. If the value of this attribute is set to NULL, then the link may be assumed to be a non-private link. Only connections of the customer identified by the customerId attribute shall be established across a private link.

Weight: This integer value describes the relative weight of using the link. The specific value of this attribute is determined by the manager who sets the linkWeight parameter. This attributed takes on a NULL value in cases where the link is not assigned a specific weight.

Notifications

Attribute Value Change: This notification is used to report changes of the bandwidth values.

State Change: This notification is used to report changes to the State attributes of this managed entity. The notification shall identify the state attribute that changed, its old value, and its new value.

Managed Entity Creation: This notification is used to report the creation of an instance of this Managed Entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this Managed Entity.

Relationships:

With linkConnections: A topologicalLink is a group of link connections sharing the same extremities. This relationship involves one and only one instance of the topologicalLink managed entity, and zero or more instances of the linkConnection managed entity.

With logicalLinkTP: A topologicalLink that represents a partitioned or composite link has two logical link termination points, one on each subnetwork that it is linking.

With linkEnd: A topologicalLink that represents a unary link has two logical end points, one on each subnetwork that it is linking.

With subnetwork: One topologicalLink has a relationship with the two and only two subnetworks that it is linking. A topologicalLink cannot exists without the subnetworks being identified.

With vcNetworkAccessProfile: Each vcTopologicalLink may use one atmNetworkAccessProfile.

semi-formal representation:

GROUPS (0..*) vcLinkConnections

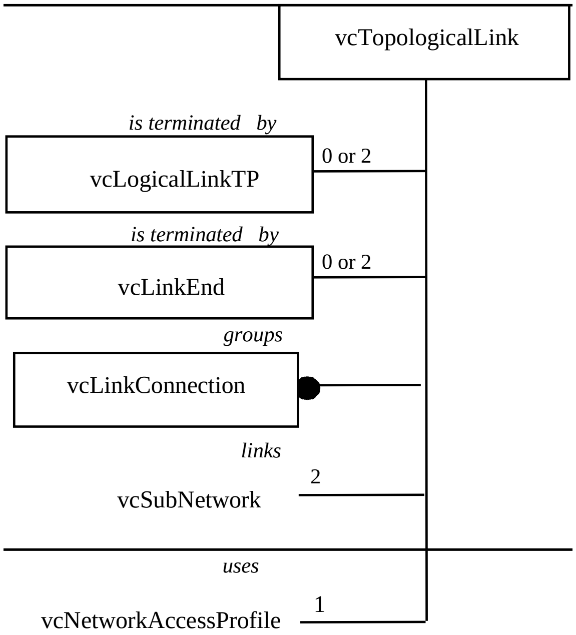
IS_TERMINATED_BY (0 or 2) vcLogicalLinkTP

IS_TERMINATED_BY (0 or 2) vcLinkEnd

LINKS (2) vcSubnetworks

USES (1) vc/vpNetworkAccessProfile

graphical representation:



vcTopologicalLink Query Operations:

Operation: query vcTopologicalLink For Contained vcLinkConnections

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
containedLinkConnections: set of linkConnectionIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

The topologicalLink plays a role of container for the existing linkConnections. This operation allows the requester (client) to query the contained managed entities. It does not affect the relationship. The reply is a set of linkConnections belonging to the topologicalLink.

Operation*:* query vcTopologicalLink For Terminating vcLinkEnds or vcLogicalLinkTPs

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatingTopologicalLinkTPs: set of linkEndIds or logicalLinkTPIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each topologicalLink is terminated by either two linkEnds or two logicalLinkTPs. This operation allows the requester (client) to query the terminating managed entities. It does not affect the relationship. It matches for the terminating linkEnds or logicalLinkTPs. The reply is a set of either linkEnds or logicalLinkTPs belonging to the topologicalLink.

Operation: query vcTopologicalLink For Delineated vcSubnetworks

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
delineatedSubnetworks: set of subnetworkIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each topologicalLink delineates two subnetworks. This operation allows the requester (client) to query the delineated managed entities. It does not affect the relationship. It matches for the two delineated subnetworks. The reply is a set of two subnetworkIds.

vcTopologicalLink: vcLinkConnection management operations

Operation: set up vcLinkConnection

INPUT PARAMETERS:
see Section 5.1.2, and
networkCTPa : choice of networkCTPId, or Descriptor
networkCTPz : choice of networkCTPId, or Descriptor
retainedResource (optional)

Descriptor:
interfaceId (Choice of networkCTP, or server TTPId, or linkEnd or logicalLinkTP,
or topologicalLink, or subnetworkId)
vpi (optional)
vci (optional)
trafficDescriptors (optional)
qos (optional)
OUTPUT PARAMETERS:
newVcLinkConnection : vcLinkConnectionId
networkCTPa: networkCTPId
networkCTPz: networkCTPId
ERROR CONDITIONS:
protocol-specific addressing errors, and
incorrectTerminationPoints : networkCTPId
reflectedTPDisabled : networkCTPId
reflectedTPLocked: networkCTPId
topologicalLinkLocked: topologicalLinkId
networkCTPInUse : networkCTPId
non-matchingDescriptors: set of networkCTPId
operationFails
BEHAVIOUR:

This operation allows the requester (user) to set-up a linkConnection between two non-connected networkCTPs of two subnetworks, identified directly or indirectly. In the latter case, a set of optional descriptors may be provided (vpi, vci, traffic descriptors, qos). An error condition will be raised if the networkCTPs are incorrect (e.g. do not belong to the subnetwork), if the two networkCTPs are already used, if they do not have matching traffic descriptors, or if the subnetwork is unable to provide sufficient bandwidth (operations failure). The result of the operation is:

- the creation of two networkCTPs and an linkConnection associated to both of them in the case of the subnetwork creating the networkCTPs
- the creation of a linkConnection associated to two existing networkCTPs in the case of networkCTPs already existing in the subnetwork.

This operation applies only if both subnetworks to be connected are visible to the requester.

Operation: modify vcLinkConnection

INPUT PARAMETERS:

see Section 5.1.2, and
networkCTPa : choice of networkCTPId, or Descriptor
networkCTPz : choice of networkCTPId, or Descriptor

Descriptor:
vpi (optional)
vci (optional)
trafficDescriptors (optional)
qos (optional)

OUTPUT PARAMETERS:

vcLinkConnection : vcLinkConnectionId

ERROR CONDITIONS:

protocol-specific addressing errors
incorrectTerminationPoints : networkCTPId
reflectedTPDisabled : networkCTPId
reflectedTPLocked: networkCTPId
non-matchingDescriptors: set of networkCTPId
operationFails

BEHAVIOUR:

This operation allows the requester (user) to modify a connection between networkCTPs of the addressed subnetwork. The networkCTPs to modify are identified directly. A set of optional descriptors may be provided (vpi, vci, traffic descriptors, qos). An error condition will be raised if the termination points are incorrect (e.g. do not belong to the subnetwork), or if the subnetwork is unable to provide sufficient bandwidth (operations failure).

Operation: release vcLinkConnection

INPUT PARAMETERS:

none (see Section 5.1.2)
Choice of networkCTP: set of networkCTPId
OUTPUT PARAMETERS:
vcSubnetworkConnectionID: vcSubnetworkConnectionId
released networkCTPs: set of networkCTPId
ERROR CONDITIONS:
protocol-specific addressing errors
incorrect networkCTPs : networkCTPIds
networkCTPNotConnected : networkCTPId
incorrectLinkConnection: linkConnectionId
BEHAVIOUR

This operation allows for the release of a linkConnection between two connected networkCTPs of the two different subnetworks, the linkConnection or the networkCTPs involved being identified directly. It matches for the subnetworkConnection or the networkCTP. The reply is the released subnetworkConnectionID and the released networkCTPs.

vcTopologicalLink: Trace Operations

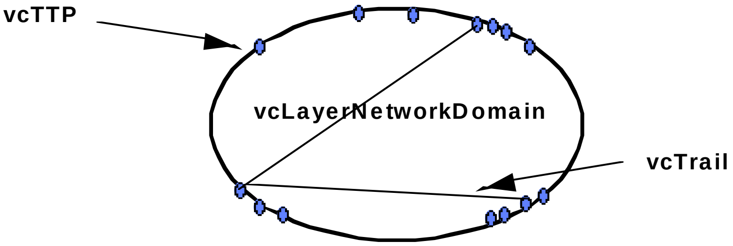
Operation: vcTopologicalLink PVC Trace

INPUT PARAMETERS:
List of vcSubnetworks
- used to scope the trace of vcSubnetworkConnections supported by the atmLink
OUTPUT PARAMETERS:
Identifier of each atmSubnetwork along with the Ids of the atmSubnetworkConnections within each atmSubnetwork that traverse the atmLink
{ vcSubnetworkId { vcSubnetworkConnectionId }* }*
ERROR CONDITIONS:
identified atmSubnetwork does not exist.
atmSubnetwork or atmLink cannot be traced by the managed system.
BEHAVIOUR:

This operation requests that the managed system perform a link PVC trace on a specific vcTopologicalLink, identifying vcSubnetworkConnections that traverse the vcTopologicalLink. The operation should identify the atmSubnetworks on which the managed system will determine the vcSubnetworkConnections that traverse the vcTopologicalLink. The request can identify multiple vcSubnetworks to scope the trace. The identified subnetworks and their connections must be visible to the managed system.

vcTrail

This managed entity represents an I.326 VC Trail. The vcTrail is always bidirectional. The vcTrail is terminated by vcTTP. This entity is specialized for the ATM VC layer. This entity is created by the management system.



Attributes

vcTrail ID: This read-only attribute provides a unique name for the managed entity instance within the management domain.

Signal Identification: This attribute represents the type of characteristic information carried by the trail. Here, it is fixed to VC.

Directionality: This attribute represents the ability of a trail to carry traffic in one or two directions. For the vcTrail, this value of this attribute is fixed to "bidirectional."

userLabel: This read/write attribute identifies the customer to which the service is delivered

Administrative State: This read/write attribute is used to lock and unlock the cell flow through the vcTrail.

availability Status: This read-only attribute identifies whether or not the managed entity is capable of performing its normal functions (Failed or no unavailability condition existing).

restorableIndicator: This read/write attribute is used to configure the trail as restorable or non-restorable.

retainedResource: This read/write attribute indicates if the managed entity instance needs to be retained when component of a composite connection (linkConnection, subnetworkConnection), or when supporting a linkConnection (trail)

Notifications

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this managed entity.

Attribute Value Change: This notification is used to report changes to the user label.

State Change: This notification is used to report changes to the states of this managed entity. The notification shall identify the state attribute that changed, its old value, and its new value.

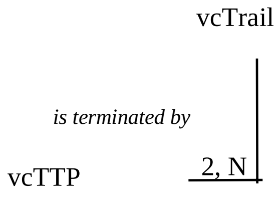
Relationships:

With vcNetworkTTP: Each VC trail is terminated by at least two *vcNetworkTTP*.

Semi-formal representation:

IS_TERMINATED_BY (2..*) vcNetworkTTP

Graphical representation:



vcTrail: vcTrail query operations

Operation: query vcTrail For terminating TTPs

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatingTTPs: set of TTPIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

A vcTrail is terminated on two vcTTPs. This operation allows the requester (client) to query the terminating TTPs. It does not affect the relationship. It matches for the associated TTPs. The reply is a set of two TTPIds.

vcTrail: Trace Operations

Operation: vcTrail Connection Trace

INPUT PARAMETERS:
no additional input parameters
OUTPUT PARAMETERS:
Sequence of atmLinks or external interface points (linkTP), and associated virtual id values. In cases of VC connection traces where VP Level information is available, the VC Link should be expanded to include the VP Level (server level) trace information.

{ (vcLinkId | vcLinkTPId) virtualId { (vpLinkId | vpLinkTPId) virtualId }* }*

ERROR CONDITIONS:

unable to perform trace on connection

BEHAVIOUR:

This operations determines the path of the vcTrail and returns the path at the lowest possible level supported by the managed system. For example, at the lowest level of subnetwork partitioning. The connection trace returns the virtual id (VPI for VP LND connections or VPI/VCI for VC LND connections) for each atmLink or external interface point (linkTP) of the connection. For VC trails, the trace should be examined at both the VC level as well as the VP Level, if both LNDs are under the purview of the managed system. In cases of multipoint connection, the results should be returned in a breadth first fashion.

vcTrailRequest

This managed entity represents a deferred request of the vcLayerNetworkDomain to either set-up, release, modify, or alter the end-points (multipoint case) of a vcTrail. If the requestType is not setup, the relationship to the vcTrail is established when the instance is created. In the case where requestType is setup, the relationship to vcTrail is established when the setup action activates a trail.
The atmTrailRequest object provides a mechanism to track scheduled requests made to the vcLayerNetworkDomain or vpLayerNetworkDomain.
It is created as result of an operation on the vcLayerNetworkDomain.

Attributes

vcTrailrequest ID: This read-only attribute provides a unique name for the managed entity instance within the management domain.

Request Status: This read-only attribute represents the status of the vcTrailRequest. It takes on values: not scheduled, scheduled, suspended, user canceled, being handled, or completed. This attribute is set when the managed entity is created.

requestType: This read-only attribute describes the type of request. It takes on values such as: setup, modify, release, addTps, or removeTps. This attribute is set when the managed entity is created.

requestCommittedTime: This read-only attribute describes the time at which the NML commits to performing the action. This attribute is set when the managed entity is created.

Notifications

Managed Entity Creation: Used to report the creation of an instance of this managed entity.

Managed Entity Deletion: Used to report the deletion of an instance of this managed entity.

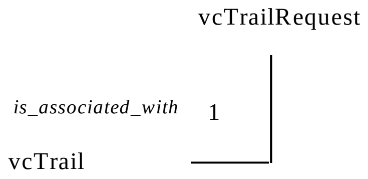
Relationships:

With vcTrail: A vcTrailRequest is associated to and an existing vcTrail. A vcTrailRequest pertains to at most one vcTrail; a vcTrail is altered by zero or more vcTrailRequests.

Semi-formal representation:

IS_ASSOCIATED_WITH (1) vcTrail

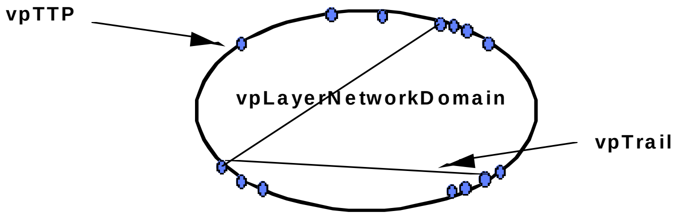
Graphical representation:



vpLayerNetworkDomain

The layer network domain is defined to support the requirement for independent layer management, here the VP layer.

An ATM layer is concerned with the generation and transfer of characteristic information, i.e. ATM cells. The layer network domain managed entity represents the part of the ATM layer which is available to a managing system through the M4 interface. It contains only managed entities from a single ATM layer, here the VP layer. Note that it is assumed that a LayerNetworkDomain contains one and only one subnetwork, which can be further decomposed.



There may be several layer network domains within a single network.

It is assumed that the layer network domain is created automatically at the installation of the superior network managed entity. The automatic creation of instances of this managed entity shall be reported over the M4 interface to the managing system.

The managing system may subsequently create and delete other instances of the layer network domain provided there are no dependent entities.

Attributes

Signal Identification: This read-only attribute represents the characteristic information of the layer network domain. Here, it is fixed to VP.

User Label: This read/write attribute allows a manager to represent additional information about the layer network domain

Notifications

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this managed entity.

Attribute Value Change: This notification is used to report changes of the user label.

Relationships:

Note that the layer network domain acts as a container for entities which provide access to a transport layer such as trails and network trail termination points. These are contained in layer network domain rather than a subnetwork because they can not be partitioned, as is possible for a subnetwork.

With vpTTP: A vpLayerNetworkDomain is delimited by zero or more vpTTPs.

With vpTrail: A vpLayerNetworkDomain groups zero or more vpTrails.

With vpSubnetwork: A vpLayerNetworkDomain is partitioned into zero or more vpSubnetworks.

With vpTrailRequest: A vpLayerNetworkDomain can have zero or more vptrailRequests modifying the trails it groups.

Semi-Formal representation:

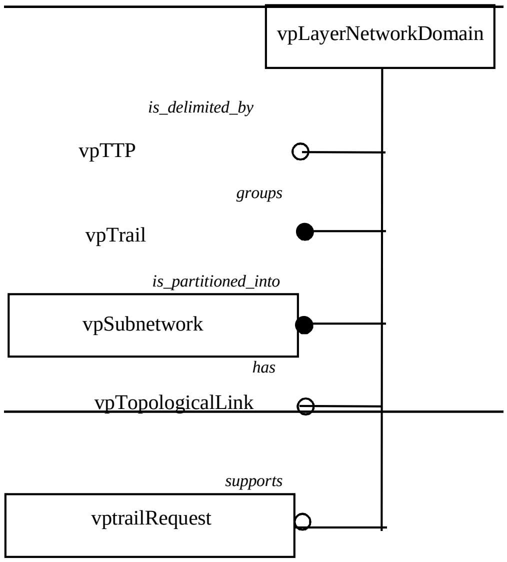
IS_DELIMITED_BY (0..*) vpTTP

GROUPS (0..*) vpTrail

IS_PARTITIONED_INTO (0..*) vpSubnetwork

SUPPORTS (0..*) vpTrailRequests

Graphical representation:



vpLayerNetworkDomain Query Operations:

Operation: query vpLayerNetworkDomain For Delimiting vpTTPs

INPUT PARAMETERS:

none (see Section 5.1.2)

OUTPUT PARAMETERS:

delimitingVpTTPs : set of vpTTPIds

ERROR CONDITIONS:

protocol-specific

BEHAVIOUR

Each vpTTP is delimiting a vpLayerNetworkDomain. This operation allows the requester (client) to query the delimiting Managed Entities. It does not affect the relationship. It matches for all the delimiting vpTTPs. The reply is a set of vpTTPIds.

Operation: query vpLayerNetworkDomain For existing vpTrails

INPUT PARAMETERS:

none (see Section 5.1.2)

OUTPUT PARAMETERS:

existingVpTrail: set of vpTrailId

ERROR CONDITIONS:

protocol-specific

BEHAVIOUR

The vpLayerNetworkDomain plays a role of container for the existing vpTrails. This operation allows the requester (client) to query the contained vpTrails. It does not affect the relationship. It matches for all the contained vpTrails. The reply is a set of vpTrailIds belonging to the vpLayerNetworkDomain.

Operation: query vpLayerNetworkDomain for component vpSubnetwork

INPUT PARAMETERS:

none (see Section 5.1.2)

OUTPUT PARAMETERS:

componentVpSubnetwork: vpSubnetworkId

ERROR CONDITIONS:

protocol-specific

BEHAVIOUR:

A vpLayerNetworkDomain contains one vpSubnetwork. This query is an downward query for the contained vpSubnetwork. It does not affect the relationship. It matches for the contained vpSubnetwork that is a direct component of the vpLayerNetworkDomain. The reply is a vpSubnetworkId.

vpLayerNetworkDomain: vpTrail set-up operations

Operation: set up vpTrail set-up

INPUT PARAMETERS:

see Section 5.1.2, and
vpTTPa : choice of vpNetworkCTPId, or Descriptor
vpTTPz : choice of vpNetworkCTPId, or Descriptor
AdministrativeState (optional)
retainedResource (optional)

Descriptor:
interfaceId (Choice of server TTPId)
vpi (optional)
vci (optional)
trafficDescriptors (optional)
qos (optional)

OUTPUT PARAMETERS:
newVpTrail : vpTrailId
vpTTPa: vpTTPId
vpTTPz: vpTTPId

ERROR CONDITIONS:
protocol-specific addressing errors
incorrectTerminationPoints : vpTTPId
reflectedTTPDisabled : vpTTPId
reflectedTTPLocked: vpTTPId
vpTrailTerminationPointConnected : vpTTPId
non-matchingDescriptors: set of vpTTPId
operationFails

BEHAVIOUR:

This operation allows the requester (user) to set-up a point-to-point Trail between two non-connected TTPs of the addressed Layer Network Domain, identified directly, or identified by an interface within which the Layer Network Domain selects a point. In the latter case, a set of optional descriptors may be provided (vpi, vci, traffic descriptors, qos). An error condition will be raised if the termination points are incorrect (i.e. do not belong to the Layer Network Domain), if the two TTPs are already used, if they do not have matching traffic descriptors, or if the Layer Network Domain is unable to provide sufficient bandwidth (operations failure). The result of the operation is:

- the creation of two TTPs and a Trail associated to both of them in the case of the Layer Network Domain creating the vpTTPs
- the creation of a Trail associated to two existing TTPs in the case of TTPs already existing in the Layer Network Domain.

Operation: setup vpTrailRequest

INPUT PARAMETERS:
none (see Section 5.1.2)
requestActionInfo: setup, modify, release, addTps, removeTps
requestCommittedTime (optional)
relatedTrails: zero or one vcTrailId (optional)

OUTPUT PARAMETERS:
vpTrailrequest: vpTrailRequestId

ERROR CONDITIONS:
protocol-specific errors, and
invalidTrailId: trailId
invalidTime
operationFails

BEHAVIOUR

This operation allows the requester (client) to setup a vpTrailRequest.. The reply is a vpTrailRequestId.

Operation: addTps To Multipoint Trail:

INPUT PARAMETERS:

see Section 5.1.2, and
trail : trailId
ttpZ : choice of ttpId, or Descriptor
AdministrativeState (optional)

Descriptor:

interfaceId (Choice of networkCTPId, or server TTPId)
vpi (optional)
vci (optional)
trafficDescriptors (optional)
qos (optional)

OUTPUT PARAMETERS:

ttpZ: ttpId

ERROR CONDITIONS:

protocol-specific addressing errors
incorrecttrail : trailId
incorrectTerminationPoints : ttpId
reflectedTPDisabled : ttpId
reflectedTPLocked: ttpId
trailTerminationPointConnected : ttpId
non-matchingDescriptors: set of ttpId
operationFails

BEHAVIOUR:

This operation allows the requester (user) to add a point to a multipoint trail. The TTP to connect are identified directly or indirectly. In the latter case, a set of optional descriptors may be provided (vpi, vci, traffic descriptors, qos). An error condition will be raised if the termination points are incorrect (e.g. do not belong to the LND), if the TTP is already used, or if the LND is unable to provide sufficient bandwidth (operations failure). The result of the operation is:

- the association of a TTP with a trail,
- possibly, the creation of a TTPTP.

Operation: release vpTrail

INPUT PARAMETERS:

Choice of VpTTP : set of vpTTPIds

OUTPUT PARAMETERS:

released vpTrail: vpTrailId
released vpTTPs: set of vpTTPId

ERROR CONDITIONS:

protocol-specific addressing errors
incorrectTerminationPoints : vpTTPId
vpTrailTerminationPointNotConnected : vpTTPId
incorrectTrails: vpTrailId

BEHAVIOUR:

This operation allows the requester (user) to release a point-to-point Trail between two connected TTPs of the addressed Layer Network Domain, identified directly, identified by one of the connected TTPs, or identified by the pair of the TTPs. An error condition will be raised if the termination points are incorrect (i.e. do not belong to the Layer Network Domain), if the two TTPs are not connected, or if the Trail is not connected. The result of the operation is:

- the deletion of two TTPs and the Trail associated to both of them.

vpLayerNetworkDomain: Link Management Operations:

Operation: Setup vpTopologicalLink

INPUT PARAMETERS:

see Section 5.1.2, and
linkTPa : choice of linkTPId, or linkDetails
linkTPz : choice of linkTPId, or linkDetails

linkTPId: linkEndId or logicalLinkTPId

linkDetail:
ingressBandwitdth
egressBandwidth
interfaceId
choice of atmNetworkAccessProfileID or profileDetails

ProfileDetails:
vpiOrVciRange,
maxNumActiveVPCAllowed,
maxEgressBandwitdth,
maxIngressBandwidth,
maxNumActiveVCCAllowed

OUTPUT PARAMETERS:

newLink : linkId
linkTPa: linkTPId
linkTPz: linkTPId
atmNetworkAccessProfileA: atmNetworkAccessProfileId
atmNetworkAccessProfileZ: atmNetworkAccessProfileId

ERROR CONDITIONS:

protocol-specific addressing errors
incorrectTerminationPoints : linkTPId

linkTerminationPointConnected : linkTPId
non-matchingDetails(profile or link): set of linkTPId, interfaceId, or atmProfileId
operationFails

BEHAVIOUR:

This operation sets up a point-to-point link between two component subnetworks or Nes in the atmSubnetwork. The linkTPs are identified either directly or indirectly with an interface identifier, available bandwidth, and an atmAccessProfile identifier, or a set of descriptors providing VPI/VCI range, etc. This approach allows to create the linkTPs at the same time if needed. An error condition is raised if the link termination points are incorrect, already used, do not have matching range or bandwidth, or if the interface is unable to provide sufficient bandwidth.

