The ATM Forum

Technical Committee

Network Management

M4 Network View
CMIP MIB
Specification
Version 1.0

af-nm-0073-000

January, 1997

M4 Network View CMIP MIB Specification

ATM_Forum af-nm-0073-000

The information in this publication is believed to be accurate as of its publication date. Such information is subject to change without notice and The ATM Forum is not responsible for any errors. The ATM Forum does not assume any responsibility to update or correct any information in this publication.

Notwithstanding anything to the contrary, neither The ATM Forum nor the publisher make representation or warranty, expressed or implied, concerning the completeness, accuracy, or applicability of any information contained in this publication. No liability of any kind shall be assumed by The ATM Forum or the publisher as a result of reliance upon any information contained in this publication.

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

o Any express or implied license or right to or under any ATM Forum member company's patent, copyright, trademark or trade secret rights which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor

o Any warranty or representation that any ATM Forum member companies will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor

o Any form of relationship between any ATM Forum member companies and the recipient or user of this document.

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

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

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

The ATM Forum
Worldwide Headquarters
2570 West El Camino Real, Ste 304
Mountain View, CA 94040-1313
Tel: +1-415-949-6700
Fax: +1-415-949-6705

SOURCE:

Andrew Mayer

Mehmet Toy

Aditya Sehgal

Kathleen Jarosinski

Patrice Lamy

Atahan Tuzel

Andrew Malis

Richard Wagner

Table of Contents

Introduction

The management functions that provide a basis for the ATM Network Level CMIP MIB presented here are based on those requirements found in the ATM Forum's network view logical MIB definition in AF-NM-0058.000 (June 1996) as well as its revision found in ATM Forum/96-1143R2.

Containment Diagram

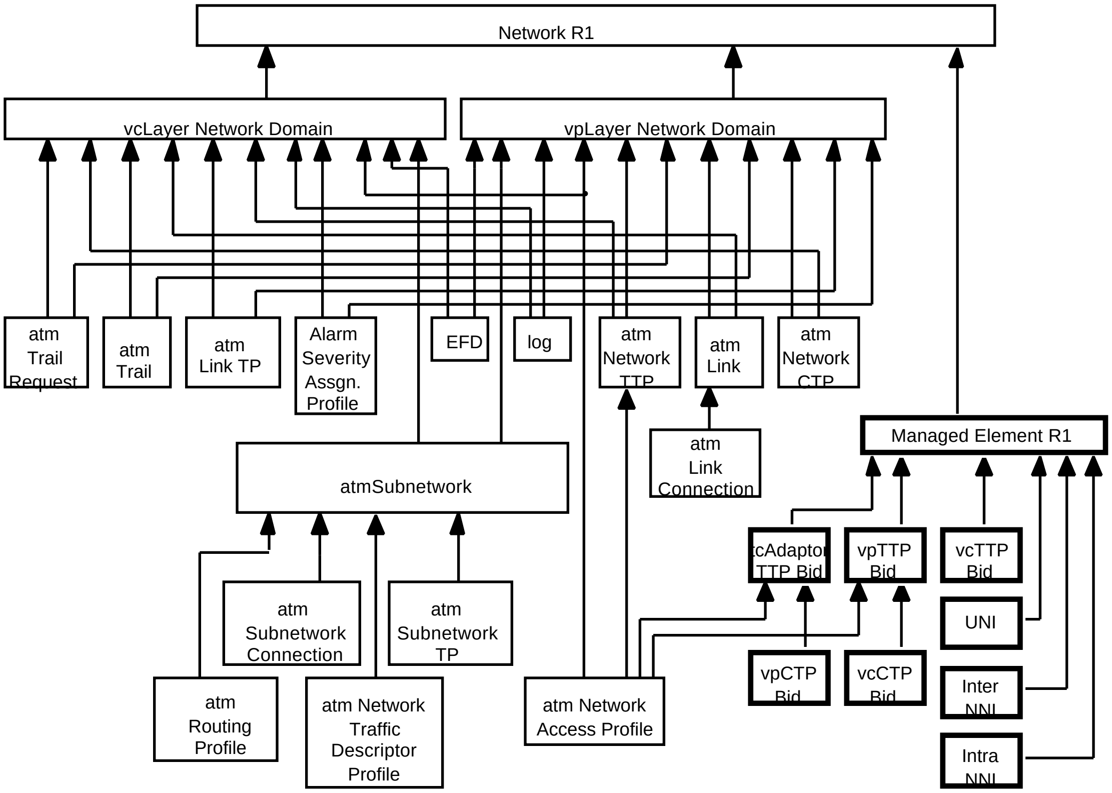

Figure 1 Containment Diagram

The objects in bold boxes on the Naming Diagram above indicate objects that are defined in the M4 NE view. These objects may be included in an implementation where they are referenced from the defined M4 Network View objects described in this document, and where both the M4 NE View and the M4 Network View are supported. Implementation of both the M4 NE View and M4 Network View together represents a specific design choice. Also, implementations that provide a "stand alone" network view (no references to M4 NE view objects) may be defined using the objects described in this document. In this case the objects in bold boxes would not be referenced by the M4 Network View objects.

Inheritance Diagram

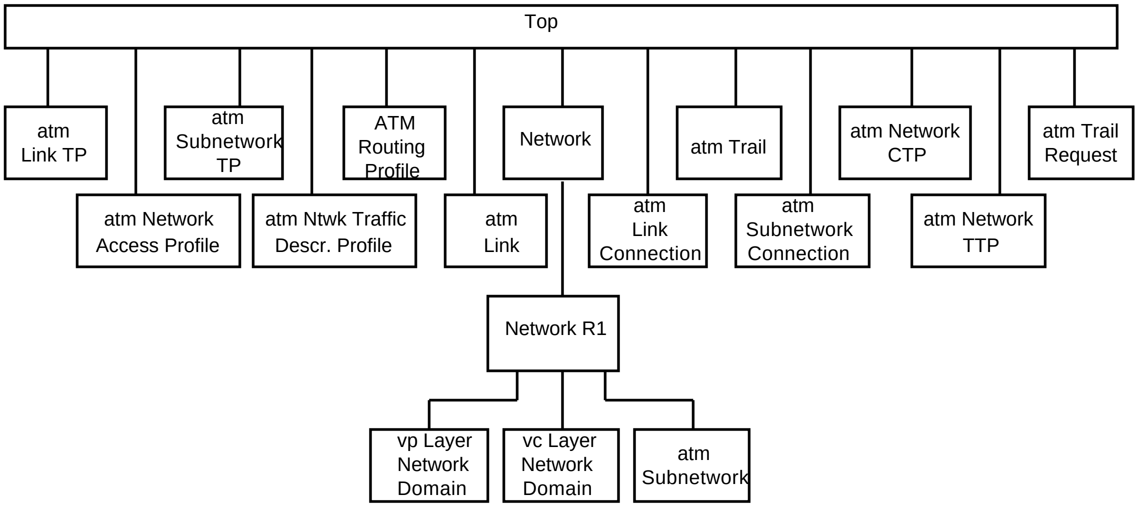

Figure 2 Inheritance Diagram

Objects

```
atmLink MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Rec. X.721 | ISO/IEC 10165-2":top;
    CHARACTERIZED BY
            “ITU-T M.3100:1995”:createDeleteNotificationsPackage,
            “ITU-T M.3100:1995”:attributeValueChangeNotificationPackage,
            “ITU-T M.3100:1995”:stateChangeNotificationPackage,
            “ITU-T Rec. X.721 | ISO/IEC 10165-2”:availabilityStatusPackage,
            “ITU-T M.3100:1995”:characteristicInformationPackage,
            atmLinkPackage  PACKAGE
                    BEHAVIOUR atmLinkBeh;
                    ATTRIBUTES
                            atmLinkId
                                GET
                                SET-BY-CREATE,
                            a-LinkTP
                                GET
                                SET-BY-CREATE,
                            z-LinkTP
                                GET
                                SET-BY-CREATE,
                            atmNetworkAccessProfilePointer
                                GET-REPLACE,
                            restorationMode
                                DEFAULT VALUE AtmNwMIBMod.restorationDefault
                                GET-REPLACE,
                            "ITU-T Rec. X.721 | ISO/IEC 10165-2":
                                        administrativeState
                                GET-REPLACE;;;
    CONDITIONAL PACKAGES
            linkConnectionManagementPackage PRESENT IF “the link
                    represented by the object instance supports connection management”;
REGISTERED AS {atmfM4NwObjectClass 1};

atmLinkBeh BEHAVIOUR
    DEFINED AS
```

" An atmLink is a topological component used to describe a fixed relationship between two atmSubnetworks (through the contained atmLinkTP instances) and represents a topological association along with capacity. Many atmLinks may exist between a pair of atmSubnetworks, although an atmLink may not exist between a composite atmSubnetwork and any of its component atmSubnetworks. An atmLink is terminated by two atmLinkTPs, one in each atmSubnetwork. These atmLinkTP instances may exist before an instance of atmLink may be created, otherwise they are created as a result of the setupLinkAction. An instance of atmLink is created by the managed system or by using the setupLink ACTION. Overlapping links (and address ranges) are not allowed.

If the availabilityStatus is failed or degraded, the atmLink object shall not allow new atmLinkConnections to be established.

Supported values for the availabilityStatus are:
- Failed: The atmLink cannot function. All underlying transport connections have failed.
- Degraded: The atmLink is degraded in some respect. For instance, the atmLink cannot perform the function of establishing new atmLinkConnections while it can still accept ACTIONS to tear down existing connections.
- Empty SET (none of the availableStatus conditions exist).

The administrativeState is used for administratively locking and unlocking the atmLink. When unlocked, the atmLink functions normally. When in the locked state, the atmLink is prohibited from the set-up, modification, or release of link connections, thus any of these actions shall be rejected. Locking an atmLink does not automatically lock the contained atmLinkConnections.

The characteristicInformation attribute describes the format of the characteristic information that the resource carries. The attribute value is set to vcCI (I.751) for VC Layer atmLinks and vpCI (I.751) for VP Layer atmLinks.

