The ATM Forum

Technical Committee

M4 Network-View Interface Requirements, and Logical MIB

AF-NM-0058.000

1. 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 [2], the scope of this specification is on requirements and logical MIB aspects needed to support the aggregated or "network view."

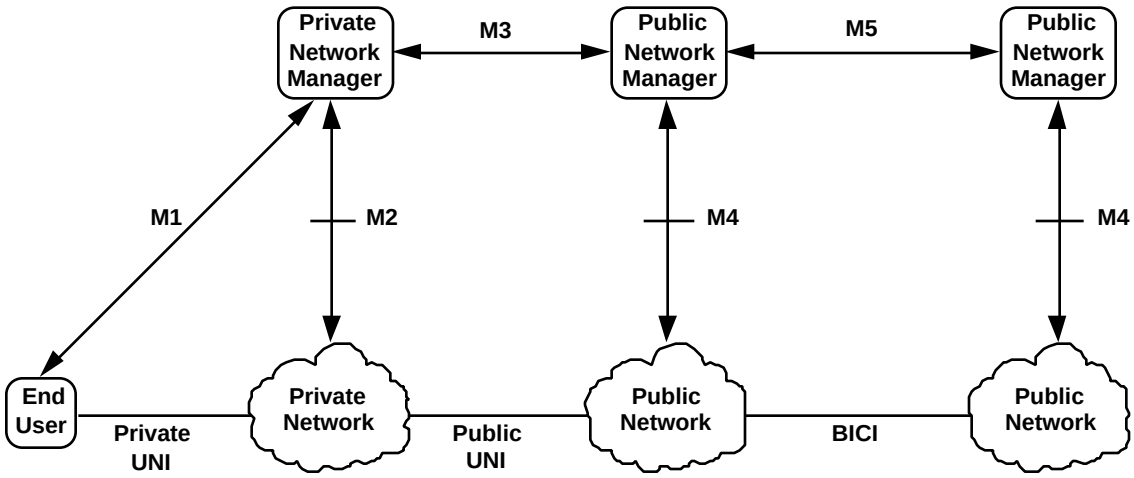


Figure 1-1 The ATM Forum Management Interface Reference Architecture

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-

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

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.

1.1 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. Although it is architecturally permissible to offer NE-view only or network-view only management services, it is to be kept in mind that the network-view is intended as an aggregate view to provide additional value. In this sense, the network-view should not constitute an opaque obstacle for reaching the NE; but it should be conceived as an organized way of seeking NE details, when needed.

This specification focuses on what is considered to be the initial functionality of 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 subnetwork, or link connections, or of trails,
- multipoint capabilities,
- routing constraints, except for topology,
- segment handling,
- scheduling or reservation,
- performance monitoring
- accounting
- security.

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

1.2 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:

1.2.1 IETF SNMP RFC

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

1.2.2 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 interfaces specification, specialized for ATM.

1.3 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 [8], 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	---
Subnetwork TP	Subnetwork TP	---	---	Connecting Point
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 [9] 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 [9].

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.

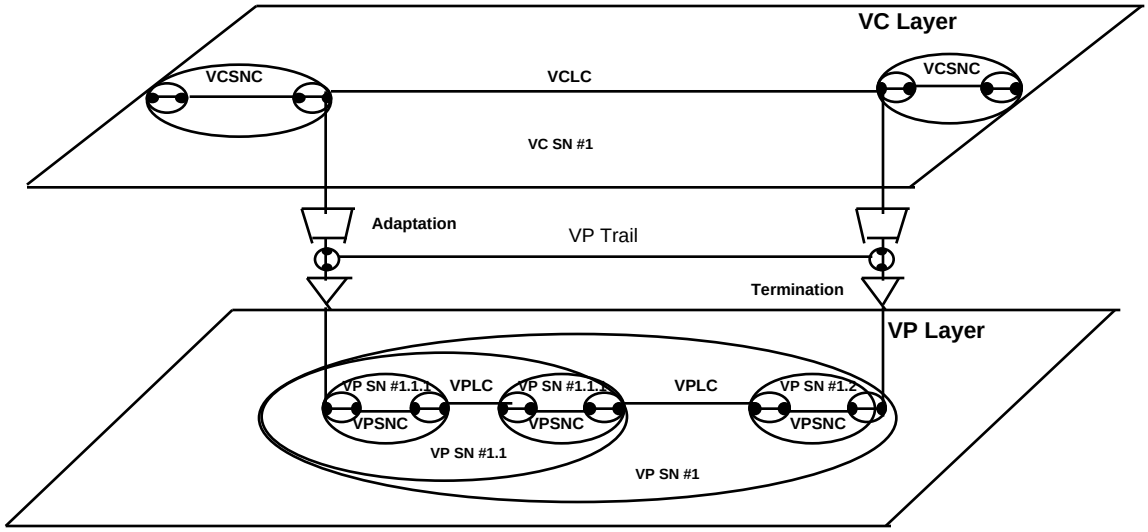
Note 3: "Link" was used in the M4 NE-view requirements and logical MIB [3], as a link between two network elements, according to I.311 [9], which is equivalent to a "link connection" in I.326 [1]/G.805 [2] language. Here, a "link" is used as a link between two subnetworks, as defined in I.326 [1]/G.805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

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.

3. 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.

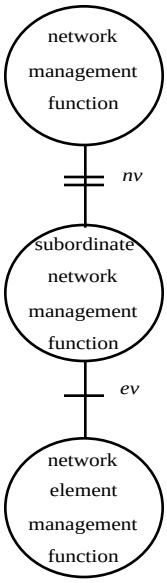


Figure 3-1: M4 Network-View Logical Architecture

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.

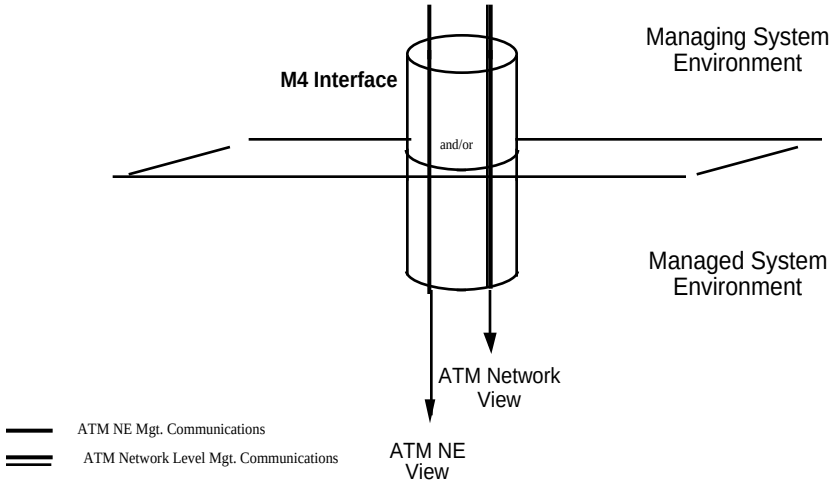


Figure 3-2: Dual Views of the M4 Interface

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

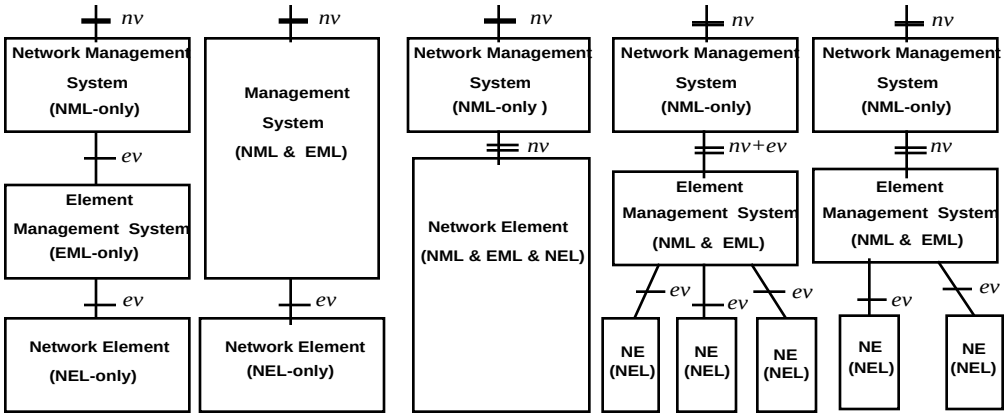


Figure 3-3-a: hierarchical examples

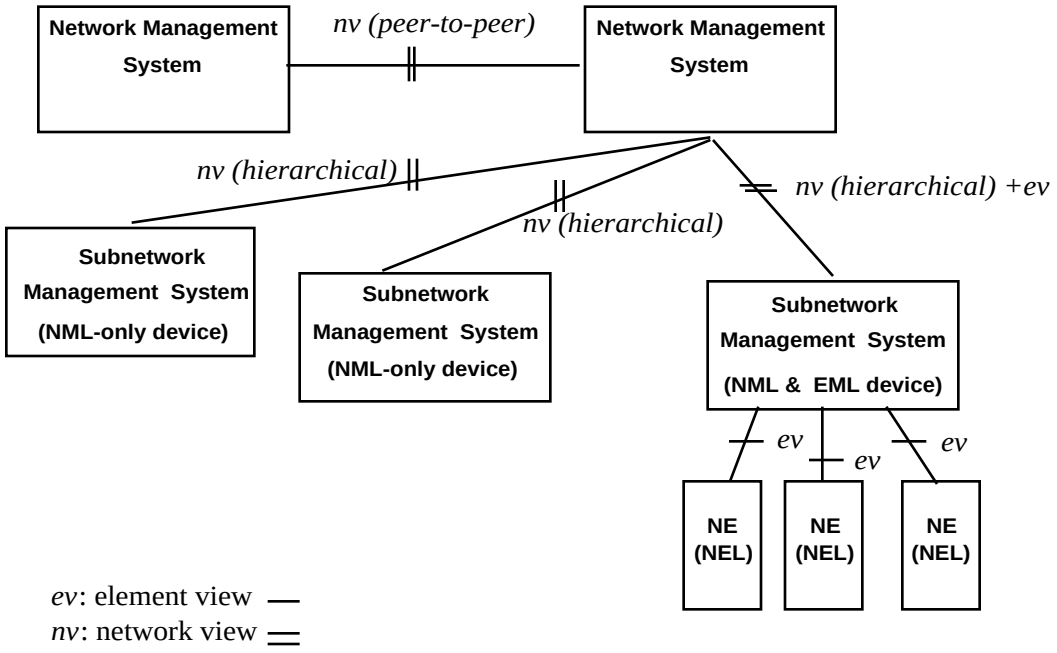


Figure 3-3-b: hierarchical and non-hierarchical examples

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:

3.1 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.

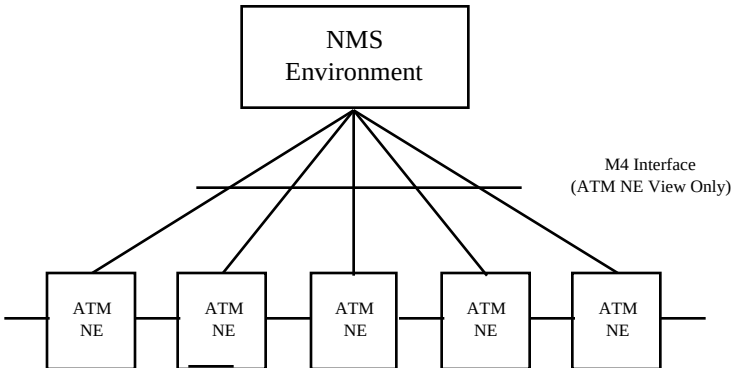


Figure 3-1-1: NE-View Management Architecture

3.2 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.

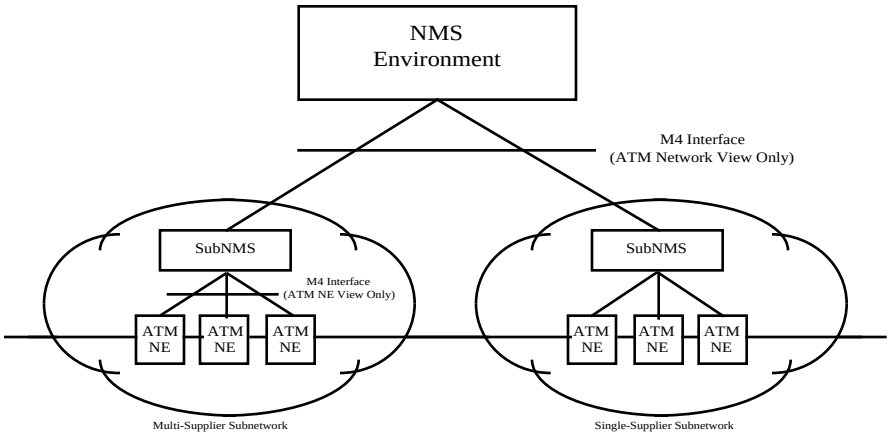


Figure 3-2-1: Example of Network-View Management Physical Configuration

3.3 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.

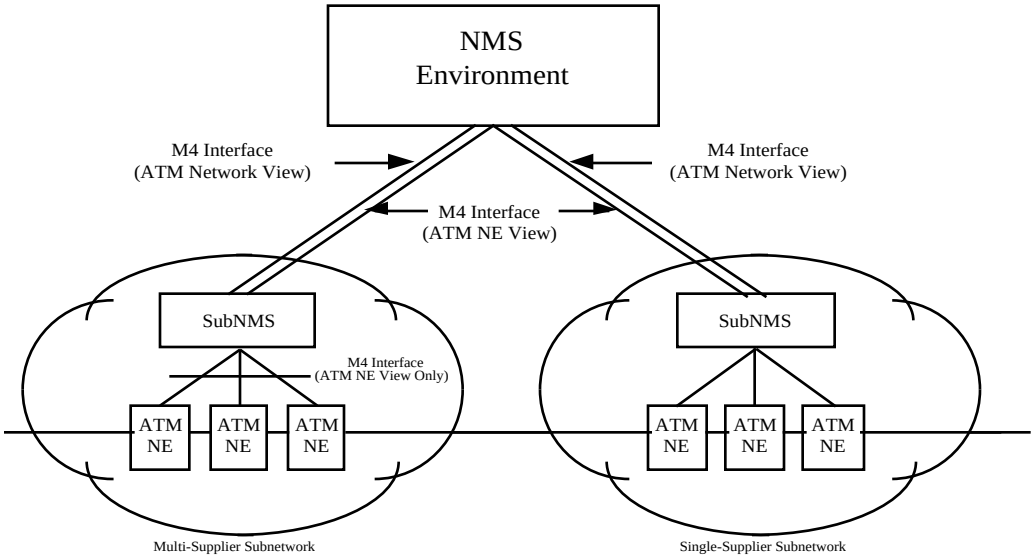


Figure 3-3-1: Example of NE+Network-Level Management Physical Configuration

3.4 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.