Operation: Make External vpLinkEnd

INPUT PARAMETERS:
linkTPendpoint: choice of server layer TTPId(s)
linkTPprofile: choice of atmNetworkAccessProfileId, or ProfileDetails
userLabel

ProfileDetails:
vpiOrVciRange,
maxNumActiveVPCAllowed,
totalEgressBandwidth,
totalIngressBandwidth,
maxNumActiveVCCAllowed

OUTPUT PARAMETERS:
newLinkTP : linkTPId

ERROR CONDITIONS:
protocol-specific addressing errors
serverTTP unavailable or used
incorrectTerminationPoints : linkTPId
linkTerminationPointConnected : linkTPId
non-matchingDetails(profile): set of linkTPId, interfaceId, or atmProfileId
operationFails

BEHAVIOUR:
This operation creates an instance of the atmLinkEnd object, that represents an interface to an external network, along with the objects in the server layer network domain, if needed, that support the external atmLinkTP. The external atmLinkEnd terminates a link that is not visible across the management interface, the link to an external network. The underlying server TTP may be identified directly (if it already exists) or indirectly, in which case the managed system may select or create the appropriate server TTP to support the atmLinkEnd. Indirect server TTP identification may indicate: interface identifier, desired total bandwidth, an atmNetworkAccessProfile identifier, or a set of descriptors providing VPI/VCI range, etc. This approach allows the possible creation of associated server layer trail terminations at the same time as the atmLinkEnd is created. An error condition is raised if the server trail

termination point is incorrect, already used, does not have matching range or bandwidth, or if the interface is unable to provide sufficient bandwidth.

Upon creation of the atmLinkEnd, the managed system is responsible for determining and setting: availableIngressBandwidth, availableEgressBandwidth, maxAssignableIngressBandwidth, maxAssignableEgressBandwidth, atmNetworkAccessProfilePointer, and the serverTTPList attributes. If a new atmNetworkAccessProfile object instance is needed, the managed system is responsible for creating it.

Operation: Remove External vpLinkEnd

INPUT PARAMETERS:

linkTPId : vpLinkEndId
removeServerTTPIndicator: Optional, Boolean

OUTPUT PARAMETERS:

removedLinkTP : linkTPId

ERROR CONDITIONS:

protocol-specific addressing errors
incorrectTerminationPoints : linkTPId
alreadyConnected: existing connections
operationFails

BEHAVIOUR:
This ACTION removes an instance of the atmLinkEnd object, that represents an interface to an external network. The external atmLinkEnd terminates a link that is not visible across the management interface, the link to an external network. An error condition is raised if the atmLinkEnd is incorrect or supports any connections.

Upon removal of the atmLinkEnd, the managed system is responsible for determining if the associated atmNetworkAccessProfile object instance is shared by another atmLinkEnd, atmLogicalLinkTP, or atmLink. If the atmNetworkAccessProfile is not shared, the managed system is responsible for deleting it. Also, the managed system shall remove the reference to the atmLinkEnd from the supportedLinkTPList attribute of all subnetworks that supported the removed external atmLinkEnd.

The removeServerTTPIndicator, may be used if it is possible to remove the server layer TTP instance while removing the external atmLinkEnd.

Input: atmLinkEndId. If removing a VC Layer atmLinkEnd: removeServerTTPIndicator

Output: Confirmation or Error Conditions.

Errors: protocol-specific addressing errors, noSuchTpInstance, alreadyConnected, operationFails.

Results: Removal of atmLinkEnd and an associated atmNetworkAccessProfile. Possible removal of VP TTP, VP CTP, and atmNetworkTrafficDescriptorProfile for VC Layer atmLinkEnd removal.

Operation: release vpTopologicalLink

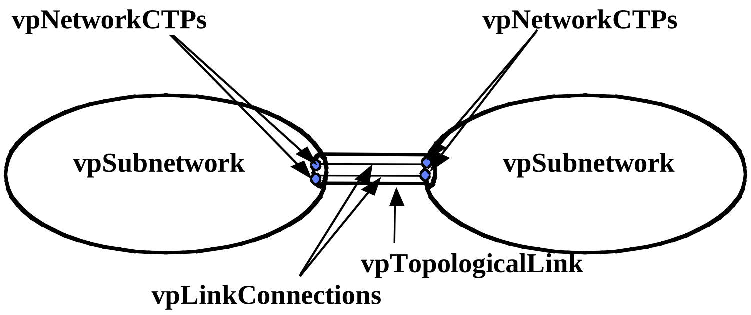
INPUT PARAMETERS:
see Section 5.1.2, and
Choice of vpLinkTPId; vpLinkEndId; linkId
OUTPUT PARAMETERS:
released vpTopologicalLink: linkId
released vpLinkTermination: set of vpLinkEnds or vpLogicalLinkTPs
released underlying server TTPs: set of TTPId (only if not a logical link)
ERROR CONDITIONS:
protocol-specific addressing errors
incorrectTerminationPoints : vcLinkTPId, or vcLinkEndId
incorrectLink: vcLinkId
BEHAVIOUR:

This operation allows the requester (user) to release a point-to-point Link between two connected LinkEnds or LogicalLinkTPs of the addressed Layer Network Domain, identified directly, identified by one of the connected endpoints. An error condition will be raised if the termination points are incorrect (i.e. do not belong to the Layer Network Domain), or if the Link is not connected. The result of the operation is:

- the deletion of two link endpoints, and the link associated to both of them. The underlying server trail is also deleted in cases where the link is not a logical link.

vpLinkConnection

This managed entity represents a I.326 link connection, derived from the G.805 definition, i.e. "a transport entity which transfers information between "ports" across a link." This entity is explicitly created by a network management function. A linkConnection cannot be created between a composite subnetwork and one of its component subnetworks. Only point-to-point linkConnections are supported.



Attributes

vpLinkConnection ID: This read-only attribute provides a unique name for the managed entity instance within the management domain.

Signal Identification: This read-only attribute, set at creation, describes the signal that is transferred across the link. Here, it is fixed to VP.

Directionality: This attribute is always set to "bidirectional."

User Label: This read/write attribute provides an arbitrary label corresponding to the connection which is established.

availability Status: This read-only attribute identifies whether or not the managed entity is capable of performing its normal functions (Failed or no unavailability condition existing).

Administrative State: This read/write attribute is used to lock and unlock cell flow through the link connection.

retainedResource: This read/write attribute indicates if the managed entity instance needs to be retained when component of a composite connection (linkConnection, subnetworkConnection), or when supporting a linkConnection (trail)

Notifications

Attribute Value Change: This notification is used to report changes of the user label.

State Change: This notification is used to report changes to the State attributes of this managed entity. The notification shall identify the state attribute that changed, its old value, and its new value.

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this managed entity.

Relationships:

With vpTopologicalLink: A topological link is a group of link connections sharing the same extremities. This relationship involves one and only one instance of the link managed entity, and zero or more instances of the linkConnection Managed Entity.

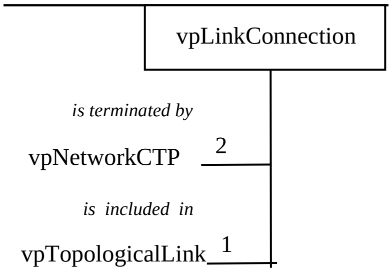
With vpNetworkCTPs: A vpLinkConnection has two vpNetworkCTPs, one on each subnetwork that it is linking.

semi-formal representation:

IS_INCLUDED_IN (1) vpTopologicalLink

IS_TERMINATED_BY (2) vpNetworkCTP

graphical representation:



vpLinkConnection Query Operations:

Operation: query vpLinkConnection For Containing vpTopologicalLink

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
containingVpTopologicalLinks: vpTopologicalLinkId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

The vpTopologicalLink plays a role of container for the existing vpLinkConnections. This operation allows the requester (client) to query the containing Managed Entities. It does not affect the relationship. It matches for the containing vpTopologicalLink. The reply is the vpTopologicalLinkId.

Operation: query vpLinkConnection For terminating vpNetworkCTPs

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatingNetworkCTPs: set of networkCTPIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

A vpLinkConnection is terminated on two vpNetworkCTPs. This operation allows the requester (client) to query the terminating Managed Entities. It does not affect the relationship. It matches for the two terminating vpNetworkCTPs. The reply is a set of two vpNetworkCTPIds.

vpLinkEnd

This managed entity is used to represent the termination of a pure topological Link at the VP-layer. In the VP LND, a Link End represents an ATM interface associated with the underlying transport facility. In addition, interface and server trail related information may be represented in the ATM Link End. That is, the Link End may be used to represent the appropriate server trail TP information, removing the need to represent the server trail TPs across the M4 network view interface.

Attributes

vpLinkEnd ID: This read-only attribute provides a unique name for the managed entity instance in the VP LND.

Administrative State: This settable attribute allows for the configuration of the administrative state of the ATM Interface represented by the vpLinkEnd.

Availability Status: This read-only attribute describes the operational status (working, degraded, not-working) of the ATM Interface represented by the vpLinkEnd.

Egress Maximum Assignable Bandwidth: This read only attribute identifies the maximum amount of bandwidth assignable on the link in the Egress direction (outbound or away from the ATM NE).

Ingress Maximum Assignable Bandwidth: This read only attribute identifies the maximum amount of bandwidth assignable on the link in the Ingress direction (inbound or towards the ATM NE).

Egress available Bandwidth: This read-only attribute identifies the amount of bandwidth left on the link in the Egress direction (outbound or away from the ATM NE).

Ingress available Bandwidth: This read-only attribute identifies the amount of bandwidth left on the link in the Ingress direction (inbound or towards the ATM NE).

User Label: This string may be used to describe additional information about the atmLinkEnd, such as a circuit identifier.

Link TP Type: Describes the interface type that the atmLinkEnd supports: UNI, inter-NNI, intra-NNI, or unconfigured.

Loopback Location Identifier: A code used for OAM cell loopback purposes. Incoming OAM Loopback cells with a Loopback Location field value that matches the value of the loopbackLocationIdentifier attribute shall be looped-back over the interface.

ILMI Virtual Identifier: This optional attribute identifies the VPI/VCI value used over the UNI to support ILMI

Supporting NE Location: This parameter identifies the location identifier of the NE that supports the vpLinkEnd.

Supporting Circuit Pack Location: This parameter identifies the location identifier of the circuit pack that supports the vpLinkEnd.

Server TTP Name: This optional attribute is a descriptive name of the Server TTP that is represented by the vpLinkEnd.

Server TTP Characteristic Information Type: This optional attribute indicates the type of the Server TTP that is represented by the vpLinkEnd.

Server TTP Port Id: This optional attribute indicates port id of the Server TTP that is represented by the vpLinkEnd.

Server TTP Operational State: This optional attribute indicates current operational state of the Server TTP that is represented by the vpLinkEnd.

Server TTP Technology Specific Additional Information: This optional attribute contains a list of server layer technology specific configuration information about the Server TTP that is represented by the vpLinkEnd.

Cell Scrambling Enable: This optional attribute allows cell scrambling to be activated or deactivated on the ATM Interface represented by the vpLinkEnd.

Subscriber Address: This optional read/write attribute describes the subscriber address associated with the vpLinkEnd.

Prefered Carrier: This optional read/write attribute provides and identification of the preferred carrier if it is directly assigned to the vpLinkEnd.

Notifications

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this managed entity.

Attribute Value Change: This notification is used to report changes to the attribute changes of this managed entity. The notification shall identify the attribute that changed, its old value, and its new value.

Relationships

With vpTopologicalLink: Each vpTopologicalLink may be terminated by two instances of the vpLinkEnd managed entity.

With vpSubnetwork: One vpLinkEnd managed entity is associated with one or more vpSubnetworks.

With serverTTPs: Each vpLinkEnd may be supported by one instance of a TTP managed entity in the serverLayer (a Transport network-level server TTP or where the NE view server TTP).

With vpNetworkAccessProfile: Each vpLinkEnd may use one atmNetworkAccessProfile.

With vpNetworkCTP: Existing Connection Termination Points: A list of CTPs within the same layer network domain that are supported by the vpLinkEnd. This provides the association between the vpLinkEnd (and underlying server trail) and the same layer network domain CTPs supported at the atmLinkEnd. That is, a VP vpLinkEnd identifies the VP CTPs supported at the interface point. In cases where VP and VC are managed together, this attribute may be used to list both VP and VC Layer networkCTPs.

semi-formal representation:

IS_ASSOCIATED_WITH (1..*) vpSubnetwork

IS_ASSOCIATED_WITH (0..1) serverTTP

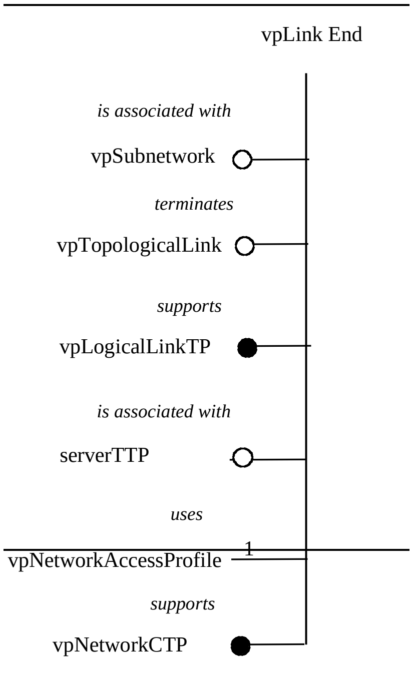
TERMINATES (0..1) vpTopologicalLink

SUPPORTS (0..*) vpLogicalLinkTP

USES (1) vpNetworkAccessProfile

SUPPORTS (0..*) vpNetworkCTPs

graphical representation:



vpLinkEnd: vpLinkEnd Query Operations

Operation: query vpLinkEnd For Terminated vpTopologicalLink

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatedTopologicalLinks: set of topologicalLinkIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR:

Each topologicalLink may be terminated by two vpLinkEnds. This operation allows the requester (client) to query the terminated topologicalLink. It does not affect the relationship. It matches for terminated topologicalLink. The reply is an topologicalLinkId.

Operation: query vpLinkEnd For Delineated vpSubnetwork

INPUT PARAMETERS:
see Section 5.1.2
OUTPUT PARAMETERS:
delineatedSubnetworks: subnetworkIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each vpLinkEnd delineates a subnetwork. This operation allows the requester (client) to query the delineated subnetwork. It does not affect the relationship. It matches for the delineated subnetwork. The reply is a subnetworkIds.

Operation*:* query vpLinkEnd For associated serverTTP

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
associatedServerTTP: serverTTPId

ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each vpLinkEnd may be associated with a serverTTP. This operation allows the requester (client) to query for the associated managed entity. It does not affect the relationship. It matches for the associated serverTTP. The reply is the associated serverTTP.

Operation: associate vpLinkEnd with supporting serverTTP

INPUT PARAMETERS:
none (see Section 5.1.2)
associatedServerTTP

OUTPUT PARAMETERS:
associatedServerTTP: serverTTPId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each vpLinkEnd at the VP-level is supported by a serverTTP managed entity, which provides transport for it. This operation allows the requester (client) to associate the vpLinkEnd with a single serverTTP. This operation may take place at the creation of the vpTopologicalLink or of the vpLinkEnd. The reply is the associated serverTTP.

vpLinkEnd: Trace Operations

Operation: vpLinkEnd PVC Trace

INPUT PARAMETERS:
List of vcSubnetworks and/or vpSubnetworks
- to be analyzed for the trace of vcSubnetworkConnections and / or vpSubnetworkConnections that terminate on the vpLinkEnd
OUTPUT PARAMETERS:
Identifier of each atmSubnetwork along with the Ids of the atmSubnetworkConnections within each atmSubnetwork that terminate on the vpLinkEnd
{ (vpSubnetworkId { vpSubnetworkConnectionId }*) |
(vcSubnetworkId { vcSubnetworkConnectionId }*) }*
ERROR CONDITIONS:
identified atmSubnetwork does not exist.
atmSubnetwork or vpLinkEnd cannot be traced by the managed system.
BEHAVIOUR:

This operation requests that the managed system perform a link PVC trace on a specific vpLinkEnd. The operation should identify the vpSubnetworks and/or vcSubnetworks on which the managed system will determine the vp and/or vc SubnetworkConnections that terminate on the vpLinkEnd. The request can identify multiple ATM Subnetworks to be analyzed. If the request is performed on a vpLinkEnd, it can identify atmSubnetworks in both the VP and VC Layer Network Domain in order to identify the VP and VC subnetwork connections that terminate on the vpLinkEnd.

The identified subnetworks and their connections must be visible to the managed system. For example, if the request is performed on vpLinkEnd and the request identifies an atmSubnetwork in the VC layer network domain, both the vpLinkEnd and the vcSubnetwork need to be under the purview of the managed system. In this case, the result of the action would identify the vcSubnetworkConnections of the identified vcSubnetwork that terminate on the vpLinkEnd.

vpLogicalLinkTP

This managed entity is used to represent the termination of a topologicalLink that is a partitioned or composite link at the VP-layer. The vpLogicalLinkTP managed entity is used to store any link-level configuration data. This managed entity is created by the management system.

Attributes

vpLogicalLinkTP ID: This read-only attribute provides a unique name for the managed entity instance in the VP LND.

Egress Maximum Assignable Bandwidth: This read only attribute identifies the maximum amount of bandwidth assignable on the logical link in the Egress direction (outbound or away from the ATM NE).

Ingress Maximum Assignable Bandwidth: This read only attribute identifies the maximum amount of bandwidth assignable on the logical link in the Ingress direction (inbound or towards the ATM NE).

Egress available Bandwidth: This read-only attribute identifies the amount of bandwidth left on the logical link in the Egress direction (outbound or away from the ATM NE).

Ingress available Bandwidth: This read-only attribute identifies the amount of bandwidth left on the logical link in the Ingress direction (inbound or towards the ATM NE).

VPI Range: This read/write parameter identifies the range of VPI values that may be used over the logical link.

User Label: This string may be used to describe additional information about the atmLogicalLinkTP, such as a circuit identifier.

Notifications

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this managed entity.

Attribute Value Change: This notification is used to report changes to the attribute changes of this managed entity. The notification shall identify the attribute that changed, its old value, and its new value.

Relationships

With vpTopologicalLink: Each vpTopologicalLink may be terminated by an instance of the vpLogicalLinkTP managed entity.

With vpLinkEnd: Each vpLogicalLinkTP of a partitioned or composite vcTopologicalLink may be supported by one or more instances of the vpLinkEnd managed entity.

With vpSubnetwork: One vpLogicalLinkTP managed entity is associated with one or more vpSubnetwork.

With vpNetworkAccessProfile: Each vpLogicalLinkTP may use one atmNetworkAccessProfile. Note that vpNetworkAccessProfile may restrict for partitioned links and extend for composite links, the characteristics described in the vpNetworkAccessProfile associated with the vpLinkEnd(s).

With vpNetworkCTP: Existing Connection Termination Points: A list of CTPs within the same layer network domain that are supported by the atmLogicalLinkTP. This attribute provides the association between the atmLogicalLinkTP and the same layer network domain CTPs supported at the atmLogicalLinkTP. That is, a VP atmLogicalLinkTP identifies the VP CTPs supported at the interface point. In cases where VP and VC are managed together, this attribute may be used to list both VP and VC Layer networkCTPs.

semi-formal representation:

TERMINATES (0..1) vpTopologicalLink

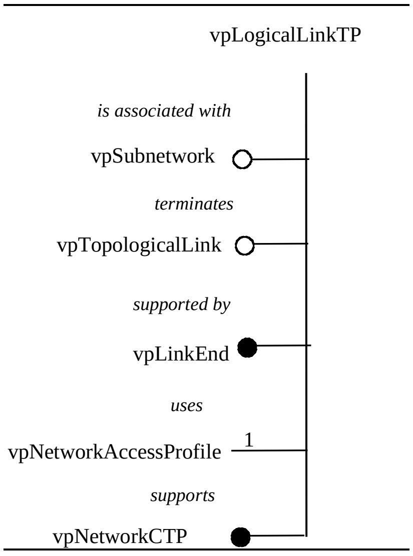
IS SUPPORTED BY (1..*) vpLinkEnd

IS_ASSOCIATED_WITH (1..*) vpSubnetwork

USES (1) vpNetworkAccessProfile

SUPPORTS (0..*) vpNetworkCTPs

graphical representation:



vcLogicalLinkTP: vcLogicalLinkTP Query Operations

Operation: query vpLogicalLinkTP For Terminated vpTopologicalLink

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatedTopologicalLinks: set of topologicalLinkIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR:

Each topologicalLink may be terminated by two LogicalLinkTPs. This operation allows the requester (client) to query the terminated topologicalLink. It does not affect the relationship. It matches for terminated topologicalLink. The reply is a topologicalLinkId.

Operation: query vpLogicalLinkTP For Delineated vpSubnetwork

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
delineatedSubnetworks: subnetworkIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each logicalLinkTP delineates a subnetwork. This operation allows the requester (client) to query the delineated subnetwork. It does not affect the relationship. It matches for the delineated subnetwork. The reply is a subnetworkIds.

Operation*:* query vpLogicalLinkTP for associated vpLinkEnds

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
associatedVpLinkEnds: vpLinkEndIds

ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each vpLogicalLinkTP is associated with one or more vpLinkEnds. This operation allows the requester (client) to query for the associated managed entity. It does not affect the relationship. It matches for the associated vpLinkEnd. The reply is the associated vpLinkEnd.

Operation: associate vpLogicalLinkTP with supporting vpLinkEnd

INPUT PARAMETERS:
none (see Section 5.1.2)

associatedVpLinkEnd
OUTPUT PARAMETERS:
associatedVpLinkEnd: vpLinkEndId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each vpLogicalLinkTP at the VP-level is supported by one or more vpLinkEnd managed entity, which represents server transport for it. This operation allows the requester (client) to associate the vpLogicalLinkTP with one or more vpLinkEnds. This operation may take place at the creation of the vpTopologicalLink or of the vpLogicalLinkTP. The reply is the associated vpLinkEnd.

vpLogicalLinkTP: Trace Operations

Operation: vcLogicalLinkTP PVC Trace

INPUT PARAMETERS:
List of vcSubnetworks and/or vpSubnetworks
- to be analyzed for the trace of vcSubnetworkConnections and / or vpSubnetworkConnections that terminate on the vpLogicalLinkTP

OUTPUT PARAMETERS:
Identifier of each atmSubnetwork along with the Ids of the atmSubnetworkConnections within each atmSubnetwork that terminate on the vpLogicalLinkTP
{ (vpSubnetworkId { vpSubnetworkConnectionId }*) |
(vcSubnetworkId { vcSubnetworkConnectionId }*) }*
ERROR CONDITIONS:
identified atmSubnetwork does not exist.
atmSubnetwork or vpLogicalLinkTP cannot be traced by the managed system.
BEHAVIOUR:

This operation requests that the managed system perform a link PVC trace on a specific vpLogicalLinkTP. The operation should identify the vpSubnetworks and/or vcSubnetworks on which the managed system will determine the vp and/or vc SubnetworkConnections that terminate on the vpLogicalLinkTP. The request can identify multiple ATM Subnetworks to be analyzed. If the request is performed on a vpLogicalLinkTP, it can identify atmSubnetworks in both the VP and VC Layer Network Domain in order to identify the VP and VC subnetwork connections that terminate on the vpLogicalLinkTP.

The identified subnetworks and their connections must be visible to the managed system. For example, if the request is performed on vpLogicalLinkTP and the request identifies an atmSubnetwork in the VC layer network domain, both the vpLogicalLinkTP and the vcSubnetwork need to be under the purview of the managed system. In this case, the result of the action would identify the vcSubnetworkConnections of the identified vcSubnetwork that terminate on the vpLogicalLinkTP.

vpNetworkAccessProfile

The vpNetworkAccessProfile managed entity contains information that describe the maximum ingress and egress bandwidth, along with the VPI values that are applies to the vpLink, vpLinkEnd, or the vpLogicalLinkTP instances that point to it.