Note that the related atmNetworkAccessProfile information is also in the NE-view atmAccessProfile object contained in the tcAdaptorTTPBidirectional or in the vpTTPBidirectional object. The characteristics described by the atmNetworkAccessProfile associated with an atmLink shall be consistent with the atmNetworkAccessProfile of the related atmLinkTPs.

The setupLinkConnection ACTION sets up a point-to-point connection between two non-connected subnetworkTPs in the each of the linked atmSubnetworks.

The modifyLinkConnection ACTION modifies the QOS and traffic descriptors of a point-to-point connection between two connected subnetworkTPs in the two linked atmSubnetworks.

The releaseLinkConnection ACTION releases a point-to-point connection between subnetworkTPs in each of the linked atmSubnetwork.

Optionally, in the case where both the NE view and the Network view are supported simultaneously over the same interface, the a-LinkTP and z-LinkTP termination pointers may point directly to instances of the NE View vpTTP or tcAdaptorTTPBidirectional object instances instead of atmLinkTPs.”;

```
atmLinkConnection MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Rec. X.721 | ISO/IEC 10165-2":top;
    CHARACTERIZED BY
            “ITU-T M.3100:1995”:createDeleteNotificationsPackage,
            “ITU-T M.3100:1995”:attributeValueChangeNotificationPackage,
            “ITU-T M.3100:1995”:characteristicInformationPackage,
            “ITU-T M.3100:1995”:userLabelPackage,
            “ITU-T Rec. X.721 | ISO/IEC 10165-2”:availabilityStatusPackage,
            atmLinkConnectionPackage  PACKAGE
                    BEHAVIOUR atmLinkConnectionBeh;
                    ATTRIBUTES
                            atmLinkConnectionId
                                GET,
                            a-TPInstance
                                GET,
                            z-TPInstance
                                GET,
                            "ITU-T Rec. X.721 | ISO/IEC 10165-2":
                                        administrativeState
                                GET-REPLACE;;;
    CONDITIONAL PACKAGES
            retainResourcesPackage PRESENT IF “retention of supporting resources
                    after atmSubnetworkConnection release is supported.”,
            “ITU-T M.3100:1995”:tmnCommunicationsAlarmInformationPackage
                    PRESENT IF “communication alarms are supported by this object.”,
            “ITU-T M.3100:1995”:alarmSeverityAssignmentPointerPackage
                    PRESENT IF “communication alarms are supported by this object.”;

REGISTERED AS {atmfM4NwObjectClass 2};

atmLinkConnectionBeh BEHAVIOUR
    DEFINED AS
" An atmLinkConnection represents a I.326 link connection.  An atmLinkConnection is
responsible for transporting characteristic information between subnetworks and is
contained within an atmLink.  It is always bidirectional and point-to-point.  An instance of
atmLinkConnection is terminated by two atmNetworkCTPs.

An instance of this object is created by an action on the atmLink object.  An
atmLinkConnection may be a component of an atmSubnetworkConnection in a composite
subnetwork.  An atmLinkConnection cannot be created between a composite subnetwork
and one of its component subnetworks.

Supported values for the availabilityStatus are:
- Failed:  The atmLinkConnection cannot function
- Empty SET (none of the availableStatus conditions exist).

The administrativeState is used for administratively locking and unlocking the
atmLinkConnection.  When unlocked, the atmLinkConnection functions normally.  When
in the locked state, the atmLinkConnection is prohibited from the transport of characteristic
information.

Optionally where both the Network View and NE View are supported, the
atmLinkConnection termination pointers may point directly to the NE View vp or vc CTP
object instances instead of atmNetworkCTPs.";



atmLinkTP MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Rec. X.721 | ISO/IEC 10165-2":top;
    CHARACTERIZED BY
            “ITU-T M.3100:1995”:createDeleteNotificationsPackage,
            “ITU-T M.3100:1995”:attributeValueChangeNotificationPackage,
            “ITU-T M.3100:1995”:characteristicInformationPackage,
            atmLinkTPPackage  PACKAGE
                    BEHAVIOUR atmLinkTPBeh;
                    ATTRIBUTES
                            atmLinkTPId
```

```
                                        GET
                                        SET-BY-CREATE,
                                    availableIngressBandwidth
                                        GET,
                                    availableEgressBandwidth
                                        GET,
                                    maxAssignableIngressBandwidth
                                        GET,
                                    maxAssignableEgressBandwidth
                                        GET,
                                    linkPointer
                                        GET,
                                    atmNetworkAccessProfilePointer
                                        GET-REPLACE,
                                    serverTTPList
                                        GET-REPLACE ADD-REMOVE;;;
    CONDITIONAL PACKAGES
            supportingUNIorNNIPackage PRESENT IF “The atmLinkTP allows a pointer
to the supporting UNI or NNI”;
REGISTERED AS {atmfM4NwObjectClass 3};

atmLinkTPBeh BEHAVIOUR
    DEFINED AS
```

" An atmLinkTP is a topological component used to represent the termination of an atmLink. Link level configuration information may be associated with the atmLinkTP object. An instance of atmLinkTP is created explicitly by the management system, the managed system, or the setupLink ACTION. Access parameters and pointers to the underlying TTPs are represented in the related atmNetworkAccessProfile objects.

The atmLinkTP terminates atmLink and provides the capability to store link level configuration information.

The available bandwidth attributes describe the aggregated amount of unallocated bandwidth in the ingress and egress directions. The maximum assignable bandwidth attributes describe for each direction the maximum bandwidth that may actually be assigned to a new connection. In the case where the atmLink is supported by a single server trail, the maximum assignable bandwidth is the same as the available bandwidth. The associated atmNetworkAccessProfile describes the total amount of bandwidth (i.e., allocated and unallocated) for the atmLink. This total bandwidth shall be consistent with the bandwidth allocated to the server trails.

The serverTTPList attribute points to the underlying or server TTPs that support the atmLinkTP. If applicable, at the vcLayer these are instances of the vpTTPBid object class. If applicable, at the vpLayer these are instance of the tcAdaptorTTPBid object class.

Attribute value change notifications are applicable to the linkPointer, atmNetworkAccessProfilePointer, and serverTTPList attributes, but not to the attributes associated with available and assignable bandwidth because of their dynamic nature.

The characteristics described by the atmNetworkAccessProfile associated with an atmLinkTP shall be consistent with the atmNetworkAccessProfile of the related atmLink. Note that the related atmNetworkAccessProfile information is also in the NE-view atmAccessProfile object contained in the tcAdaptorTTPBidirectional or in the vpTTPBidirectional object:

- If the profiling information in the atmNetworkAccessProfile object includes attribute values that are more constraining than those in the Access Profile contained in the server TTP and if the corresponding atmLinkTP points to multiple server TTPs, then the server TTP Access Profile information takes precedence over the one in the atmNetworkAccessProfile. If not, the atmNetworkAccessProfile information applies.

- If the profiling information in the atmNetworkAccessProfile object includes attribute values that are less constraining than those in the Access Profile contained in the server TTP and if the multiple atmLinkTP points to a single server TTP, then the atmNetwork AccessProfile information takes precedence over the server TTP Access Profile. If not, the server TTP Access Profile information applies.”;

```
atmNetworkAccessProfile MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Rec. X.721 | ISO/IEC 10165-2":top;
    CHARACTERIZED BY
            “ITU-T M.3100:1995”:createDeleteNotificationsPackage,
            “ITU-T M.3100:1995”:attributeValueChangeNotificationPackage,
            atmNetworkAccessProfilePackage  PACKAGE
                    BEHAVIOUR atmNetworkAccessProfileBeh;
                    ATTRIBUTES
                            atmNetworkAccessProfileId
                                GET
                                SET-BY-CREATE,
                            totalEgressBandwidth
                                GET-REPLACE,
                            totalIngressBandwidth
                                GET-REPLACE,
                            maxNumActiveConnectionsAllowed
                                GET-REPLACE,
                            vpiOrVciRange
                                GET-REPLACE;;;
    CONDITIONAL PACKAGES
            “ATMF M4 NEView”: atmSubscriberAddressPkg PRESENT IF “The
atmLinkTP has a subscriber address assigned directly.”,
            “ATMF M4 NEView”: preferredCarrierPkg PRESENT IF “The atmLinkTP has
a preferred carrier assigned directly.”;
REGISTERED AS {atmfM4NwObjectClass 4};
```

```
atmNetworkAccessProfileBeh BEHAVIOUR
    DEFINED AS
```

" An atmNetworkAccessProfile contains information that describe the maximum ingress and egress bandwidth, along with the range of VPI or VCI values that are applied to the atmLink or atmLinkTP object instances that point to it.

Note that NE-view atmAccessProfile object contained in the tcAdaptorTTPBidirectional or in the vpTTPBidirectional object contains information that shall be consistent with the atmNetworkAccessProfile:

- If the profiling information in the atmNetworkAccessProfile object includes attribute values that are more constraining than those in the Access Profile contained in the server TTP and if the corresponding atmLinkTP points to multiple server TTPs, then the server

TTP AccessProfile information takes precedence over the one in the atmNetworkAccessProfile. If not, the atmNetworkAccessProfile information may apply.