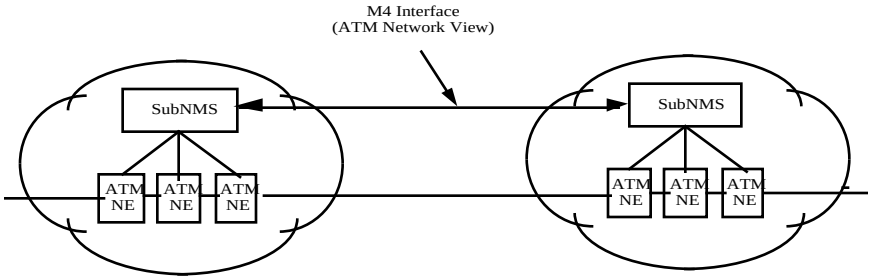


Figure 3-4-1: Example of Non-hierarchical Management Physical Configuration

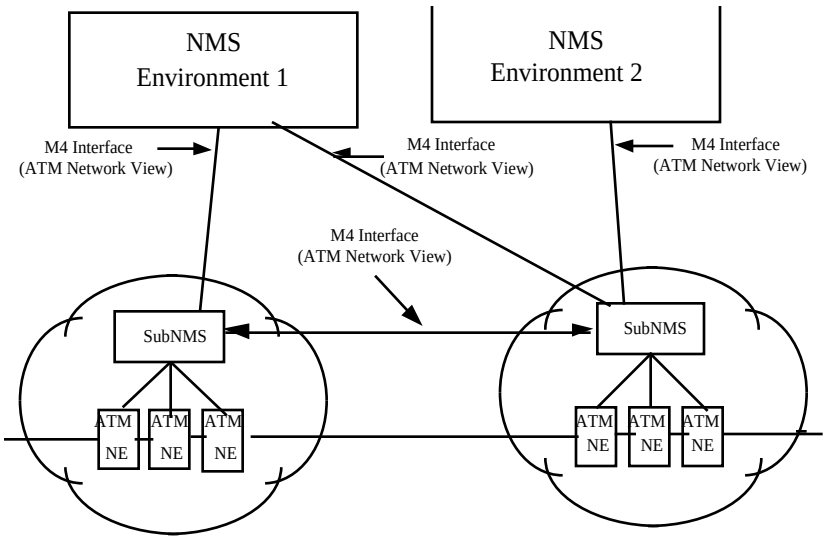


Figure 3-4-2: Example of Multi-Manager Management Physical Configuration

4. 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.

4.1 Transport Network Configuration Management

4.1.1 Transport Network Provisioning/Layered Network Provisioning

4.1.1.1 Subnetwork Provisioning

In the first phase of this specification, it is assumed that a subnetwork is 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.

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 -9: A termination point representing an actual resource can be a member of more than one subnetwork. Where this is the case, a unique subnetwork termination point shall be provisioned for each occurrence with a relationship to the underlying network termination point which it represents.

4.1.1.2 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 requirements below support that functionality.

ATM Link:

(R-4.1.1.2)-cm -1: The M4 interface shall support requests to set-up an ATM link between two subnetworks. In order to establish a ATM 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 provisioned bandwidth, i.e. the bandwidth allocated to the link

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 -2: The M4 interface shall support requests to modify the provisioned bandwidth of an ATM link between two subnetworks.

(R-4.1.1.2)-cm -3: 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 -4 : The M4 interface shall support requests to retrieve the provisioned and the available bandwidth of a ATM link.

4.1.1.3 Transport Path Provisioning:

FFS

4.1.2 Subnetwork Connection Management

The M4 network-view managed entities supports 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 supports 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 being Terminated

2. VPI and/or VCI Value
3. Ingress and Egress Peak Cell Rate for CLP=0 and CLP=0+1 Traffic
4. Ingress and Egress Sustainable Cell Rate for CLP=0 and CLP=0+1 Traffic
5. Ingress and Egress Maximum Burst Tolerance for CLP=0 and CLP=0+1 Traffic
6. Ingress and Egress CDV Tolerance for CLP=0 and CLP=0+1 Traffic
7. Ingress and Egress QOS class

4.1.2.1 Connection Set-Up

4.1.2.1.1 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:

Note that multipoint connections are not currently supported by the managed entities defined in Section 5. The definition of the necessary managed entities is FFS. These managed entities shall be consistent with I.326 definitions [1].

(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. an unique identifier for each leg established
4. a list of the endpoints which were not setup, if any

4.1.2.1.2 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.

4.1.2.1.3 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.

4.1.2.1.4 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.

4.1.2.2 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. 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. 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-4.1.2.2)-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. 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. 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-4.1.2.2)-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.

4.1.2.3 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 trail is released, as an option, the M4 interface should allow the link connections and subnetwork connections that were part of the trail to remain.

4.1.2.4 Subnetwork State Management

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

(R-4.2.2)-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-4.2.2)-sm-2 : The M4 interface shall be autonomously be notified of subnetwork connection protection switches, if any.

(R-4.2.2)-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-4.2.2)-sm-1: The M4 interface should be capable of autonomously notifying trail operational state changes.

(R-4.2.2)-cm -5: The M4 interface shall be support requests to suspend autonomous reporting of subnetwork connections.

(R-4.2.2)-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.

4.2 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.

4.2.1 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.

4.2.2 Connection Testing

Loopback Test

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

(R-4.2.2.2) -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

4.3 Transport Network Performance Management

4.3.1 Congestion Performance Monitoring

FFS

4.3.2 Connection Performance Monitoring

FFS

4.3.3 Segment Performance Monitoring

FFS

4.4 Network Accounting Management

FFS

4.5 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.

5. Protocol Independent MIB

5.1 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 subnetwork, or link connections, or of trails,
- multipoint capabilities,
- routing constraints, except for topology,
- segment handling,
- scheduling or reservation,
- performance monitoring
-

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.

5.1.1 Managed Entity List:

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

network
vcLayerNetworkDomain
vcLinkConnection
vcSubnetwork
vcSubnetworkConnection
vcSubnetworkTP
vcTopologicalLink
vcTopologicalLinkTP
vcTrail
vpLayerNetworkDomain
vpLinkConnection
vpSubnetwork
vpSubnetworkConnection
vpSubnetworkTP
vpTopologicalLink
vpTopologicalLinkTP
vpTrail

The managed entities below have the same information content as in the NE-view, and are specified here because of their relationships with the network-view managed entities.

vcCTP
vcTTP
vpCTP
vpTTP

The following managed entities needed in the M4 network-view are already specified in the NE-view:

alarmRecord
alarmSeverityAssignmentProfile
eventForwardingDiscriminator
log

5.1.2 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

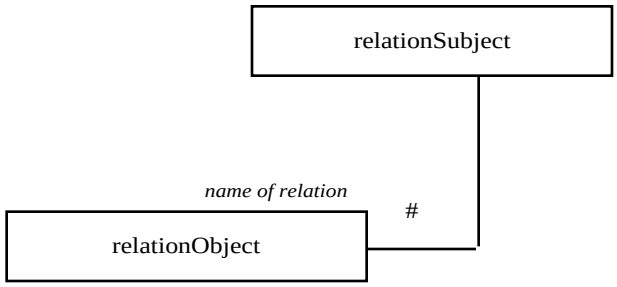
(n..m) n to m

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

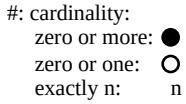
4/ The containment relationship is denoted by "GROUPS," "IS_MADE_OF" so as not to imply the use of containment as defined in OSI management.

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:



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.

5.2 Managed Entity Descriptions:

5.2.1 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.

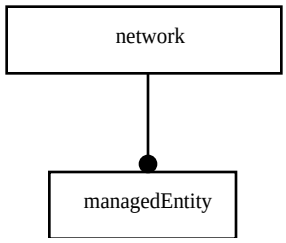
Relationships:

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

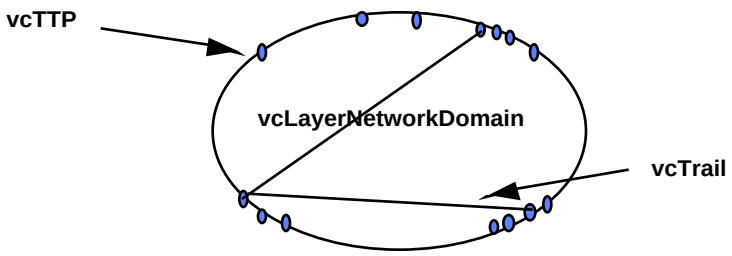
BEHAVIOUR

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.

5.2.2 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.

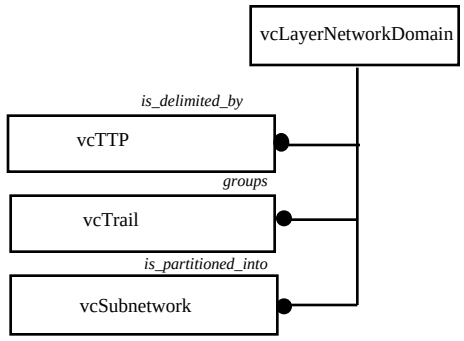
Semi-Formal representation:

IS_DELIMITED_BY (0..*) vcTTP

GROUPS (0..*) vcTrail

IS_PARTITIONED_INTO (0..*) vcSubnetwork

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 vcSubnetworkTPId, or Descriptor
vcTTPz : choice of vcSubnetworkTPId, or Descriptor
AdministrativeState (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: release vcTrail

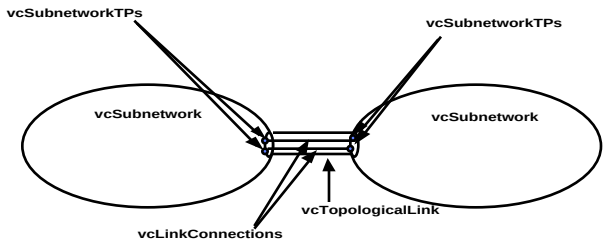
INPUT PARAMETERS:
see Section 5.1.2, and
Choice of VpTTP : set of vcTTPIds
OUTPUT PARAMETERS:
released vcTrail: vcTrailId
released vcTTPs: set of vcTTPId
ERROR CONDITIONS:
protocol-specific addressing errors
incorrectTerminationPoints : vcTTPId
vcTrailTerminationPointNotConnected : vcTTPId
incorrectTrails: 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.

5.2.3 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.

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

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

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.

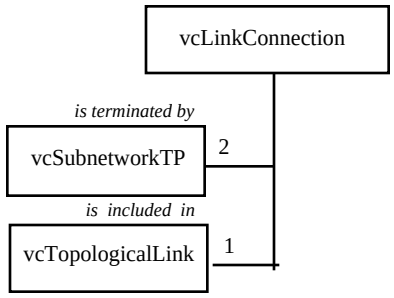
With vcSubnetworkTPs: A vcLinkConnection has two vcSubnetworkTPs, one on each subnetwork that it is linking. The two vcSubnetworkTerminationPoints belong to two different subnetworks.

semi-formal representation:

IS_INCLUDED_IN (1) vcTopologicalLink

IS_TERMINATED_BY (2) vcSubnetworkTP

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.

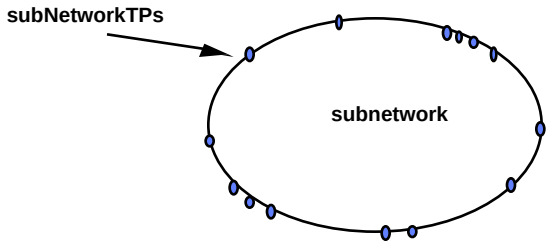
Operation: query vcLinkConnection For terminating vcSubnetworkTPs

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

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

5.2.4 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 subnetwork termination points, modeled by subnetwork TP managed entities. 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.



This phase of the specification only supports automatic creation of this managed entity. Deletion is not supported.

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.

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 vcSubnetworkTPs: a vcSubnetwork is delineated by zero or more vcSubnetworkTPs.

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 vcTopologicalLinkTP: a subnetwork is delineated by zero or more vcTopologicalLinkTPs.

Semi-Formal representation:

IS_DELINEATED_BY (0..*) vcSubnetworkTP

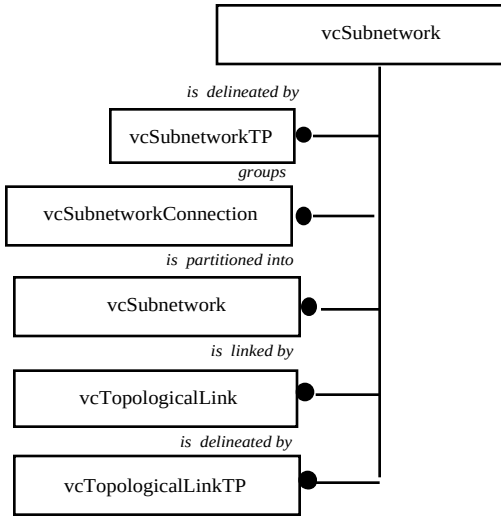
GROUPS (0..*) vcSubnetworkConnection

IS_PARTITIONED_INTO (0..*) vcSubnetwork

IS_LINKED_BY (0..*) vcTopologicalLink

IS_DELINEATED_BY (0..*) vcTopologicalLinkTP

Graphical representation:



Subnetwork Query Operations

Operation: query vcSubnetwork For delimiting vcSubnetworkTPs

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

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

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 vcTopologicalLinkTPs

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
delineatingvcTopologicalLinkTPs : set of vcTopologicalLinkTPIds

ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each vcTopologicalLinkTP is delineating a vcSubnetwork. This operation allows the requester (client) to query the delineating Managed Entities. It does not affect the relationship. It matches for all the delineating vcTopologicalLinkTPs. The reply is a set of vcTopologicalLinkTPIds.

Subnetwork: SubnetworkConnection management operations

Operation: set up vcSubnetworkConnection

INPUT PARAMETERS:
see Section 5.1.2, and
snTPa : choice of subnetworkTPId, or Descriptor
snTPz : choice of subnetworkTPId, or Descriptor
AdministrativeState (optional)

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

This operation allows the requester (user) to set-up a point-to-point connection between two non-connected subnetworkTPs of the addressed subnetwork. The subnetworkTPs 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 subnetworkTPs 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 subnetworkTPs and a subnetworkConnection associated to both of them in the case of the subnetwork creating the subnetworkTPs
- the creation of a subnetworkConnection associated to two existing subnetworkTPs in the case of subnetworkTPs already existing in the subnetwork.

Operation*:* release vcSubnetworkConnection

INPUT PARAMETERS:

none (see Section 5.1.2)
Choice of subnetworkTP : set of subnetworkTPIds

OUTPUT PARAMETERS:

vcSubnetworkConnectionID: vcSubnetworkConnectionId
released subnetworkTPs: set of subnetworkTPId

ERROR CONDITIONS:

protocol-specific addressing errors
incorrectvcsubnetworkTPs : subnetworkTPIds
subnetworkTPNotConnected : subnetworkTPId
incorrectsubnetworkConnection: subnetworkConnectionId

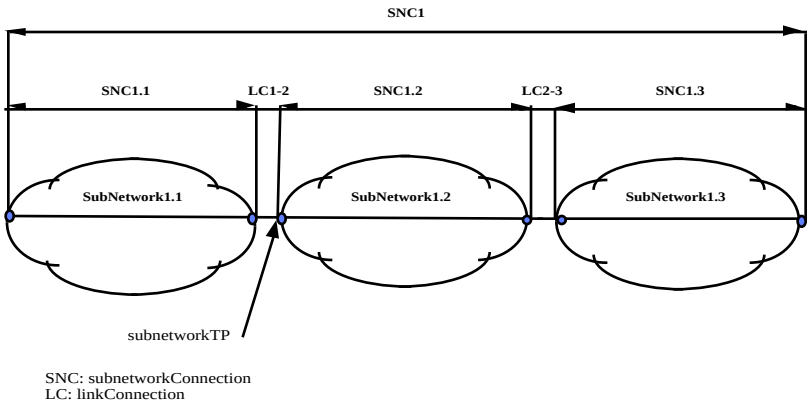
BEHAVIOUR

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

5.2.5 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).



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."

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

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.

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 subnetworkTPs: A subnetwork connection has at least two subnetworkTPs.

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.