This managed entity is created by the managing system.

Attributes

vpNetworkAccessProfile ID: This read-only attribute provides a unique name for the managed entity instance.

total Egress Bandwidth: This read/write attribute identifies the total aggregate egress bandwidth for a link or a linkTP (linkEnd or logicalLinkTP).

total Ingress Bandwidth: This read/write attribute identifies the total aggregate ingress bandwidth for a link or a linkTP (linkEnd or logicalLinkTP).

maximum Number of Active Connection Allowed: This read/write attribute identifies the maximum number of concurrently active VP (for a vpLayerNetworkDomain) or VC (for a vcLayerNetworkDomain) connections that a link or a linkTP (linkEnd or logicalLinkTP) may support.

VPI/VCI Range: This read/write attribute describes the virtual ID range (VPIs and/ or VCIs) that may be used for linkConnections associated with a link or linkTP

Notifications

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

Relationships:

None originating from this managed entity.

Operations:

None beyond the setting and querying of the read/write attributes.

vpRoutingProfile

This managed entity a set of routing constraints that can be applied to a new connection or trail during setup. A routing profile may be created automatically based on the routing description in the setup operation. The management system may also create profiles directly. Each vpSubnetworkConnection or vpTrail may point to an vpRoutingProfile. Connections should not be established (or re-established) if the routing criteria cannot be met. If maxHops is specified, the connection should not be established (or re-established) if the maximum number of hops is exceeded

The maxHops attribute is the maximum number of hops between nodes that the new connection may traverse. This attribute may be set to NULL to indicate that the maxHops criteria does not apply.

The routeDescriptionList attribute is a list of objects (such as Links, Subnetworks, existing connections) and their use in routing (exclude, mandatory, preferred, same route, diverse route).

The connection types that the routing profile supports are indicated in the connectionTypesSupported attribute. For all types of multipoint connections at least the sameRoute criteria may be applied. All of the criteria may be applied to point-to-point connections.

Managed entities (such as vpSubnetwork, vpLink, or managedElement, etc) may be referenced by the routeDescriptionList as being excluded, mandatory, or preferred. If a managed entity is described as mandatory it must be used in setting up a new connection. An attempt must be made during setup to include a managed entity described as preferred. An excluded managed entity must not be used in a connection.

Connection objects (such as vpTrail, vpSubnetworkConnection, etc) may be referenced by the routeDescriptionList as same route or diverse route. A new connection being created should follow the same route as a sameRoute referenced managed entity. A new connection must follow a different route than a referenced managed entity referred to as diverseRoute.

The routing information in setup operations may be either explicitly stated in the operation or the operation can point to an existing instance of the vpRoutingProfile managed entity.

Attributes

vpRoutingProfile ID: This read-only attribute provides a unique name for the managed entity instance within the management domain.

connectionTypeSupported: This read/write attribute represents the type of connection supported (e.g. point-to-point, full multipoint,...).

routeDescriptionList: This read/write attribute is a list of objects (such as Links, Subnetworks, existing connections) and their use in routing (exclude, mandatory, preferred, same route, diverse route).

maxHops: This read/write attribute is the maximum number of hops between nodes that the new connection may traverse. This attribute may be set to NULL to indicate that the maxHops criteria does not apply

Notifications

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this managed entity.

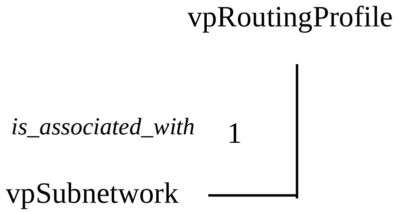
Relationships:

With vpSubnetwork: A vpRoutingProfile is associated with a given subnetwork. It can apply to any subnetworkConnection within that network.

Semi-formal representation:

IS_ASSOCIATED_WITH (1) vpSubnetwork

Graphical representation:



Additional Subnetwork Operation:

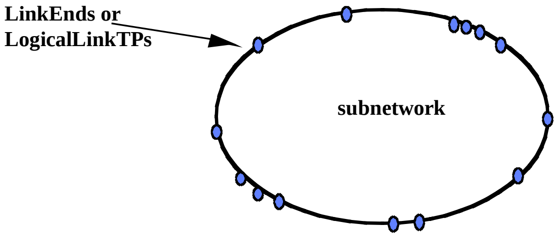
Operation: setup vpRoutingProfile

INPUT PARAMETERS:
none (see Section 5.1.2)
routeDescriptionList: list of subnetworkConnectionId, LinkId, SubnetworkId (optional)
connectionTypeSupported: broadcast, merge, composite, multipoint, pt-to-pt
maxHops: integerorNULL (optiona)
OUTPUT PARAMETERS:
vpRoutingProfile: vpRoutingProfileId
ERROR CONDITIONS:
protocol-specific errors, and
invalidSubnetworkId: subnetworkId
operationFails
BEHAVIOUR

This operation allows the requester (client) to setup a vpRoutingProfile. The reply is a vpTrailRequestId.

vpSubnetwork

A subnetwork (according to G.805) is a topological component used for carrying characteristic information. An ATM subnetwork carries ATM cells. Subnetwork are delineated by Link Ends or Logical Link TPs. Note that a subnetwork may be empty. Subnetworks are used for making subnetwork connections. This Managed Entity is specialized per layer, here the VP layer.



Attributes

Subnetwork ID: This read-only attribute provides a unique name for the managed entity instance within the management domain.

Signal Identification: This attribute represents the specific format that the resource carries.
It is fixed here to the vpLayer.

user Label: This read/write attribute identifies the managing organization.

availability Status: This read-only attribute identifies whether or not the managed entity is capable of performing its normal functions (Failed, degraded, or no unavailability condition existing).

Supported by Object List: The supportedByObjectList points to managed elements that support the subnetwork. (specific information about these elements is available through the M4 NE view)

Notifications

Attribute Value Change: This notification is used to report changes of the user label.

Managed Entity Creation: This notification is used to report the creation of an instance of this Managed Entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this Managed Entity.

Relationships:

With vpSubnetworkConnection: A subnetwork contains zero or more subnetwork connections. Note that the verb “contain” does not imply here a containment relationship in the OSI management sense.

With vpSubnetworks: A subnetwork may be partitioned into one or more subnetworks.

With vpTopologicalLink: A composite vpSubnetwork contains vpTopologicalLinks between its component subnetworks.

With vpLogicalLinkTP: a subnetwork is delineated by zero or more vpLogicalLinkTPs.

With vpLinkEnd: a subnetwork is delineated by zero or more vpLinkEnds.

Semi-Formal representation:

GROUPS (0..*) vpSubnetworkConnection

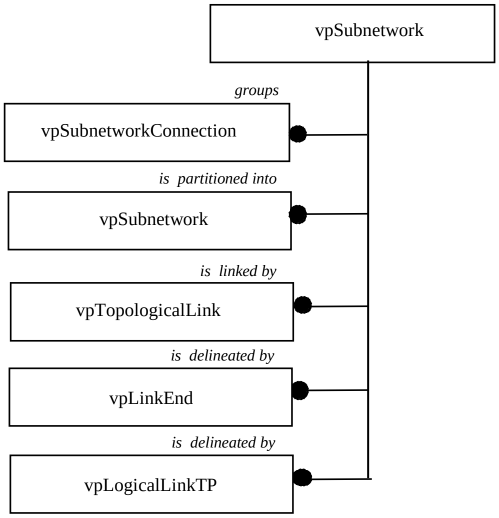
IS_PARTITIONED_INTO (0..*) vpSubnetwork

IS_LINKED_BY (0..*) vpTopologicalLink

IS_DELINEATED_BY (0..*) vpLogicalLinkTP

IS_DELINEATED_BY (0..*) LinkEnd

Graphical representation:



Subnetwork Query Operations

Operation: query vpSubnetwork For delimiting vpNetworkCTPs

INPUT PARAMETERS:

none (see Section 5.1.2)
OUTPUT PARAMETERS:
delineatingVpNetworkCTPs : set of vpNetworkCTPIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR:

The subnetwork plays a role of grouping the networkCTPs. This operation allows the requester (client) to query the grouped networkCTPs. It does not affect the relationship. It matches
for all the grouped networkCTPs. The reply is a set of networkCTPs belonging to the subnetwork.

Operation: query vpSubnetwork For existing vpSubnetworkConnections

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
existingVpSubnetworkConnection: set of vpSubnetworkConnectionId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

The subnetwork plays a role of container for the existing subnetworkConnections. This operation allows the requester (client) to query the contained subnetworkConnections. It does not affect the relationship. It matches for all the contained subnetworkConnections. The reply is a set of subnetworkConnectionIds belonging to the subnetwork.

Operation: query vpSubnetwork for component vpSubnetworks

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
componentVpSubnetworks : set of vpSubnetworkIds
ERROR CONDITIONS:
protocol-specific, and
unpartitionedSubnetwork : SubnetworkId
BEHAVIOUR:
A subnetwork may be partitioned in lower-level subnetworks. This query is an downward query the corresponding direct component subnetworks for component subnetworks. There may be more than one component subnetworks. The addresses subnetwork plays a role of container for the existing subnetworkConnections. This operation allows the requester (client) to query the component subnetworks. It does not affect the relationship. The query raises an error condition if the subnetwork is at the lowest-level of partitioning. It matches for all the component Subnetworks. The reply is a set of component subnetworks.

Operation: query vpSubnetwork for vpTopologicalLinks between its component vpSubnetworks

INPUT PARAMETERS:
set of component vpSubnetworkIds (optional)
OUTPUT PARAMETERS:

containedVpTopologicalLinks : set of vpTopologicalLinkIds

ERROR CONDITIONS:

protocol-specific, and
unpartitionedSubnetwork : SubnetworkId

BEHAVIOUR

Each topologicalLink connects two component subnetworks. This operation allows the requester (client) to query the composite subnetwork for the contained topologicalLinks. It does not affect the relationship. It matches for all the contained topologicalLinks. The reply is a set of topologicalLinkIds.

Operation: query vpSubnetwork For Connecting vpLinkEnds or vpLogicalLinkTPs

INPUT PARAMETERS:

none (see Section 5.1.2)

OUTPUT PARAMETERS:

connectingVpTopologicalLinkTPs : set of vpLinkEndIds or vpLogicalLinkTPs

ERROR CONDITIONS:

protocol-specific

BEHAVIOUR

Each vpTopologicalLinkTP is associated with vpSubnetworks. This operation allows the requester (client) to query the associated Managed Entities. It does not affect the relationship. It matches for all the associated vpLinkEnds and vpLogicalLinkTPs. The reply is a set of vpLinkEndIds and vpLogicalLinkTPs.

Subnetwork: SubnetworkConnection management operations

Operation: set up vpSubnetworkConnection

INPUT PARAMETERS:

see Section 5.1.2, and
networkCTPa : choice of networkCTPId, or Descriptor
networkCTPz : choice of networkCTPId, or Descriptor
AdministrativeState (optional)
retainedResource (optional)

Descriptor:

interfaceId (Choice of networkCTPId, or server TTPId)
vpi (optional)
vci (optional)
trafficDescriptors (optional)
qos (optional)

OUTPUT PARAMETERS:

newSNC : SNCId
networkCTPa: networkCTPId
networkCTPz: networkCTPId

ERROR CONDITIONS:

protocol-specific addressing errors

incorrectTerminationPoints : networkCTPId
reflectedTPDisabled : networkCTPId
reflectedTPLocked: networkCTPId
networkCTPConnected : networkCTPId
non-matchingDescriptors: set of networkCTPId
operationFails

BEHAVIOUR:

This operation allows the requester (user) to set-up a point-to-point connection between two non-connected networkCTPs of the addressed subnetwork. The networkCTPs to connect are identified directly or indirectly. In the latter case, a set of optional descriptors may be provided (vpi, vci, traffic descriptors, qos). An error condition will be raised if the termination points are incorrect (e.g. do not belong to the subnetwork), if the two networkCTPs are already used, if they do not have matching traffic descriptors, or if the subnetwork is unable to provide sufficient bandwidth (operations failure). The result of the operation is:

- the creation of two networkCTPs and a subnetworkConnection associated to both of them in the case of the subnetwork creating the networkCTPs
- the creation of a subnetworkConnection associated to two existing networkCTPs in the case of networkCTPs already existing in the subnetwork.

Operation: modify vpSubnetworkConnection

INPUT PARAMETERS:

see Section 5.1.2, and
networkCTPa : choice of networkCTPId, or Descriptor
networkCTPz : choice of networkCTPId, or Descriptor

Descriptor:
vpi (optional)
vci (optional)
trafficDescriptors (optional)
qos (optional)

OUTPUT PARAMETERS:

SNC : SNCId

ERROR CONDITIONS:

protocol-specific addressing errors
incorrectTerminationPoints : networkCTPId
reflectedTPDisabled : networkCTPId
reflectedTPLocked: networkCTPId
non-matchingDescriptors: set of networkCTPId
operationFails

BEHAVIOUR:

This operation allows the requester (user) to modify a connection between networkCTPs of the addressed subnetwork. The networkCTPs to modify are identified directly. A set of optional descriptors may be provided (vpi, vci, traffic descriptors, qos). An error condition will be raised if the termination points are incorrect (e.g. do not belong to the subnetwork), or if the subnetwork is unable to provide sufficient bandwidth (operations failure).

Operation: addTps To SubnetworkConnection:

INPUT PARAMETERS:
see Section 5.1.2, and
subnetworkConnection : subnetworkConnectionId
networkCTPz : choice of networkCTPId, or Descriptor
AdministrativeState (optional)

Descriptor:
interfaceId (Choice of networkCTPId, or server TTPId)
vpi (optional)
vci (optional)
trafficDescriptors (optional)
qos (optional)
OUTPUT PARAMETERS:
networkCTPz: networkCTPId
ERROR CONDITIONS:
protocol-specific addressing errors
incorrectSubnetworkConnection : subnetworkConnectionId
incorrectTerminationPoints : networkCTPId
reflectedTPDisabled : networkCTPId
reflectedTPLocked: networkCTPId
networkCTPConnected : networkCTPId
non-matchingDescriptors: set of networkCTPId
operationFails
BEHAVIOUR:

This operation allows the requester (user) to add a point to a multipoint connection. The networkCTP to connect are identified directly or indirectly. In the latter case, a set of optional descriptors may be provided (vpi, vci, traffic descriptors, qos). An error condition will be raised if the termination points are incorrect (e.g. do not belong to the subnetwork), if the networkCTP is already used, or if the subnetwork is unable to provide sufficient bandwidth (operations failure). The result of the operation is:

- the association of a networkCTP with a subnetworkConnection,
- possibly, the creation of a networkCTP.

Operation*:* release vpSubnetworkConnection

INPUT PARAMETERS:
none (see Section 5.1.2)
Choice of networkCTP : set of networkCTPIds
OUTPUT PARAMETERS:
vpSubnetworkConnectionID: vpSubnetworkConnectionId

released networkCTPs: set of networkCTPId

ERROR CONDITIONS:

protocol-specific addressing errors
incorrectVpNetworkCTPs : networkCTPIds
networkCTPNotConnected : networkCTPId
incorrectsubnetworkConnection: subnetworkConnectionId

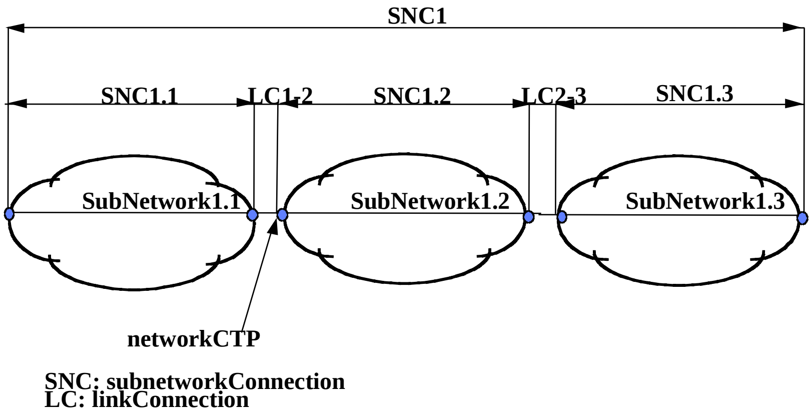
BEHAVIOUR

This operation allows for the release of an vpSubnetworkConnection between connected networkCTPs, the subnetworkConnection or the networkCTPs involved being identified directly. It matches for the subnetworkConnection or the networkCTP. The reply is the released subnetworkConnectionID and the released networkCTPs.

vpSubnetworkConnection

This managed entity represents a G.805 subnetwork connection (SNC), i.e. "a transport entity which transfers information across a subnetwork. It is formed by the association of "ports" at the boundary of the subnetwork." This entity is explicitly created by a network management function.

A subnetwork connection in a composite subnetwork consists of a series of subnetworkConnections and vpLinkConnections. A subnetworkConnection cannot be created between a composite subnetwork and one of its component subnetwork. The figure below shows this relationship (in this Figure, subnetwork connection SNC1 is decomposed as follows: SNC1=SNC1.1+LC1-2+SNC1.2+LC2-3+SCN1.3).



Attributes

vpSubnetworkConnection ID: This read-only attribute provides a unique name for the managed entity instance within the management domain.

Directionality: This attribute is always set to "bidirectional."

availability Status: This read-only attribute identifies whether or not the managed entity is capable of performing its normal functions (Failed or no unavailability condition existing).

Administrative State: This read/write attribute is used to lock and unlock cell flow through the subnetwork connection.

User Label: This read/write attribute identifies the customer to which the service is delivered.

restorableIndicator: This read/write attribute is used to connection the entity as restorable or non-restorable.

retainedResource: This read/write attribute indicates if the managed entity instance needs to be retained when component of a composite connection (linkConnection, subnetworkConnection), or when supporting a linkConnection (trail) *provisionType:* This read/write attribute indicates whether the route for the associated subnetworkConnection is specified by the administrator (manual) or determined by the system (automatic) that may include managing and managed entities of the subnetwork

Notifications

State Change: This notification is used to report changes to the State attributes of this managed entity. The notification shall identify the state attribute that changed, its old value, and its new value.

Attribute Value Change: This notification is used to report changes of the user label.

Managed Entity Creation: This notification is used to report the creation of an instance of this Managed Entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this Managed Entity.

Relationships:

With networkCTPs: A subnetwork connection has at least two networkCTPs.

With subnetworkConnections and vpLinkConnections: A composite subnetworkConnection is made of multiple linkConnections (at least one) and inner subnetworkConnections (at least two).

With routingProfiles: A subnetwork connection may be constrained by a routingProfile.

Semi-formal representation:

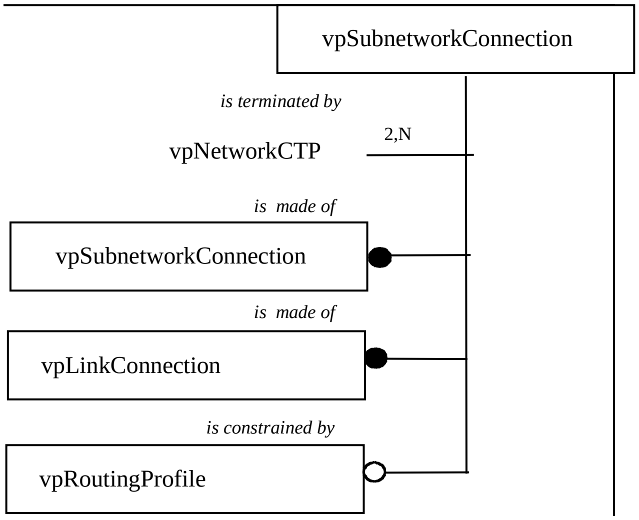
IS_TERMINATED_BY (2..*) networkCTPs

IS_MADE_OF (0..*) linkConnection -- composite case: at least one

IS_MADE_OF (0..*) subnetworkConnection -- composite case: at least two

IS_CONSTRAINED_BY (0..1) routingProfile

Graphical representation:



subnetworkConnection: subnetworkConnection query operations

Operation: query subnetworkConnection For terminating networkCTPs

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatingNetworkCTPs: set of networkCTPIds
ERROR CONDITIONS:
protocol-specific

BEHAVIOUR:

A point-to-point subnetworkConnection is terminated on two networkCTPs. This operation allows the requester (client) to query the terminating Managed Entities. It does not affect the relationship. It matches for the two terminating networkCTPs. The reply is a set of networkCTPIds.

Operation: query vpSubnetworkConnection For Component vpSubnetworkConnections

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
componentSubnetworkConnections: set of subnetworkConnectionIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR:

A subnetworkConnection in a partitioned subnetwork can be decomposed into link connections and subnetwork connections. This query allows the requester (client) to query the component subnetwork connections (linkConnection can be derived indirectly, once the component subnetwork connections are known). It does not affect the relationship. It matches for all subnetworkConnections. The reply is a set of subnetworkConnectionIds.

vpSubnetworkConnection: Trace Operations

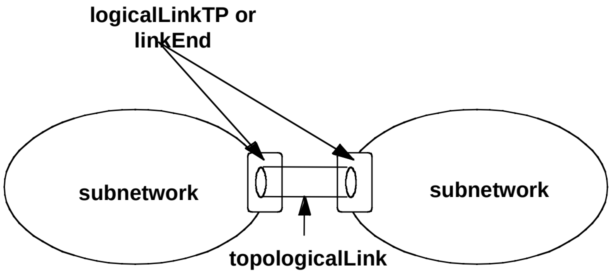
Operation: vpSubnetworkConnection Connection Trace

INPUT PARAMETERS:
no additional input parameters
OUTPUT PARAMETERS:
Sequence of atmLinks or external interface points (linkTP), and associated virtual id values.
{ (vcLinkId | vcLinkTPId) virtualId }*
ERROR CONDITIONS:
unable to perform trace on connection
BEHAVIOUR:

This operations determines the path of the vpSubnetworkConnection and returns the path at the lowest possible level supported by the managed system. For example, at the lowest level of subnetwork partitioning. The connection trace returns the virtual id (VPI for VP LND connections or VPI/VCI for VC LND connections) for each atmLink or external interface point (linkTP) of the connection. In cases of multipoint connection, the results should be returned in a breadth first fashion.

vpTopologicalLink

A topological link is a link between two subnetworks. A unary link, terminated by linkEnds, is a topological link that corresponds directly with an underlying server trail. A topological link terminated by logicalLinkTPs represents either a composite link (a group of unary links) or partitioned link (a portion of a unary link). There can be multiple topological links between subnetworks. A topological link cannot be created between a composite subnetwork and one of its component subnetworks. This entity may be explicitly created by the network management system.



Attributes

vpTopologicalLink ID: This read-only attribute provides a unique name for the managed entity instance within the management domain.

Signal Identification: This read-only attribute, set at creation, describes the signal that is transferred across the link. Here, it is fixed to VP.

Directionality: This attribute is always set to "bidirectional."

Operational State: This read-only attribute identifies whether or not this instance of the link managed entity is capable of performing its normal function (i.e., transport ATM cells).

provisioned Bandwidth: This read/write attribute identifies the maximum amount of bandwidth configured for the link.

available Bandwidth: This read-only attribute identifies the amount of bandwidth left on the link.

restorationMode: This read/write attribute is used to configure the restoration mode of a link as: unavailable for routing and re-routing, available for routing and not re-routing; available for re-routing and not routing; or available for both routing and rerouting.

Customer Identification: This string identifies the customer who may use a private link. If the value of this attribute is set to NULL, then the link may be assumed to be a non-private link. Only connections of the customer identified by the customerId attribute shall be established across a private link.

Weight: This integer value describes the relative weight of using the link. The specific value of this attribute is determined by the manager who sets the linkWeight parameter. This attributed takes on a NULL value in cases where the link is not assigned a specific weight.

Notifications

Attribute Value Change: This notification is used to report changes of the bandwidth values.

State Change: This notification is used to report changes to the State attributes of this managed entity. The notification shall identify the state attribute that changed, its old value, and its new value.

Managed Entity Creation: This notification is used to report the creation of an instance of this Managed Entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this Managed Entity.

Relationships:

With linkConnections: A topologicalLink is a group of link connections sharing the same extremities. This relationship involves one and only one instance of the topologicalLink managed entity, and zero or more instances of the linkConnection managed entity.

With logicalLinkTP: A topologicalLink that represents a partitioned or composite link has two logical link termination points, one on each subnetwork that it is linking.