- If the profiling information in the atmNetworkAccessProfile object includes attribute values that are less constraining than those in the Access Profile contained in the server TTP and if the multiple atmLinkTP points to a single server TTP, then the atmNetworkAccessProfile information takes precedence over the server TTP Access Profile. If not, the server TTP Access Profile information applies.”;

```
atmNetworkCTP MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Rec. X.721 | ISO/IEC 10165-2":top;
    CHARACTERIZED BY
            "ITU-T M.3100:1995":attributeValueChangeNotificationPackage,
            "ITU-T M.3100:1995":createDeleteNotificationsPackage,
            “ITU-T M.3100:1995”:characteristicInformationPackage,
            atmNetworkCTPPkg PACKAGE
                    BEHAVIOUR atmNetworkCTPBeh;
                    ATTRIBUTES
                            atmNetworkCTPId
                                GET,
                            virtualId
                                GET,
                            “ATMF M4 NEView”: segmentEndPoint
                                GET-REPLACE,
                            relatedAtmTTP
                                GET-REPLACE,
                            relatedTrafficDescriptors
                                GET-REPLACE;;;
            CONDITIONAL PACKAGES
            “ATMF M4 NEView”: oamCellLoopbackPkg
               PRESENT IF "the termination point supports OAM cell Loopbacks",
            “ITU-T M.3100:1995”:tmnCommunicationsAlarmInformationPackage
                    PRESENT IF “communication alarms are supported by this object.”,
            “ITU-T M.3100:1995”:alarmSeverityAssignmentPointerPackage
                    PRESENT IF “communication alarms are supported by this object.”;

REGISTERED AS {atmfM4NwObjectClass 5};

atmNetworkCTPBeh BEHAVIOUR
    DEFINED AS
```

" The atmNetworkCTP object class is used when the Network View only is provided. Every client CTP is supported by an underlying server layer TTP (identified through the linkTP).

The relatedAtmTTP attribute is used to associate the final CTP of a VCC or VPC with the same layer Trail Termination Point. Other attributes reflect the VCI/VPI (depending on network layer), traffic descriptors, and quality of service class.

The relatedTrafficDescriptors attribute may be used to point to the traffic descriptor profile at points where ingress and/or egress UPC/NPC functions are performed or when the relatedAtmTTP attribute points to an instance of the atmNetworkTTP object class. The object pointed to by the relatedTrafficDescriptors attribute may also contain QOS information.

The tmnCommunicationsAlarmInformationPackage allows the reporting of communications alarms associated with the atmNetworkCTP. When an AIS or RDI failure is detected and this package is present, the atmNetworkCTP object shall generate a communicationsAlarm notification with the probableCause parameter value set equal to aIS or farEndReceiverFailure, respectively.

The conditional package oamCellLoopbackPkg provides the M-ACTION used to request the termination point to insert an OAM cell for downstream loopback and to report whether or not the cell was returned within the required time.";

```
atmNetworkTTP MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Rec. X.721 | ISO/IEC 10165-2":top;
    CHARACTERIZED BY
            "ITU-T M.3100:1995": attributeValueChangeNotificationPackage,
            "ITU-T M.3100:1995": createDeleteNotificationsPackage,
            “ITU-T M.3100:1995”:characteristicInformationPackage,
            atmNetworkTTPPkg PACKAGE
                    BEHAVIOUR atmNetworkTTPBeh;
                    ATTRIBUTES
                            atmNetworkTTPId
                                GET,
                            relatedAtmCTP
                                GET-REPLACE;;;
            CONDITIONAL PACKAGES
            “ATMF M4 NEView”: oamCellLoopbackPkg
               PRESENT IF "the termination point supports OAM cell Loopbacks",
            “ITU-T M.3100:1995”:tmnCommunicationsAlarmInformationPackage
                    PRESENT IF “communication alarms are supported by this object.”,
            “ITU-T M.3100:1995”:alarmSeverityAssignmentPointerPackage
               PRESENT IF “communication alarms are supported by this object.”,
            “ITU-T Rec. X.721 | ISO/IEC 10165-2”:availabilityStatusPackage
               PRESENT IF "the object supports an indication of a degraded or failure
               state.";
REGISTERED AS {atmfM4NwObjectClass 6};

atmNetworkTTPBeh BEHAVIOUR
    DEFINED AS
```

" The atmNetworkTTP object class is used when the Network View only is provided.

The relatedAtmCTP attribute is used to associate the final CTP of a VCC or VPC with the Trail Termination Point.

The conditional package oamCellLoopbackPkg provides the M-ACTION used to request the termination point to insert an OAM cell for downstream loopback and to report whether or not the cell was returned within the required time.

The availabilityStatusPackage is a conditional package that may be used to indicate the availability of the atmNetworkTTP. Changes in the availabilityStatus are reported using the attributeValueChangeNotification.

```
Supported values for the availabilityStatus are:
- Failed:  The atmLinkConnection cannot function
- Empty SET (none of the availableStatus conditions exist). ";



atmNetworkTrafficDescriptorProfile MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Rec. X.721 | ISO/IEC 10165-2":top;
    CHARACTERIZED BY
            "ITU-T M.3100": attributeValueChangeNotificationPackage,
            "ITU-T M.3100": createDeleteNotificationsPackage,
            atmNetworkTrafficDescriptorProfilePkg PACKAGE
                    BEHAVIOUR atmNetworkTrafficDescriptorProfileBeh;
                    ATTRIBUTES
                            atmNetworkTrafficDescriptorProfileId
                                GET;;;
            CONDITIONAL PACKAGES
            “ATMF M4 NEView”: egressTrafficDescriptorPkg
               PRESENT IF " This package must be present at points where egress traffic
               descriptors need to be provided.",
            “ATMF M4 NEView”: ingressTrafficDescriptorPkg
               PRESENT IF " This package must be present at points where ingress traffic
               descriptors need to be provided.",
            “ATMF M4 NEView”: qosClassesPkg
               PRESENT IF " This package must be present at points where quality of
               service descriptors need to be provided.";
    REGISTERED AS {atmfM4NwObjectClass 7};

atmNetworkTrafficDescriptorProfileBeh BEHAVIOUR
    DEFINED AS
" An atmNetworkTrafficDescriptorProfile contains information that describes the ingress
and egress Peak Cell Rate, ingress and egress CDV Tolerance, ingress and egress
Sustainable Cell Rate, and the ingress and egress Maximum Burst Size.”;



atmRoutingProfile MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Rec. X.721 | ISO/IEC 10165-2":top;
    CHARACTERIZED BY
            "ITU-T M.3100:1995": createDeleteNotificationsPackage,
            atmRoutingProfilePkg PACKAGE
                    BEHAVIOUR atmRoutingProfileBeh;
                    ATTRIBUTES
                            atmRoutingProfileId
                                GET,
                            routeDescriptionList
                                GET-REPLACE ADD-REMOVE,
                            maxHops
                                GET-REPLACE,
                            connectionTypesSupported
                                GET-REPLACE ADD-REMOVE;;;
REGISTERED AS {atmfM4NwObjectClass 8};
```

atmRoutingProfileBeh BEHAVIOUR
    DEFINED AS
" The atmRoutingProfile object class represents a set of topological routing constraints that can be applied to a new connection or trail during setup. A routing profile may be created automatically based on the routing description in the setup action. The management system may also create profiles directly. Each atmSubnetworkConnection or atmTrail may point to an atmRoutingProfile. Connections shall not be established (or re-established) if the routing criteria cannot be met. If maxHops is specified, the connection shall not be established (or re-established) if the maximum number of hops is exceeded (including hops that may not be visible to the managing systems).

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

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

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

Objects (such as atmSubnetwork, atmLink, or managedElement, etc) may be referenced by the routeDescriptionList as being excluded, mandatory, or preferred. If an object is described as mandatory it must be used in setting up a new connection. An attempt must be made during setup to include an object described as preferred. An excluded object must not be used in a connection.

Connection objects (such as atmTrail, atmSubnetworkConnection, etc) may be referenced by the routeDescriptionList as same route or diverse route. A new connection being created shall follow the same route as a sameRoute referenced object. A new connection must follow a different route than a referenced object referred to as diverseRoute.

The routing information in setup actions may be either explicitly stated in the action or the action can point to an existing instance of the atmRoutingProfile object class. ";

```
atmSubnetwork MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T M.3100:1995":networkR1;
    CHARACTERIZED BY
            “ITU-T M.3100:1995”:createDeleteNotificationsPackage,
            “ITU-T M.3100:1995”:attributeValueChangeNotificationPackage,
            “ITU-T Rec. X.721 | ISO/IEC 10165-2”:availabilityStatusPackage,
            “ITU-T M.3100:1995”:userLabelPackage,
            “ITU-T M.3100:1995”:characteristicInformationPackage,
            atmSubnetworkPackage  PACKAGE
                    BEHAVIOUR atmSubnetworkBeh;
                    ATTRIBUTES
                            “ITU-T M.3100:1995”:supportedByObjectList
                                GET-REPLACE ADD-REMOVE,
                            containedLinkList
```

```
                                        GET-REPLACE ADD-REMOVE,
                                    containedSubnetworkList
                                        GET-REPLACE ADD-REMOVE,
                                    supportedLinkTPList
                                        GET-REPLACE ADD-REMOVE;;;
    CONDITIONAL PACKAGES
            subnetworkMultipointActionsPackage PRESENT IF “the atmSubnetwork
                    supports multipoint connections.”,
            subnetworkConnectionManagementPackage PRESENT IF “the subnetwork
                    represented by the object instance supports connection management”;
REGISTERED AS {atmfM4NwObjectClass 9};

atmSubnetworkBeh BEHAVIOUR
    DEFINED AS
```

"An atmSubnetwork is a topological component used for carrying characteristic information(ATM cells within a layer network). The atmSubnetwork is delineated by ATM Subnetwork Termination Points (atmSubnetworkTP). An atmSubnetwork may: be empty containing no atmSubnetworkTP instances; associated with a single linkTP in which case it referred to as a point subnetwork; or associated with many termination points. Subnetworks are used for making subnetwork connections. An instance of atmSubnetwork is specific to the VC or VP layer and is contained in the appropriate vcLayerNetworkDomain or vpLayerNetworkDomain. A point subnetwork does not contain any visible subnetwork connections.

The atmSubnetwork object provides an abstraction that allows the establishment and removal of connections across the atmSubnetwork.

characteristicInformation describes the format of the characteristic information that the resource carries. This is set to vcCI (I.751) for VC Layer atmSubnetworks and vpCI (I.751) for VP Layer atmSubnetworks. The characteristicInformation, where present, for dependent objects shall match this attribute

The userLabel may be used to describe the managing organization. In cases where the atmSubnetwork is managed by a different system the inherited systemTitle may be used.