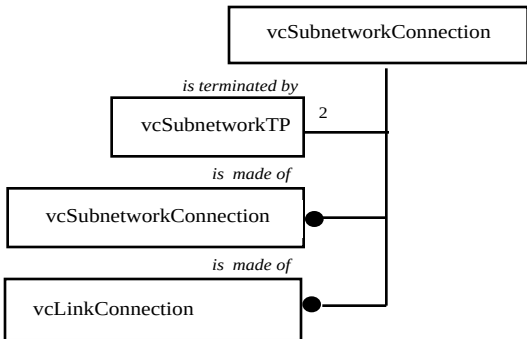
Semi-formal representation:

IS_TERMINATED_BY (2..*) subnetworkTP

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

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

Graphical representation:



subnetworkConnection: subnetworkConnection query operations

Operation: query subnetworkConnection For terminating subnetworkTPs

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatingSubnetworkTPs: set of subnetworkTPIds
ERROR CONDITIONS:
protocol-specific

BEHAVIOUR:

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

Operation: query vcSubnetworkConnection For Component vcSubnetworkConnections

INPUT PARAMETERS:

none (see Section 5.1.2)

OUTPUT PARAMETERS:

componentSubnetworkConnections: set of subnetworkConnectionIds

ERROR CONDITIONS:

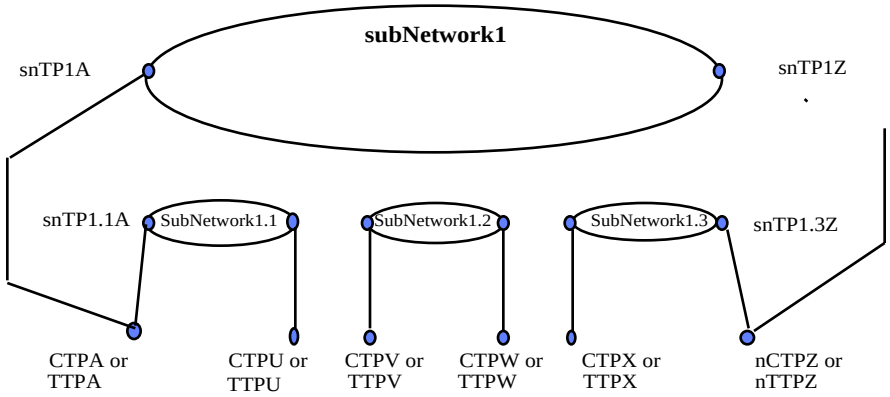
protocol-specific

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.

5.2.6 vcSubnetworkTP

The subnetworkTP Managed Entity terminates a subnetworkConnection or linkConnection on a subnetwork. Each subnetworkTP refers to a CTP or a TTP managed entity, and therefore allows a single representation of the network resources. As an example, a subnetwork may be partitioned into component subnetwork several times. Rather than creating a new CTP or TTP for each layer of decomposition, this approach to keep the CTP or TTP at the lowest layer, and to refer to them using subnetworkTPs. It represents the relationship between the endpoints of composite subnetworks supported by the same resource. This entity is explicitly created by a network management function.



Attributes:

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

Relationships :

With vcCTP/TTPs: Each subnetworkTP reflects the underlying CTP or possibly the underlying TTP.

With vcSubnetwork: each subnetworkTP delineates a subnetwork.

With vcLinkConnection: each subnetworkTP terminates a linkConnection.

With vcSubnetworkConnection: each subnetworkTP terminates a subnetworkConnection.

Semi-formal representation:

REFLECTS (0..1) vcCTP

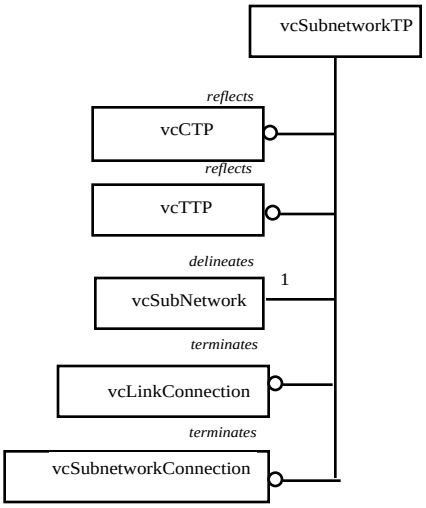
REFLECTS (0..1) vcTTP

DELINEATES (1) subnetwork

TERMINATES (0..1) vcLinkConnection

TERMINATES (0..1) vcSubnetworkConnection

Graphical representation:



subnetworkTP: subnetworkTP Query Operations

Operation: query subnetworkTPs For delimited subnetwork

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

The subnetwork plays a role of container for the subnetworkTPs. This operation allows the requester (client) to query the containing subnetwork for a given subnetworkTP. It does not affect the relationship. It matches for the delineated subnetwork. The reply is a subnetworkId.

Operation: query subnetworkTP For terminated vcLinkConnection

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatedVcLinkConnections: vcLinkConnectionIds
ERROR CONDITIONS:

protocol-specific

BEHAVIOUR

A linkConnection is terminated on two subnetworkTPs. This operation allows the requester (client) to query the terminated linkConnection. It does not affect the relationship. It matches for the terminated linkConnection. The reply is a linkConnectionId.

Operation: query subnetworkTP For terminated subnetworkConnection

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatedSubnetworkConnection: subnetworkConnectionId
ERROR CONDITIONS:
protocol-specific

A point-to-point subnetworkConnection is terminated on two subnetworkTPs. This operation allows the requester (client) to query the terminated subnetwork Connection. It does not affect the relationship. It matches for the terminated subnetworkConnection. The reply is a subnetworkConnectionId.

Operation: query subnetworkTP For associated TTPs and CTPs

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
associatedTP: CTPId or TTPId
ERROR CONDITIONS:
protocol-specific

Each subnetworkTP is associated with a CTP or TTP. This operation allows the requester (client) to query the associated Managed Entity. It does not affect the relationship. It matches for the associated CTP or TTP. The reply is a CTPId or TTPId.

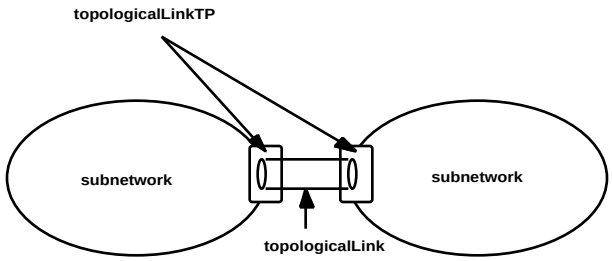
Operation*:* associate subnetworkTP with supporting CTP or TTP

INPUT PARAMETERS:
none (see Section 5.1.2)
associatedCTPorTTPId
OUTPUT PARAMETERS:
associatedTTPorCTP: associated CTPId or TTPId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each subnetworkTP is supported by either a CTP or a TTP, which provides transport for it. This operation allows the requester (client) to associate the subnetworkTP with a single desired CTP or TTP. This operation may take place at the creation of the subnetworkTP. The reply is the associated CTP or TTP Id.

5.2.7 vcTopologicalLink

A link is a topological component which describes a fixed relationship between two subnetworks. A topological link is a link between two subnetworks. 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 is 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.

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 topologicalLinkTP: A topologicalLink has two termination 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.

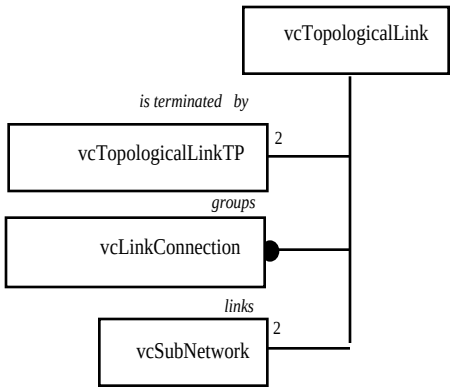
semi-formal representation:

GROUPS (0..*) vcLinkConnections

IS_TERMINATED_BY (2) vcTopologicalLinkTP

LINKS (2) vcSubnetworks

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 vcTopologicalLinkTPs

INPUT PARAMETERS:

none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatingTopologicalLinkTPs: set of topologicalLinkTPIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each topologicalLink is terminated by two topologicalLinkTPs. This operation allows the requester (client) to query the terminating managed entities. It does not affect the relationship. It matches for the terminating topologicalLinkTPs. The reply is a set of topologicalLinkTPs 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
snTPa : choice of subnetworkTPId, or Descriptor
snTPz : choice of subnetworkTPId, or Descriptor

Descriptor:
interfaceId (Choice of subnetworkTP, or server TTPId, or topologicalLinkTP, or topologicalLink, or subnetworkId)
vpi (optional)
vci (optional)
trafficDescriptors (optional)
qos (optional)
OUTPUT PARAMETERS:
newVcLinkConnection : vcLinkConnectionId
snTPa: subnetworkTPId
snTPz: subnetworkTPId
ERROR CONDITIONS:
protocol-specific addressing errors, and
incorrectTerminationPoints : subnetworkTPId
reflectedTPDisabled : subnetworkTPId
reflectedTPLocked: subnetworkTPId
topologicalLinkLocked: topologicalLinkId
subnetworkTerminationPointInUse : subnetworkTPId
non-matchingDescriptors: set of subnetworkTPId
operationFails

BEHAVIOUR:

This operation allows the requester (user) to set-up a linkConnection between two non-connected subnetworkTPs 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 subnetworkTPs are incorrect (e.g. do not belong to the subnetwork), if the two subnetworkTPs 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 subnetworkTPs and an linkConnection associated to both of them in the case of the subnetwork creating the subnetworkTPs
- the creation of a linkConnection associated to two existing subnetworkTPs in the case of subnetworkTPs already existing in the subnetwork.

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

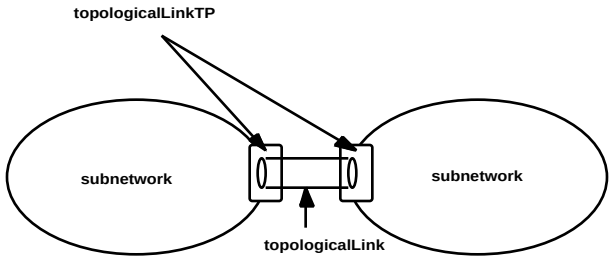
Operation: release vcLinkConnection

INPUT PARAMETERS:
none (see Section 5.1.2)
Choice of subnetworkTP : set of subnetworkTPIds
OUTPUT PARAMETERS:
vcSubnetworkConnectionID: vcSubnetworkConnectionId
released subnetworkTPs: set of subnetworkTPId
ERROR CONDITIONS:
protocol-specific addressing errors
incorrectSubnetworkTPs : subnetworkTPIds
subnetworkTPNotConnected : subnetworkTPId
incorrectLinkConnection: linkConnectionId
BEHAVIOUR

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

5.2.8 vcTopologicalLinkTP

This managed entity is used to represent the termination of a topologicalLink at the VC-layer. When a network is managed only at the VC-layer, the vcLink managed entity is used to store any link-level configuration data. This managed entity is created by the management system.



Attributes

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

Provisioned Bandwidth: see vcTopologicalLink managed entity definition

Available Bandwidth: see vcTopologicalLink managed entity definition

VCI Range: This read/write parameter identifies the maximum number of contiguous VPI bits, starting from the least significant bit, that may be used over the link.

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 vcTopologicalLinkTP managed entity.

With vcSubnetwork: One vcTopologicalLinkTP managed entity is associated with one and only one vcSubnetwork.

semi-formal representation:

IS_ASSOCIATED_WITH (1) vcSubnetwork

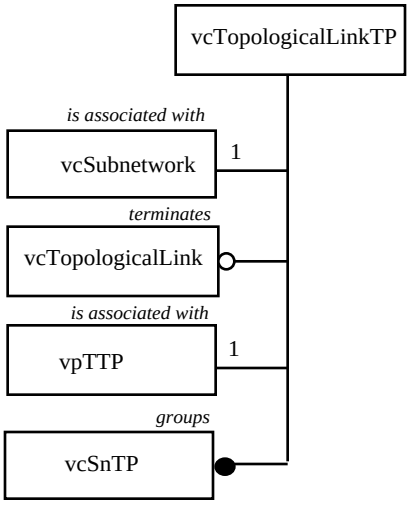
IS_ASSOCIATED_WITH (1) vpTTP

TERMINATES (0..1) vcTopologicalLink

Add relationship between linkTP and snTP YES

GROUPS (0..*) vcSnTP

graphical representation:



vcTopologicalLinkTP: vcTopologicalLinkTP Query Operations

Operation: query vcTopologicalLinkTP For Terminated vcTopologicalLink

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

Each topologicalLink is terminated by two topologicalLinkTPs. 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 vcTopologicalLinkTP For Delineated vcSubnetwork

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
delineatedSubnetworks: subnetworkIds
ERROR CONDITIONS:

protocol-specific

BEHAVIOUR

Each topologicalLinkTP 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 vcTopologicalLinkTP For associated vpTTP

INPUT PARAMETERS:

none (see Section 5.1.2)

OUTPUT PARAMETERS:

associatedVcTTP: vcTTPId

ERROR CONDITIONS:

protocol-specific

BEHAVIOUR

Each vcTopologicalLinkTP 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 vcTopologicalLinkTP with supporting vpTTP

INPUT PARAMETERS:

none (see Section 5.1.2)
associatedVpTTP

OUTPUT PARAMETERS:

associatedVpTTP: vpTTPId

ERROR CONDITIONS:

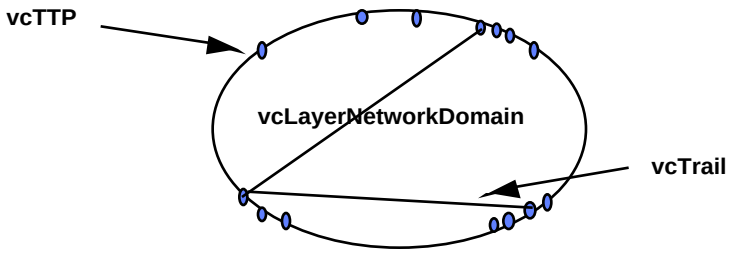
protocol-specific

BEHAVIOUR

Each vcTopologicalLinkTP 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 vcTopologicalLinkTP with a single vpTTP. This operation may take place at the creation of the vcTopologicalLinkTP or of the vcTopologicalLinkTP. The reply is the associated vcTTP.

5.2.9 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.

Operational State: This read-only attribute identifies whether or not the managed entity is capable of performing its normal functions (i.e., in-service or out-of-service).

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.

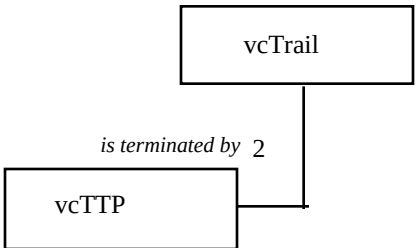
Relationships:

With vcTTPBidirectional: Each VC trail is terminated by *vcTrailTerminationPointsBidirectional.*

Semi-formal representation:

IS_TERMINATED_BY (2) vcTTP

Graphical representation:



vcTrail: vcTrail query operations

Operation: query vcTrail For terminating TTPs

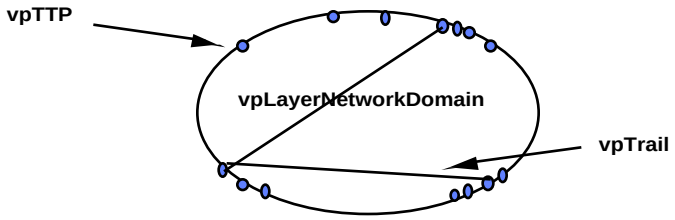
INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatingTTPs: set of TTPIds
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

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.

5.2.10 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.