With linkEnd: A topologicalLink that represents a unary link has two logical end points, one on each subnetwork that it is linking.

With subnetwork: One topologicalLink has a relationship with the two and only two subnetworks that it is linking. A topologicalLink cannot exists without the subnetworks being identified.

With vpNetworkAccessProfile: Each vpTopologicalLink may use one atmNetworkAccessProfile.

semi-formal representation:

GROUPS (0..*) vpLinkConnections

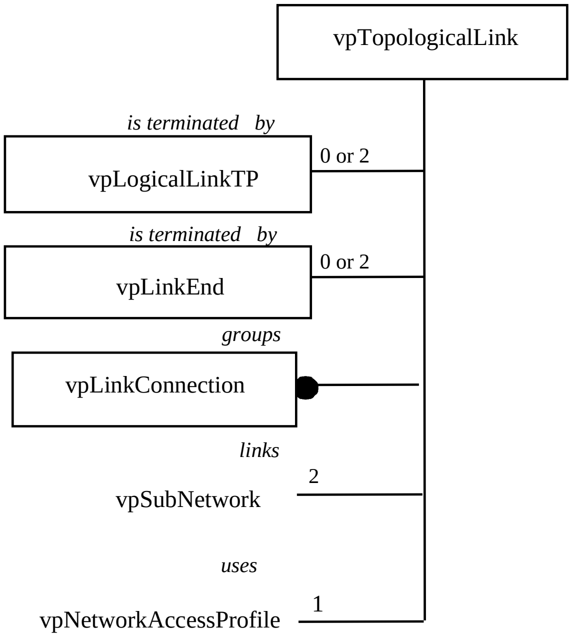
IS_TERMINATED_BY (0 or 2) vpLogicalLinkTP

IS_TERMINATED_BY (0 or 2) vpLinkEnd

LINKS (2) vpSubnetwork

USES (1) vpNetworkAccessProfile

graphical representation:



vpTopologicalLink Query Operations:

Operation: query vpTopologicalLink For Contained vpLinkConnections

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
containedLinkConnections: set of linkConnectionIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

The topologicalLink plays a role of container for the existing linkConnections. This operation allows the requester (client) to query the contained managed entities. It does not affect the relationship. The reply is a set of linkConnections belonging to the topologicalLink.

Operation*:* query vpTopologicalLink For Terminating vpLinkEnds or vpLogicalLinkTPs

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatingTopologicalLinkTPs: set of logicalLinkTPIds and linkEndIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each topologicalLink is terminated by either two linkEnds or two logicalLinkTPs. This operation allows the requester (client) to query the terminating managed entities. It does not affect the relationship. It matches for the terminating linkEnds and logicalLinkTPs. The reply is a set of either linkEnds (unary links) or logicalLinkTPs (composite or partitioned links) belonging to the topologicalLink.

Operation: query vpTopologicalLink For Delineated vpSubnetworks

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
delineatedSubnetworks: set of subnetworkIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each topologicalLink delineates two subnetworks. This operation allows the requester (client) to query the delineated managed entities. It does not affect the relationship. It matches for the two delineated subnetworks. The reply is a set of two subnetworkIds.

vpTopologicalLink: vpLinkConnection management operations

Operation: set up vpLinkConnection

INPUT PARAMETERS:
see Section 5.1.2, and
networkCTPa : choice of networkCTPId, or Descriptor
networkCTPz : choice of networkCTPId, or Descriptor
retainedResource (optional)

Descriptor:
interfaceId (Choice of networkCTP, or server TTPId, or linkEnd or logicalLinkTP,
or topologicalLink, or subnetworkId)
vpi (optional)
vci (optional)
trafficDescriptors (optional)
qos (optional)
OUTPUT PARAMETERS:
newVpLinkConnection : vpLinkConnectionId
networkCTPa: networkCTPId
networkCTPz: networkCTPId
ERROR CONDITIONS:
protocol-specific addressing errors, and
incorrectTerminationPoints : networkCTPId
reflectedTPDisabled : networkCTPId
reflectedTPLocked: networkCTPId
topologicalLinkLocked: topologicalLinkId
networkCTPInUse : networkCTPId
non-matchingDescriptors: set of networkCTPId

operationFails

BEHAVIOUR:

This operation allows the requester (user) to set-up a linkConnection between two non-connected networkCTPs of two subnetworks, identified directly or indirectly. In the latter case, a set of optional descriptors may be provided (vpi, vci, traffic descriptors, qos). An error condition will be raised if the networkCTPs are incorrect (e.g. do not belong to the subnetwork), if the two networkCTPs are already used, if they do not have matching traffic descriptors, or if the subnetwork is unable to provide sufficient bandwidth (operations failure). The result of the operation is:

- the creation of two networkCTPs and an linkConnection associated to both of them in the case of the subnetwork creating the networkCTPs
- the creation of a linkConnection associated to two existing networkCTPs in the case of networkCTPs already existing.

This operation applies only if both subnetworks to be connected are visible to the requester.

Operation: modify vpLinkConnection

INPUT PARAMETERS:

see Section 5.1.2, and
networkCTPa : choice of networkCTPId, or Descriptor
networkCTPz : choice of networkCTPId, or Descriptor

Descriptor:
vpi (optional)
vci (optional)
trafficDescriptors (optional)
qos (optional)

OUTPUT PARAMETERS:

vpLinkConnection : vpLinkConnectionId

ERROR CONDITIONS:

protocol-specific addressing errors
incorrectTerminationPoints : networkCTPId
reflectedTPDisabled : networkCTPId
reflectedTPLocked: networkCTPId
non-matchingDescriptors: set of networkCTPId
operationFails

BEHAVIOUR:

This operation allows the requester (user) to modify a connection between networkCTPs of the addressed subnetwork. The networkCTPs to modify are identified directly. A set of optional descriptors may be provided (vpi, vci, traffic descriptors, qos). An error condition will be raised if the termination points are incorrect (e.g. do not belong to the subnetwork), or if the subnetwork is unable to provide sufficient bandwidth (operations failure).

Operation: release vpLinkConnection

INPUT PARAMETERS:

none (see Section 5.1.2)
Choice of networkCTP : set of networkCTPIds

OUTPUT PARAMETERS:

vpSubnetworkConnectionID: vpSubnetworkConnectionId
released networkCTPs: set of networkCTPId

ERROR CONDITIONS:

protocol-specific addressing errors
incorrectNetworkCTPs : networkCTPIds
networkCTPNotConnected : networkCTPId
incorrectLinkConnection: linkConnectionId

BEHAVIOUR

This operation allows for the release of a linkConnection between two connected networkCTPs of the two different subnetworks, the linkConnection or the networkCTPs involved being identified directly. It matches for the subnetworkConnection or the networkCTP. The reply is the released subnetworkConnectionID and the released networkCTPs.

vpTopologicalLink: Trace Operations

Operation: vpTopologicalLink PVC Trace

INPUT PARAMETERS:

List of vcSubnetworks and/or vpSubnetworks
- to be analyzed for the trace of vcSubnetworkConnections and / or vpSubnetworkConnections supported by the atmLink

OUTPUT PARAMETERS:

Identifier of each atmSubnetwork along with the Ids of the atmSubnetworkConnections within each atmSubnetwork that traverse the atmLink

{ (vpSubnetworkId { vpSubnetworkConnectionId }*) |
(vcSubnetworkId { vcSubnetworkConnectionId }*) }*

ERROR CONDITIONS:

identified atmSubnetwork does not exist.
atmSubnetwork or atmLink cannot be traced by the managed system.

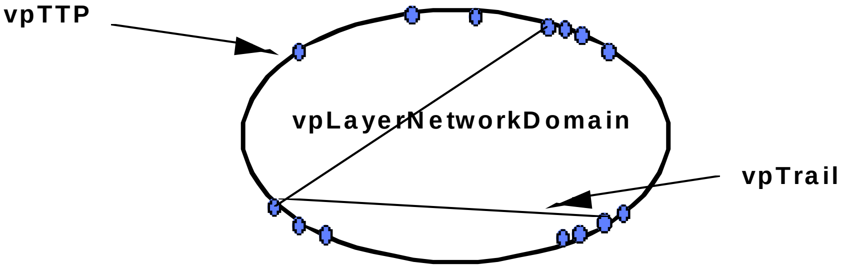
BEHAVIOUR:

This operation requests that the managed system perform a link PVC trace on a specific vpTopologicalLink. The operation should identify the vpSubnetworks and/or vcSubnetworks on which the managed system will determine the vp and/or vc SubnetworkConnections that traverse the vpTopologicalLink. The request can identify multiple ATM Subnetworks to be analyzed. If the request is performed on a vpTopologicalLink, it can identify atmSubnetworks in both the VP and VC Layer Network Domain in order to identify the VP and VC subnetwork connections that traverse the vpTopologicalLink.

The identified subnetworks and their connections must be visible to the managed system. For example, if the request is performed on vpTopologicalLink and the request identifies an atmSubnetwork in the VC layer network domain, both the vpTopologicalLink and the vcSubnetwork need to be under the purview of the managed system. In this case, the result of the action would identify the vcSubnetworkConnections of the identified vcSubnetwork that traverse the vpTopologicalLink.

vpTrail

This managed entity represents an I.326 VP Trail. The vpTrail is always bidirectional. The vpTrail is terminated by vpTTP. This entity is specialized for the ATM VP layer. This entity is created by the management system.



Attributes

vpTrail ID: This read-only attribute provides a unique name for the managed entity instance within the management domain.

Signal Identification: This attribute represents the type of characteristic information carried by the trail. Here, it is fixed to VP.

Directionality: This attribute represents the ability of a trail to carry traffic in one or two directions. For the vpTrail, this value of this attribute is fixed to "bidirectional."

User Label: This read/write attribute identifies the customer to which the service is delivered

Administrative State: This read/write attribute is used to lock and unlock the cell flow through the vpTrail.

availability Status: This read-only attribute identifies whether or not the managed entity is capable of performing its normal functions (Failed or no unavailability condition existing).

restorableIndicator: This read/write attribute is used to configure the trail as restorable or non-restorable.

retainedResource: This read/write attribute indicates if the managed entity instance needs to be retained when component of a composite connection (linkConnection, subnetworkConnection), or when supporting a linkConnection (trail)

Notifications

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this managed entity.

Attribute Value Change: This notification is used to report changes to the user label.

State Change: This notification is used to report changes to the states of this managed entity. The notification shall identify the state attribute that changed, its old value, and its new value.

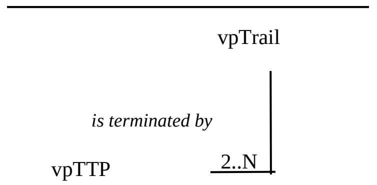
Relationships:

With vpNetworkTTP: Each Vp trail is terminated by at least two *vpNetworkTTP*.

Semi-formal representation:

IS_TERMINATED_BY (2..*) vpNetworkTTP

Graphical representation:



vpTrail: vpTrail query operations

Operation: query vpTrail For terminating TTPs

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatingTTPs: set of TTPIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

A vpTrail is terminated on two vpTTPs. This operation allows the requester (client) to query the terminating TTPs. It does not affect the relationship. It matches for the associated TTPs. The reply is a set of two TTPIds.

vpTrail: vpTrail trace operations

Operation: vpTrail Connection Trace

INPUT PARAMETERS:
no additional input parameters
OUTPUT PARAMETERS:
Sequence of atmLinks or external interface points (linkTP), and associated virtual id values.
{ (vcLinkId | vcLinkTPId) virtualId }*
ERROR CONDITIONS:
unable to perform trace on connection

BEHAVIOUR:

This operations determines the path of the vpTrail and returns the path at the lowest possible level supported by the managed system. For example, at the lowest level of subnetwork partitioning. The connection trace returns the virtual id (VPI for VP LND connections or VPI/VCI for VC LND connections) for each atmLink or external interface point (linkTP) of the connection. In cases of multipoint connection, the results should be returned in a breadth first fashion.

vpTrailRequest

This managed entity represents a deferred request of the vpLayerNetworkDomain to either set-up, release, modify, or alter the end-points (multipoint case) of an atmTrail. If the requestType is not setup, the relationship to the vpTrail is established when the instance is created. In the case where requestType is setup, the relationship to vpTrail is established when the setup action activates a trail.
The vpTrailRequest managed entity provides a mechanism to track scheduled requests made to the vpLayerNetworkDomain.
It is created as result of an operation on the vpLayerNetworkDomain or vpLayerNetworkDomain object.

Attributes

vpTrailrequest ID: This read-only attribute provides a unique name for the managed entity instance within the management domain.

Request Status: This read-only attribute represents the status of the vpTrailRequest. It takes on values: not scheduled, scheduled, suspended, user canceled, being handled, or completed. This attribute is set when the managed entity is created.

requestType: This read-only attribute describes the type of request. It takes on values such as: setup, modify, release, addTps, or removeTps. This attribute is set when the managed entity is created.

requestCommittedTime: This read-only attribute describes the time at which the NML can commit to performing the action. This attribute is set when the managed entity is created.

Notifications

Managed Entity Creation: Used to report the creation of an instance of this managed entity.

Managed Entity Deletion: Used to report the deletion of an instance of this managed entity.

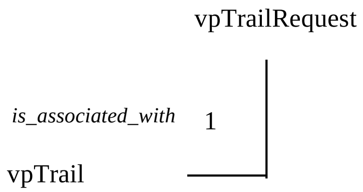
Relationships:

With vpTrail: A vpTrailRequest is associated to and an existing vpTrail. An vpTrailRequest pertains to at most one vpTrail; a vpTrail is altered by zero or more vpTrailRequests.

Semi-formal representation:

IS_ASSOCIATED_WITH (1) vpTrail

Graphical representation:



vcNetworkCTP

This managed entity is used to represent the termination of VC connections on an ATM subnetwork. An instance of the vcSubnetworkConnection or of an vcLinkConnection managed entity may be used to relate two instances of the VC Network Connection Termination Point managed entity (i.e., for point-to-point cross connection).

Instances of this managed entity may be created automatically by the subnetwork, as a result of a connection (link or subnetwork) being created, or explicitly by the management system. Similarly, instances of this managed entity may be deleted automatically by the subnetwork, as a result of a connection release request, or explicitly by the management system.

Attributes

vcCTP ID: This read-only attribute provides a unique name for the managed entity instance in the subnetwork.

VPI/VCI Value: This read-only attribute identifies the VPI/VCI value associated with the VC connection being terminated.

User Label: This read/write attribute identifies the customer to which the service is delivered.

Segment Endpoint: This boolean attribute indicates whether the vcNetworkCTP object instance has been configured to represent an end-point of a VCC Segment.

Ingress Tagging Indicator: This optional boolean attribute specifies if tagging is being used on the receive side of an ATM VCC. A value of true indicates that tagging is being used. A value of false indicates it is not being used.

Egress Tagging Indicator: This optional boolean attribute specifies if tagging is being used on the transmit side of an ATM VCC. A value of true indicates that tagging is being used. A value of false indicates it is not being used.

PM OAM Method: This optional attribute indicates the method used to setup and terminate the PM OAM monitoring activity. Valid values are TMN, OAM, or notSupported. If the value is notSupported, then PM OAM is not supported on the endpoint.

PM OAM Direction: This optional attribute indicates the desired direction(s) of transmission to monitor PM OAM. Valid directions are: away from activator (transmit), towards activator (receive), or both.

PM OAM Block Size: This optional attribute indicates the PM OAM nominal block size choice for both the receive and transmit dirctions.

PM OAM Forward Active: This optional boolean attribute is used to initiate generation of PM OAM cells in the forward direction by setting the value to true.

PM OAM Backward Active: This optional boolean attribute is used to initiate generation of PM OAM cells in the backward direction by setting the value to true.

Notifications

Alarm: This message is used to notify the management system when a failure has been detected or cleared. The following parameters shall be supplied with this notification:

- The Nature of the Alarm (i.e., see generic trouble list)
- Specific Problems (optional)
- The ID of the Managed Entity Reporting the Alarm
- The Failed Switch Component or List of Failed (or Possibly Failed) Components
- Back-up Status (optional)
  This is a Boolean indication as to whether or not the failed entity has been backed-up.
- Back-up Entity (optional)
  This is the ID of the managed entity providing back-up services to the failed entity. This parameter shall be NULL when the value of the "Back-up Status" parameter is *false*.
- Severity of Failure (critical, major, minor, warning, indeterminate, and cleared)
- Additional Information (optional)
- Proposed Repair Actions (optional)
- Time and Date Failure was Detected

Attribute Value Change: This notification is used to report changes to the attributes of this managed entity. The notification shall identify the attribute that changed, its old value, and its new value.

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this managed entity.

Relationships

.

With vcNetworkTTP: Zero or one instance of the vcNetworkTTP managed entity may exist for each instance of a vcCTP managed entity.

With subnetworkConnectiont: Zero or more of the vcNetworkCTP managed entity may exist for each instance of a vcSubnetworkConnection managed entity.

With trafficDescriptorProfile: Zero or one instance of the traffic descriptor managed entity may characterize the CTP.

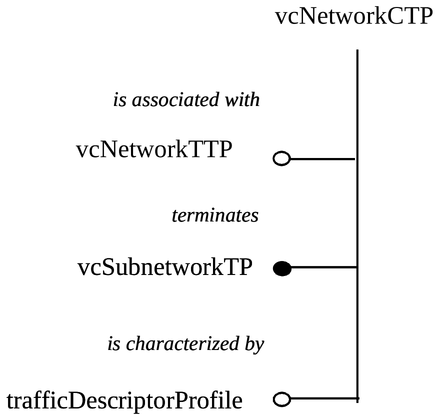
Semi-formal representation:

IS_ASSOCIATED_WITH (0..1) vcNetworkTTP

SUPPORTS (0..*) vcSubnetworkConnection

IS_CHARACTERIZED_BY (0..1) trafficDescriptorProfile

Graphical representation:



vcNetworkCTP: vcNetworkCTP Query/Association Operations

Operation: associate vcNetworkCTP with vcNetworkTTP

INPUT PARAMETERS:
none (see Section 5.1.2)
associatedVcNetworkCTP: vcNetworkCTPId
OUTPUT PARAMETERS:
associatedVcNetworkTTP : vcNetworkTTPId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR:

Each vcNetworkCTP can be associated with zero or one vcNetworkTTP. This operation allows the requester (client) to associate a vcNetworkCTP to a given vcNetworkTTP. The reply is a vcNetworkTTPId.

Operation: query vcNetworkCTP for associated vcNetworkTTP

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
associatedVcNetworkTTP : vcNetworkTTPId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR:

Each vcNetworkCTP can be associated with zero or one vcNetworkTTP. This operation allows the requester (client) to query the associated vcNetworkTTP for a given vcNetworkCTP. It does not affect the relationship. It matches for the associated vcNetworkTTP. The reply is a vcNetworkTTPId.

Operation: query vcNetworkCTP for associated subnetworkConnections

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
associatedSubnetworkConnections : set of subnetworkConnectionId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR:

Each vcNetworkCTP is associated with zero or more subnetworkConnecitons. This operation allows the requester (client) to query the associated subnetworkConnection for a given vcNetworkCTP. It does not affect the relationship. It matches for the associated subnetworkConnection. The reply is a set of subnetworkConnectionIds.

vcConnectionTerminationPoint: vcConnectionTerminationPoint Loopback Operation

Operation: loopback vcTrail at vcNetworkCTP

INPUT PARAMETERS:
none (see Section 5.1.2)
loopbackType: end-to-end, segment
loopbackLocation: interfaceId
OUTPUT PARAMETERS:
loopbackResults: passed or failed
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

This operation is used to request that the vcCTP insert a loopback OAM cell into the ATM cell stream, verify its return, and report the results of the loopback (i.e., passed or failed) back to the management system. Along with each request will be the location where the inserted OAM cell shall loop-back and an indication as to whether a segment or end-to-end OAM cell shall be used. The Loopback Location Code which indicates where the loopback is to take place may be used to identify the loopback location. Additionally, a globally unique default value (e.g., "end-point") may also be used to perform a loopback at the other end of a vcTrail.

vcNetworkTTP

This managed entity represents the point in the ATM subnetwork where the VC Trail and associated overhead (F5 OAM cells) are terminated/originated. Management systems shall configure/remove VC Trail Terminations in the ATM subnetwork by creating/deleting instances of this managed entity.

Managed entities that represent AAL functions performed above VC Trail termination points are for further study.

Attributes

vcTTP ID: This read-only attribute provides a unique name for the managed entity instance in the ATM subnetwork.

availability Status: This read-only attribute identifies whether or not the managed entity is capable of performing its normal functions (Failed or no unavailability condition existing).

PM OAM Method: This optional attribute indicates the method used to setup and terminate the PM OAM monitoring activity. Valid values are TMN, OAM, or notSupported. If the value is notSupported, then PM OAM is not supported on the endpoint.

PM OAM Direction: This optional attribute indicates the desired direction(s) of transmission to monitor PM OAM. Valid directions are: away from activator (transmit), towards activator (receive), or both.

PM OAM Block Size: This optional attribute indicates the PM OAM nominal block size choice for both the receive and transmit dirctions.

PM OAM Forward Active: This optional boolean attribute is used to initiate generation of PM OAM cells in the forward direction by setting the value to true.

PM OAM Backward Active: This optional boolean attribute is used to initiate generation of PM OAM cells in the backward direction by setting the value to true.

Notifications

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this managed entity.

Attribute Value Change Notification: This notification is used to report changes to the availability status attribute of this managed entity.

Relationships

With vcNetworkCTPs: Zero or one instance of the VC Trail Termination managed entity may exist for each instance of a vcNetworkCTP managed entity.

With vcTrail: A vcTrail is terminated by two vcTTPs.

with AAL Profile: Zero or one instance of either AAL1 Profile, AAL3/4 Profile, or AAL5 Profile describes the AAL parameters associated with the trail termination.

with Service Profile: Zero or one instance of a Service Profile describes the service interworking parameters associated with the trail termination.

Semi-formal representation:

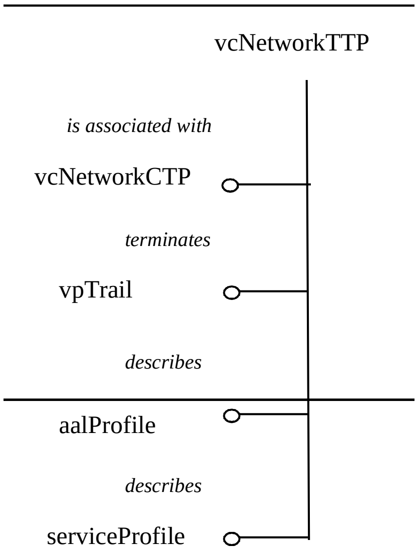
IS_ASSOCIATED_WITH (0..1) vcCTP

TERMINATES (0..1) vcTrail

DESCRIBED BY (0..1) aalProfile

DESCRIBED BY (0..1) serviceProfile

Graphical representation:



vcNetworkTTP: vcNetworkTTP Query Operations

Operation: query vcNetworkTTP for associated vcNetworkCTP

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
associatedVcCTP : vcCTPId
ERROR CONDITIONS:

protocol-specific

BEHAVIOUR:

Each vcNetworkTTP is associated with zero or one vcCTP. This operation allows the requester (client) to query the associated vcNetworkCTP. It does not affect the relationship. It matches for the associated vcNetworkCTP. The reply is a vcNetworkCTPId.

Operation: query vcTTP For terminated vcTrail

INPUT PARAMETERS:

none (see Section 5.1.2)

OUTPUT PARAMETERS:

terminatedTrail: vcTrailId

ERROR CONDITIONS:

protocol-specific

BEHAVIOUR

An vcTrail is terminated on two vcTTPs. This operation allows the requester (client) to query the terminated vcTrail. It does not affect the relationship. It matches for the associated vcTrail. The reply is a vcTrailId.

vcNetworkTTP: vcNetworkTTP Loopback Operation

Operation: loopback vcTrail at vcTTP

INPUT PARAMETERS:

see Section 5.1.2, and
loopbackType: end-to-end, segment
loopbackLocation: interfaceId

OUTPUT PARAMETERS:

loopbackResults: passed or failed

ERROR CONDITIONS:

protocol-specific

BEHAVIOUR

This operation is used to request that the vcTTP insert a loopback OAM cell into the ATM cell stream, verify its return, and report the results of the loopback (i.e., passed or failed) back to the management system. Along with each request will be the location where the inserted OAM cell shall loop-back and an indication as to whether a segment or end-to-end OAM cell shall be used. The Loopback Location Code attribute value of the UNI, interNNI, or intraNNI where the loopback is to take place may be used to identify the loopback location. Additionally, a globally unique default value (e.g., "end-point") may also be used to perform a loopback at the other end of a vcTrail.

vpNetworkCTP

This managed entity is used to represent the termination of VP connections on an ATM subnetwork. An instance of the vpSubnetworkConnection or of an vpLinkConnection managed entity may be used to relate two instances of the VP Network Connection Termination Point managed entity (i.e., for point-to-point cross connection).