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

Supported values for the availabilityStatus are:
- Degraded: The atmSubnetwork is degraded in some respect. For instance, the atmSubnetwork cannot perform the function of establishing new atmSubnetworkConnections while it can still accept ACTIONS to tear down existing connections.
- Empty SET (none of the availableStatus conditions exist).

The setupSubnetworkConnection ACTION sets up a point-to-point or a multipoint connection between non-connected subnetworkTPs in the atmSubnetwork.

The modifySubnetworkConnection ACTION modifies the QOS and traffic descriptors of a point-to-point or a multipoint connection between subnetworkTPs in the atmSubnetwork.

The addTpsToSubnetworkConnection ACTION adds atmSubnetworkTPs to an existing multipoint subnetwork connection.

The removeTpsFromSubnetworkConnection ACTION releases atmSubnetworkTPs from a multipoint atmSubnetworkConnection.

The releaseSubnetworkConnection ACTION releases a point-to-point or a multipoint connection between subnetworkTPs in the atmSubnetwork.";

```
atmSubnetworkConnection MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Rec. X.721 | ISO/IEC 10165-2":top;
    CHARACTERIZED BY
            "ITU-T M.3100:1995":createDeleteNotificationsPackage,
            "ITU-T M.3100:1995":attributeValueChangeNotificationPackage,
            "ITU-T M.3100:1995":characteristicInformationPackage,
            "ITU-T M.3100:1995":userLabelPackage,
            "ITU-T Rec. X.721 | ISO/IEC 10165-2":availabilityStatusPackage,
            atmSubnetworkConnectionPackage  PACKAGE
                    BEHAVIOUR atmSubnetworkConnectionBeh;
                    ATTRIBUTES
                            atmSubnetworkConnectionId
                                GET,
                            a-TPInstance
                                GET,
                            z-TPList
                                GET,
                            connectionType
                                GET-REPLACE,
                            restorableIndicator
                                GET-REPLACE,
                            componentSubnetworkConnectionList
                                GET-REPLACE ADD-REMOVE,
                            componentLinkConnectionList
                                GET-REPLACE ADD-REMOVE,
                            provisionType
                                GET-REPLACE,
                            relatedAtmRoutingProfile
                                GET-REPLACE,
                            "ITU-T Rec. X.721 | ISO/IEC 10165-2":
                                        administrativeState
                                GET-REPLACE;;;
    CONDITIONAL PACKAGES
            retainResourcesPackage PRESENT IF "retention of supporting resources
                    after atmTrail or containing atmSubnetworkConnection release is
                    supported.",
            "ITU-T M.3100:1995":tmnCommunicationsAlarmInformationPackage
                    PRESENT IF "communication alarms are supported by this object.",
            "ITU-T M.3100:1995":alarmSeverityAssignmentPointerPackage
                    PRESENT IF "communication alarms are supported by this object.";
REGISTERED AS {atmfM4NwObjectClass 10};

atmSubnetworkConnectionBeh BEHAVIOUR
    DEFINED AS
```

" An atmSubnetworkConnection represents a connection across a subnetwork. An atmSubnetworkConnection is responsible for transporting cells across a subnetwork. It is always bidirectional. An instance of atmSubnetworkConnection is terminated by atmSubnetworkTPs.

An instance of this object is created by the managed system or by an action on the atmSubnetwork object. An atmSubnetworkConnection in a composite subnetwork is made up of a series of atmSubnetworkConnections and atmLinkConnections. An atmSubnetworkConnection cannot be created between a composite subnetwork and one of its component subnetworks.

Supported values for the availabilityStatus are:
- Failed: The atmSubnetworkConnection cannot function
- Empty SET (none of the availableStatus conditions exist).

The administrativeState is used for administratively locking and unlocking the atmSubnetworkConnection. When unlocked, the atmSubnetworkConnection functions normally. When in the locked state, the atmSubnetworkConnection is prohibited from the transport of characteristic information.

For point to point Subnetwork Connections the a-TPInstance and a single entry in the z-TPList are used to indicate the endpoints. Multiple entries in the z-TPList and the a-TPInstance are used to represent the end points of broadcast (point-to-multipoint), merge (multipoint-to-point), and composite connections. The a-TPInstance identifies the primary endpoint. Only the z-TPList is used identify all end points in a multipoint-to-multipoint connection (there is no primary end point). In this case the a-TPInstance shall be NULL.

Optionally where both the Network View and NE View are supported and the atmSubnetworkConnection is not broken down into component network level connections, the componentSubnetworkConnectionList of an atmSubnetwork Connection may point directly to the NE View Cross Connection object instances instead of atmSubnetworkConnections, while the componentLinkConnectionList points to atmLinkConnections whose termination points are instances of the NE View vp or vc CTP object class. ";

```
atmSubnetworkTP MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Rec. X.721 | ISO/IEC 10165-2":top;
    CHARACTERIZED BY
            “ITU-T M.3100:1995”:createDeleteNotificationsPackage,
            “ITU-T M.3100:1995”:attributeValueChangeNotificationPackage,
            “ITU-T M.3100:1995”:characteristicInformationPackage,
            “ITU-T M.3100:1995”:userLabelPackage,
            atmSubnetworkTPPackage  PACKAGE
                    BEHAVIOUR atmSubnetworkTPBeh;
                    ATTRIBUTES
                            atmSubnetworkTPId
                                GET
                                SET-BY-CREATE,
                            relatedLinkConnection
                                GET,
                            relatedLinkTP
                                GET,
                            relatedSubnetworkConnection
                                GET,
                        reflectedCTP
                            GET-REPLACE;;;
REGISTERED AS {atmfM4NwObjectClass 11};

atmSubnetworkTPBeh BEHAVIOUR
    DEFINED AS
```

"An atmSubnetworkTP represent the termination of an atmSubnetworkConnection or an atmLinkConnection on an atmSubnetwork. Each atmSubnetworkTP instance refers to a CTP object instance allowing a single representation of the network resources. That is, through the layers of subnetwork decomposition, the instances of atmSubnetworkTP at each layer all point to the same CTP object instance, thus representing the relationship among endpoints of composite subnetworks supported by the same network resource. An instance of atmSubnetworkTP is created by the managed system during the set-up of link connections and/or subnetwork connections. However, this does not preclude the pre-provisioning of atmSubnetworkTPs.

The atmSubnetworkTP terminates atmSubnetworkConnections and atmLinkConnections and provides the capability to associate endpoints of composite subnetworks with the supporting network resource (CTP).

In cases where the M4 NE view is also supported, the reflectedCTP attribute may point directly to an instance of the vcCTPBidirectional or vpCTPBidirectional object classes.";

```
atmTrail MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Rec. X.721 | ISO/IEC 10165-2":top;
    CHARACTERIZED BY
            "ITU-T M.3100:1995":createDeleteNotificationsPackage,
            "ITU-T M.3100:1995":attributeValueChangeNotificationPackage,
            "ITU-T M.3100:1995":characteristicInformationPackage,
            "ITU-T M.3100:1995":userLabelPackage,
            atmTrailPackage  PACKAGE
                    BEHAVIOUR atmTrailBeh;
                    ATTRIBUTES
                        atmTrailId
                            GET,
                        a-TPInstance
                            GET,
                        z-TPList
                            GET,
                        connectionType
                            GET-REPLACE,
                        restorableIndicator
                            GET-REPLACE,
                        provisionType
                            GET-REPLACE,
                        relatedAtmRoutingProfile
                            GET-REPLACE,
                        "ITU-T Rec. X.721 | ISO/IEC 10165-2":
                                    administrativeState
                            GET-REPLACE;;;
    CONDITIONAL PACKAGES
            retainConnectionsPackage PRESENT IF "retention of supporting connections
```

```
                after trail release is supported.”,
        “ITU-T M.3100:1995”:tmnCommunicationsAlarmInformationPackage
                PRESENT IF “communication alarms are supported by this object.”,
        “ITU-T M.3100:1995”:alarmSeverityAssignmentPointerPackage
                PRESENT IF “communication alarms are supported by this object.”;
REGISTERED AS {atmfM4NwObjectClass 12};

atmTrailBeh BEHAVIOUR
    DEFINED AS
" An atmTrail represents an I.326 VC Trail or VP Trail.  An atmTrail is always
bidirectional.  An instance is terminated by an atmNetworkTTP.  An instance of this object
is created by the managed system or by an action on the vcLayerNetworkDomain or
vpLayerNetworkDomain object.

The atmTrail object provides connectivity across a specific vcLayerNetworkDomain or
vpLayerNetworkDomain.

The administrativeState is used for administratively locking and unlocking the atmTrail.
When unlocked, the atmTrail functions normally.  When in the locked state, the atmTrail is
prohibited from the transport of characteristic information.
For point to point Subnetwork Connections the a-TPInstance and a single entry in the z-
TPList are used to indicate the endpoints.  Multiple entries in the z-TPList and the a-
TPInstance are used to represent the end points of broadcast (point-to-multipoint), merge
(multipoint-to-point), and composite connections.  The a-TPInstance identifies the primary
endpoint.  Only the z-TPList is used identify all end points in a multipoint-to-multipoint
connection (there is no primary end point).  In this case the a-TPInstance shall be NULL.";



atmTrailRequest MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Rec. X.721 | ISO/IEC 10165-2":top;
    CHARACTERIZED BY
        “ITU-T M.3100:1995”:createDeleteNotificationsPackage,
        “ITU-T M.3100:1995”:attributeValueChangeNotificationPackage,
        atmTrailRequestPackage  PACKAGE
                BEHAVIOUR atmTrailRequestBeh;
                ATTRIBUTES
                        atmTrailRequestId
                            GET,
                        requestStatus
                            GET,
                        requestType
                            GET,
                        requestCommittedTime
                            GET,
                        relatedAtmTrail
                            GET,
                        requestActionInfo
                            GET;;;
REGISTERED AS {atmfM4NwObjectClass 13};

atmTrailRequestBeh BEHAVIOUR
    DEFINED AS
```

"This object class represents a deferred request of the vcLayerNetworkDomain or vpLayerNetworkDomain to either set-up, release, modify, or alter the end-points (multipoint case) of an atmTrail. If the requestType is not setup, the relationship to the atmTrail is established when the instance is created. In the case where requestType is setup, the relationship to atmTrail is established when the setup action activates a trail.