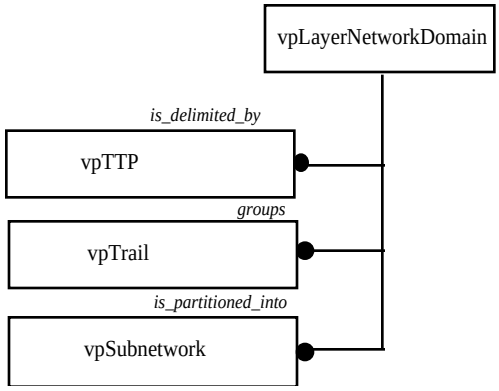
Semi-Formal representation:

IS_DELIMITED_BY (0..*) vpTTP

GROUPS (0..*) vpTrail

IS_PARTITIONED_INTO (0..*) vpSubnetwork

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 vpSubnetworkTPId, or Descriptor
vpTTPz : choice of vpSubnetworkTPId, or Descriptor
AdministrativeState (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: release vpTrail

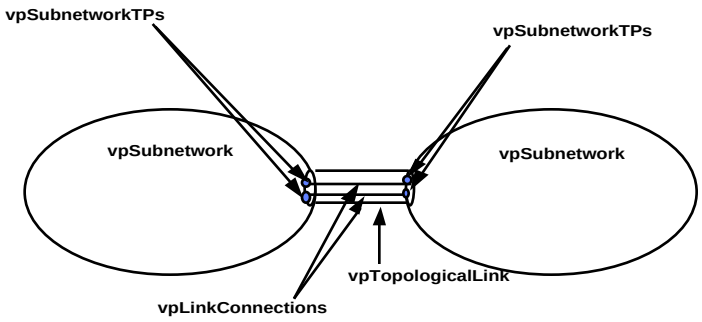
INPUT PARAMETERS:
Choice of VpTTP : set of vpTTPIds
OUTPUT PARAMETERS:
released vpTrail: vpTrailId
released vpTTPs: set of vpTTPId
ERROR CONDITIONS:
protocol-specific addressing errors
incorrectTerminationPoints : vpTTPId
vpTrailTerminationPointNotConnected : vpTTPId
incorrectTrails: vpTrailId
BEHAVIOUR:

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.

5.2.11 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 subnetwork. 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.

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

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

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 vpTopologicalLinks: 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.

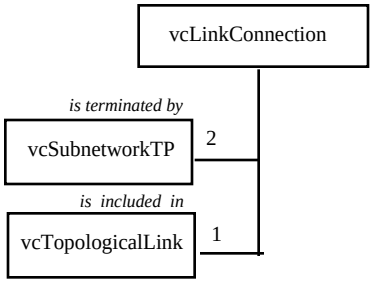
With vpSubnetworkTPs: A vpLinkConnection has two vpSubnetworkTPs, one on each subnetwork that it is linking. The two vpSubnetworkTerminationPoints belong to two different subnetworks.

semi-formal representation:

IS_INCLUDED_IN (1) vpTopologicalLink

IS_TERMINATED_BY (2) vpSubnetworkTP

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.

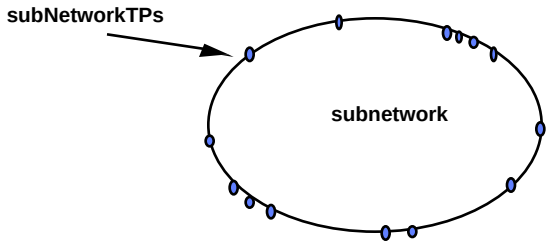
Operation: query vpLinkConnection For terminating vpSubnetworkTPs

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

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

5.2.12 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 subnetwork termination points, modeled by subnetwork TP managed entities. 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.



This phase of the specification only supports automatic creation of this managed entity. Deletion is not supported.

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.

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 vpSubnetworkTPs: a vpSubnetwork is delineated by zero or more vpSubnetworkTPs.

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 vpTopologicalLinkTP: a subnetwork is delineated by zero or more vpTopologicalLinkTPs.

Semi-Formal representation:

IS_DELINEATED_BY (0..*) vpSubnetworkTP

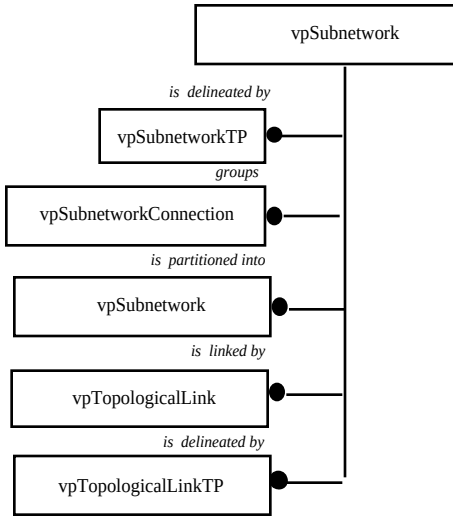
GROUPS (0..*) vpSubnetworkConnection

IS_PARTITIONED_INTO (0..*) vpSubnetwork

IS_LINKED_BY (0..*) vpTopologicalLink

IS_DELINEATED_BY (0..*) vpTopologicalLinkTP

Graphical representation:



Subnetwork Query Operations

Operation: query vpSubnetwork For delimiting vpSubnetworkTPs

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

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

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
connectingVpTopologicalLinkTPs : set of vpTopologicalLinkTPIds

ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each vpTopologicalLinkTP is connecting vpSubnetworks. This operation allows the requester (client) to query the connecting Managed Entities. It does not affect the relationship. It matches for all the connecting vpTopologicalLinkTPs. The reply is a set of vpTopologicalLinkTPIds.

Subnetwork: SubnetworkConnection management operations

Operation: set up vpSubnetworkConnection

INPUT PARAMETERS:
see Section 5.1.2, and
snTPa : choice of subnetworkTPId, or Descriptor
snTPz : choice of subnetworkTPId, or Descriptor
AdministrativeState (optional)

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

This operation allows the requester (user) to set-up a point-to-point connection between two non-connected subnetworkTPs of the addressed subnetwork. The subnetworkTPs 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 subnetworkTPs 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 subnetworkTPs and a subnetworkConnection associated to both of them in the case of the subnetwork creating the subnetworkTPs
- the creation of a subnetworkConnection associated to two existing subnetworkTPs in the case of subnetworkTPs already existing in the subnetwork.

Operation*:* release vpSubnetworkConnection

INPUT PARAMETERS:

none (see Section 5.1.2)
Choice of subnetworkTP : set of subnetworkTPIds

OUTPUT PARAMETERS:

vpSubnetworkConnectionID: vpSubnetworkConnectionId
released subnetworkTPs: set of subnetworkTPId

ERROR CONDITIONS:

protocol-specific addressing errors
incorrectVpsubnetworkTPs : subnetworkTPIds
subnetworkTPNotConnected : subnetworkTPId
incorrectsubnetworkConnection: subnetworkConnectionId

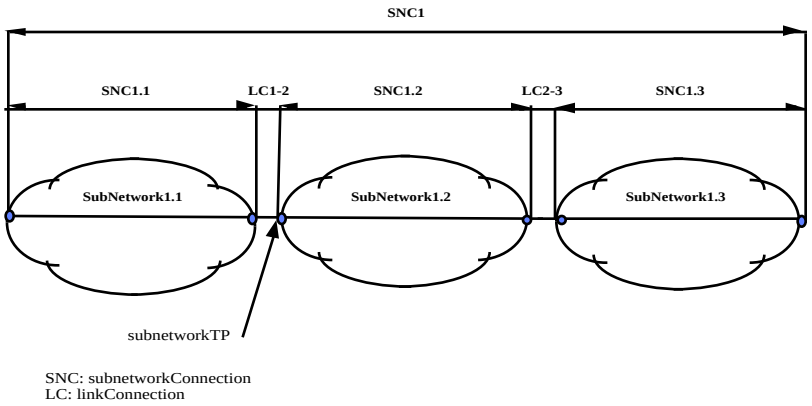
BEHAVIOUR

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

5.2.13 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."

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

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.

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 subnetworkTPs: A subnetwork connection has at least two subnetworkTPs.

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

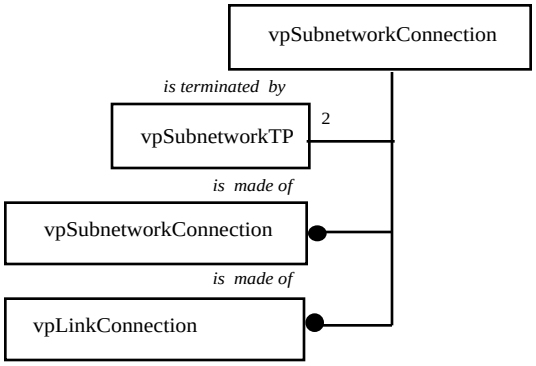
Semi-formal representation:

IS_TERMINATED_BY (2) subnetworkTPs

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

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

Graphical representation:



subnetworkConnection: subnetworkConnection query operations

Operation: query subnetworkConnection For terminating subnetworkTPs

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatingSubnetworkTPs: set of subnetworkTPIds
ERROR CONDITIONS:
protocol-specific

BEHAVIOUR:

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

Operation: query vpSubnetworkConnection For Component vpSubnetworkConnections

INPUT PARAMETERS:
none (see Section 5.1.2)

OUTPUT PARAMETERS:

componentSubnetworkConnections: set of subnetworkConnectionIds

ERROR CONDITIONS:

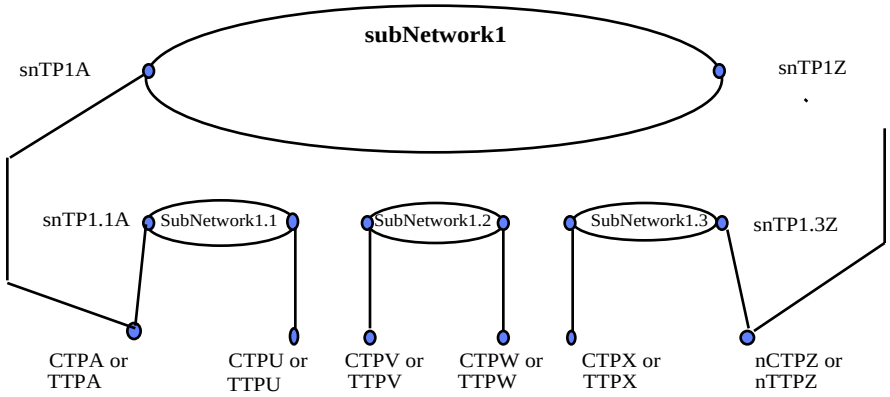
protocol-specific

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.

5.2.14 vpSubnetworkTP

The subnetworkTP Managed Entity terminates a subnetworkConnection or linkConnection on a subnetwork. Each subnetworkTP refers to a CTP or a TTP managed entity, and therefore allows a single representation of the network resources. As an example, a subnetwork may be partitioned into component subnetwork several times. Rather than creating a new CTP or TTP for each layer of decomposition, this approach to keep the CTP or TTP at the lowest layer, and to refer to them using subnetworkTPs. It represents the relationship between the endpoints of composite subnetworks supported by the same resource. This entity is explicitly created by a network management function.



Attributes:

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

Relationships :

With vpCTP/TTPs: Each subnetworkTP reflects the underlying CTP, or possibly underlying TTP.

With vpSubnetwork: each subnetworkTP delineates a subnetwork.

With vpLinkConnection: each subnetworkTP terminates a linkConnection.

With vpSubnetworkConnection: each subnetworkTP terminates a subnetworkConnection.

Semi-formal representation:

REFLECTS (0..1) vpCTP

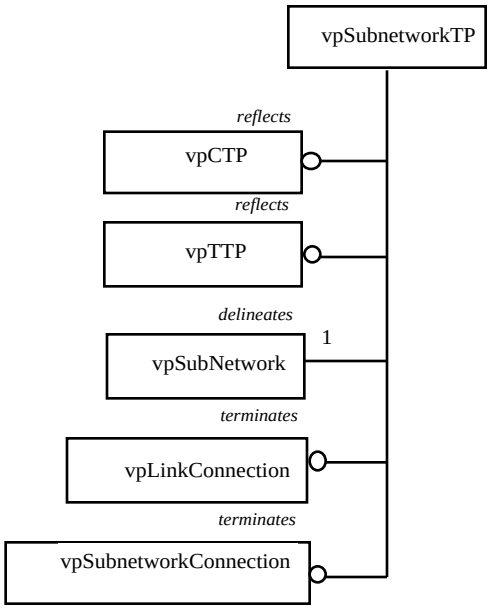
REFLECTS (0..1) vpTTP

DELINEATES (1) subnetwork

TERMINATES (0..1) vpLinkConnection

TERMINATES (0..1) vpSubnetworkConnection

Graphical representation:



subnetworkTP: subnetworkTP Query Operations

Operation: query subnetworkTPs For delimited subnetwork

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

The subnetwork plays a role of container for the subnetworkTPs. This operation allows the requester (client) to query the containing subnetwork for a given subnetworkTP. It does not affect the relationship. It matches for the delineated subnetwork. The reply is a subnetworkId.

Operation: query subnetworkTP For terminated vpLinkConnection

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

A linkConnection is terminated on two subnetworkTPs. This operation allows the requester (client) to query the terminated linkConnection. It does not affect the relationship. It matches for the terminated linkConnection. The reply is an linkConnectionId.

Operation: query subnetworkTP For terminated subnetworkConnection

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
terminatedSubnetworkConnection: subnetworkConnectionId
ERROR CONDITIONS:
protocol-specific

A point-to-point subnetworkConnection is terminated on two subnetworkTPs. This operation allows the requester (client) to query the terminated subnetwork Connection. It does not affect the relationship. It matches for the terminated subnetworkConnection. The reply is a subnetworkConnectionId.

Operation: query subnetworkTP For associated TTPs and CTPs

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
associatedTP:CTPId or TTPId
ERROR CONDITIONS:
protocol-specific

Each subnetworkTP is associated with a CTP or TTP. This operation allows the requester (client) to query the associated Managed Entity. It does not affect the relationship. It matches for the associated CTP or TTP. The reply is a CTPId or TTPId.

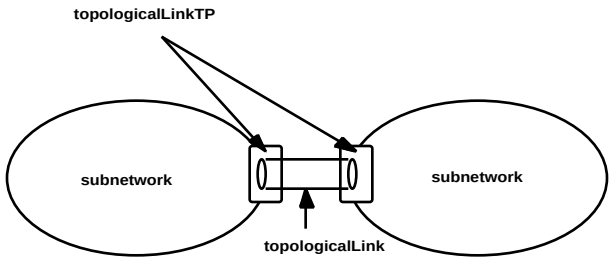
Operation*:* associate subnetworkTP with supporting CTP or TTP

INPUT PARAMETERS:
none (see Section 5.1.2)
associatedCTPorTTPId
OUTPUT PARAMETERS:
associatedTTPorCTP: associated CTPId or TTPId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

Each subnetworkTP is supported by either a CTP or a TTP, which provides transport for it. This operation allows the requester (client) to associate the subnetworkTP with a single desired CTP or TTP. This operation may take place at the creation of the subnetworkTP. The reply is the associated CTP or TTP Id.

5.2.15 vpTopologicalLink

A link is a topological component which describes a fixed relationship between two subnetworks. A topological link is a link between two subnetworks. 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 is 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.

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 topologicalLinkTP: A topologicalLink has two termination 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.