Instances of this managed entity may be created automatically by the subnetwork, as a result of a connection (link or subnetwork) being created, or explicitly by the management system. Similarly, instances of this managed entity may be deleted automatically by the subnetwork, as a result of a connection release request, or explicitly by the management system.

Attributes

vpCTP ID: This read-only attribute provides a unique name for the managed entity instance in the subnetwork.

VPI Value: This read-only attribute identifies the VCI value associated with the VC connection being terminated.

User Label: This read/write attribute identifies the customer to which the service is delivered.

Segment Endpoint: This boolean attribute indicates whether the vpNetworkCTP object instance has been configured to represent an end-point of a VPC Segment.

Ingress Tagging Indicator: This optional boolean attribute specifies if tagging is being used on the receive side of an ATM VPC. A value of true indicates that tagging is being used. A value of false indicates it is not being used.

Egress Tagging Indicator: This optional boolean attribute specifies if tagging is being used on the transmit side of an ATM VPC. A value of true indicates that tagging is being used. A value of false indicates it is not being used.

PM OAM Method: This optional attribute indicates the method used to setup and terminate the PM OAM monitoring activity. Valid values are TMN, OAM, or notSupported.   If the value is notSupported, then PM OAM is not supported on the endpoint.

PM OAM Direction: This optional attribute indicates the desired direction(s) of transmission to monitor PM OAM. Valid directions are: away from activator (transmit), towards activator (receive), or both.

PM OAM Block Size: This optional attribute indicates the PM OAM nominal block size choice for both the receive and transmit dirctions.

PM OAM Forward Active: This optional boolean attribute is used to initiate generation of PM OAM cells in the forward direction by setting the value to true.

PM OAM Backward Active: This optional boolean attribute is used to initiate generation of PM OAM cells in the backward direction by setting the value to true.

Notifications

Alarm: This message is used to notify the management system when a failure has been detected or cleared. The following parameters shall be supplied with this notification:

- The Nature of the Alarm (i.e., see generic trouble list)
- Specific Problems (optional)
- The ID of the Managed Entity Reporting the Alarm
- The Failed Switch Component or List of Failed (or Possibly Failed) Components
- Back-up Status (optional)
  This is a Boolean indication as to whether or not the failed entity has been backed-up.
- Back-up Entity (optional)
  This is the ID of the managed entity providing back-up services to the failed entity. This parameter shall be NULL when the value of the "Back-up Status" parameter is *false*.
- Severity of Failure (critical, major, minor, warning, indeterminate, and cleared)
- Additional Information (optional)
- Proposed Repair Actions (optional)
- Time and Date Failure was Detected

Attribute Value Change: This notification is used to report changes to the attributes of this managed entity. The notification shall identify the attribute that changed, its old value, and its new value.

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this managed entity.

Relationships

With vpNetworkTTP: Zero or one instance of the vpNetworkTTP managed entity may exist for each instance of a vpCTP managed entity.

With subnetworkConnectiont: Zero or more of the vpNetworkCTP managed entity may exist for each instance of a subnetworkConnection managed entity.

With trafficDescriptorProfile: Zero or one instance of the Traffic Descriptor managed entity may characterize the CTP.

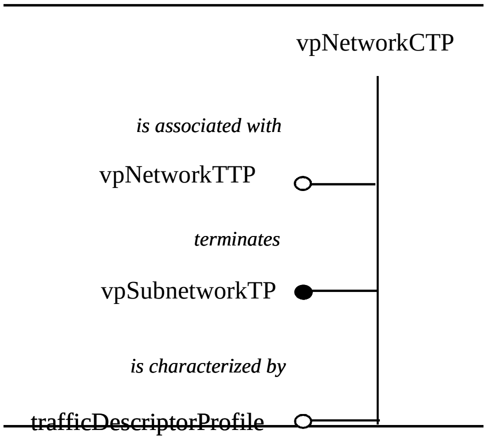
Semi-formal representation:

IS_ASSOCIATED_WITH (0..1) vpNetworkTTP

SUPPORTS (0..*) vpSubnetworkConnection

IS_CHARACTERIZED_BY (0..1) trafficDescriptorProfile

Graphical representation:



vpConnectionTerminationPoint: vcConnectionTerminationPoint Query/Association Operations

Operation: associate vpNetworkCTP with vpNetworkTTP

INPUT PARAMETERS:
none (see Section 5.1.2)
associatedVpTTP: vpTTPId
OUTPUT PARAMETERS:
associatedVpTTP : vpTTPId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR:

Each vpNetworkCTP can be associated with zero or one vpNetworkTTP. This operation allows the requester (client) to associate a vpCTP to a given vpTTP. The reply is a vpTTPId.

Operation: query vpNetworkCTP for associated vpNetworkTTP

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
associatedVpTTP : vpTTPId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR:

Each vpNetworkCTP can be associated with zero or one vpNetworkTTP. This operation allows the requester (client) to query the associated vpNetworkTTP for a given vpNetworkCTP. It does not affect the relationship. It matches for the associated vpNetworkTTP. The reply is a vpNetworkTTPId.

Operation: query vpNetworkCTP for associated subnetworkConnections

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
associatedSubnetworkConnections : list of subnetworkConnectionIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR:

Each vpNetworkCTP is associated with zero or more subnetworkConnections. This operation allows the requester (client) to query the associated subnetworkConnections for a given vpNetworkCTP. It does not affect the relationship. It matches for the associated subnetworkConnections. The reply is a subnetworkConnectionId.

vpNetworkCTP: vpNetworkCTP Loopback Operation

Operation: loopback vpTrail at vpNetworkCTP

INPUT PARAMETERS:
none (see Section 5.1.2)
loopbackType: end-to-end, segment
loopbackLocation: interfaceId
OUTPUT PARAMETERS:
loopbackResults: passed or failed
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

This operation is used to request that the vpNetworkCTP insert a loopback OAM cell into the ATM cell stream, verify its return, and report the results of the loopback (i.e., passed or failed) back to the management system. Along with each request will be the location where the inserted OAM cell shall loop-back and an indication as to whether a segment or end-to-end OAM cell shall be used. The Loopback Location Code which indicates where the loopback is to take place may be used to identify the loopback location. Additionally, a globally unique default value (e.g., "end-point") may also be used to perform a loopback at the other end of a vcTrail.

vpNetworkTTP

This managed entity represents the point in the ATM subnetwork where the VP Trail and associated overhead (F4 OAM cells) are terminated/originated. Management systems shall configure/remove VP Trail Terminations in the ATM subnetwork by creating/deleting instances of this managed entity.

Attributes

VP TTP ID: This read-only attribute provides a unique name for the managed entity instance in the ATM subnetwork.

availability Status: This read-only attribute identifies whether or not the managed entity is capable of performing its normal functions (Failed or no unavailability condition existing).

PM OAM Method: This optional attribute indicates the method used to setup and terminate the PM OAM monitoring activity. Valid values are TMN, OAM, or notSupported. If the value is notSupported, then PM OAM is not supported on the endpoint.

PM OAM Direction: This optional attribute indicates the desired direction(s) of transmission to monitor PM OAM. Valid directions are: away from activator (transmit), towards activator (receive), or both.

PM OAM Block Size: This optional attribute indicates the PM OAM nominal block size choice for both the receive and transmit dirctions.

PM OAM Forward Active: This optional boolean attribute is used to initiate generation of PM OAM cells in the forward direction by setting the value to true.

PM OAM Backward Active: This optional boolean attribute is used to initiate generation of PM OAM cells in the backward direction by setting the value to true.

Notifications

Managed Entity Creation: This notification is used to report the creation of an instance of this managed entity.

Managed Entity Deletion: This notification is used to report the deletion of an instance of this managed entity.

Attribute Value Change Notification: This notification is used to report changes to the availability status attribute of this managed entity.

Relationships

With vpNetworkCTPs: Zero or one instance of the VP Trail Termination managed entity may exist for each instance of a vcNetworkCTP managed entity.

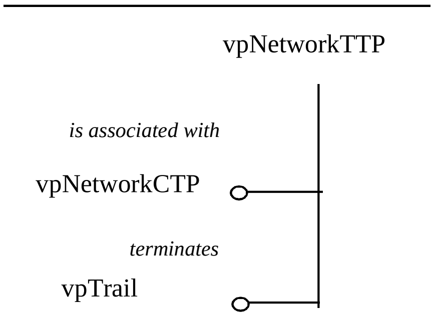
With vpTrail: A vcTrail is terminated by two vpTTPs.

Semi-formal representation:

IS_ASSOCIATED_WITH (0..1) vpNetworkCTP

TERMINATES (0..1) vpTrail

Graphical representation:



vpNetworkTTP: vpNetworkTTP Query Operations

Operation: query vpNetworkTTP for associated vpNetworkCTP

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
associatedVcCTP : vcCTPId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR:

Each vpNetworkTTP is associated with zero or one vpNetworkCTP. This operation allows the requester (client) to query the associated vpNetworkCTP. It does not affect the relationship. It matches for the associated vpNetworkCTP. The reply is a vpNetworkCTPId.

Operation: query vpTTP For terminated vpTrail

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatedTrail: vcTrailId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

An vcTrail is terminated on two vc- or vp-TrailTerminationPoints. This operation allows the requester (client) to query the terminated vcTrail. It does not affect the relationship. It matches for the associated vcTrail. The reply is a vcTrailId.

vpTTP: vpTTP Loopback Operation

Operation: loopback vcTrail at vcTTP

INPUT PARAMETERS:
see Section 5.1.2, and
loopbackType: end-to-end, segment
loopbackLocation: interfaceId
OUTPUT PARAMETERS:
loopbackResults: passed or failed
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

This operation is used to request that the vpTTP insert a loopback OAM cell into the ATM cell stream, verify its return, and report the results of the loopback (i.e., passed or failed) back to the management system. Along with each request will be the location where the inserted OAM cell shall loop-back and an indication as to whether a segment or end-to-end OAM cell shall be used. The Loopback Location Code attribute value of the UNI, interNNI, or intraNNI where the loopback is to take place may be used to identify the loopback location. Additionally, a globally unique default value (e.g., "end-point") may also be used to perform a loopback at the other end of a vcTrail.

Outline Ensembles[3]

The outline ensembles defined below are provided only as examples of how the managed entities and operations defined in Section 5 can be used in different operational scenarios. These outline ensembles are outlines of work that needs to be done together with the protocol specific work; this is to indicate how the specifications will be used. This section is for information ONLY. The outline ensembles are NOT normative.

As well as Managed Entities from Section 5, the following Managed Entities from the M4 NE-view may also be reused, depending on the applications and the network management physical architecture**:**

atmAccessProfile
equipment
equipmentHolder
interNNI
intraNNI
plug-inUnit
software
tcAdaptorBidirectional
uni

Each outline ensemble consists of:

- a description of the management architecture used
- a description of the transport architecture being managed, using the concepts and the symbols defined in ITU-T Recommendation I.326
- a description of the management function performed, and of the corresponding requirements.
- the managed entities used, and the corresponding operations. Note that the managed entities provide some functionality which may not be needed in every ensemble. However, there is no attempt at profiling in this document. This may be done once the protocol-specific managed entities are defined.
- the different steps followed to perform the management function, and the operations used to perform these steps (scenario).

There are three options to use the protocol-independent Managed Entities and the outline ensembles to aid interoperability:

- All the protocol-independent MIB Managed Entities are always required, and one or more of the managed entities are used in each outline ensemble.

[3] The term “outline ensembles” was selected here to avoid confusion with the “ensembles” defined in the Network Management (NM) Forum. The outline ensembles defined here are strongly inspired by the work of the NM Forum. However, they are simplified. As an example, the outline ensembles do not contain any MOC.

- The protocol-independent MIB Managed Entities within the M4 network-view are profiled (or selected without profiling) by the future outline ensembles, for the specific applications covered in the ensemble.
- The protocol-independent MIB Managed Entities within the M4 network-view are profiled (or selected without profiling) by the future outline ensembles, along with additional application-specific Managed Entities, for the applications covered in the outline ensemble

All options are allowed, although the second one in the intended one for the outline ensembles defined in this section.

The Managed Entities which are created or deleted in each scenario, fall into two categories:

- Those Entities which, in the scenario, appear only in the management system.
- The Managed Entities which can be manipulated over the M4 interface.

Network management assumptions applying to all outline ensembles defined below:

- The network management architecture (management systems involved, interfaces) is pre-established and beyond the scope of this scenario**.**

- The fault management architecture and the connection management architectures are generally the same here, but this is not necessarily so. The only requirement is that both fault and connection management have knowledge of the resources they manage.

- The creation of Connection or Trail Termination Points, Subnetwork Connections, Link Connections, and Trails (transport entities) is notified to the fault management application, and may result in the instantiation of the above managed entities at all levels.

- The failure of the transport entities may be notified to connection management applications, and may result in a state change of the instantiation of the resources in the connection management application.

Subnetwork Provisioning Outline Ensembles

Subnetwork Creation/Deletion Outline Ensemble

It is assumed that subnetworks are auto-instantiated, meaning that the Network Management Systems are not allowed to create or delete subnetworks.

Fault Management And Alarm Surveillance Outline Ensembles

Equipment Fault Outline Ensemble (Network-view only)

Network Management Context

The network management context selected here is the network-level management architecture, i.e. a subnetwork Management System (subNMS) is interfacing with a set of subtending subNMSs, to which it delegates the control of their subtending NEs or further partitioned subNMSs. Only a subnetwork (or network) view is exposed. The "top" subNMS may itself be a server to a higher-level Network Management System. This ensemble applies to faults detected at either the VP or VC level.

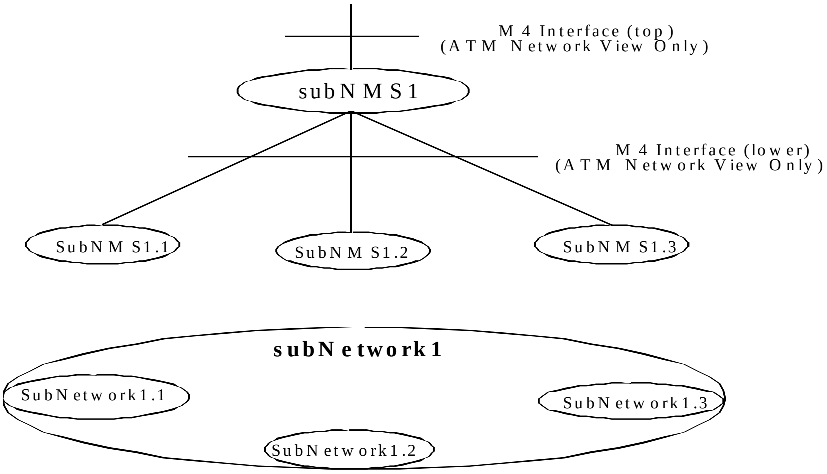


Figure 6.2.2.1-a: Network Management Architecture

Transport Architecture

The transport architecture consists of a large subnetwork partitioned into three small subnetworks. The goal of this outline ensemble is to show how a failure detected at an intermediate network element is notified.

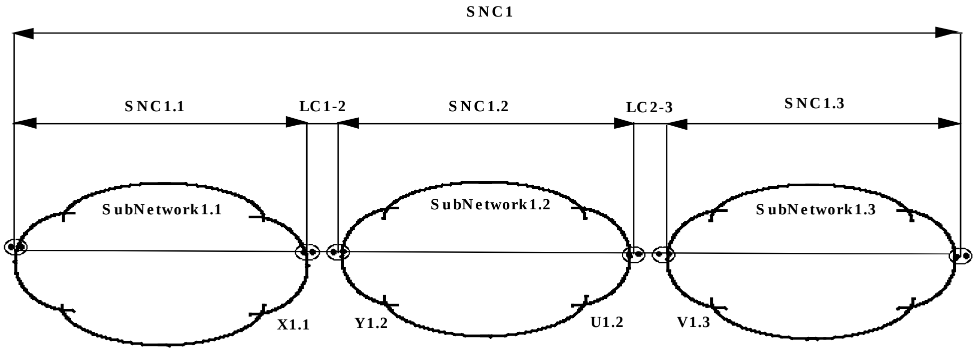


Figure 6.2.2.2-a: Network Transport Architecture

Functional Requirements

The goal of this outline is to show how, in the case of a network-view architecture, a failure detected at an intermediate NE triggers a notification, using subnetwork connection-level state change notification. The requirements involved are detailed in Section 4.2.1 (R -fm- 1).

Managed Entities

Topological Managed Entities involved in this scenario:

- vpSubnetwork or vcSubnetwork
- vpNetworkCTP or vcNetworkCTP
- vpSubnetworkConnection or vcSubnetworkConnection
- vpLinkConnection or vcLinkConnection

Management Entities involved in this scenario:

- Network

Scenario

Assumption: To each subnetwork Management System corresponds an administrative domain (“Network” Managed Entity) which contains an already-existing subnetwork.

Step 1: Interface Y1.2 detects an equipment failure**.**

Step 2: SubNMS1.2 identifies the affected managed entities (e.g. vpNetworkCTP Y1.2, in the incoming direction from subnetwork1.1)

Step 3: SubNMS1.2 sends an alarm to SubNMS1 against vp/vcNetworkCTP Y1.2.

Step 4: SubNMS1 concludes that the vp/vcLinkConnection LC1-2 is failed.

Step 5: SubNMS1 changes the state of the vp/vcSubnetworkConnection SNC1 and notifies its client of the state change.

Equipment Fault Outline Ensemble (network-view + NE-view)

Network Management Context

The network management context selected here is the network-view + NE-view architecture, i.e. the higher-level management system has full visibility into not only the lower-layer subnetworks, but also into their constituent NEs. This Outline Ensemble describes an equipment failure using the M4 Network Element view. This ensemble applies to faults detected at either the VP or VC level.

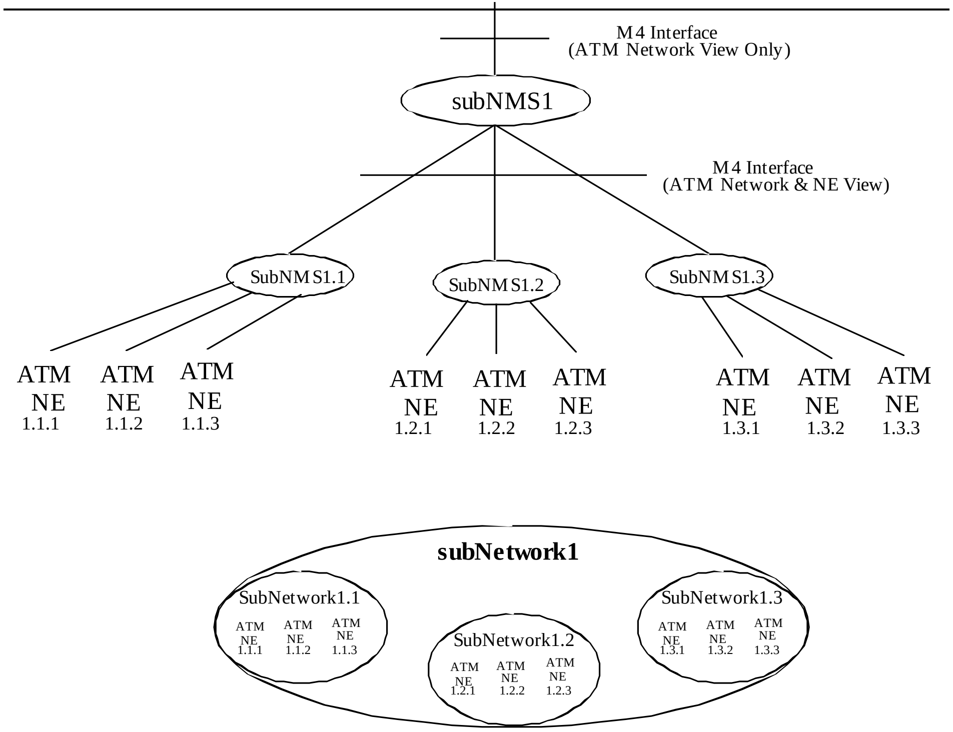


Figure 6.3.2.1-a: Network Management Architecture

Transport Architecture

The transport architecture consists of a large subnetwork partitioned into three small subnetworks. The goal of this outline ensemble is to show how a failure detected at an intermediate network element (ATM NE1.2.1) is notified.

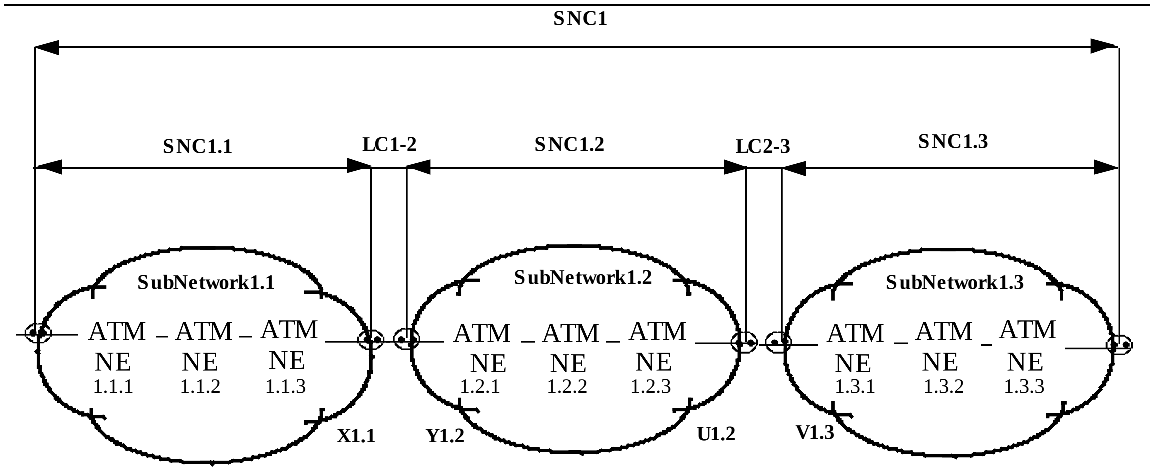


Figure 6.2.3.2-a: Network Transport Architecture

Functional Requirements

The goal of this outline is to show how, in the case of a network-view + NE-view architecture, a failure detected at an intermediate NE triggers a notification, using subnetwork connection-level state change notification. The requirements involved are detailed in Section 4.2.1 (R -fm- 2).

Managed Entities

Topological Managed Entities involved in this scenario:

- vpSubnetwork or vcSubnetwork
- vpNetworkCTP or vcNetworkCTP
- vpSubnetworkConnection or vcSubnetworkConnection
- vpLinkConnection or vcLinkConnection
- atmNE (from NE view)

Management Entities involved in this scenario:

- Network

Scenario

Assumption: To each subnetwork Management System corresponds an administrative domain ("Network" Managed Entity) which contains an already-existing subnetwork.

Step 1: NE 1.2.1 detects an equipment failure**.**

Step 2: NE 1.2.1 notifies subNMS1.2 of the failure of the equipment (note that this notification is not visible to SubNMS1).

Step 3: SubNMS1.2 identifies the affected managed entities (e.g. vpNetworkCTP Y1.2 within ATM NE 1.2.1, in the incoming direction from NE1.1.3)

Step 4: SubNMS1.2 sends an alarm to SubNMS1 against vp/vcNetworkCTP Y1.2, and its location.

Step 5: SubNMS1 concludes that the vp/vcLinkConnection LC1-2 is failed.

Step 6: SubNMS1 changes the state of the vp/vcSubnetworkConnection SNC1 and notifies its client of the state change.

Transport Failure Alarming Outline Ensemble (network-view only)

Network Management Context

Same as for Section 6.2.2.1

Transport Architecture

Same as for Section 6.2.2.2

Functional Requirements

The goal of this outline is to show how, in the case of a network-view architecture, a failure detected at an intermediate NE triggers a notification, using subnetwork connection-level state change notification. The requirements involved are detailed in Section 4.2.2 (R -fm- 2).