The atmTrailRequest object provides a mechanism to track deferred requests made to the vcLayerNetworkDomain or vpLayerNetworkDomain.

The requestCommittedTime attribute describe the time at which the agent commits to performing the requested action. This may differ from the requested activation time on the initial request. The requestStatus describes the status of the deferred request.

This object is created as a result of an action on the vcLayerNetworkDomain or vpLayerNetworkDomain object.";

```
vcLayerNetworkDomain MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T M.3100:1995":networkR1;
    CHARACTERIZED BY
            “ITU-T M.3100:1995”:createDeleteNotificationsPackage,
            “ITU-T M.3100:1995”:userLabelPackage,
            vcLayerNetworkDomainPackage  PACKAGE
                    BEHAVIOUR vcLayerNetworkDomainBeh;;;
    CONDITIONAL PACKAGES
            layerNetworkDomainTrailActionsPackage PRESENT IF “the vc layer
                    network domain object instance supports management of ATM trails.”,
            layerNetworkDomainLinkActionsPackage PRESENT IF “the vc layer network
                    domain object instance supports management of ATM links.”,
            layerNetworkDomainMultipointActionsPackage PRESENT IF “the vc layer
                    network domain object instance supports management of multipoint
                    trails.”;
REGISTERED AS {atmfM4NwObjectClass 14};

vcLayerNetworkDomainBeh BEHAVIOUR
    DEFINED AS
```

"The vcLayerNetworkDomain object class represents the part of the VC Layer which is available to a managing system through the M4 interface. The vcLayerNetworkDomain corresponds to an administration. A vcLayerNetworkDomain is defined to support the requirement for independent layer management of the VC Layer. The vcLayerNetworkDomain object represents part of an administration’s portion of the VC Layer which is available to a managing system through the M4 interface. It contains only objects from a single VC Layer. In this model, a VC Layer Network Domain is associated with one and only one top subnetwork, which can be further decomposed. Portions of a trail that traverse another administration are represented by subnetwork connections in subnetworks named by the vcLayerNetworkDomain of that administration. These foreign subnetworks are component subnetworks that have topological relationships with other component subnetworks of the vcLayerNetworkDomain. There may be several Layer Network Domains within a single Network.

A vcLayerNetworkDomain is defined to support the requirement for independent layer management of the VC Layer. The vcLayerNetworkDomain object provides an abstraction that allows the establishment and removal of atmTrails across the vcLayerNetworkDomain.

The userLabel may be used to represent additional information about the layer network domain. In cases where the vcLayerNetworkDomain is managed by a different system the inherited systemTitle may be used.

Created automatically at the initialization of the superior (network) object. The automatic creation of instances of this object shall be reported over the M4 interface to the managing system. The managing system may subsequently create instances of vcLayerNetworkDomain.

The setupTrail ACTION sets up, or schedules the setup of a point-to-point or a multipoint connection between non-connected TTPs in the layer network.

The modifyTrail ACTION modifies, or schedules the modification of the parameters of a point-to-point or a multipoint connection between connected TTPs in the layer network.

The addTpsToTrail ACTION adds, or schedules the addition of TTPs to an existing multipoint trail connection in the layer network.

The removeTpsFromTrail ACTION releases, or schedules the release of TTPs from a multipoint trail in the layer network.

The releaseTrail ACTION releases, or schedules the release of a point-to-point or a multipoint trail between TTPs in the layer network.

The cancelTrailRequest ACTION cancels a deferred trail setup, modification, release, addTps, or removeTps request.

The setupLink ACTION sets up a point-to-point link between two subnetworks or NEs in the atmSubnetwork";

```
vpLayerNetworkDomain MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T M.3100:1995":networkR1;
    CHARACTERIZED BY
            “ITU-T M.3100:1995”:createDeleteNotificationsPackage,
            “ITU-T M.3100:1995”:userLabelPackage,
            vpLayerNetworkDomainPackage  PACKAGE
                    BEHAVIOUR vpLayerNetworkDomainBeh;;;
    CONDITIONAL PACKAGES
            layerNetworkDomainTrailActionsPackage PRESENT IF “the vp layer
                    network domain object instance supports management of ATM trails.”,
            layerNetworkDomainLinkActionsPackage PRESENT IF “the vp layer
                    network domain object instance supports management of ATM links.”,
            layerNetworkDomainMultipointActionsPackage PRESENT IF “the vp layer
                    network domain object instance supports management of multipoint
                    trails.”;
REGISTERED AS {atmfM4NwObjectClass 15};

vpLayerNetworkDomainBeh BEHAVIOUR
    DEFINED AS
```

" The vpLayerNetworkDomain object class represents the part of the VP Layer which is available to a managing system through the M4 interface. The vpLayerNetworkDomain corresponds to an administration A vpLayerNetworkDomain is defined to support the requirement for independent layer management of the VP Layer. The vpLayerNetworkDomain object represents part of an administration’s portion of the VP Layer which is available to a managing system through the M4 interface. It contains only objects from a single VP Layer. In this model, a VP Layer Network Domain is associated with one and only one top subnetwork, which can be further decomposed. Portions of a trail that traverse another administration are represented by subnetwork connections in subnetworks named by the vpLayerNetworkDomain of that administration. These foreign subnetworks are related components that have topological relationships with other component subnetworks of the vpLayerNetworkDomain. There may be several Layer Network Domains within a single Network.

A vpLayerNetworkDomain is defined to support the requirement for independent layer management of the VP Layer. The vpLayerNetworkDomain object provides an abstraction that allows the establishment and removal of atmTrails across the vpLayerNetworkDomain.

The userLabel may be used to represent additional information about the layer network domain. In cases where the vcLayerNetworkDomain is managed by a different system the inherited systemTitle may be used.

Created automatically at the initialization of the superior (network) object. The automatic creation of instances of this object shall be reported over the M4 interface to the managing system. The managing system may subsequently create instances of vpLayerNetworkDomain.

The setupTrail ACTION sets up, or schedules the setup of a point-to-point or a multipoint connection between non-connected TTPs in the layer network.

The modifyTrail ACTION modifies, or schedules the modification of the parameters of a point-to-point or a multipoint connection between connected TTPs in the layer network.

The addTpsToTrail ACTION adds, or schedules the addition of TTPs to an existing multipoint trail connection in the layer network.

The removeTpsFromTrail ACTION releases, or schedules the release of TTPs from a multipoint trail in the layer network.

The releaseTrail ACTION releases, or schedules the release of a point-to-point or a multipoint trail between TTPs in the layer network.

The cancelTrailRequest ACTION cancels a deferred trail setup, modification, release, addTps, or removeTps request.";

Packages

```
layerNetworkDomainLinkActionsPackage PACKAGE
    ACTIONS
            setupLink;
REGISTERED AS {atmfM4NwPackage 1};


layerNetworkDomainMultipointActionsPackage PACKAGE
    ACTIONS
            addTpsToTrail,
            removeTpsFromTrail;
REGISTERED AS {atmfM4NwPackage 2};


layerNetworkDomainTrailActionsPackage PACKAGE
    ACTIONS
            setupTrail,
            modifyTrail,
            releaseTrail,
            cancelTrailRequest;
REGISTERED AS {atmfM4NwPackage 3};


linkConnectionManagementPackage PACKAGE
    ACTIONS
            setupLinkConnection,
            modifyLinkConnection,
            releaseLinkConnection;
REGISTERED AS {atmfM4NwPackage 4};


retainConnectionsPackage PACKAGE
    ATTRIBUTES
            retainConnectionsIndicator
                    GET-REPLACE;
REGISTERED AS {atmfM4NwPackage 5};


retainResourcesPackage PACKAGE
    ATTRIBUTES
            retainResourcesIndicator
                    GET-REPLACE;
REGISTERED AS {atmfM4NwPackage 6};


subnetworkConnectionManagementPackage PACKAGE
    ACTIONS
            setupSubnetworkConnection,
            modifySubnetworkConnection,
            releaseSubnetworkConnection;
REGISTERED AS {atmfM4NwPackage 7};
```

```
subnetworkMultipointActionsPackage PACKAGE
    ACTIONS
            addTpsToSubnetworkConnection,
            removeTpsFromSubnetworkConnection;
REGISTERED AS {atmfM4NwPackage 8};


supportingUNIorNNIPackage PACKAGE
    ATTRIBUTES
            supportingUNIorNNI
                    GET-REPLACE ADD-REMOVE;
REGISTERED AS {atmfM4NwPackage 9};
```