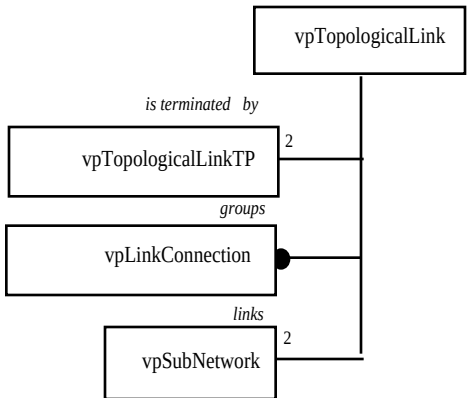
semi-formal representation:

GROUPS (0..*) vpLinkConnections

IS_TERMINATED_BY (2) vpTopologicalLinkTP

LINKS (2) vpSubnetwork

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 vpTopologicalLinkTPs

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

Each topologicalLink is terminated by two topologicalLinkTPs. This operation allows the requester (client) to query the terminating managed entities. It does not affect the relationship. It matches for the terminating topologicalLinkTPs. The reply is a set of topologicalLinkTPs 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
snTPa : choice of subnetworkTPId, or Descriptor
snTPz : choice of subnetworkTPId, or Descriptor

Descriptor:
interfaceId (Choice of subnetworkTP, or server TTPId, or topologicalLinkTP, or topologicalLink, or subnetworkId)
vpi (optional)
vci (optional)
trafficDescriptors (optional)
qos (optional)
OUTPUT PARAMETERS:
newVpLinkConnection : vpLinkConnectionId
snTPa: subnetworkTPId
snTPz: subnetworkTPId
ERROR CONDITIONS:
protocol-specific addressing errors, and
incorrectTerminationPoints : subnetworkTPId
reflectedTPDisabled : subnetworkTPId
reflectedTPLocked: subnetworkTPId

topologicalLinkLocked: topologicalLinkId
subnetworkTerminationPointInUse : subnetworkTPId
non-matchingDescriptors: set of subnetworkTPId
operationFails

BEHAVIOUR:

This operation allows the requester (user) to set-up a linkConnection between two non-connected subnetworkTPs 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 subnetworkTPs are incorrect (e.g. do not belong to the subnetwork), if the two subnetworkTPs 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 subnetworkTPs and an linkConnection associated to both of them in the case of the subnetwork creating the subnetworkTPs
- the creation of a linkConnection associated to two existing subnetworkTPs in the case of subnetworkTPs already existing in the subnetwork.

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

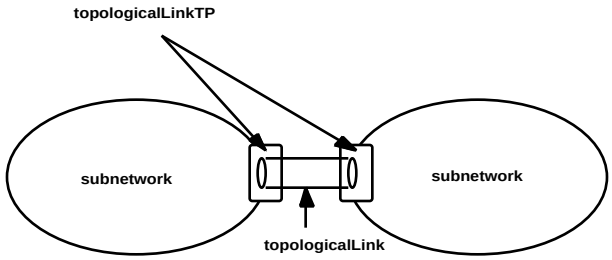
Operation: release vpLinkConnection

INPUT PARAMETERS:
none (see Section 5.1.2)
Choice of subnetworkTP : set of subnetworkTPIds
OUTPUT PARAMETERS:
vpSubnetworkConnectionID: vpSubnetworkConnectionId
released subnetworkTPs: set of subnetworkTPId
ERROR CONDITIONS:
protocol-specific addressing errors
incorrectSubnetworkTPs : subnetworkTPIds
subnetworkTPNotConnected : subnetworkTPId
incorrectLinkConnection: linkConnectionId
BEHAVIOUR

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

5.2.16 vpTopologicalLinkTP

This managed entity is used to represent the termination of a topologicalLink at the VP-layer. When a network is managed only at the VP-layer, the vpLink managed entity is used to store any link-level configuration data. This managed entity is created by the management system.



Attributes

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

Provisioned Bandwidth: see vpTopologicalLink managed entity definition

Available Bandwidth: see vpTopologicalLink managed entity definition

VPI Range: This read/write parameter identifies the maximum number of contiguous VPI bits, starting from the least significant bit, that may be used over the link.

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 vpTopologicalLinkTP managed entity.

With vpSubnetwork: One vpTopologicalLinkTP managed entity is associated with one and only one vpSubnetwork.

semi-formal representation:

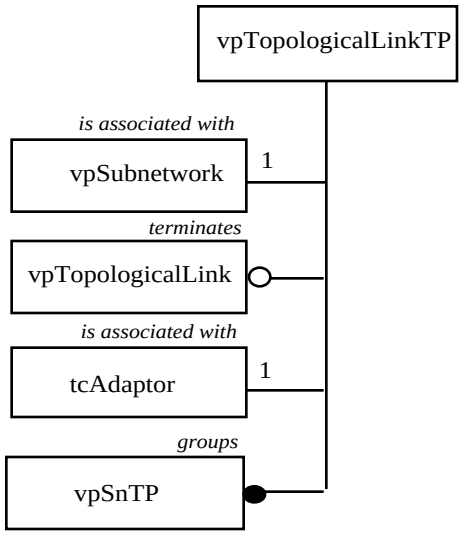
IS_ASSOCIATED_WITH (1) vpSubnetwork

IS_ASSOCIATED_WITH (1) tcAdaptorTTP

TERMINATES (0..1) vpTopologicalLink

GROUPS (0..*) vpSnTPs

graphical representation:



vpTopologicalLinkTP: vpTopologicalLinkTP Query Operations

Operation: query vpTopologicalLinkTP For Terminated vpTopologicalLink

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

Each topologicalLink is terminated by two topologicalLinkTPs. 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 vpTopologicalLinkTP For Delineated vpSubnetwork

INPUT PARAMETERS:
see Section 5.1.2
OUTPUT PARAMETERS:
delineatedSubnetworks: subnetworkIds
ERROR CONDITIONS:
protocol-specific

BEHAVIOUR

Each topologicalLinkTP 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 vpLinkTP For associated tcAdaptorTTP

INPUT PARAMETERS:
none (see Section 5.1.2)
OUTPUT PARAMETERS:
associatedVpTTP: tcAdaptorTTPId

ERROR CONDITIONS:
protocol-specific
BEHAVIOUR

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

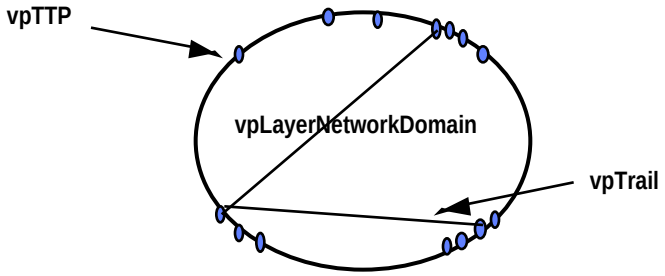
Operation: associate vpTopologicalLinkTP with supporting tcAdaptorTTP

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

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

5.2.17 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.

Operational State: This read-only attribute identifies whether or not the managed entity is capable of performing its normal functions (i.e., in-service or out-of-service).

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.

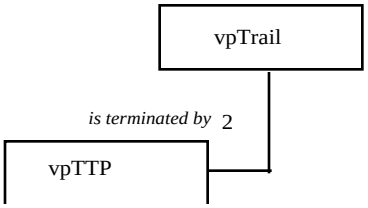
Relationships:

With vpTTPBidirectional: Each Vp trail is terminated by *vpTrailTerminationPointsBidirectional.*

Semi-formal representation:

IS_TERMINATED_BY (2) vpTTPBidirectional

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.

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.

For example a managing system may wish to view the VP layer independently of the VP layer. In that case all the VP resources are part of the VP layer network domain. The VPs in the VP layer network domain may be configured independently of the use of the VPs within that layer network domain.

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. 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.

5.2.18 vcCTP

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 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.

VCI Value: This read-only attribute identifies the VCI value associated with the VC connection being terminated.

Traffic Descriptors: This read/write attribute identifies values for the following traffic descriptors:

- Ingress and Egress Peak Cell Rate for CLP=0+1 Traffic
- Ingress and Egress Peak Cell Rate for CLP=0 Traffic (Optional)
- Ingress and Egress CDV Tolerance for CLP=0+1 Traffic
- Ingress and Egress CDV Tolerance for CLP=0 Traffic (Optional)
- Ingress and Egress Sustainable Cell Rate for CLP=0 and CLP=0+1 Traffic (Optional)
- Ingress and Egress Burst Tolerance for CLP=0 and CLP=0+1 Traffic (Optional)

QOS Class: This read/write attribute identifies the QOS class assigned to the VC connection.

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

Operational State: This read-only attribute identifies whether or not the VC connection termination is capable of performing its normal functions (in-service or out-of-service).

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 Traffic Descriptor changes. The notification shall identify the attribute that changed, its old value, and its new value.

State Change: This notification is used to report changes to the states 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 vcTTPBidirectional: Zero or one instance of the vcTTP managed entity may exist for each instance of a vcCTP managed entity.

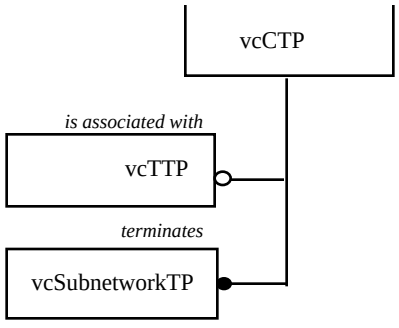
With subnetworkTerminationPoint: Zero or more of the vcCTP managed entity may exist for each instance of a subnetworkTP managed entity.

Semi-formal representation:

IS_ASSOCIATED_WITH (0..1) vcTTP

SUPPORTS (0..*) vcSubnetworkTP

Graphical representation:



vcConnectionTerminationPoint: vcConnectionTerminationPoint Query Operations

Operation: query vcCTP for associated vcTTP

INPUT PARAMETERS:

none (see Section 5.1.2)

OUTPUT PARAMETERS:
associatedVcTTP : vcTTPId
ERROR CONDITIONS:
protocol-specific
BEHAVIOUR:

Each vcCTP is associated with zero or one vcTTP. This operation allows the requester (client) to query the associated vcTTP for a given vcCTP. It does not affect the relationship. It matches for the associated vcTTP. The reply is a vcTTPId.

Operation: query vcCTP for associated subnetworkTP

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

Each vcCTP is associated with zero or one subnetworkTP. This operation allows the requester (client) to query the associated subnetworkTP for a given vcCTP. It does not affect the relationship. It matches for the associated subnetworkTP. The reply is a subnetworkTPId.

vcConnectionTerminationPoint: vcConnectionTerminationPoint Loopback Operation

Operation: loopback vcTrail at vcCTP

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.

5.2.19 vcTTP

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

VC TTP ID: This read-only attribute provides a unique name for the managed entity instance in the ATM subnetwork.

Operational State: This read-only attribute identifies whether or not the managed entity is capable of performing its normal functions (i.e., enabled or disabled).

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.

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

Relationships

With vcCTPs: Zero or one instance of the VC Trail Termination managed entity may exist for each instance of a vcConnectionTerminationPoint managed entity.

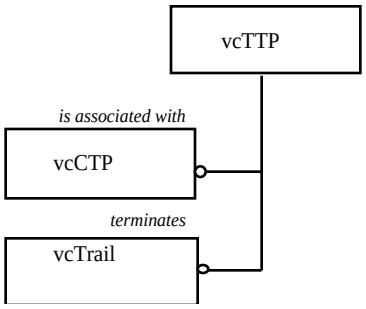
With vcTrail: A vcTrail is terminated by two vcTrailTerminationPoints.

Semi-formal representation:

IS_ASSOCIATED_WITH (0..1) vcCTP

TERMINATES (0..1) vcTrail

Graphical representation:



vcTTP: vcTTP Query Operations

Operation: query vcTTP for associated vcCTP

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

Each vcTTP is associated with zero or one vcCTP. This operation allows the requester (client) to query the associated vcCTP. It does not affect the relationship. It matches for the associated vcCTP. The reply is a vcCTPId.

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 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.

vcTTP: vcTTP 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.

5.2.20 vpCTP

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 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 VPI value associated with the VP connection being terminated.

Traffic Descriptors: This read/write attribute identifies values for the following traffic descriptors:

- Ingress and Egress Peak Cell Rate for CLP=0+1 Traffic
- Ingress and Egress Peak Cell Rate for CLP=0 Traffic (Optional)
- Ingress and Egress CDV Tolerance for CLP=0+1 Traffic
- Ingress and Egress CDV Tolerance for CLP=0 Traffic (Optional)
- Ingress and Egress Sustainable Cell Rate for CLP=0 and CLP=0+1 Traffic (Optional)
- Ingress and Egress Burst Tolerance for CLP=0 and CLP=0+1 Traffic (Optional)

QOS Class: This read/write attribute identifies the QOS class assigned to the VP connection.

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

Operational State: This read-only attribute identifies whether or not the VP connection termination is capable of performing its normal functions (in-service or out-of-service).

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 Traffic Descriptors of this managed entity . The notification shall identify the attribute that changed, its old value, and its new value.

State Change: This notification is used to report changes of the states 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 vpTrailTerminationPoint: Zero or one instance of the vpTTP managed entity may exist for each instance of a vcCTP managed entity.

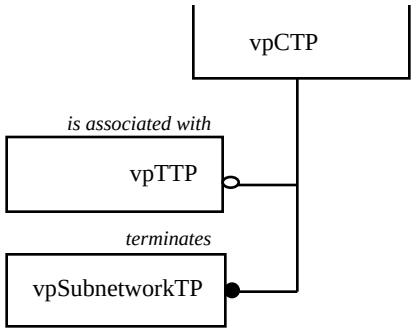
With subnetworkTerminationPoint: Zero or more instance of the vpCTP managed entity may exist for each instance of a subnetworkTP managed entity.

Semi-formal representation:

IS_ASSOCIATED_WITH (0..1) vpTTP

SUPPORTS (0..*) vpSubnetworkTP

Graphical representation:



vpConnectionTerminationPoint: vpConnectionTerminationPoint Query Operations

Operation: query vpCTP for associated vpTTP

INPUT PARAMETERS:

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

Each vpCTP is associated with zero or one vpTTP. This operation allows the requester (client) to query the associated vpTTP for a given vpCTP. It does not affect the relationship. It matches for the associated vpTTP. The reply is a vpTTPId.

Operation: query vpCTP for associated subnetworkTP

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

Each vpCTP is associated with zero or one subnetworkTP. This operation allows the requester (client) to query the associated subnetworkTP for a given vpCTP. It does not affect the relationship. It matches for the associated subnetworkTP. The reply is a subnetworkTPId.

vpConnectionTerminationPoint: vpConnectionTerminationPoint Loopback Operation

Operation: loopback vpTrail at vpCTP

INPUT PARAMETERS:
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 vpCTP 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.

5.2.21 vpTTP

This managed entity represents the point in the ATM subnetwork where the vpTrail and associated overhead (F4 OAM cells) are terminated/originated. Management systems shall configure/remove vpTrail Terminations in the ATM subnetwork by creating/deleting instances of this managed entity.

Attributes

vpTTP ID: This read-only attribute provides a unique name for the managed entity instance in the ATM subnetwork.

Operational State: This read-only attribute identifies whether or not the managed entity is capable of performing its normal functions (i.e., in-service or out-of-service).

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.

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

Relationships

With vpCTP: Zero or one instance of the vpTTP managed entity may exist for each instance of a vpConnectionTerminationPoint managed entity.

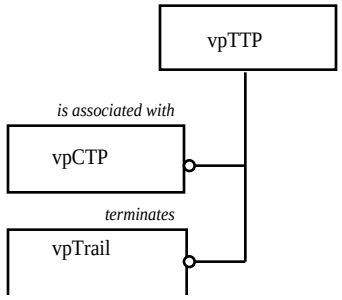
With vpTrail: Two vpTTPs terminate a vpTrail.

Semi-formal representation:

IS_ASSOCIATED_WITH (0..1) vpCTP

TERMINATES (0..1) vpTrail

Graphical representation:



vpTTPBidirectional Query Operations vpTTPBidirectional:

Operation: query vpTTPbidirectional for associated vpCTP

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

Each vpTTP is associated with zero or one vpCTP. This operation allows the requester (client) to query the associated vpCTP for a given vpTTP. It does not affect the relationship. It matches for the associated vpCTP. The reply is a vpCTPId.