Managed Entities

Topological Managed Entities involved in this scenario:

- vpSubnetwork or vcSubnetwork
- vpNetworkCTP or vcNetworkCTP
- vpSubnetworkConnection or vcSubnetworkConnection
- vpLinkConnection or vcLinkConnection

Management Entities involved in this scenario:

- Network

Scenario

Assumption: To each subnetwork Management System corresponds an administrative domain (“Network” Managed Entity) which contains an already-existing subnetwork.

Step 1: The vp/vcLinkConnection between X1.1. and Y1.2 fails, bidirectionally.

Step 2: SubNMS1.1 is informed of a state change in vp/vcNetworkCTP X.1.1.

Step 3: SubNMS1.1 sends an alarm against vp/vcNetworkCTP X1.1 to SubNMS1.

Additional steps for subNMS1.2: steps 2 to 3 are repeated asynchronously for subNMS1.2 and vp/vcNetworkCTP Y1.2.

Step 4: SubNMS1 receives alarm from SubNMS1.1 and SubNMS1.2 to indicate the failure of X1.1. and Y1.2 respectively.

Step 5: SubNMS1 concludes that the vp/vcLinkConnection LC1-2 is failed.

Step 6: SubNMS1 changes the state of the vp/vcSubnetworkConnection SNC1 and notifies its client of the state change.

Subnetwork Connection Management Outline Ensembles

Simple VP or VC Single-layer Subnetwork Connection Set-Up Outline Ensemble (network-view only)

Network Management Context

This outline ensemble involves the M4 interface between a network management system and a subnetwork management system as detailed in the figure below. It fits into the network-level management architecture defined in Section 3.2. This ensemble is intended to fulfill the requirements of ITU-T Recommendation G.852-01 [5].

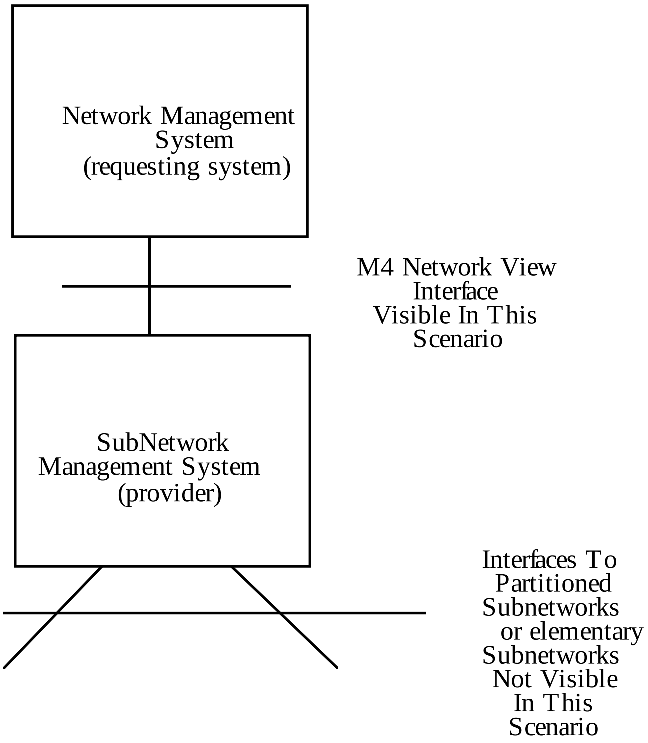

Network Management System (requesting system)
M4 Network View Interface Visible In This Scenario
SubNetwork Management System (provider)
Interfaces To Partitioned Subnetworks or elementary Subnetworks Not Visible In This Scenario

Figure 6.3.1.1-a: Network Management Architecture

Transport Architecture

The transport architecture consists of a single subnetwork. The goal of this outline ensemble is to show how to set-up a VP or VC subnetwork connection over this subnetwork.

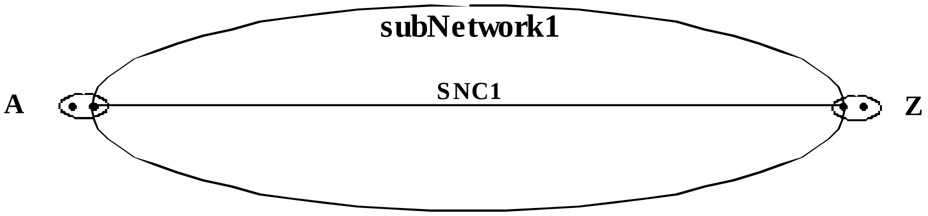


Figure 6.3.1.2-a: Network Transport Architecture

Functional Requirements

The goal of this outline is to set-up a subnetwork connection between two VP or VC network connection termination points. As an example, this set-up may be needed to fulfill a customer order for a subnetwork connection between two UNIs. The requirements involved are detailed in Section 4.1.2.1 (R-cm -1 and R-cm -2).

Managed Entities

Topological Managed Entities involved in this scenario:

- vpSubnetwork or vcSubnetwork
- vpNetworkCTP or vcNetworkCTP
- vpSubnetworkConnection or vcSubnetworkConnection

Scenario

Assumption: A subnetwork pre-exists in the subnetwork management system of which the network management system has knowledge. This knowledge is acquired through the Query Operations identified in the Network Managed Entity. The following diagram outlines the knowledge assumed to be in place to support this scenario.

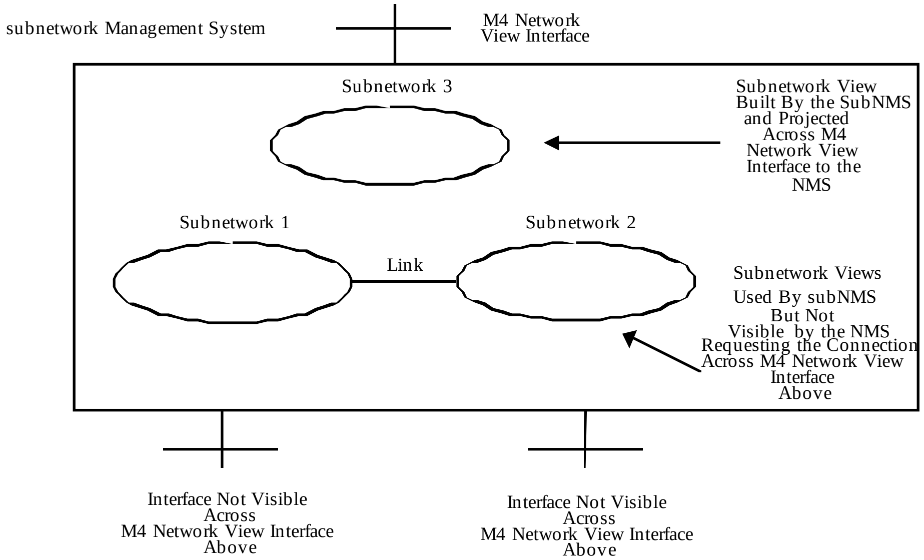

subnetwork Management System
M4 Network View Interface
Subnetwork 3
Subnetwork View Built By the SubNMS and Projected Across M4 Network View Interface to the NMS
Subnetwork 1
Subnetwork 2
Link
Subnetwork Views Used By subNMS But Not Visible by the NMS Requesting the Connection Across M4 Network View Interface Above
Interface Not Visible Across M4 Network View Interface Above
Interface Not Visible Across M4 Network View Interface Above

Figure 6.3.1.5-a: Information Applicable To This Scenario

Step #1: The NMS builds the following:

vpSubnetworkConnection, or

vcSubnetworkConnection

across a subnetwork known to it. Each request has the following required attributes:

Required:
- User Label
- Source/Destination (UNIs or NNIs)
- Source/Destination Service Descriptors
  - QoS Class
  - UPC Treatment
  - Traffic Descriptor
    - Both Cell Loss Priority
    - Peak Cell Rate
    - Sustainable Cell Rate
    - Maximum Burst Size
    - CDVT

Optional:
- Source/Destination VPIs/VCIs
- Administrative State (Locked means to Reserve the SNC)

Step#2: The NMS sends one of the two types of request and the subNMS receives the request and validates the incoming parameters of the request.

Step#3: The subNMS either begins processing the request or returns an error condition if the input parameters cannot be validated.

Step#4: If the source/destination VPIs/VCIs are not specified the subNMS selects the VPIs/VCIs and creates the required vp/vcNetworkCTPs and vp/vc CTP/TTPs which need to be connected.

Step#5: If the Administrative State of Locked is specified the Subnetwork Connection will be setup in a state which does not permit user cell flow until Unlocked.

Step#6: The subNMS then selects/finds a route through the subnetwork and then creates the Subnetwork Connection and any supporting objects.

Step#7: The subNMS informs the NMS of the Subnetwork Connection ID.

Multi-Level VP or VC Single-layer Subnetwork Connection Set-Up Outline Ensemble   (network-view only)

Network Management Context

The network management context selected here is the network-view only architecture, i.e. a subnetwork management System (subNMS) is interfacing with a set of subtending subNMSs, to which it delegates the control of their subtending NEs or further partitioned subNMSs. Only a subnetwork (or network) view is exposed. The “top” subNMS may itself be a server to a higher-level Network Management System.

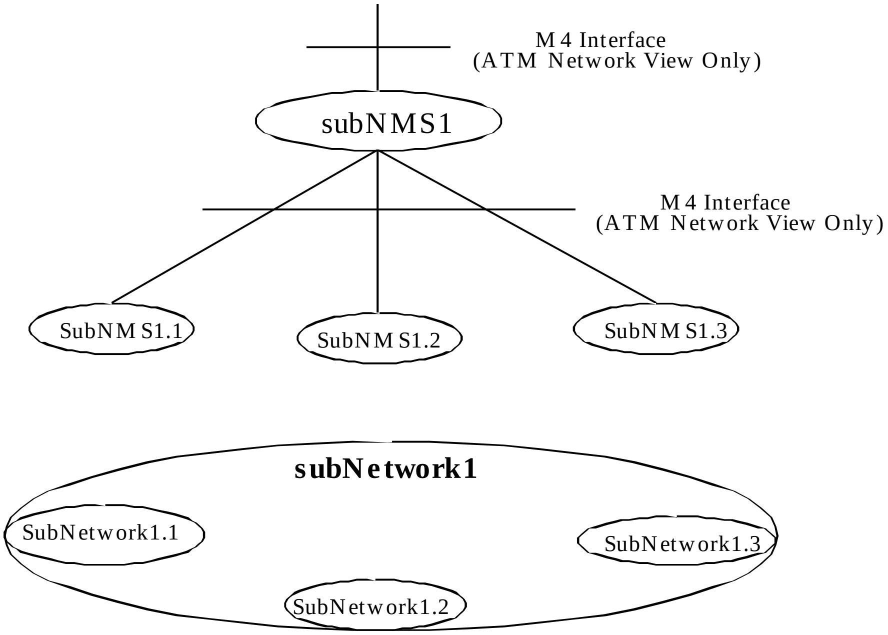


Figure 6.3.1.1-a: Network Management Architecture

Transport Architecture

The transport architecture consists of a large subnetwork partitioned into three small subnetworks. The goal of this outline ensemble is to show how to set-up a VP or VC subnetwork connection over the three subnetworks.

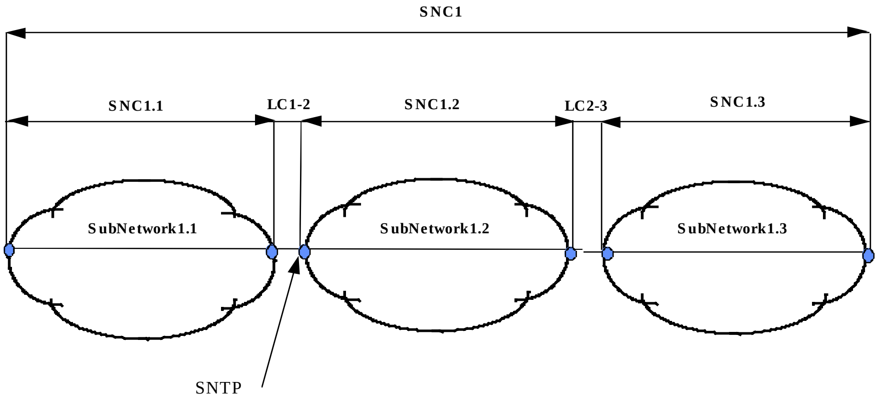


Figure 6.3.2.2-a: Network Transport Architecture

Functional Requirements

The goal of this outline is to set-up a subnetwork connection between two VP or VC connection termination points. As an example, this set-up may be needed to fulfill a customer order for a subnetwork connection between two UNIs. The requirements involved are detailed in Section 4.1.2.1 (R-cm -1 and R-cm -2).

Managed Entities

Topological Managed Entities involved in this scenario:

- vpSubnetwork or vcSubnetwork
- vpNetworkCTP or vcNetworkCTP
- vpSubnetworkConnection or vcSubnetworkConnection
- vpLinkConnection or vcLinkConnection

Management Entities involved in this scenario:

- Network

Scenario

Assumption: To each subnetwork Management System corresponds an administrative domain which contains an already-existing subnetwork. However, the Network Connection Termination Points used to create the Subnetwork Connection are not pre-existing**.** The following example is for VP subnetwork connections.

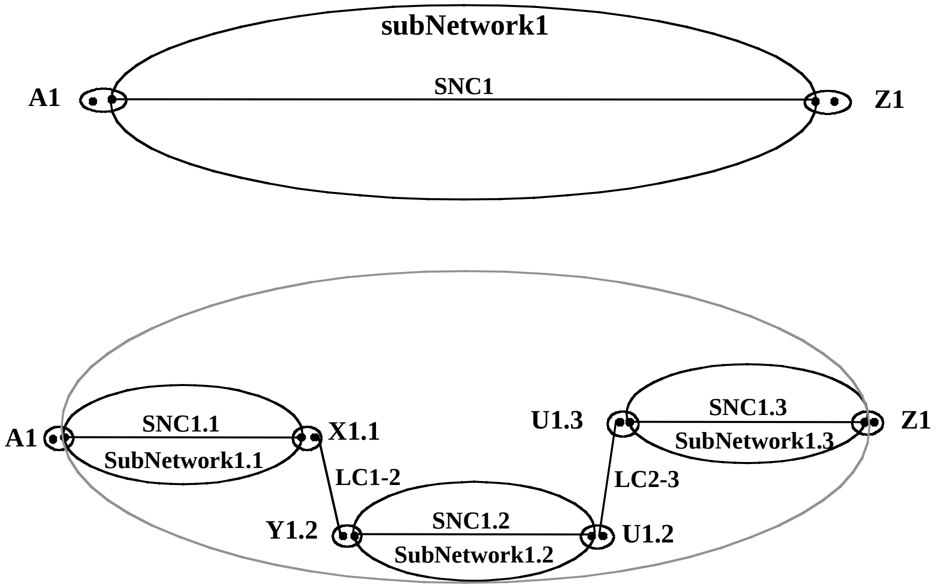


Figure 6.3.1.5-a: Transport Managed Entities

Step 1: Subnetwork Management System #1 (subNMS1) receives a request to set-up a VP subnetwork connection between A and Z within subnetwork #1 (SN1). The termination points A and Z may be identified by their VPI within the Transport Path or by a specific name, and are qualified by their traffic descriptors and QoS**.**

Step 2: subNMS1 instantiates two network Connection Termination Points (CTPs), A1, and Z1. subNMS1 also identifies which route to use, over which lower-level subnetworks, and using which links.

Step 3: subNMS1 requests subNMS1.1 to create two network connection termination points (A1 and X1.1), which are qualified by their traffic descriptors and QOS.

Step 4: subNMS1 requests subNMS1.1 to set-up a VP subnetwork connection between network connection termination points A1 and X1.1 within subNMS1.1. The termination points A1 and X1.1 may be identified by their VPI within the Transport Path or by a specific name, within subnetwork subNetwork1.1**.**

Step 5: subNMS1.1 creates two VP network CTPs (A1 and X1.1).

Step 6: subNMS1.1 creates a vpSubnetworkConnection terminated by A1 and X1.1.

Step 7: subNMS1.1 acknowledges the creation of A1, X1.1 and of the vpSubnetworkConnection SNC1.1 terminated by A1 and X1.1 to subNMS1.

Additional steps for subNMS1.2 and subNMS1.3: steps 3 to 7 are repeated asynchronously for subNMS1.2 and subNMS1.3.

Step 8: After subNMS1 receives acknowledgment from subNMS1.1 and subNMS1.2 for the creation of X1.1 and Y1.2, a vpLinkConnection, within subnetwork 1, between X1.1 and Y1.2, is created. The “vpLinkConnection” is a management object for subNMS1, which may be made visible to its clients. The same applies to the other link from U1.2 to U1.3.

Step 9: subNMS1 creates vpSubnetworkConnection atmSNC1 between A1 and Z1.

Step 10: subNMS1 acknowledges the creation of vpSubnetworkConnection1, A1 and Z1 to its client.

Note that if a subnetwork connection cannot be created at a lower level (i.e. further subnetwork partitioning), all associated managed entities at the higher level shall be removed, the initial request denied, and the denial logged (or its equivalent).

VP or VC Single-layer Piece-By-Piece Connection Set-Up Outline Ensemble (NE-view + NW-view)

This ensemble assumes that the Network Management System (NMS) wishes to specifically direct the routing of the PVC across the individual ATM NEs and not leave this control to a subnetwork management system. The result may however create a network-view subnetwork connection, if both the network-view and the NE-view are supported.

Network Management Context

The network management context selected here is a full network-view + NE-view M4 interface with a hierarchical arrangement of ATM management systems. There is a non-homogeneous mixture of multi-vendor subnetwork management systems in the network being managed and it is necessary for the NMS to direct the specific route of a connection over this network. While subnetwork controllers (or sub NMS) may be present to control a specific vendor's network elements, they are used and viewed primarily as a command pass-through mechanism in this ensemble. This allows the NMS to directly control the various ATM Network Elements (NE) in the network. Figure 6.3.3.1-a shows the relationship of the Network Management System and the subnetwork management systems.

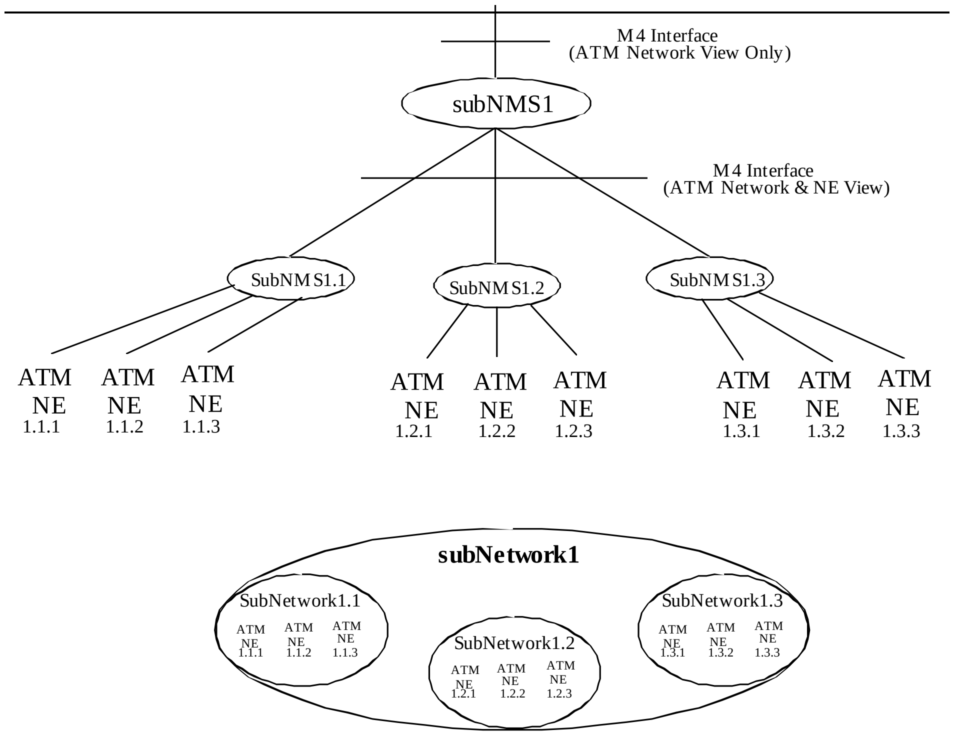


Figure 6.3.3.1-a Network Management Architecture

Transport Architecture

The transport architecture for an ATM connection used here is shown in figure 6.3.3.2-a. In this architecture the network operator is using 3 different ATM network element vendors. Vendor A (subnetwork management system 1.1) provides ATM access functionality such as multiplexing lots of lower speed ATM access connections into a higher speed ATM trunk. Vendor B's ATM network element (subnetwork management system 1.2) provides the functionality of high speed backbone switching. Vendor C (subnetwork management system 1.3) provides additional customer oriented functionality such as protocol or media conversion. A typical ATM connection comes from a source end user through vendor A's ATM access device (NE-1.1.1), where it is multiplexed into a trunk running into and through a high speed backbone network from vendor B (NE-1.2.1 through NE-1.2.3). Then the ATM connection goes back to another ATM access device from vendor A (NE-1.1.3) and then through an ATM protocol or media conversion device from vendor C (NE-1.3.3) before terminating on the user's destination host.

Such a mix of vendors can and will occur because:

1. Need to address specific market functionality with individual equipment,
2. Evolutionary introduction of new more powerful network elements,
3. Specific functional cost-effectiveness of particular network element vendors,
4. Merging of companies with existing ATM networks from different vendors.

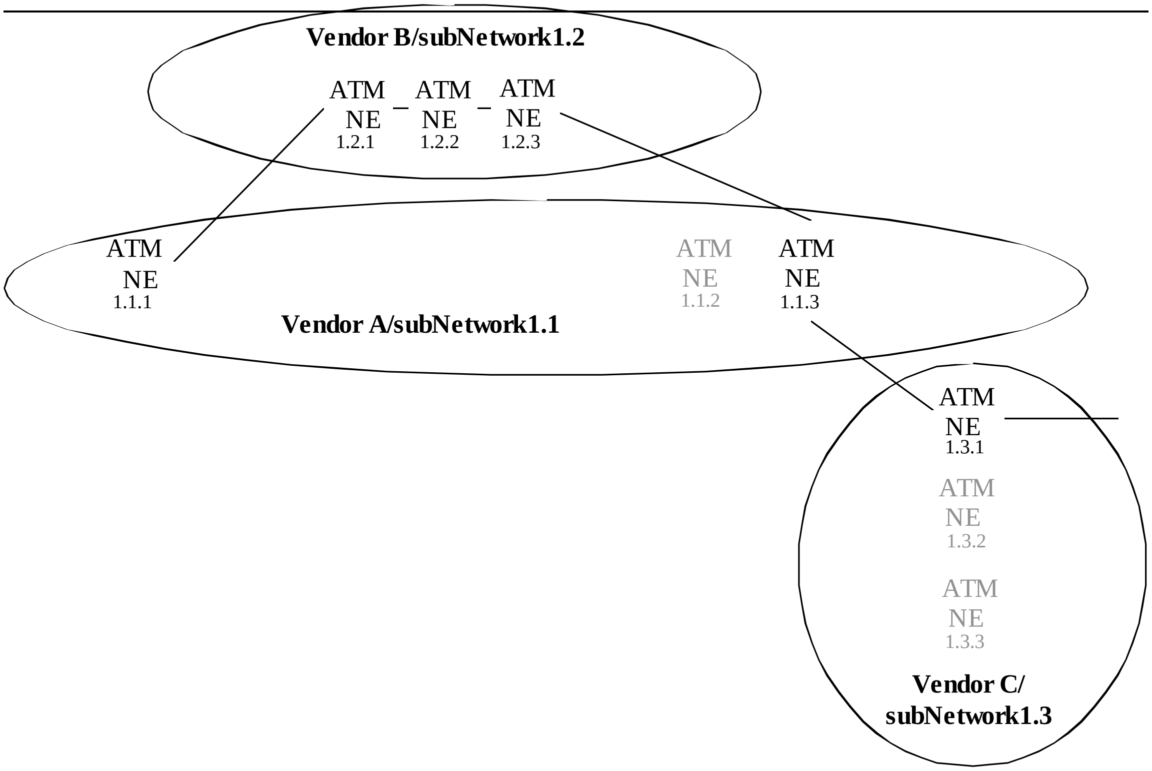


Figure 6.3.3.2-a Network Transport Architecture

Functional Requirements

This ensemble supports requirement (R-4.1.1.2)-cm -6, (R-4.1.1.2)-cm -11, (R-4.1.1.2)-cm -12, and (R-4.1.1.2)-cm -13.