Attributes

```
a-LinkTP ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.ObjectList;
        MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR a-LinkTPBeh;
REGISTERED AS {atmfM4NwAttribute 1};

a-LinkTPBeh BEHAVIOUR
  DEFINED AS
" This attribute represents the association between an atmLink and an existing atmLinkTP.
An atmLink links exactly 2 atmLinkTPs;  an atmLinkTP is linked by zero or one atmLinks.
This attribute points to one of the two linked atmLinkTPs.  In this case the a-LinkTP points
to a single atmLinkTP object instance.  Usually this attribute points to a single termination
point.  However, in cases where the NE view is also supported and the atmLink is
supported by more than one server trail, this attribute may point to more than one
termination point. ";


a-TPInstance ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.PointerOrNull;
        MATCHES FOR EQUALITY;
        BEHAVIOUR a-TPInstanceBeh;
REGISTERED AS {atmfM4NwAttribute 2};

a-TPInstanceBeh BEHAVIOUR
  DEFINED AS
" For point to point connections the a-TPInstance is used to indicate one of the endpoints.
The a-TPInstance is used to represent the primary end-point of broadcast (point-to-
multipoint), merge (multipoint-to-point), and composite connections.  In the case of
multipoint-to-multipoint connections there is no primary end point, and the a-TPInstance
shall be NULL. ";


atmLinkConnectionId ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.NameType;
        MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
        BEHAVIOUR atmLinkConnectionIdBeh;
REGISTERED AS {atmfM4NwAttribute 3};

atmLinkConnectionIdBeh BEHAVIOUR
  DEFINED AS
" This attribute is used to name instances of the atmLinkConnection managed object class.";


atmLinkId ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.NameType;
        MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
        BEHAVIOUR atmLinkIdBeh;
REGISTERED AS {atmfM4NwAttribute 4};

atmLinkIdBeh BEHAVIOUR
```

```
  DEFINED AS
" This attribute is used to name instances of the atmLink managed object class.";


atmLinkTPId ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.NameType;
        MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
        BEHAVIOUR atmLinkTPIdBeh;
REGISTERED AS {atmfM4NwAttribute 5};

atmLinkTPIdBeh BEHAVIOUR
  DEFINED AS
" This attribute is used to name instances of the atmLinkTP managed object class.";


atmNetworkAccessProfileId ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.NameType;
        MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
        BEHAVIOUR atmNetworkAccessProfileIdBeh;
REGISTERED AS {atmfM4NwAttribute 6};

atmNetworkAccessProfileIdBeh BEHAVIOUR
  DEFINED AS
" This attribute is used to name instances of the atmNetworkAccessProfile managed object
class.";


atmNetworkAccessProfilePointer ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.PointerOrNull;
        MATCHES FOR EQUALITY;
        BEHAVIOUR atmNetworkAccessProfilePointerBeh;
REGISTERED AS {atmfM4NwAttribute 7};

atmNetworkAccessProfilePointerBeh BEHAVIOUR
  DEFINED AS
" This attribute represents the association between an atmLinkTP and an
atmNetworkAccessProfile.  If this association is not supported by the object class this
attribute shall take on the value NULL.";


atmNetworkCTPId ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.NameType;
        MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
        BEHAVIOUR atmNetworkCTPIdBeh;
REGISTERED AS {atmfM4NwAttribute 8};

atmNetworkCTPIdBeh BEHAVIOUR
  DEFINED AS
" This attribute is used to name instances of the atmNetworkCTP managed object class.";


atmNetworkTTPId ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.NameType;
        MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
```

```
            BEHAVIOUR atmNetworkTTPIdBeh;
REGISTERED AS {atmfM4NwAttribute 9};

atmNetworkTTPIdBeh BEHAVIOUR
  DEFINED AS
" This attribute is used to name instances of the atmNetworkTTP managed object class.";


atmNetworkTrafficDescriptorProfileId ATTRIBUTE
            WITH ATTRIBUTE SYNTAX AtmNwMIBMod.NameType;
            MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
            BEHAVIOUR atmNetworkTrafficDescriptorProfileIdBeh;
REGISTERED AS {atmfM4NwAttribute 10};

atmNetworkTrafficDescriptorProfileIdBeh BEHAVIOUR
  DEFINED AS
" This attribute is used to name instances of the atmNetworkTrafficDescriptorProfile
managed object class.";


atmRoutingProfileId ATTRIBUTE
            WITH ATTRIBUTE SYNTAX AtmNwMIBMod.NameType;
            MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
            BEHAVIOUR atmRoutingProfileIdBeh;
REGISTERED AS {atmfM4NwAttribute 11};

atmRoutingProfileIdBeh BEHAVIOUR
  DEFINED AS
" This attribute is used to name instances of the atmRoutingProfile managed object class.";


atmSubnetworkConnectionId ATTRIBUTE
            WITH ATTRIBUTE SYNTAX AtmNwMIBMod.NameType;
            MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
            BEHAVIOUR atmSubnetworkConnectionIdBeh;
REGISTERED AS {atmfM4NwAttribute 12};

atmSubnetworkConnectionIdBeh BEHAVIOUR
  DEFINED AS
" This attribute is used to name instances of the atmSubnetworkConnection managed object
class.";


atmSubnetworkTPId ATTRIBUTE
            WITH ATTRIBUTE SYNTAX AtmNwMIBMod.NameType;
            MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
            BEHAVIOUR atmSubnetworkTPIdBeh;
REGISTERED AS {atmfM4NwAttribute 14};

atmSubnetworkTPIdBeh BEHAVIOUR
  DEFINED AS
" This attribute is used to name instances of the atmSubnetwork managed object class.";
```

```
atmTrailId ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.NameType;
        MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
        BEHAVIOUR atmTrailIdBeh;
REGISTERED AS {atmfM4NwAttribute 15};

atmTrailIdBeh BEHAVIOUR
  DEFINED AS
" This attribute is used to name instances of the atmTrail managed object class.";


atmTrailRequestId ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.NameType;
        MATCHES FOR EQUALITY, ORDERING, SUBSTRINGS;
        BEHAVIOUR atmTrailRequestIdBeh;
REGISTERED AS {atmfM4NwAttribute 16};

atmTrailRequestIdBeh BEHAVIOUR
  DEFINED AS
" This attribute is used to name instances of the atmTrailRequest managed object class.";


availableEgressBandwidth ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.Integer;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR availableEgressBandwidthBeh;
REGISTERED AS {atmfM4NwAttribute 17};

availableEgressBandwidthBeh BEHAVIOUR
  DEFINED AS
" The available bandwidth attributes describe the aggregated amount of unallocated
bandwidth in the egress direction on the Object Class.";


availableIngressBandwidth ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.Integer;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR availableIngressBandwidthBeh;
REGISTERED AS {atmfM4NwAttribute 18};

availableIngressBandwidthBeh BEHAVIOUR
  DEFINED AS
" The available bandwidth attributes describe the aggregated amount of unallocated
bandwidth in the ingress direction on the Object Class.";


componentLinkConnectionList ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.ObjectList;
        MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR componentLinkConnectionListBeh;
REGISTERED AS {atmfM4NwAttribute 19};

componentLinkConnectionListBeh BEHAVIOUR
  DEFINED AS
```

```
"This attribute represents the association between an atmSubnetworkConnection and its
component atmLinkConnections.  An atmSubnetworkConnection may be made up of zero
or more atmLinkConnections. ";


componentSubnetworkConnectionList ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.ObjectList;
        MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR componentSubnetworkConnectionListBeh;
REGISTERED AS {atmfM4NwAttribute 20};

componentSubnetworkConnectionListBeh BEHAVIOUR
  DEFINED AS
"This attribute represents the association between an atmSubnetworkConnection and its
component atmSubnetworkConnections or NE View cross connections if applicable.  An
atmSubnetworkConnection may be made up of zero, two or more
atmSubnetworkConnections. ";


connectionType ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.ConnectionType;
        MATCHES FOR EQUALITY;
        BEHAVIOUR connectionTypeBeh;
REGISTERED AS {atmfM4NwAttribute 21};

connectionTypeBeh BEHAVIOUR
  DEFINED AS
"This attribute describes the type of trail or connection: point-to-point, multicast,
merge,multicast/merge, full multipoint.";


connectionTypesSupported ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.ConnectionTypesSupported;
        MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR connectionTypesSupportedBeh;
REGISTERED AS {atmfM4NwAttribute 22};

connectionTypesSupportedBeh BEHAVIOUR
  DEFINED AS
"This attribute represents the types of connection that are supported by an
atmRoutingProfile. ";


containedLinkList ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.ObjectList;
        MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR containedLinkListBeh;
REGISTERED AS {atmfM4NwAttribute 23};

containedLinkListBeh BEHAVIOUR
  DEFINED AS
"This attribute represents the association between an atmSubnetwork and its component
atmLinks.  These component atmLinks link the partitioned atmSubnetworks within a given
atmSubnetwork.  An atmSubnetwork may include zero or more atmLinks. ";
```

```
containedSubnetworkList ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.ObjectList;
        MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR containedSubnetworkListBeh;
REGISTERED AS {atmfM4NwAttribute 24};

containedSubnetworkListBeh BEHAVIOUR
  DEFINED AS
"This attribute represents the association between an atmSubnetwork and its component
atmSubnetworks.  An atmSubnetwork may be partitioned into zero or more
atmSubnetworks. ";


linkPointer ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.PointerOrNull;
        MATCHES FOR EQUALITY;
        BEHAVIOUR linkPointerBeh;
REGISTERED AS {atmfM4NwAttribute 25};

linkPointerBeh BEHAVIOUR
  DEFINED AS
" This attribute represents the association between an atmLinkTP and an atmLink.  An
atmLinkTP terminates at most 1 atmLink;  an atmLink is terminated by exactly 2
atmLinkTPs, one in each atmSubnetwork.  If this association is not supported by the object
class this attribute shall take on the value NULL.";


maxAssignableEgressBandwidth ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.Integer;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR maxAssignableEgressBandwidthBeh;
REGISTERED AS {atmfM4NwAttribute 26};

maxAssignableEgressBandwidthBeh BEHAVIOUR
  DEFINED AS
" This attribute is used to identify the maximum amount of bandwidth that is available for
assignment in the egress direction on the Object Class.  This may be smaller than the total
available bandwidth.";


maxAssignableIngressBandwidth ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.Integer;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR maxAssignableIngressBandwidthBeh;
REGISTERED AS {atmfM4NwAttribute 27};

maxAssignableIngressBandwidthBeh BEHAVIOUR
  DEFINED AS
" This attribute is used to identify the maximum amount of bandwidth that is available for
assignment in the ingress direction on the Object Class.  This may be smaller than the total
available bandwidth.";
```

```
maxHops ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.IntegerOrNull;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR maxHopsBeh;
REGISTERED AS {atmfM4NwAttribute 28};

maxHopsBeh BEHAVIOUR
  DEFINED AS
" The maxHops attribute is the maximum number of hops between nodes that a connection
may traverse.";


maxNumActiveConnectionsAllowed ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.Integer;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR maxNumActiveConnectionsAllowedBeh;
REGISTERED AS {atmfM4NwAttribute 29};

maxNumActiveConnectionsAllowedBeh BEHAVIOUR
  DEFINED AS
" The maxNumActiveConnectionsAllowed attribute identifies the maximum number of
concurrently active Virtual Channel Connections (VCCs) (for vcLayerNetworkDomain
Links or Link TPs), or Virtual Path Connections (VPCs) (for vpLayerNetworkDomain
Links or Link TPs) that the Link or Link TP may support.";


provisionType ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.ProvisionType;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR provisionTypeBeh;
REGISTERED AS {atmfM4NwAttribute 30};

provisionTypeBeh BEHAVIOUR
  DEFINED AS
" The provisionType attribute indicates whether the route for the associated
subnetworkConnection is specified by the administrator (manual) or determined by the
system (automatic) that may include managing and managed entities of the subnetwork.";