Operation: query vpTTPbidirectional For terminated vpTrail

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

An vpTrail is terminated on two vc- or vp-TrailTerminationPoints. This operation allows the requester (client) to query the terminated vpTrail. It does not affect the relationship. It matches for the associated vpTrail. The reply is an vpTrailId.

vpTTPBidirectional: vpTTPBidirectional Loopback Operation

Operation: loopback vpTrail at vpTTPBidirectional

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 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 vpTrail.

6. 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.
- 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

[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.

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.

6.1 Subnetwork Provisioning Outline Ensembles

6.1.1 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.

6.2 Fault Management And Alarm Surveillance Outline Ensembles

6.2.1 Equipment Fault Outline Ensemble (Network-view only)

6.2.1.1 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.

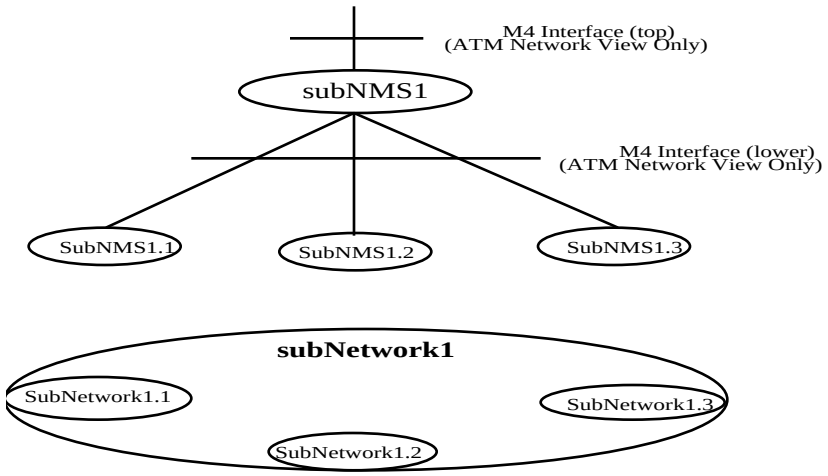


Figure 6.2.2.1-a: Network Management Architecture

6.2.1.2 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.

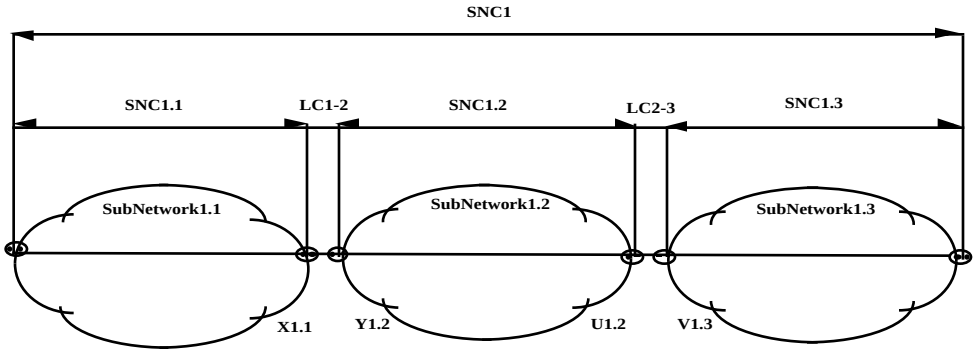


Figure 6.2.2.2-a: Network Transport Architecture

6.2.1.3 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).

6.2.1.4 Managed Entities

Topological Managed Entities involved in this scenario:

- vpSubnetwork or vcSubnetwork
- vpSubnetworkTP or vcSubnetworkTP
- vpSubnetworkConnection or vcSubnetworkConnection
- vpLinkConnection or vcLinkConnection

Management Entities involved in this scenario:

- Network

6.2.1.5 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. vpSubnetworkTP Y1.2, in the incoming direction from subnetwork1.1)

Step 3: SubNMS1.2 sends an alarm to SubNMS1 against vp/vcSubnetworkTP 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.

6.2.2 Equipment Fault Outline Ensemble (network-view + NE-view)

6.2.2.1 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.

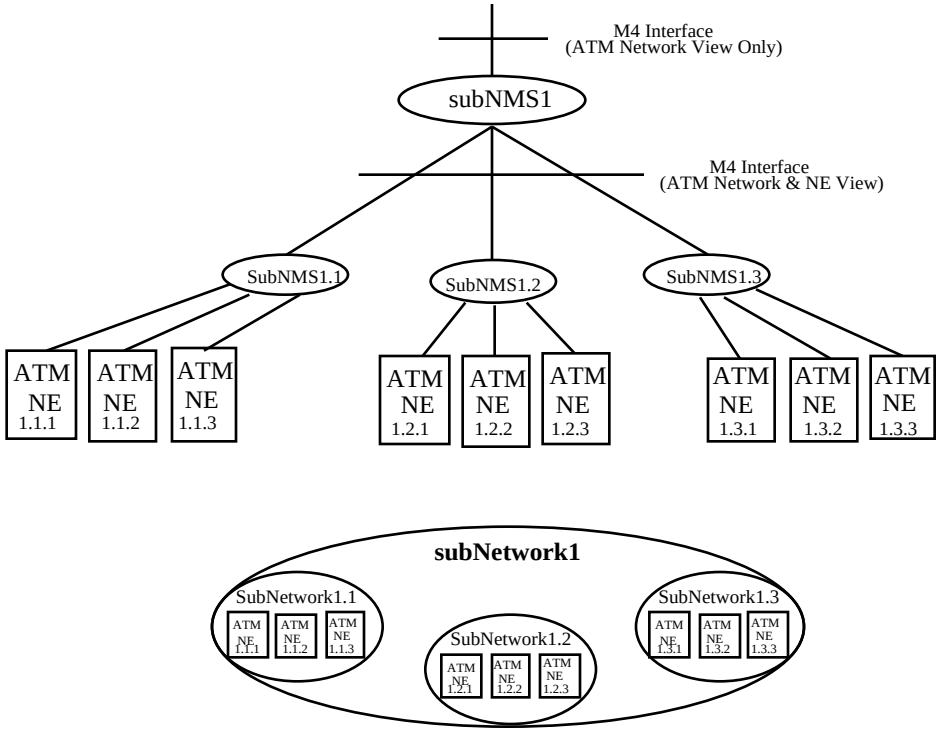


Figure 6.3.2.1-a: Network Management Architecture

6.2.2.2 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.

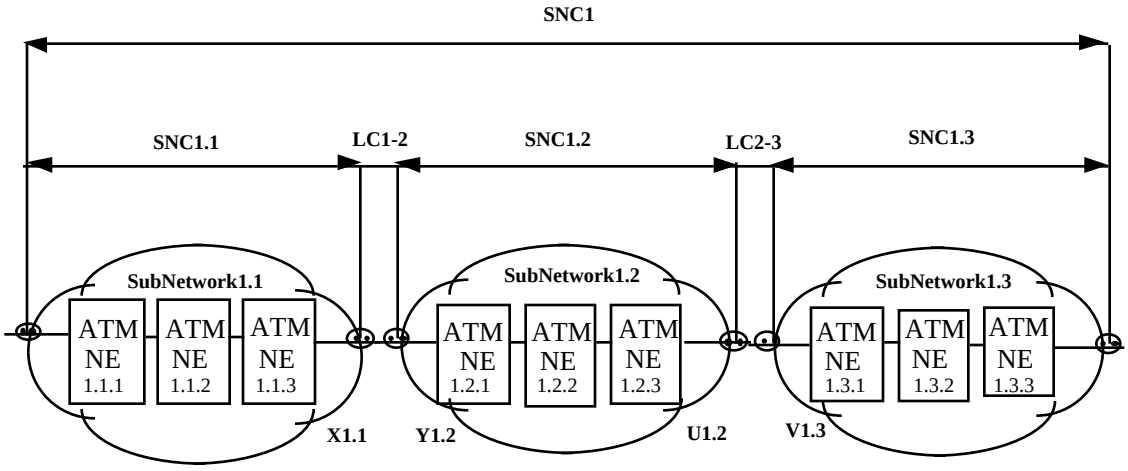


Figure 6.2.3.2-a: Network Transport Architecture

6.2.2.3 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).

6.2.2.4 Managed Entities

Topological Managed Entities involved in this scenario:

- vpSubnetwork or vcSubnetwork
- vpSubnetworkTP or vcSubnetworkTP
- vpSubnetworkConnection or vcSubnetworkConnection
- vpLinkConnection or vcLinkConnection
- atmNE (from NE view)

Management Entities involved in this scenario:

- Network

6.2.2.5 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. vpSubnetworkTP 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/vcSubnetworkTP 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.

6.2.3 Transport Failure Alarming Outline Ensemble (network-view only)

6.2.3.1 Network Management Context

Same as for Section 6.2.2.1

6.2.3.2 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).

6.2.3.3 Managed Entities

Topological Managed Entities involved in this scenario:

- vpSubnetwork or vcSubnetwork
- vpSubnetworkTP or vcSubnetworkTP
- vpSubnetworkConnection or vcSubnetworkConnection
- vpLinkConnection or vcLinkConnection

Management Entities involved in this scenario:

- Network

6.2.3.4 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/vcSubnetworkTP X.1.1.

Step 3: SubNMS1.1 sends an alarm against vp/vcSubnetworkTP X1.1 to SubNMS1.

Additional steps for subNMS1.2: steps 2 to 3 are repeated asynchronously for subNMS1.2 and vp/vcSubnetworkTP 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.

6.3 Subnetwork Connection Management Outline Ensembles

6.3.1 Simple VP or VC Single-layer Subnetwork Connection Set-Up Outline Ensemble (network-view only)

6.3.1.1 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].

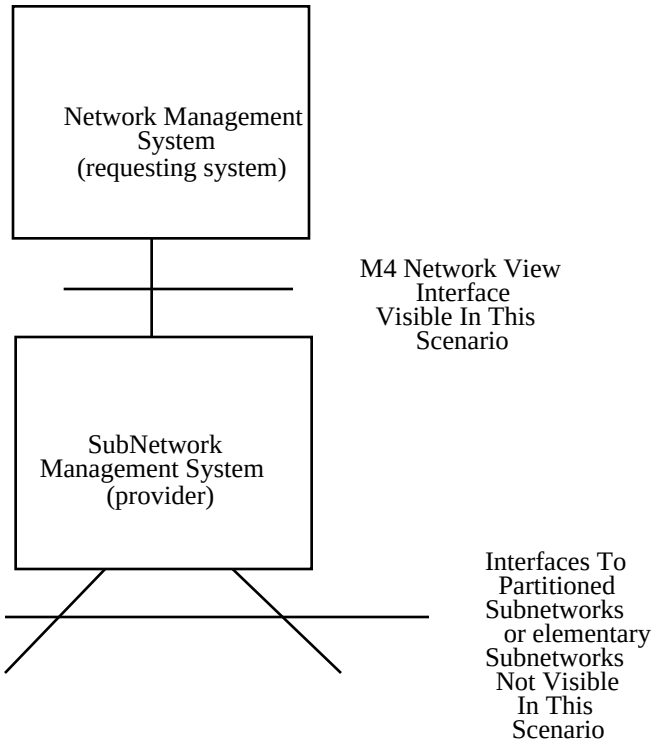


Figure 6.3.1.1-a: Network Management Architecture

6.3.1.2 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.

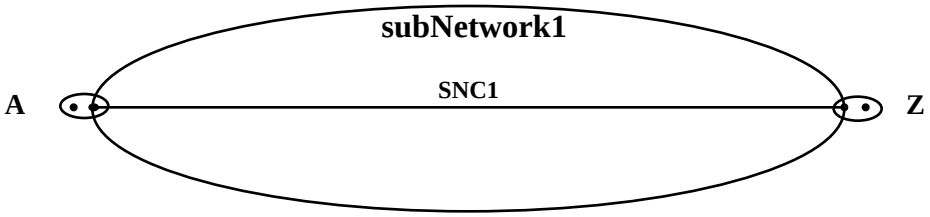


Figure 6.3.1.2-a: Network Transport Architecture

6.3.1.3 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).

6.3.1.4 Managed Entities

Topological Managed Entities involved in this scenario:

- vpSubnetwork or vcSubnetwork
- vpSubnetworkTP or vcSubnetworkTP
- vpSubnetworkConnection or vcSubnetworkConnection

6.3.1.5 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.

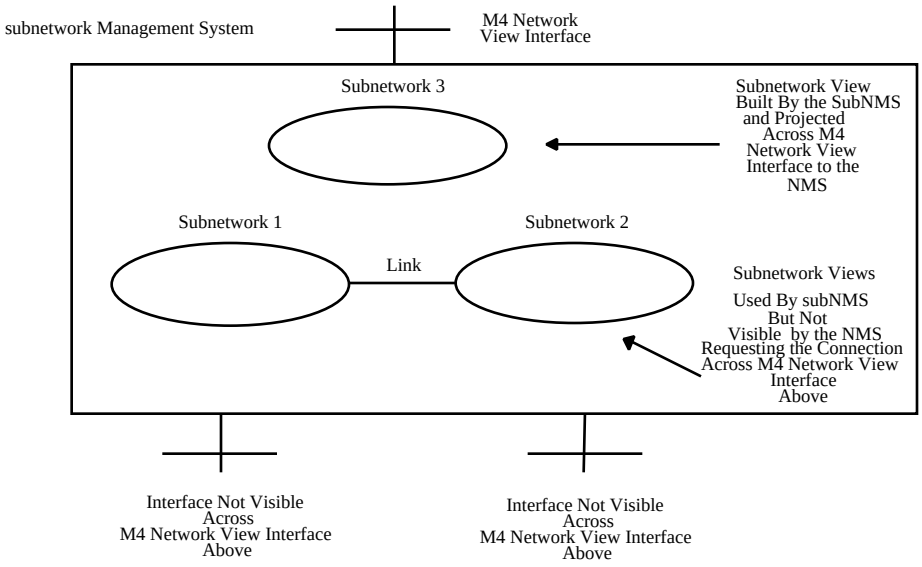


Figure 6.3.1.5-a: Information Applicable To This Scenario

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/vcSubnetworkTPs 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.

6.3.2 Multi-Level VP or VC Single-layer Subnetwork Connection Set-Up Outline Ensemble (network-view only)

6.3.2.1 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.

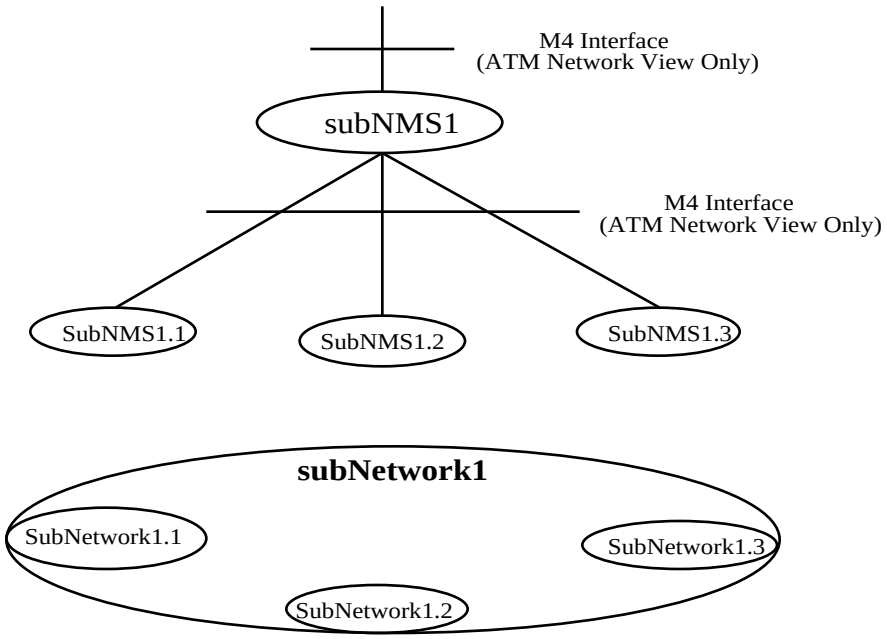


Figure 6.3.1.1-a: Network Management Architecture

6.3.2.2 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.