Managed Entities

- atmNE
- vpSubnetworkConnection or vcSubnetworkConnection
- VPCrossConnect or VCCrossConnect
- vpLinkConnection or vcLinkConnection
- vcTrail or vpTrail

Scenario

Assumptions: This scenario involves many of the very real service drivers that a carrier or large network operator must consider in building a subnetwork connection in a public network to meet a customer's needs. As such, it discusses service management related items that affect the proper set up of a subnetwork connection.

This ensemble is focused on the Network Element (NE) by NE set up of a trail by a network management system (NMS) across a non-homogenous set of subnetwork management systems. As such, it puts the route selection control in the NMS and not in the subnetwork management system.

Note: the following steps take place within the NMS, or above, and do not require interactions over the M4 interface in this ensemble.

- The Network Management System (NMS) is requested to set up an end-to-end subnetwork connection (VP or VC) across the ATM network for a point-to-point connection. The termination points A and Z are identified along with desired class of service information.

- The NMS creates a connection identifier for this subnetwork connection. This identifier provides a name or administrative label (master connection name) for this particular or connection.

- The NMS assigns connection ownership to a particular customer, organization or user. This requires a user label for this subnetwork connection that identifies the name of the owner. This ownership attribute, while having many uses, will be required later if diverse routing of the subnetwork connection is required. (Diverse routing route this subnetwork connection over a physically different route than that of another trail belonging to the same customer.)

Note: the following steps take place within the NMS, or above, and do not require interactions over the M4 interface.

- The NMS determines the relevant parameters affecting the quality of service that is required by this trail. These parameters can affect the global (subnetwork-to-subnetwork) and specific (NE-to-NE) route selection and the physical termination points to be used for the trail.

- The NMS determines the termination points (networkCTP) for the subnetwork connection (A & Z). For security and customer satisfaction reasons, ownership of these connection termination points should be the same as ownership of the connection as established in step 3. If ownership of logical termination points does not match the ownership of the connection, then an administrative check should occur to determine if the connection is going to an acceptable "public network" or a shared facility or if the connection is an inter-organization facility. A logical termination point may be a direct mapping to a physical termination point or it may be a label for a group of physical termination points.

- The NMS maps the logical termination points to physical termination points. This mapping may be an administration dependent process and may be very simple or quite complex. It is typically beyond the capabilities of a subnetwork management system.

This step handles a variety of service drivers including, but not limited to:

1 a computer with multiple ATM physical links to the network, that desires that the network provide a basic load leveling service of connections.

2 a customer who wants traffic delivered to different physical destinations based on time of day and day of week considerations.

3 a customer who is on the move and desires that the traffic follow the person from home to office to distant locations.

- The NMS determines relevant possible routes available between the physical termination points for this subnetwork connection. This route selection may be an administration dependent process and involve a variety of factors including, but not limited to, quality of service parameters, network hierarchy, trail ownership, security, cost of route, protocol conversion requirements, geo-graphical diversity, network loading, service premiums, network engineering rules, available capacity, etc. This outline ensemble assumes that because of so many factors, it is desirable for the carrier or network operator to do this routing on a NE-by-NE basis at the NMS level and not leave this to a subnetwork management system. [This ensemble was written because it is often beyond the scope of many subnetwork management systems to handle such a wide variety of factors in doing route determination.]

- The NMS will select one of the possible routes determined in step 7 for this trail. The route selected is a set of network elements (NE), and can be represented as NE-1, NE-2, NE-3, ..., NE-N, where N is the number of nodes or network elements being traversed in the entire network.

The following steps, 1 through 3, will be repeated N times, once for each network element (NE) being traversed. X in these steps will increment from 1 to N with each pass through the steps. After the Nth pass through the steps is completed, proceed to step 7.

Step 1: For the selected route and node X, NE-X will have two termination points, TP-X1 and TP-X2 which identify the two physical ports on the NE that need to be connected as part of this trail. The NMS must command NE-X (or its subnetwork controller) to provide a VP or VC cross connection from TP-X1 to TP-X2, where TP-X1 is a physical connection to NE-(X-1) and TP-X2 is physical connection to NE-(X+1). [Special cases: Where X-1 is equal to zero, the connection is to termination point A. Where X+1 is equal to N+1, the connection is to termination point Z.] The VP and VC values assigned by the NMS must not have already been in use of these ports and must be consistent between any two adjacent NEs. In addition, some administration specific rules may be applied by the NMS in the selection of VP and VC numbers.

Step 2: The NMS commands NE-X or its subnetwork management system to set the correct quality of service parameters for each of the connected VPs or VCs on each of the cross connections for NE-X.

Step 3: The NMS commands NE-X or its subnetwork management system to assign the master connection name and the user label of this connection.

Step 4: Should the NE-X or its subnetwork control be unable to perform any of the actions commanded by the NMS, it should return an error status to the NMS indicating the cause of the action failure. The NMS will then select an alternate route and repeat the necessary steps or return a failure status to the requesting entity.

Step 5: For each of the two physical ports at the edge of the network, the NMS commands the corresponding network element (NE-1.1.1 & NE-1.3.1) or the NE's subnetwork management system to interrogate the connection parameters in the customer's device. This is achieved by the NE using the correct port's ILMI to compare the relevant MIB parameters in the customer's device to the corresponding MIB parameters in the NE. Once the comparison is achieved, the results of the comparison (good and bad) should be reported back to the NMS by the NE.

Step 5: The NMS should update its vpTopologicalLink or vcTopologicalLink information for each endpoint and check the available bandwidth used the end connections for NE-1 and NE-N.

Step 6: Optionally the NMS may need to assign the vpLinkConnection or vcLinkConnection values for this trail if the trail terminates to another carrier.

Step 7: The NMS should optionally command a NE involved in the subnetwork connection (or its subnetwork management system) to initiate a customer loopback test to insure proper end to end conductivity of the trail. This sub procedure or ensemble would entail, sending out OAM cells at the F4 or F5 level to insure a correct end-to-end path. This step involves the use of the VC or VPTTPl.

Step 8: The NMS commands the end of trail NEs (NE-1.1.1 & NE-1.3.1) to administratively lock the ATM trail to prevent customer traffic from flowing over the connection. This is done to prevent connection use by the customer until the desired start date.

Note: the following two steps take place within the NMS, or above, and do not require interactions over the M4 interface in this ensemble.

- The NMS reports the success or failure of the creation of the desired trail for the desired customer, along with its master connection identifier to the requesting entity.

- The NMS logs the original build connection request action and its result, along with the assigned route to the appropriate administrative log.

VP or VC Single-layer Piece-By-Piece Connection Set-Up Outline Ensemble (NE-view)

Network Management Context

The network management context selected here is the NE-view management architecture, i.e. the higher-level management system has full visibility into not only the lower-layer subnetworks, but also into their constituent NEs. This Outline Ensemble describes set-up of a point to point subnetwork connection using the M4 Network Element view. The higher-level management system selects not to use the network-level capabilities to set-up the connection. This assumption results in the utilization of NE-level managed entities, which implies that the NE-view management architecture is required as in a Network-level Managed Architecture where the network manager does not have direct access to the NE-level managed entities.

The following Outline Ensemble focuses on the set-up of a subnetwork connection at the VC level; prerequisites at lower network levels (VP, physical level) are addressed where necessary.

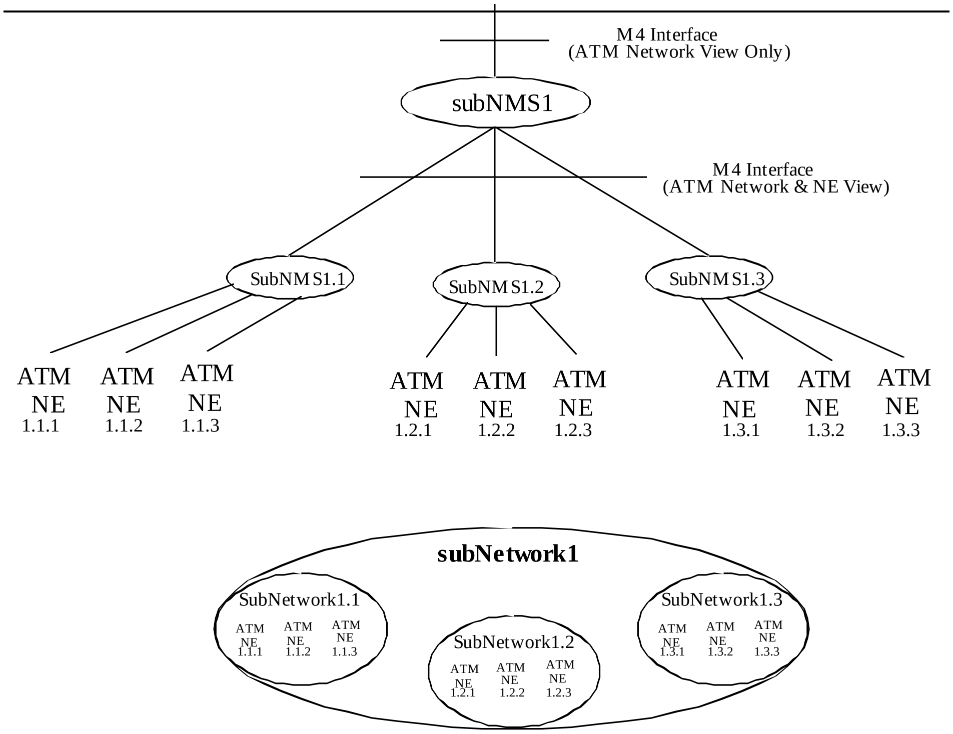


Figure 6.3.4.1-a:Network Management Architecture

Architecture

The transport architecture consists of a large subnetwork partitioned into three small subnetworks, themselves made-up of three NEs. The goal of this outline ensemble is to show how to set-up a VP or VC subnetwork connection over the nine corresponding NEs.

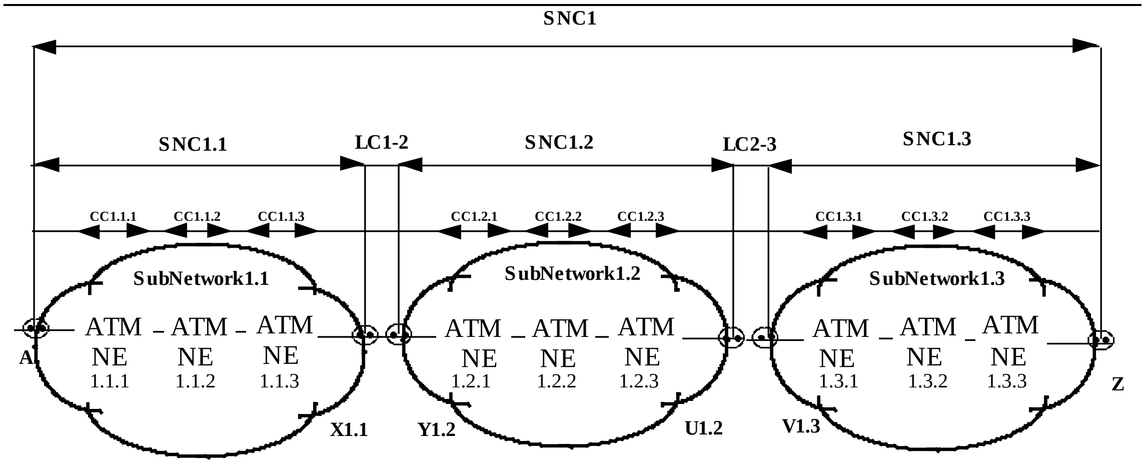


Figure 6.3.4.2-a: Transport Architecture

Functional Requirements

The management task to be achieved is the set-up of a subnetwork connection.

The following M4 NE view related functions are used:

- Provisioning and configuration of physical access points

- Provisioning and configuration of connection points at VP and/or VC level

- Establishment of cross connections in the involved NEs

Managed Entities

The following M4 NE-view Managed Entities are relevant to this Outline Ensemble (please refer to the M4 NE view requirements and protocol-independent MIB specification for the definition of these entities), along with M4 Network-View Managed Entities:

NE-View Managed Entities involved in this scenario:

Note: The CMIP object name is provided in parenthesis, if different from the protocol-independent MIB name)

- atmAccessProfile
- ATM Cross Connection (atmCrossConnection)
- ATM Cross Connection Control (atmFabric)
- Physical access points, e.g. for SDH/SONET or PDH
- BICI (interNNI)
- BISSI (intraNNI)
- TC Adaptor (tcAdaptorTTPBidirectional, shorten as tcAdaptor below)
- UNI
- VCC Termination Point (vcTTPBidirectional)
- VCL Termination Point (vcCTPBidirectional)
- VPC Termination Point (vpTTPBidirectional)
- VPL Termination Point (vpCTPBidirectional)

Scenario: Set-up of a point-to-point subnetwork connection at VC level

Assumptions:

- The respective physical layer resources are configured.
- The respective interface managed entities (UNI etc.) are configured.

- The respective access and adaptation managed entities (tcAdaptor etc.) are configured.

The subnetwork connection set-up is performed in two phases:

- Configuration of the VP and/or VC level managed entities at the terminating NEs
- Selection of transit NEs, configuration of needed VP/VC level managed entities and establishment of the corresponding cross-connections.

Phase 1, steps (only VC-level shown):

Step 1: Retrieve UNI information to identify corresponding tcAdaptor

Step 2: Retrieve connectivity pointers to vpTTP

Step 3: Establish vcCTP at both Terminating NEs

Phase 2, steps (only VC-level shown):

These steps have to be performed for each transit NE where VC cross connection is performed, i.e. not in those transit NEs where only VP cross connection is done.

Step 1: Determine ingress vcCTP by use of corresponding egress of that NE which precedes in the route.

Step 2: Determine egress vcTTP

Step 3: Cross-connect ingress and egress Managed Entities.

Note that this is a simplified scenario which shows the set-up of a subnetwork connection at VC level using the M4 NE view. No error cases have been shown. It is clear, however, that in case one step fails the operator has to undo a number of steps. Depending on the capabilities of the management system and the NEs involved this may be done automatically or manually. Furthermore the operator may abort the set-up at any step. Depending on the capabilities of the involved systems this may be done automatically or manually.

VP or VC Subnetwork Connection Modification Outline Ensemble

Network Management Context

This Outline Ensemble describes the modification of selected properties of subnetwork connections. Both a network-level view scenario and a NE-view scenario are outlined. Please refer to the previous sections for a description of these network management architectures.

Transport Architecture

This Outline Ensemble describes the modification of selected properties of subnetwork connections. Both a network-level view scenario and a NE-view scenario are outlined. Please refer to the previous sections for a description of these transport architectures. Note than in the case of an M4 interface supporting both views, attributes in both the NE-view and the Network-view may be affected. Therefore, both sets of transport entities, at the NE-level, and at the network-level, are involved.

Functional Requirements

This Outline Ensemble deals with the modification of the traffic-affecting parameters, such as Traffic Descriptors. It does not cover other modifications, such as route modification.

The requirements on subnetwork connection modification provided in Section 4.1.2.2
apply here.

Managed Entities

The following M4 NE-view Managed Entities are relevant to this Outline Ensemble (please refer to the M4 NE view requirements and protocol-independent MIB specification for the definition of these entities), along with M4 Network-View Managed Entities:

NE-View Managed Entities involved in this scenario:

Note: The CMIP object name is provided in parenthesis, if different from the protocol-independent MIB name)

- atmAccessProfile
- ATM Cross Connection (atmCrossConnection)
- ATM Cross Connection Control (atmFabric)
- Physical access points, e.g. for SDH/SONET or PDH
- BICI (interNNI)
- BISSI (intraNNI)
- TC Adaptor (tcAdaptorTTPBidirectional)
- UNI
- VCC Termination Point (vcTTPBidirectional)
- VCL Termination Point (vcCTPBidirectional)
- VPC Termination Point (vpTTPBidirectional)
- VPL Termination Point (vpCTPBidirectional)

Network-Level Topological Managed Entities involved in this scenario:

- vpSubnetwork or vcSubnetwork
- vpSubnetworkConnection or vcSubnetworkConnection
- vpLinkConnection or vcLinkConnection
- vp or vcTrail

Management Entities involved in this scenario:

- Network

Scenario:

In analogy to the set-up of a subnetwork connection, the modification of a subnetwork connection can be done at the network level, i.e. using managed entities of the M4 network level view or at the NE level**.** Depending on the scenario (see below) the M4 operations and managed entities which are used via the M4 interface belong to the M4 network level or the M4 NE level. In the case of a change at the network-level (case 1), both a NE+Network-level Managed Architecture and a Network-level Managed Architecture are applicable; in the case of a change at the NE-level, a NE+Network-level Managed Architecture is required because NE-level view resources are directly addressed.

Case 1: Network level view, steps:

Step 1: The subnetwork Management System receives a request to modify an existing subnetwork connection (at that VP or VC level).

Step 2: The subnetwork Management System requests the subordinate subNMS to modify their portion of the subnetwork connection.

Step 3: If the subNMS receives acknowledgment of the changes from its subtending subNMS, the subnetwork connection modification is completed, and an attribute change modification is sent out.

Step 4: If any of the subtending subNMS is unable to perform the change, the subnetwork connection modification is failed. Changes which have been at a lower level are undone.

Case 2: NE level view; steps:

Step 1: Determine the NE involved

Step 2: Modify the corresponding vcCTP attributes

Step 3: If the modification is accepted by each involved NE, the modification is put into effect. An attribute change modification is sent out by the NE.

Step 4: If any NE is unable to perform the change, the modification request is failed. The Operator is then responsible to undo the accepted changes in the NE which were able to perform the change.

Interlayer Trail Connection Management Outline Ensemble

Interlayer Trail Set-Up In a Single Management Domain Outline Ensemble

Network Management Context

This Outline Ensemble addresses a subset of the capabilities of the M4 management interface which performs the Trail connection set-up service. ATM networks can be decomposed into the virtual path layer and the virtual channel layer. It is possible to build a subnetwork Management System (subNMS) for the management of each layer network. Those two layers have a client-server relationship as described in the ITU-T Recommendation G.803, Architectures of Transport Networks Based on the Synchronous Digital Hierarchy (SDH). Similar relationships can be applied between the ATM layer network and the transmission network such as SONET or SDH.

This Outline Ensemble shows one example of Trail connection setup between two layer networks with the client-server relationship. A layer network (VC or VP) can be partitioned into subnetworks and links interconnecting them. A VC link may use a VP layer network to interconnect the VC subnetwork. In this case a VC link is supported by a VP trail in the VP layer (see Appendix A).

Note: The described mechanism to provide and manage VP-trails is same for VC-trails at VC-layer.

Relationships with Other Outline Ensembles: Note that this Outline Ensemble is closely related to the link provisioning.

The Network Management architecture selected here is the Network-View-only Management Architecture.

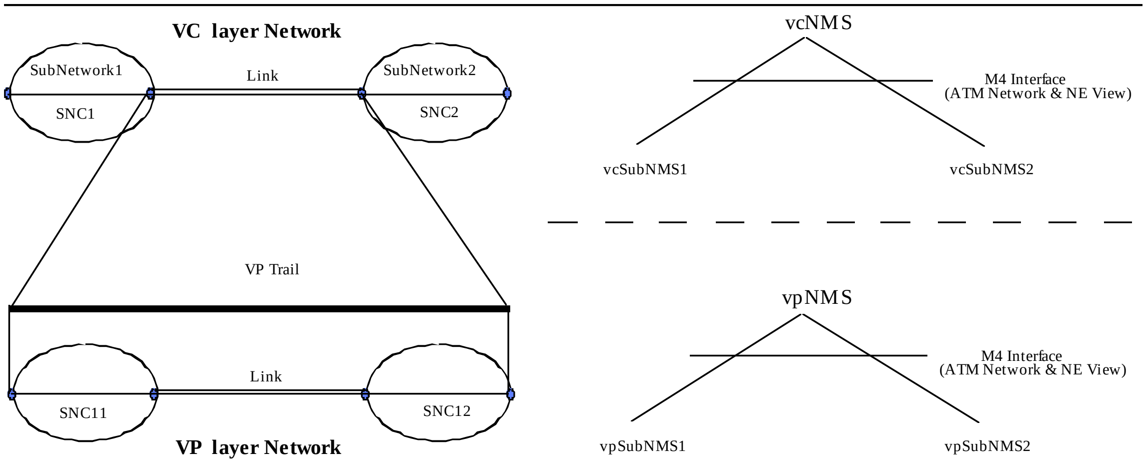


Figure 6.5.1.1: Interlayer Trail Set-Up in a single Management Domain

Transport Architecture

See above Figure (left side)

Functional Requirements

The following functional requirements are covered by this Outline Ensemble.

- (R-4.1.2.1.2)-cm -1
- (R-4.1.2.1.2)-cm -2
- (R-4.1.2.1.2)-cm -3
- (R-4.1.2.1.2)-cm -4
- (R-4.1.2.1.2)-cm -5

Managed Entities

Managed Entities:

The following Managed Entities are used in this ensemble:

- vcTopologicalLink
- vcLinkConnection
- vpNetworkCTP
- vpSubnetworkConnection
- vpTrail
- vpLinkEnd
- vpLinkConnection

Scenarios

This scenario covers an interlayer trail set-up in a single Management Domain

When a VC link is created between two VC subnetworks that are connected through VP subnetwork a VP trail has to be set up through the VP subnetwork.

Steps:

Step 1: The VC-layer Network Management System informs the provision function that a link is required between VC subnetworks SNW1 and SNW2. VC-layer Network Manager specifies the end-points to the link, VCI range, maximum number of simultaneously active VCC, etc.

Step 2: The provision function requests the VP-layer Network Manager to set up a VP-trail with identity of the ATM interface termination point, or with the VPI value of a VP termination within a specific ATM interface and other necessary parameters.

Step 3: The VP-layer Network Management System transforms the trail request into subnetwork and link connection set up requests to set up a Virtual Path Subnetwork Connection through the VP layer network.

Step 4: VP-layer subnetwork management Systems create the requested subnetwork connections and the VP-layer network manager creates the link connections between the subnetworks.

Step 5: At the edges of the Virtual Path Subnetwork Connection the VP-layer Network Management System creates VP Trail Termination points, associates VP Trail termination functions to them and binds them to corresponding VP connection termination points.

Step 6: The VP-layer Network Manager notifies the provision function that the VP trail has been set up by returning the VP trail identifier to the provisioning function. Now, the VC link creation can proceed.

Interlayer Trail Set-Up Across Multiple Management Domain Outline Ensemble

Network Management Context

This Outline Ensemble addresses a subset of the capabilities of the M4 management interface which performs the Trail connection set-up service. In this, both subNMS are peers. It is an example of network-level distributed management architecture. Note that the trail Managed Entity is needed only if the vc-layer and the vp-layer are managed independently.