reflectedCTP ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.ObjectInstance;
        MATCHES FOR EQUALITY;
        BEHAVIOUR reflectedCTPBeh;
REGISTERED AS {atmfM4NwAttribute 31};

reflectedCTPBeh BEHAVIOUR
  DEFINED AS
" This attribute represents the association between an atmSubnetworkTP and an
atmNetworkCTP.  An atmSubnetworkTP is reflected by exactly one atmNetworkCTP. ";


relatedAtmCTP ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.ObjectInstance;
        MATCHES FOR EQUALITY;
        BEHAVIOUR relatedAtmCTPBeh;
REGISTERED AS {atmfM4NwAttribute 32};

relatedAtmCTPBeh BEHAVIOUR
  DEFINED AS
" This attribute represents the association between an atmNetworkTTP and an
atmNetworkCTP.";


relatedAtmRoutingProfile ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.PointerOrNull;
        MATCHES FOR EQUALITY;
        BEHAVIOUR relatedAtmRoutingProfileBeh;
REGISTERED AS {atmfM4NwAttribute 33};

relatedAtmRoutingProfileBeh BEHAVIOUR
  DEFINED AS
" This attribute represents the association between an atmTrail or
atmSubnetworkConnection and an atmRoutingProfile.  An atmTrail or
atmSubnetworkConnection may point to at most one atmRoutingProfile.  If this association
is not supported by the object class this attribute shall take on the value NULL. ";


relatedAtmTrail ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.PointerOrNull;
        MATCHES FOR EQUALITY;
        BEHAVIOUR relatedAtmTrailBeh;
REGISTERED AS {atmfM4NwAttribute 34};

relatedAtmTrailBeh BEHAVIOUR
  DEFINED AS
" This attribute represents the association between an atmTrailRequest and an existing
atmTrail.  An atmTrailRequest pertains to at most one atmTrail;  an atmTrail is altered by
zero or more atmTrailRequests.  If this association is not supported by the object class this
attribute shall take on the value NULL. ";


relatedAtmTTP ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.PointerOrNull;
        MATCHES FOR EQUALITY;
        BEHAVIOUR relatedAtmTTPBeh;
REGISTERED AS {atmfM4NwAttribute 35};

relatedAtmTTPBeh BEHAVIOUR
  DEFINED AS
" This attribute represents the association between an atmNetworkCTP and an
atmNetworkTTP.  If this association is not supported by the object class this attribute shall
take on the value NULL.";


relatedLinkConnection ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.PointerOrNull;
        MATCHES FOR EQUALITY;
        BEHAVIOUR relatedLinkConnectionBeh;
REGISTERED AS {atmfM4NwAttribute 36};

relatedLinkConnectionBeh BEHAVIOUR
  DEFINED AS
" This attribute represents the association between an atmSubnetworkTP and the
atmLinkConnection it terminates.  If this association is not supported by the object class
this attribute shall take on the value NULL. ";


relatedLinkTP ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.PointerOrNull;
        MATCHES FOR EQUALITY;
        BEHAVIOUR relatedLinkTPBeh;
REGISTERED AS {atmfM4NwAttribute 37};

relatedLinkTPBeh BEHAVIOUR
  DEFINED AS
" This attribute represents the association between an atmSubnetworkTP and the
atmLinkTP that supports it.  If this association is not supported by the object class this
attribute shall take on the value NULL. ";


relatedSubnetworkConnection ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.PointerOrNull;
        MATCHES FOR EQUALITY;
        BEHAVIOUR relatedSubnetworkConnectionBeh;
REGISTERED AS {atmfM4NwAttribute 38};

relatedSubnetworkConnectionBeh BEHAVIOUR
  DEFINED AS
" This attribute represents the association between an atmSubnetworkTP and the
atmSubnetworkConnection it terminates.  If this association is not supported by the object
class this attribute shall take on the value NULL. ";


relatedTrafficDescriptors ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.PointerOrNull;
        MATCHES FOR EQUALITY;
        BEHAVIOUR relatedTrafficDescriptorsBeh;
REGISTERED AS {atmfM4NwAttribute 39};

relatedTrafficDescriptorsBeh BEHAVIOUR
  DEFINED AS
" This attribute represents the association between an atmNetworkCTP and the Traffic
Descriptor Profile that contains its traffic descriptors.  If this association is not supported
by the object class this attribute shall take on the value NULL. ";


requestActionInfo ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.RequestActionInfo;
        MATCHES FOR EQUALITY;
        BEHAVIOUR requestActionInfoBeh;
REGISTERED AS {atmfM4NwAttribute 40};
```

```
requestActionInfoBeh BEHAVIOUR
  DEFINED AS
" This attribute describes action information associated with a deferred request.  This is the
information that is contained within the deferred M-Action request.";


requestCommittedTime ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.Time;
        MATCHES FOR EQUALITY;
        BEHAVIOUR requestCommittedTimeBeh;
REGISTERED AS {atmfM4NwAttribute 41};

requestCommittedTimeBeh BEHAVIOUR
  DEFINED AS
" This attribute describes the time at which the NML commits to performing the action.";


requestStatus ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.RequestStatus;
        MATCHES FOR EQUALITY;
        BEHAVIOUR requestStatusBeh;
REGISTERED AS {atmfM4NwAttribute 42};

requestStatusBeh BEHAVIOUR
  DEFINED AS
" This attribute is used to show the status of the atmTrailRequest.  Takes on values as:  not
scheduled, scheduled, suspended, user canceled, being handled, or completed.";


requestType ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.RequestType;
        MATCHES FOR EQUALITY;
        BEHAVIOUR requestTypeBeh;
REGISTERED AS {atmfM4NwAttribute 43};

requestTypeBeh BEHAVIOUR
  DEFINED AS
" This attribute describes the type of request.  Set to:  setup, modify, release, addTps, or
removeTps.";


restorableIndicator ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.Boolean;
        MATCHES FOR EQUALITY;
        BEHAVIOUR restorableIndicatorBeh;
REGISTERED AS {atmfM4NwAttribute 44};

restorableIndicatorBeh BEHAVIOUR
  DEFINED AS
" This attribute is used to configure the connection as restorable or not-restorable.";


restorationMode ATTRIBUTE
```

```
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.RestorationMode;
        MATCHES FOR EQUALITY;
        BEHAVIOUR restorationModeBeh;
REGISTERED AS {atmfM4NwAttribute 45};

restorationModeBeh BEHAVIOUR
  DEFINED AS
" This attribute is used to configure the restoration mode of a link as:  unavailable for
routing and re-routing;  available for routing and not re-routing;  available for re-routing
and not routing;  or available for both routing and re-routing.";


retainConnectionsIndicator ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.Boolean;
        MATCHES FOR EQUALITY;
        BEHAVIOUR retainConnectionsIndicatorBeh;
REGISTERED AS {atmfM4NwAttribute 46};

retainConnectionsIndicatorBeh BEHAVIOUR
  DEFINED AS
" This attribute is used to show that some of the atmSubnetworkConnections and
atmLinkConnections supporting the atmTrail may be retained when the atmTrail is released.
If the value of this attribute is TRUE, the connection objects with their respective
retainIndicator attributes set to TRUE will be retained after the atmTrail is released.";


retainResourcesIndicator ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.Boolean;
        MATCHES FOR EQUALITY;
        BEHAVIOUR retainResourcesIndicatorBeh;
REGISTERED AS {atmfM4NwAttribute 47};

retainResourcesIndicatorBeh BEHAVIOUR
  DEFINED AS
" This attribute is used to show that this connection supporting an atmTrail may be retained
when the atmTrail is released.  If the value of the retainConnectionsIndicator attribute of the
supported atmTrail is TRUE, this connection object with will be retained after the atmTrail
is released only if the retainResourcesIndicator attribute is set to TRUE.";


routeDescriptionList ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.RouteDescriptionList;
        MATCHES FOR EQUALITY;
        BEHAVIOUR routeDescriptionListBeh;
REGISTERED AS {atmfM4NwAttribute 48};

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


serverTTPList ATTRIBUTE
```

```
            WITH ATTRIBUTE SYNTAX AtmNwMIBMod.ObjectList;
            MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
            BEHAVIOUR serverTTPListBeh;
REGISTERED AS {atmfM4NwAttribute 49};

serverTTPListBeh BEHAVIOUR
  DEFINED AS
" This attribute represents the association between an atmLinkTP and the underlying or
server TTP instances. If applicable, the atmServerTTPList points to vpLayer TTP instances
if the atmLinkTP is at the vcLayer, or the underlying tcAdaptorTTPBid instances if the
atmLinkTP is at the vpLayer.";


supportedLinkTPList ATTRIBUTE
            WITH ATTRIBUTE SYNTAX AtmNwMIBMod.ObjectList;
            MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
            BEHAVIOUR supportedLinkTPListBeh;
REGISTERED AS {atmfM4NwAttribute 50};

supportedLinkTPListBeh BEHAVIOUR
  DEFINED AS
" This attribute represents the association between an atmSubnetwork and the atmLinkTPs
it supports.";


supportingUNIorNNI ATTRIBUTE
            WITH ATTRIBUTE SYNTAX AtmNwMIBMod.ObjectList;
            MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
            BEHAVIOUR supportingUNIorNNIBeh;
REGISTERED AS {atmfM4NwAttribute 51};

supportingUNIorNNIBeh BEHAVIOUR
  DEFINED AS
" This attribute represents the association between an atmLinkTP and its supporting UNI or
NNI instances.";


totalEgressBandwidth ATTRIBUTE
   WITH ATTRIBUTE SYNTAX AtmNwMIBMod.Integer;
   MATCHES FOR EQUALITY, ORDERING;
   BEHAVIOUR totalEgressBandwidthBeh;
REGISTERED AS {atmfM4NwAttribute 52};

totalEgressBandwidthBeh BEHAVIOUR
  DEFINED AS
"This attribute identifies the maximum egress bandwidth for an ATM Link or Link TP.”;


totalIngressBandwidth ATTRIBUTE
   WITH ATTRIBUTE SYNTAX AtmNwMIBMod.Integer;
   MATCHES FOR EQUALITY, ORDERING;
   BEHAVIOUR totalIngressBandwidthBeh;
REGISTERED AS {atmfM4NwAttribute 53};
```

```
totalIngressBandwidthBeh BEHAVIOUR
  DEFINED AS
"This attribute identifies the maximum ingress bandwidth for an ATM Link or Link TP. ";


virtualId ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.VirtualId;
        MATCHES FOR EQUALITY;
        BEHAVIOUR virtualIdBeh;
REGISTERED AS {atmfM4NwAttribute 55};

virtualIdBeh BEHAVIOUR
  DEFINED AS
" This attribute describes VPI value for an atmNetworkCTP at the vpLayer, or the VPI/VCI
for an atmNetworkCTP at the vcLayer";


vpiOrVciRange ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.VpiOrVciRange;
        MATCHES FOR EQUALITY, ORDERING;
        BEHAVIOUR vpiOrVciRangeBeh;
REGISTERED AS {atmfM4NwAttribute 56};

vpiOrVciRangeBeh BEHAVIOUR
  DEFINED AS
" The vpiOrVciRange attribute describes the virtual ID range (VCIs in the
vcLayerNetworkDomain or VPIs in the vpLayerNetworkDomain) that may be used for
atmLinkConnection associated with the atmLink or atmLinkTP that points to the
atmNetworkAccessProfile.";


z-LinkTP ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.ObjectList;
        MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR z-LinkTPBeh;
REGISTERED AS {atmfM4NwAttribute 58};

z-LinkTPBeh BEHAVIOUR
  DEFINED AS
" This attribute represents the association between an atmLink and an existing atmLinkTP.
An atmLink links exactly 2 atmLinkTPs;  an atmLinkTP is linked by zero or one atmLinks.
This attribute points to one of the two atmLinkTPs. In this case the z-LinkTP points to a
single atmLinkTP object instance.  Usually this attribute points to a single termination
point.  However, in cases where the NE view is also supported and the atmLink is
supported by more than one server trail, this attribute may point to more than one
termination point. ";


z-TPInstance ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.PointerOrNull;
        MATCHES FOR EQUALITY;
        BEHAVIOUR z-TPInstanceBeh;
REGISTERED AS {atmfM4NwAttribute 59};
```

```
z-TPInstanceBeh BEHAVIOUR
  DEFINED AS
" The z-TPInstance is used to indicate one of the endpoints of an atmLinkConnection.";


z-TPList ATTRIBUTE
        WITH ATTRIBUTE SYNTAX AtmNwMIBMod.ObjectList;
        MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
        BEHAVIOUR z-TPListBeh;
REGISTERED AS {atmfM4NwAttribute 60};

z-TPListBeh BEHAVIOUR
  DEFINED AS
" This attribute represents the association between a multipoint connection and the
underlying termination points.  For point to point connections the a single entry in the z-
TPList is used to indicate one of the endpoints.  Multiple entries in the z-TPList are used to
represent the secondary end points of broadcast (point-to-multipoint), merge (multipoint-to-
point), and composite connections.  The a-TPInstance identifies the primary endpoint.
Only the z-TPList is used identify all end points in a multipoint-to-multipoint connection
(there is no primary end point). ";
```