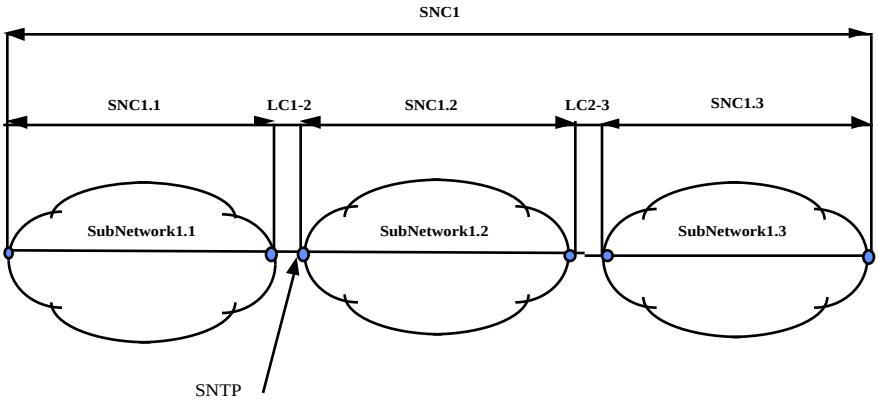


Figure 6.3.2.2-a: Network Transport Architecture

6.3.2.3 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).

6.3.2.4 Managed Entities

Topological Managed Entities involved in this scenario:

- vpSubnetwork or vcSubnetwork
- vpSubnetworkTP or vcSubnetworkTP
- vpSubnetworkConnection or vcSubnetworkConnection
- vpLinkConnection or vcLinkConnection

Management Entities involved in this scenario:

- Network

6.3.2.5 Scenario

Assumption: To each subnetwork Management System corresponds an administrative domain which contains an already-existing subnetwork. However, the subnetwork Termination points and the corresponding Connection Termination Points used to create the Subnetwork Connection are not pre-existing**.** The following example is for VP subnetwork connections.

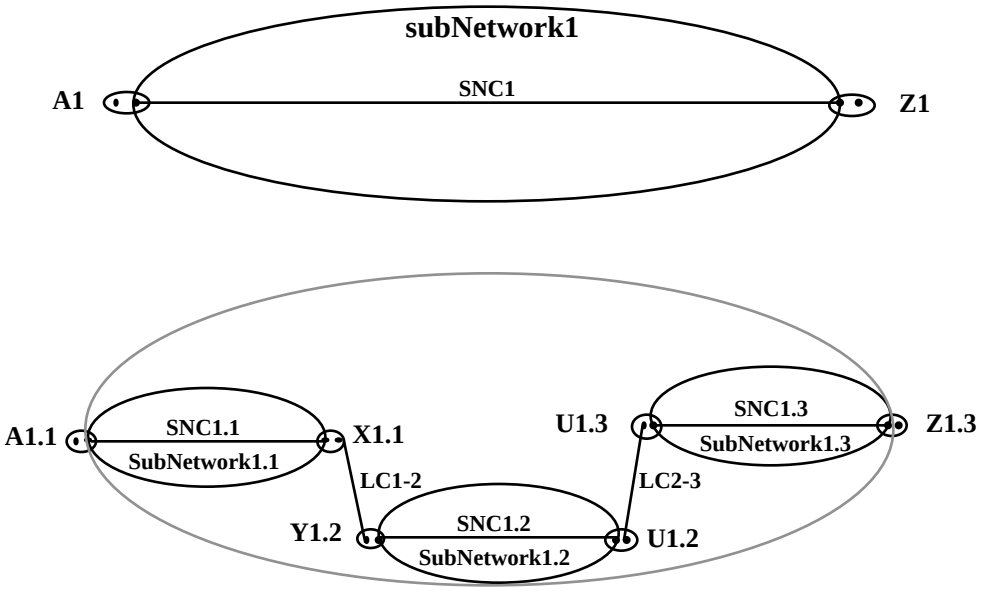


Figure 6.3.1.5-a: Transport Managed Entities

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 subnetwork Termination Point, 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 subnetwork Termination Points (A1.1. and X1.1), and the corresponding Connection Termination Points, which are qualified by their traffic descriptors and QOS.

Step 4: subNMS1 requests subNMS1.1 to set-up a VP subnetwork connection between subnetwork termination points A1.1 and X1.1 within subNMS1.1. The termination points A1.1 and Z1.1 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 subnetwork Termination Points (A1.1. and X1.1.) and the corresponding vpCTP.

Step 6: subNMS1.1 creates an vpSubnetworkConnection terminated by A1.1 and X.1.1.

Step 7: subNMS1.1 acknowledges the creation of A1.1, X.1.1 and of the vpSubnetworkConnection SNC1.1 terminated by A1.1 and X.1.1 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, an 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).

6.3.3 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.

6.3.3.1 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.

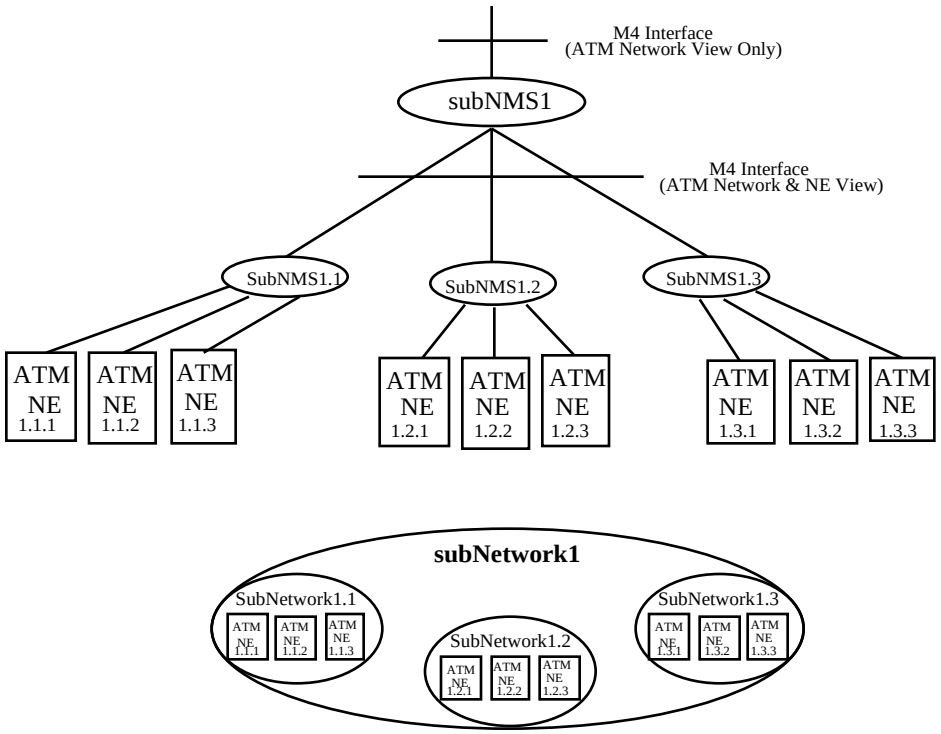


Figure 6.3.3.1-a Network Management Architecture

6.3.3.2 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 of :

1. the need to address specific market functionality with individual equipment,
2. the evolutionary introduction of new more powerful network elements,
3. the specific functional cost-effectiveness of particular network element vendors,
4. the merging of companies with existing ATM networks from different vendors.

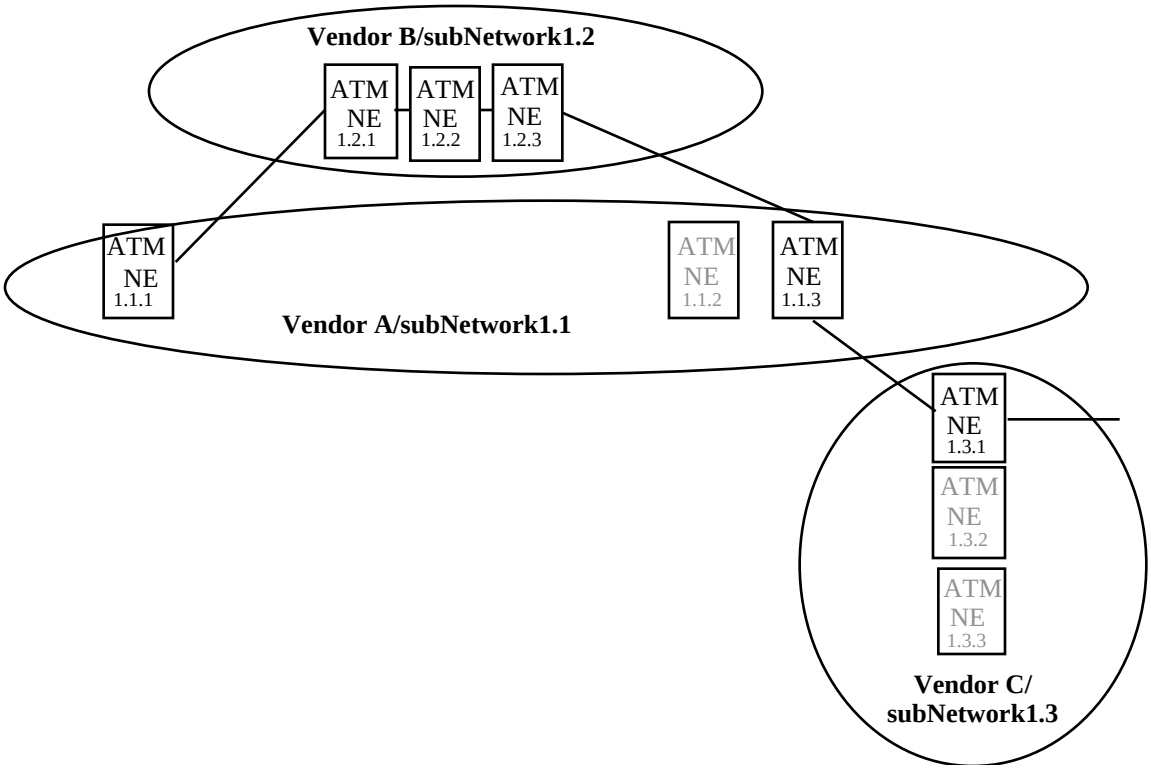


Figure 6.3.3.2-a Network Transport Architecture

6.3.3.3 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.

6.3.3.4 Managed Entities

- atmNE
- vpSubnetworkConnection or vcSubnetworkConnection
- VPCrossConnect or VCCrossConnect
- vpLinkConnection or vcLinkConnection
- vcTrail or vpTrail

6.3.3.5 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 logical termination points (subnetworkTP) 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.

6.3.4 VP or VC Single-layer Piece-By-Piece Connection Set-Up Outline Ensemble (NE-view)

6.3.4.1 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.

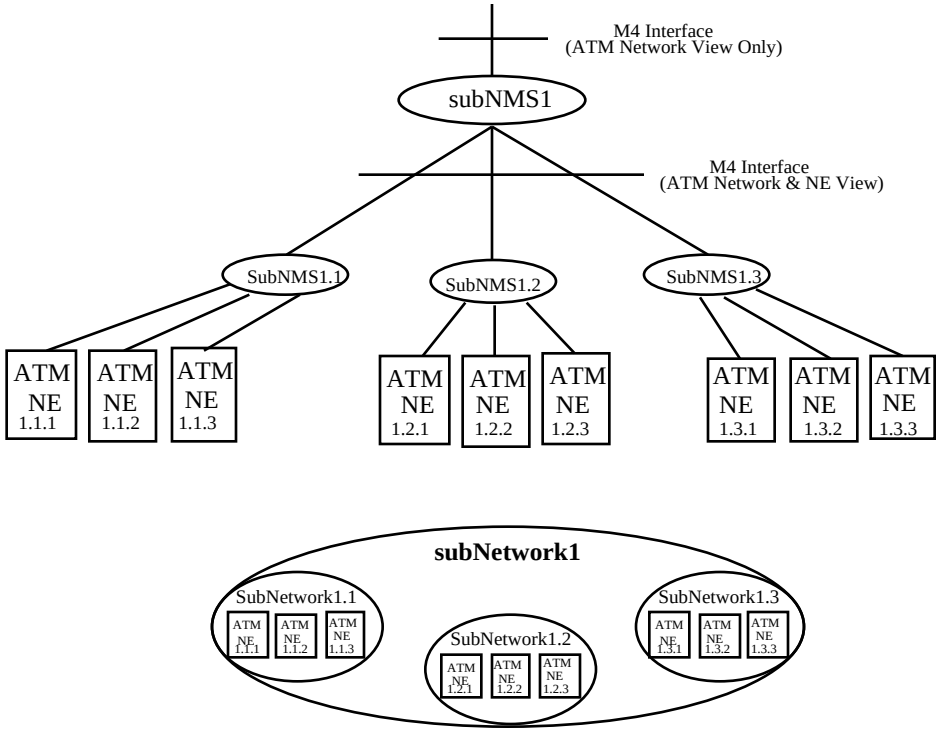


Figure 6.3.4.1-a:Network Management Architecture

6.3.4.2 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.

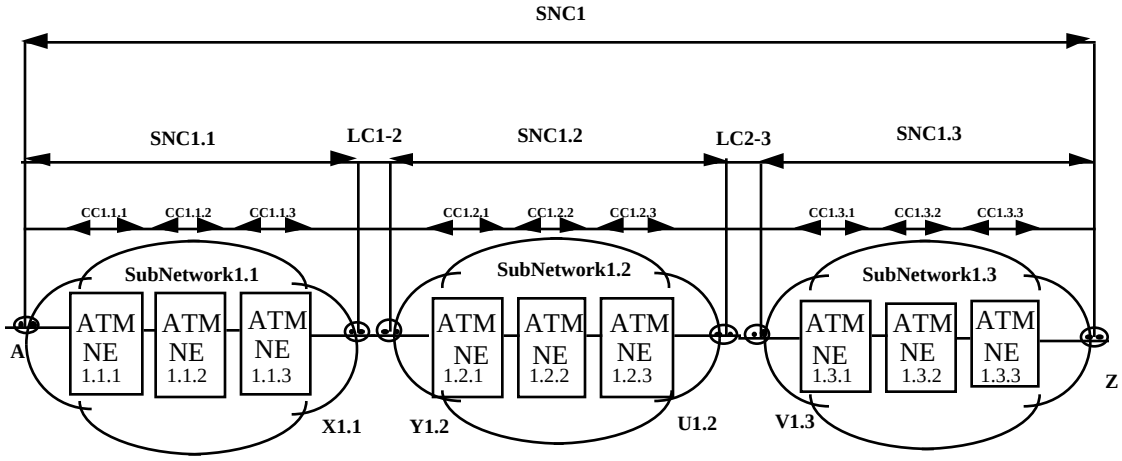


Figure 6.3.4.2-a: Transport Architecture

6.3.4.3 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

6.3.4.4 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 (vcCTPBidirectional)
- VCL Termination Point (vcTTPBidirectional)
- VPC Termination Point (vpCTPBidirectional)
- VPL Termination Point (vpTTPBidirectional)

6.3.4.5 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.

6.3.5 VP or VC Subnetwork Connection Modification Outline Ensemble

6.3.5.1 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.

6.3.5.2 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.

6.3.5.3 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.

6.3.5.4 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 (vcCTPBidirectional)
- VCL Termination Point (vcTTPBidirectional)

- VPC Termination Point (vpCTPBidirectional)
- VPL Termination Point (vpTTPBidirectional)

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

6.3.5.5 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.