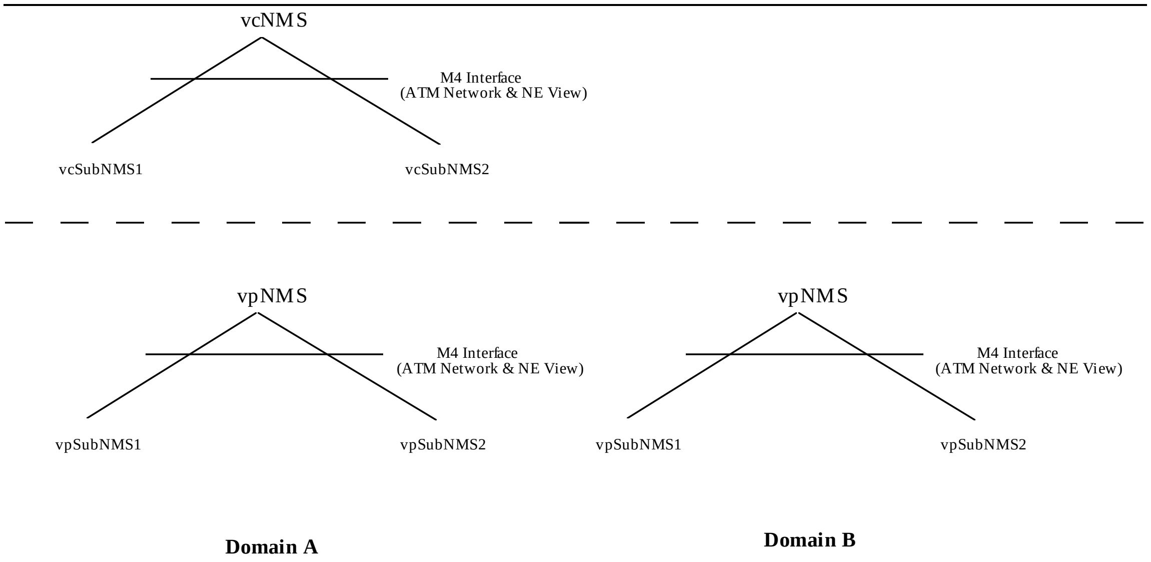


Figure 6.5.2.1: Interlayer Trail Set-Up across Management Domain (NMS view)

Transport Architecture

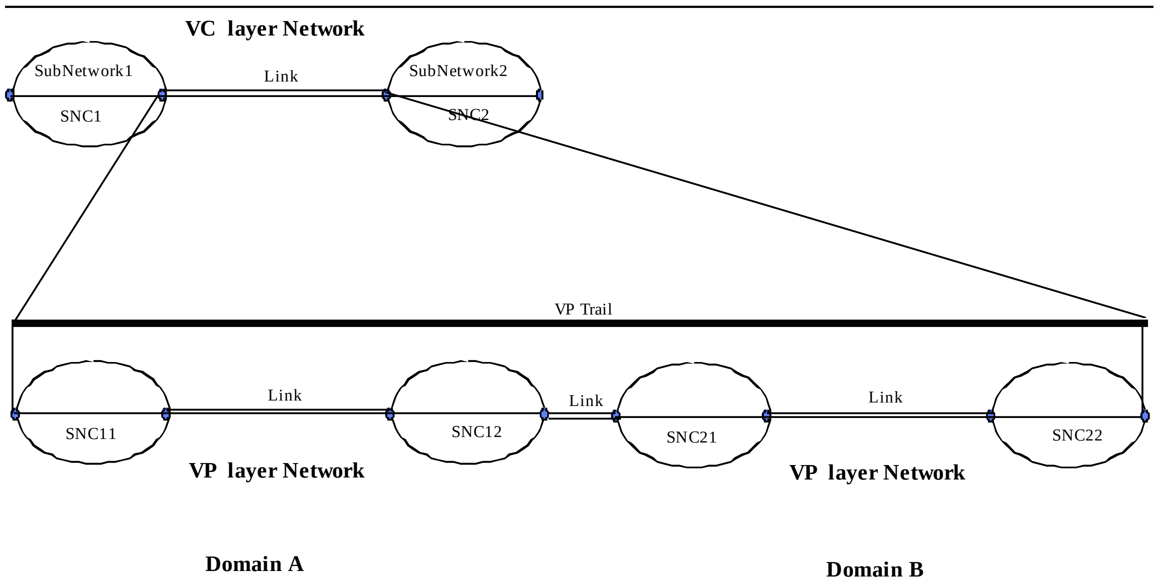


Figure 6.5.2.1: Interlayer Trail Set-Up across Management Domain (Transport Entities)

Functional Requirements

Same as in Section 6.5.1.3

Managed Entities

Managed Entities:

The following Managed Entities are used in this ensemble:

- vcTopologicalLink
- vcLinkConnection
- vpNetworkCTP
- vpSubnetworkConnection
- vpTrail
- vpLinkEnd
- vpLinkConnection

Scenarios

This scenario covers an interlayer trail set-up across two Management Domains

Steps:

Step 1: The VC-layer Network Manager informs the provision function that a link is required between VC subnetworks SNW1 and SNW2. VC-layer Network Manager specifies the end-points to the link, VCI range, maximum number of simultaneously active VCC, etc.

Step 2: The provision function requests one of the VP-layer subnetwork Management System at the edges of the trail to set up a VP-trail with identity of the ATM interface termination point, or with the VPI value of a VP termination within a specific ATM interface and other necessary parameters.

Step 3: The VP-layer subnetwork Management System that is requested to set-up the VP-trail transforms the trail request into subnetwork and link connection set up requests to set up a Virtual Path Subnetwork Connection through the VP layer network.

Step 4: The VP-layer subnetwork management system that receives the requests for subnetwork and link connections creates the requested subnetwork connections within its own domain and the link connection, in association with the requesting the VP-layer subNMS.

Step 5: VP-layer subnetwork management system that receives the requests for subnetwork and link connections creates the requested subnetwork connections within its own domain and the link connection, in association with the requesting the VP-layer subNMS.

Step 5: At the edges of the Virtual Path Subnetwork Connection the VP-layer Network Manager creates VP Trail Termination points, associates VP Trail termination functions to them and binds them to corresponding VP connection termination points.

Step 6: The VP-layer Network Manager notifies the provision function that the VP trail has been set up by returning the VP trail identifier to the provisioning function. Now, the VC link creation can proceed.

Acronyms

ATM: Asynchronous Transfer Mode

BICI: Broadband Inter-Carrier Interface

BISSI: Broadband Inter-Switching System Interface

CTP: Connection Termination Point

EML: Element Management Layer

EMS: Element Management System

EV: Element View

LC: Link Connection

NE: Network Element

NV: Network View

NEF: Network Element Function

NEL: Network Element Layer

NML: Network Management Layer

NMS: Network Management System

PVC: Permanent Virtual Circuit

QOS: Quality Of Service

SNC: Subnetwork connection

subNMS: Subnetwork Management System

TP: Termination Point

TTP: Trail Termination Point

UPC/NPC: User Parameter Control/Network Parameter Control

UNI: User Network Interface

VC: Virtual Channel

VCI: Virtual Channel Identifier

VP: Virtual Path

References:

[1] ITU-T Recommendation I.326, "Functional Architecture of Transport Network Based on ATM," 1995.

[2] ITU-T Recommendation G.805, "Functional Architecture of Transport Network," 1995.

[3] ATM-Forum af-nm-0020.001, "M4 Interface Requirements and Logical MIB; ATM Network Element View," August 1998.

[4] IETF RFC1695, Ahmed, M., Tesink, K., "Definition Functional Architecture of Transport Network."

[5] ITU-T Recommendation G.852-01, "Management of the Transport Network - Entrprise Viewpoint for Simple Subnetwork Connection Management," Draft For Determination, 1996.

[6] ITU-T Recommendation G.853-01, "Common Elements of the Information Viewpoint for the Management of a Transport Network," Draft For Determination, 1996.

[7] ITU-T Recommendation G.854-01, "Computational Interfaces for Basic Transport Network Model," Draft For Determination, 1996.

[8] ATM-Forum af-nm-0073.000, "M4 Network View CMIP MIB Specification Version 1.0" January 1997.

[9] ITU-T Recommendation I.311, "B-ISDN general network aspects," 1993.

[10] ITU-T Recommendation I.610, " B-ISDN operation and maintenance principles and functions," 1995.

Appendix A: Transport Network Architecture Functional Model

This Appendix provides a functional description of the network based on ITU-T Recommendations I.326 [1] and G.805 [2]. Recommendation G.805 provides a generic functional description of the network. Recommendation I.326 focuses specifically on the description of the ATM network functional architecture. In these recommendations, the following Network architectural components have been identified[4].

1 Topological Components

The topological components provide the most abstract description of a network. Four topological components have been distinguished; these are the layer network, the sub-network, the link and the port. Using these components it is possible to completely describe the logical topology of a layer network.

1.1 Layer Network

A layer network is defined by the complete set of like ports which may be associated for the purpose of transferring information. The information transferred is characteristic of the layer and is termed characteristic information. Port associations may be made and broken by a layer management process thus changing its connectivity. A layer network is made up of sub-networks and links between them.

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

1.2 subnetwork

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

[4] The symbols used in the figures of this Appendix are defined in ITU-T Recommendation G.805.

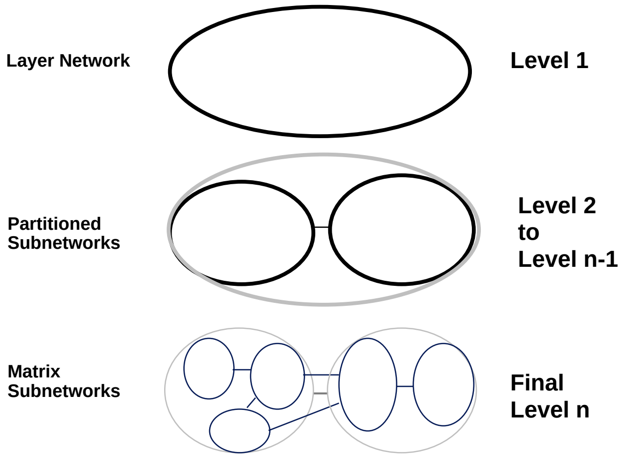


Figure 1 Example of subnetwork Partitioning

1.3 Link

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

1.4 Port

A port represents the output of a trail termination source or unidirectional link connection, or the input to a trail termination sink or unidirectional link connection.

2. Transport Entities

The transport entities provide transparent information transfer within a layer network.

Two basic entities are distinguished according to whether the information transferred is being monitored for integrity or not. These are termed trails and connections. Connections are further distinguished into sub-network connections and link connections according to the topological component to which they belong.

Trails and connections are characterized by their directionality. Connections are also characterized by their modes.

The directionality of a connection or of a trail indicates whether transmission is uni-directional or bi-directional. In this specification, all connections are assumed to be bidirectional, with different bandwidth characteristic in each direction. That way, a unidirectional connection can be represented as a bidirectional connection, with zero return bandwidth.

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

2.1 Trail

A trail in a server layer network is responsible for the integrity of transfer of characteristic information from one or more client layer networks between the server layer ports, utilizing the characteristic information of its own layer. Trail termination functions at either end of the trail monitor the integrity of transfer by adding incremental information to the adapted characteristic information from the client layer networks. These trail termination functions are thought of as being part of the trail. A trail may support one or more links in each client layer network.

2.2 Sub-network Connection

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

2.3 Link Connection

A link connection is supported by a link which 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 trail connection point and a connection point in the case of a link connection at the boundary of a layer network.

3 Transport Processing Functions

Two generic processing functions of adaptation and trail termination are distinguished in describing the architecture of layer networks.

3.1 Adaptation Function

The Adaptation Function is a "transport processing function" which adapts a server layer to the needs of a client layer.

Adaptation source: Adapts the client layer network characteristic information into a form suitable for transport over a trail in the server layer network.

Adaptation sink: Converts the server layer network trail information into the characteristic information of the client layer network

Bidirectional Adaptation: This function is performed by a co-located pair of adaptation source and sink functions.

"Adaptation" functions have been defined for many "client/server" interactions. The following are examples of processes which may occur singly or in combination in an adaptation function; coding, cell extraction; rate changing, aligning, justification, multiplexing.

3.3.2 Trail Termination Function

The Trail termination function is a "transport processing function" which generates the "characteristic information" of a layer network and ensures integrity of that "characteristic information".

Trail termination source: a "transport processing function" which accepts adapted "characteristic information" from a client layer networks at its input, adds information to allow the "trail" to be monitored and presents the characteristic information of the layer network at its output. The trail termination source can operate without an input from a client layer network.

Trail termination sink; a "transport processing function" which accepts the characteristic information of the layer network at its input, removes the information related to "trail" monitoring and presents the remaining information at its output.

Bidirectional Trail termination: a "transport processing function" that consists of a pair of co-located trail terminations source and sink functions.

4 Architecture of an ATM Network

4.1 General

The functional architecture of an ATM network shall follow the generic rules defined here which are consistent with ITU Recommendation G.805 [2]. It is described in Recommendation G.atma [1]. The specific aspects regarding the layer structure, the partitioning and the client/server associations of ATM networks are given below.

4.2 Layer Structure

ATM networks can be decomposed into two independent layers with a client/server association between them namely the virtual path layer and the virtual channel layer. The virtual path layer is a server layer for the virtual channel layer.

4.2.1 Virtual Path Layer Network

The VP layer network allows the transport of ATM cells through VP trail between access points. Access points can be located in the user domain and/or in the network provider domain. The VP layer network contains the following transport functions and transport entities:

- VP trail termination (VPT): generates and terminate the F4 end-to-end OAM cells [I.610].
- VP network connection (VPNC).
- VP link connection (VPLC).
- VP sub-network connection (VPSC).
- VP connection points (VPCP): binds link and subnetwork connections.
- Access points (AP): a reference point that binds the trail termination and adaptation functions.

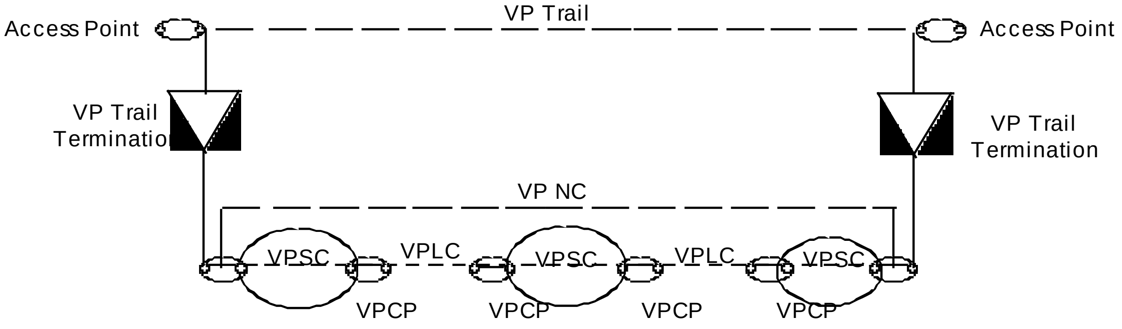


Figure 2 ATM VP Layer Network

3.4.2.2 Virtual channel layer network

The VC layer network allows the transport of ATM cells through VC trail between access points. Access points can be located in the user domain and/or in the network operator domain. The VC layer network contains the following transport functions and transport entities (see figure 2):

- VC trail termination(VCT): generates and terminate the F5 end-to-end OAM cells [I.610].
- VC network connection (VCNC).
- VC link connection (VCLC).
- VC sub-network connection (VCSC).
- VC connection points (VCCP): binds link and subnetwork connections.
- Access points (AP): a reference point that binds the trail termination and adaptation functions.

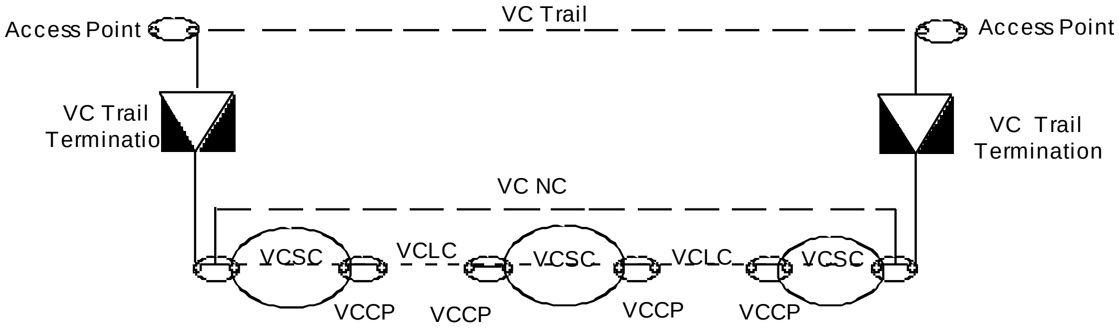


Figure 3 ATM VC Layer Network

3.4.3 ATM Layer Network Adaptations

The overall layer networks relationship of the ATM network is shown in figure 4. A VP link connection is provided by a trail in a transmission path (T-path) layer network. This T-path layer network can be SONET/SDH-based, PDH-based or cell-based. A VC link connection is provided by a trail in the VP layer network. A VC trail is used to transport specific service characteristic information.

Therefore, three types of adaptation functions, described below, have to be defined.

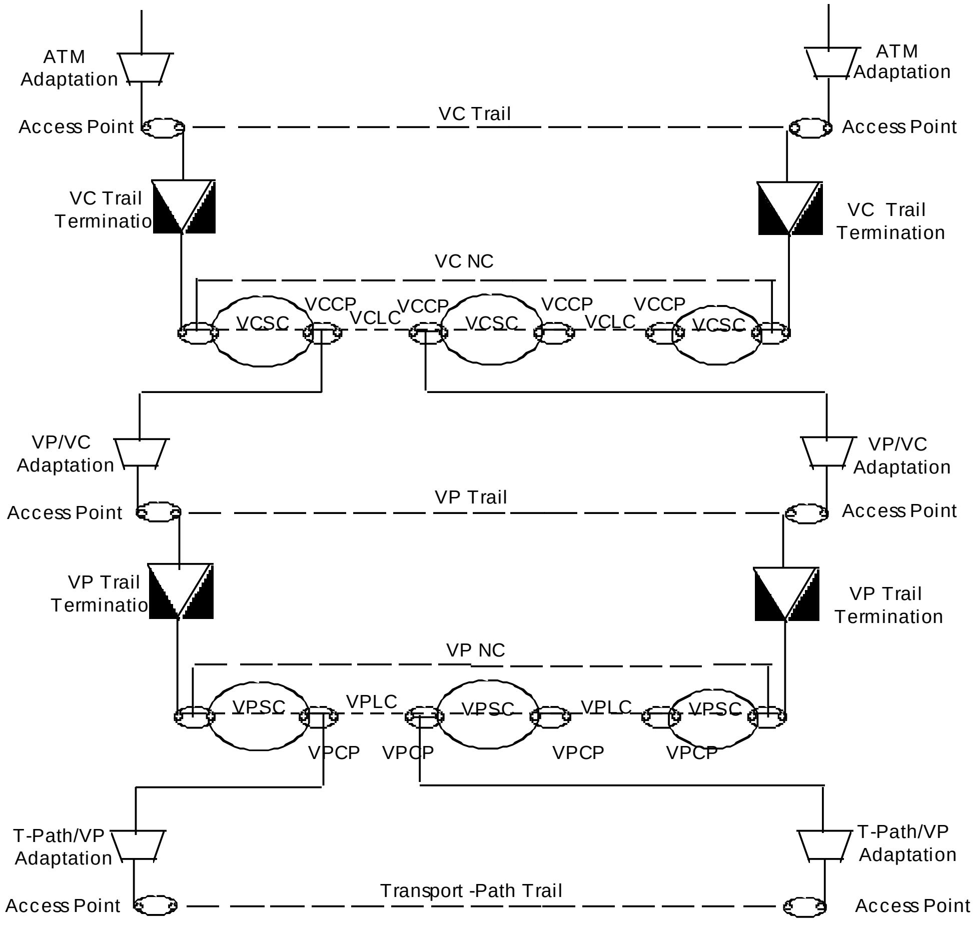


Figure 4 Example of Layer Networks in ATM Context

4.3.1 VP/T-path adaptation

The source VP/T-path adaptation performs, between its input (VP connection point) and its output (T-path access point), VPI allocation, cell multiplexing, including selective cell discards and GFC setting or unassigned cell insertion, idle cell insertion, cell scrambling, HEC generation and cell stream mapping into the T-path payload.

The sink VP/T-path adaptation performs, between its input (T-path access point) and its output (VP connection point), cell stream extraction from the T-path payload, cell delineation, cell descrambling, HEC processing, idle cell removal, cell demultiplexing according to the VPI value, including unmatched VPI cell discards and selective cell discard.

4.3.2 VC/VP adaptation

The source VC/VP adaptation performs, between its input (VC connection point) and its output (VP access point), VCI allocation, and cell multiplexing, including cell discard and meta signaling insertion.

The sink VC/VP adaptation performs, between its input (VP access point) and its output (VC connection point), cell demultiplexing according to the VCI value, meta-signaling extraction and unmatched VCI cell discard.

4.3.3 ATM Adaptation (AAL)

ATM adaptation are defined in ITU-T Recommendation I.363.

Appendix B: TMN Model of the M4 and M5 Interfaces

This Appendix shows the ATM-Forum M4 and M5 interfaces modeled as TMN interfaces[5]. The M4 interface is a TMN Q3 interface between a public Network Management System (or TMN Operation System (OS)), and a TMN Network Element (NE), an Element Management System (EMS), or another Network Management System (NMS). Figure B-1 shows the example of an interface between an NE and an NMS. The M5 interface between two public NMSs of different network providers is a TMN X interface (an interface between OSs in different TMNs).

In the TMN, interfaces (such as Q3 and X) are always between physical building blocks (such as OSs and NEs). The TMN physical building blocks contain one or more function blocks which may communicate amongst each other within the same building block or with function blocks in other building blocks. When function blocks in different building blocks communicate with each other, they use a TMN interface. TMN function blocks include the Network Element Function block (NEF), and the Operation System Function block (OSF). The OSF function block may be specialized into business (B-OSF), Service (S-OSF), Network (N-OSF), and Element (E-OSF) function blocks.

ITU-T Recommendation M.3100, the *Generic Information Model,* defines three management information views:

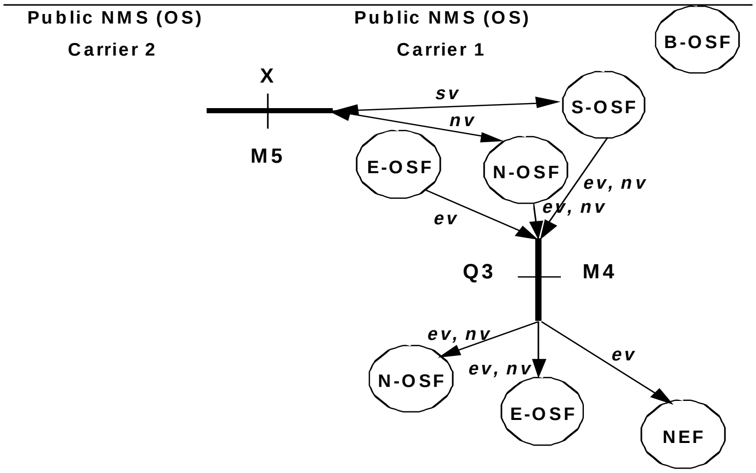
1. "The Network Element view is concerned with the information that is required to manage a Network Element (NE)." This includes "the information required to manage the NEF and the physical aspects of the NE."
2. "The Network view is concerned with the information representing the network, both physically and logically. It is concerned with how network element entities are related, topographically interconnected, and configured to provide and maintain end-to-end connectivity."
3. "The service view is concerned with how network view aspects (such as an end-to-end path) are utilized to provide a network service, and, as such, is concerned with the requirements of a network service (e.g. availability, cost, etc.), and how these requirements are met through the use of the network, and all related customer information."

"Objects defined for a given view may be used in others, and any object may be used by any interface which require it.

The M4 NE-view (*ev*) and the network view (*nv*) are ATM-specific implementations of the Network Element view and of the Network view respectively. As an example, in Figure B-1, an Element Management Function block (an E-OSF) in the public network provider OS communicates with the NEF in the ATM Network using objects from the M4 Network Element view (*ev*). An N-OSF may use both

[5] Note that the terminology used in this Appendix is the terminology defined in ITU-T Recommendation M.3010. It refers to Operation Systems (OS) which map into Network Management Systems (NMS) and Element Management Systems (EMS). In the same fashion, Operation System Functions (OSF) map to the Network Management Layer (NML) functions and to the Element Management Layer (EML) functions.

the M4 *ev* and *nv* when communicating with an OSF in the ATM network, but would only use the M4 *ev* when communication with the NEF. Note that *nv* denotes only objects which are unique to the network view, and that the complete network view uses both the *ev* and the *nv*.



TMN Keys:

OS: Operation System
OSF: Operation System Function
B-OSF: Business OSF
S-OSF: Service OSF
N-OSF: Network OSF
E-OSF: Element OSF
NE: Network Element
NEF: Network Element Function
Q3: TMN Q3 Interface
X: TMN X Interface

ATM-Forum Keys:

sv: Service View
nv: Network View
ev: Network Element View
M4: Network Provider NMS to ATM NE, EMS, (sub)NMS interface
M5: Network Provider to Network Provider Management System interface

Figure B-1: M4 and M5 Interfaces Modeled In terms of the TMN Architecture (Example)

Appendix C: Naming

This information is provided here as a place-holder. It may better belong in the NE MIB or in an implementor’s guide:

(R-App-C)-M4: Only local naming is required. The local distinguished name shall start at the NE RDN itself. A network object may be implemented at the subnetwork level.

(R-App-C)-M5: Managed entities names shall be unique.