6.4 Interlayer Trail Connection Management Outline Ensemble

6.4.1 Interlayer Trail Set-Up In a Single Management Domain Outline Ensemble

6.4.1.1 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) [3]. 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.

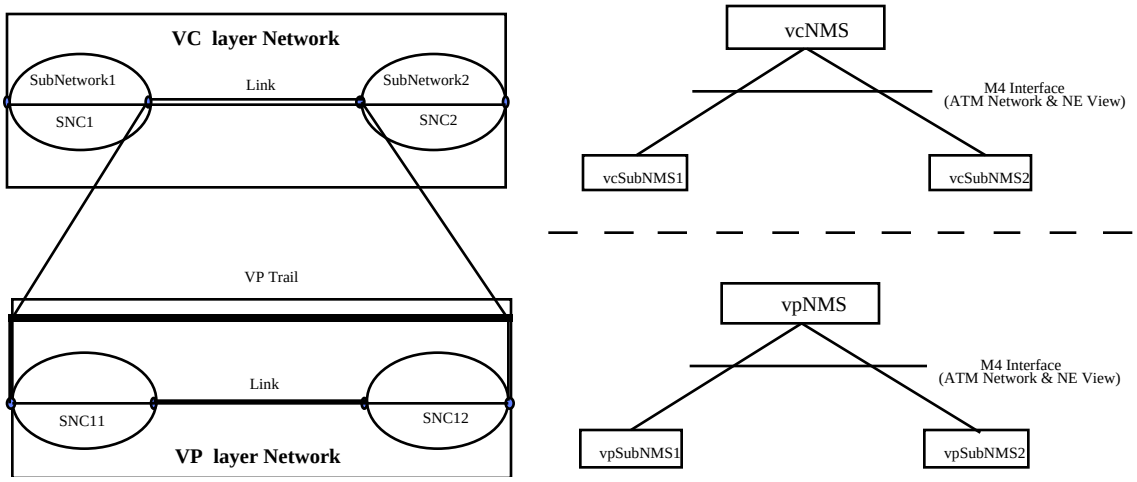


Figure 6.5.1.1: Interlayer Trail Set-Up in a single Management Domain

6.4.1.2 Transport Architecture

See above Figure (left side)

6.4.1.3 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

6.4.1.4 Managed Entities

Managed Entities:

The following Managed Entities are used in this ensemble:

- vcTopologicalLink
- vcLinkConnection
- vpSubnetworkTP
- vpSubnetworkConnection
- vpTrail
- vpTopologicalLinkTP
- vpLinkConnection

6.4.1.5 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.

6.4.2 Interlayer Trail Set-Up Across Multiple Management Domain Outline Ensemble

6.4.2.1 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.

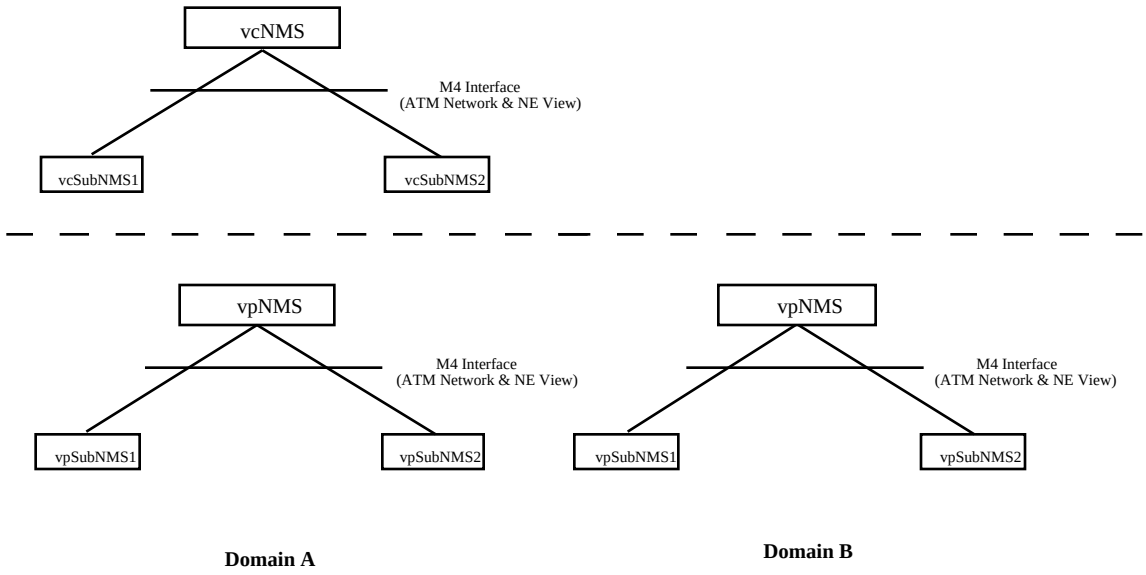


Figure 6.5.2.1: Interlayer Trail Set-Up across Management Domain (NMS view)

6.4.2.2 Transport Architecture

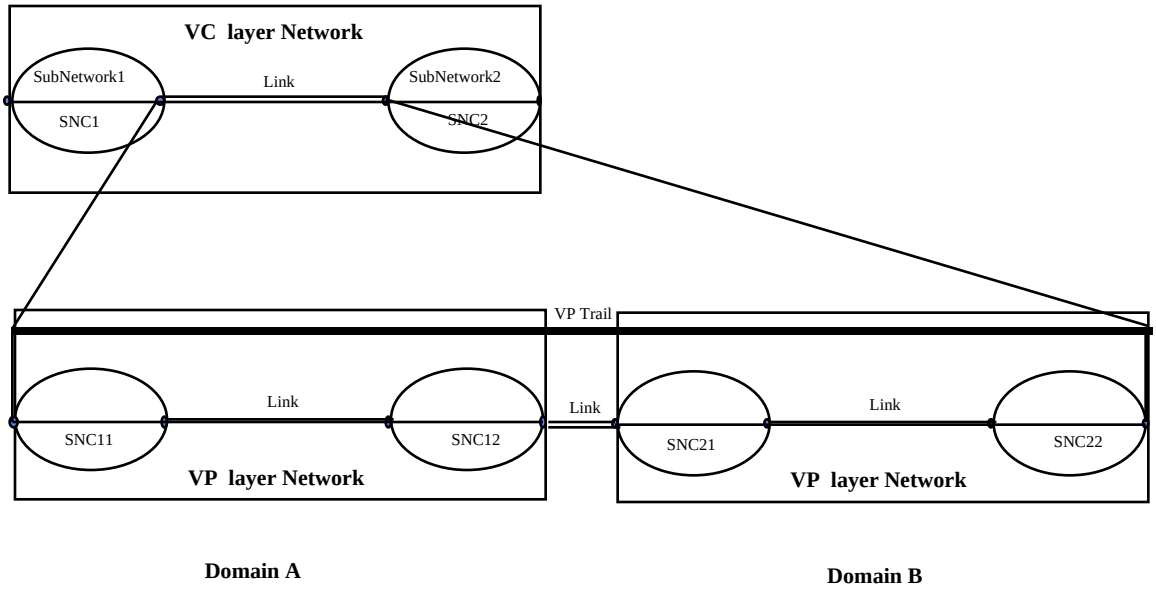


Figure 6.5.2.1: Interlayer Trail Set-Up across Management Domain (Transport Entities)

6.4.2.3 Functional Requirements

Same as in Section 6.5.1.3

6.4.2.4 Managed Entities

Managed Entities:

The following Managed Entities are used in this ensemble:

- vcTopologicalLink
- vcLinkConnection
- vpSubnetworkTP
- vpSubnetworkConnection
- vpTrail
- vpTopologicalLinkTP
- vpLinkConnection

6.4.2.5 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.

7. 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

8. 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, S.Farkouh, "M4 Interface Requirements and Logical MIB; ATM NE-view," November 1994.

[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] ITU-T Recommendation I.311, "B-ISDN general network aspects," 1993.

[9] 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.

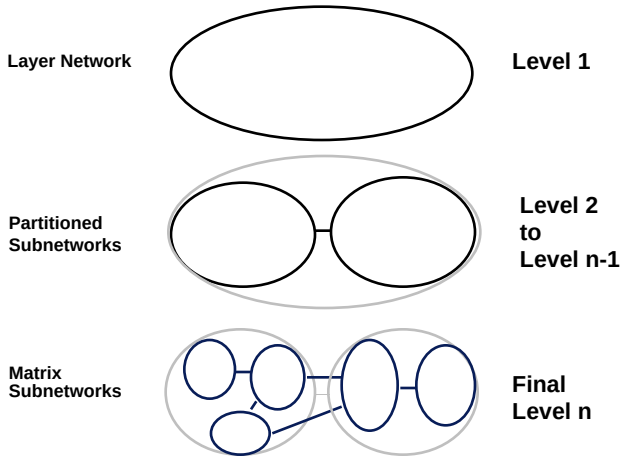
1.1 Layer Network

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.

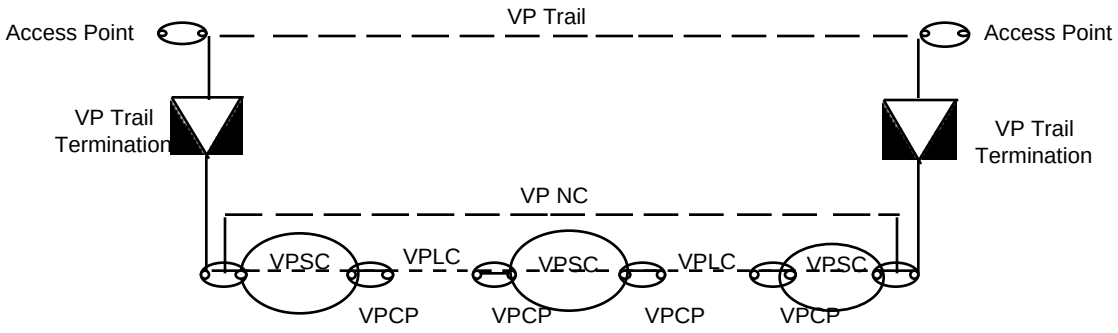


Figure 2 ATM VP Layer Network

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.

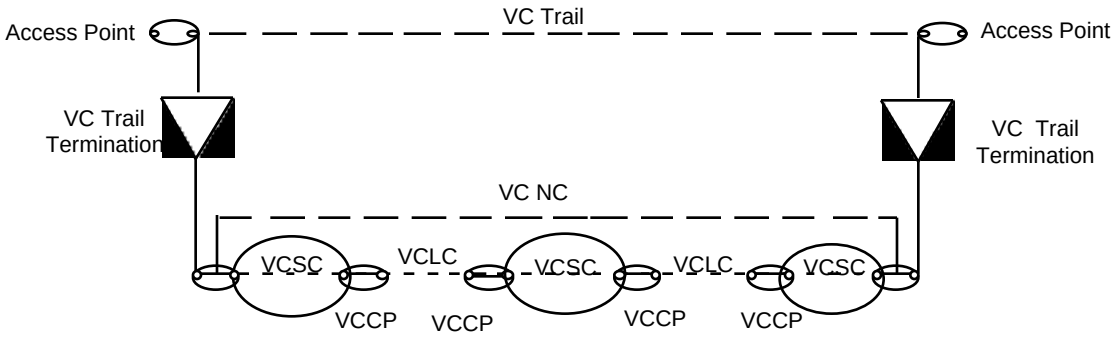


Figure 3 ATM VC Layer Network

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.

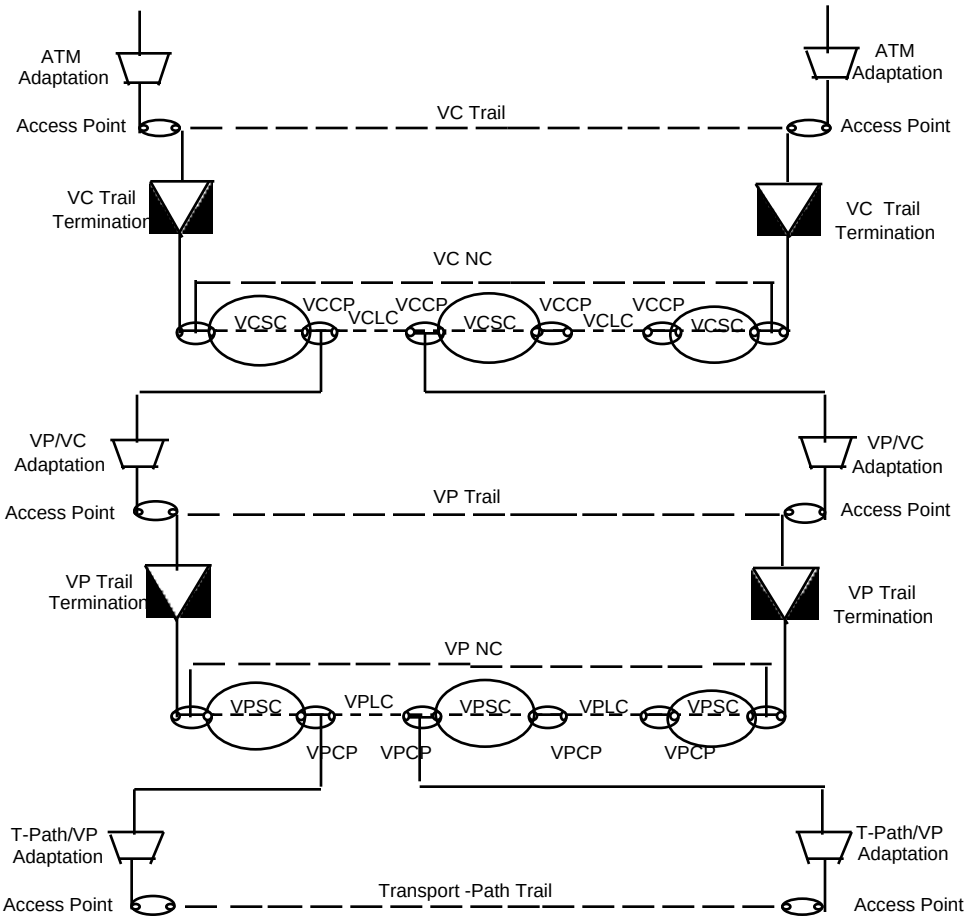


Figure 4 Example of Layer Networks in ATM Context

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.

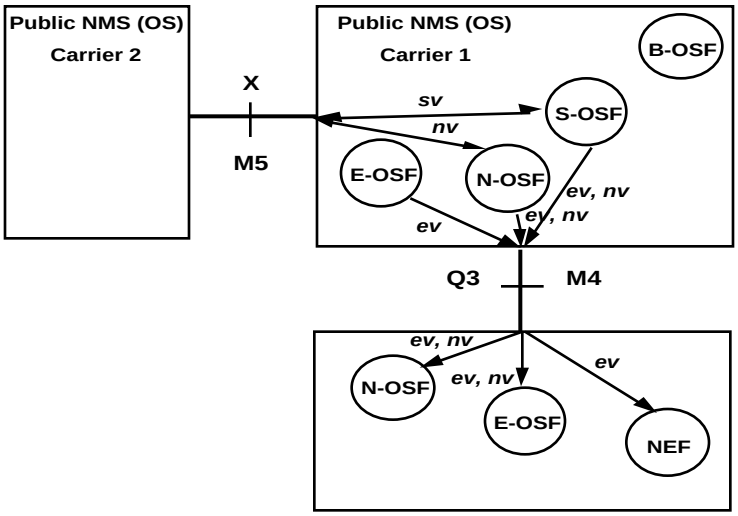
ITU-T Recommendation M.3100, the *Generic Information Model*, defines three management information views:

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 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*.

[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.



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.