Name Bindings

```
alarmSeverityAssignmentProfile-vcLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS "ITU-T M.3100:1995":
        alarmSeverityAssignmentProfile AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vcLayerNetworkDomain
  AND SUBCLASSES;
  WITH ATTRIBUTE "ITU-T M.3100:1995":alarmSeverityAssignmentProfileId;
CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 1};


alarmSeverityAssignmentProfile-vpLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS "ITU-T M.3100:1995":
        alarmSeverityAssignmentProfile AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vpLayerNetworkDomain
  AND SUBCLASSES;
  WITH ATTRIBUTE "ITU-T M.3100:1995":alarmSeverityAssignmentProfileId;
CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 2};


atmLink-vcLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS atmLink AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vcLayerNetworkDomain
  AND SUBCLASSES;
  WITH ATTRIBUTE atmLinkId;
DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 3};


atmLink-vpLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS atmLink AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vpLayerNetworkDomain
  AND SUBCLASSES;
  WITH ATTRIBUTE atmLinkId;
DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 4};


atmLinkConnection-atmLink NAME BINDING
  SUBORDINATE OBJECT CLASS atmLinkConnection
  AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS atmLink AND SUBCLASSES;
  WITH ATTRIBUTE atmLinkConnectionId;
DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 5};


atmLinkTP-vcLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS atmLinkTP AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vcLayerNetworkDomain AND
SUBCLASSES;
  WITH ATTRIBUTE atmLinkTPId;
CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 6};


atmLinkTP-vpLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS atmLinkTP AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vpLayerNetworkDomain AND
SUBCLASSES;
  WITH ATTRIBUTE atmLinkTPId;
CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 7};


atmNetworkAccessProfile-tcAdaptorTTPBidirectional NAME BINDING
  SUBORDINATE OBJECT CLASS atmNetworkAccessProfile AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS “ATMF M4 NEView”:
        tcAdaptorTTPBidirectional AND SUBCLASSES;
  WITH ATTRIBUTE atmNetworkAccessProfileId;
  CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 8};


atmNetworkAccessProfile-atmNetworkTTP NAME BINDING
  SUBORDINATE OBJECT CLASS atmNetworkAccessProfile AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS atmNetworkTTP AND SUBCLASSES;
  WITH ATTRIBUTE atmNetworkAccessProfileId;
  CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 9};


atmNetworkAccessProfile-vcLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS atmNetworkAccessProfile AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vcLayerNetworkDomain AND
        SUBCLASSES;
  WITH ATTRIBUTE atmNetworkAccessProfileId;
  CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 10};


atmNetworkAccessProfile-vpLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS atmNetworkAccessProfile AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vpLayerNetworkDomain AND
        SUBCLASSES;
  WITH ATTRIBUTE atmNetworkAccessProfileId;
  CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 11};


atmNetworkAccessProfile-vpTTPBidirectional NAME BINDING
  SUBORDINATE OBJECT CLASS atmNetworkAccessProfile AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS “ATMF M4 NEView”:
        vpTTPBidirectional AND SUBCLASSES;
  WITH ATTRIBUTE atmNetworkAccessProfileId;
  CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 12};


atmNetworkCTP-vcLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS atmNetworkCTP AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vcLayerNetworkDomain
  AND SUBCLASSES;
  WITH ATTRIBUTE atmNetworkCTPId;
  CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        DELETES-CONTAINED-OBJECTS;
```

```
REGISTERED AS {atmfM4NwNameBinding 13};


atmNetworkCTP-vpLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS atmNetworkCTP AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vpLayerNetworkDomain
  AND SUBCLASSES;
  WITH ATTRIBUTE atmNetworkCTPId;
  CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 14};


atmNetworkTrafficDescriptorProfile-atmSubnetwork NAME BINDING
  SUBORDINATE OBJECT CLASS atmNetworkTrafficDescriptorProfile
  AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS atmSubnetwork AND SUBCLASSES;
  WITH ATTRIBUTE atmNetworkTrafficDescriptorProfileId;
  CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 15};


atmNetworkTTP-vcLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS atmNetworkTTP AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vcLayerNetworkDomain
  AND SUBCLASSES;
  WITH ATTRIBUTE atmNetworkTTPId;
  CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 16};


atmNetworkTTP-vpLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS atmNetworkTTP AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vpLayerNetworkDomain
  AND SUBCLASSES;
  WITH ATTRIBUTE atmNetworkTTPId;
  CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 17};
```

```
atmRoutingProfile-atmSubnetwork NAME BINDING
  SUBORDINATE OBJECT CLASS atmRoutingProfile AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS atmSubnetwork
  AND SUBCLASSES;
  WITH ATTRIBUTE atmRoutingProfileId;
CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 18};


atmSubnetwork-vcLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS atmSubnetwork AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vcLayerNetworkDomain
  AND SUBCLASSES;
  WITH ATTRIBUTE “ITU-T M.3100:1995”: networkId;
CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 19};


atmSubnetwork-vpLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS atmSubnetwork AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vpLayerNetworkDomain
  AND SUBCLASSES;
  WITH ATTRIBUTE “ITU-T M.3100:1995”: networkId;
CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 20};


atmSubnetworkConnection-atmSubnetwork NAME BINDING
  SUBORDINATE OBJECT CLASS atmSubnetworkConnection AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS atmSubnetwork AND SUBCLASSES;
  WITH ATTRIBUTE atmSubnetworkConnectionId;
DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 21};


atmSubnetworkTP-atmSubnetwork NAME BINDING
  SUBORDINATE OBJECT CLASS atmSubnetworkTP AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS atmSubnetwork AND SUBCLASSES;
  WITH ATTRIBUTE atmSubnetworkTPId;
CREATE
```

```
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 22};


atmTrail-vcLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS atmTrail AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vcLayerNetworkDomain
  AND SUBCLASSES;
  WITH ATTRIBUTE atmTrailId;
DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 23};


atmTrail-vpLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS atmTrail AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vpLayerNetworkDomain
  AND SUBCLASSES;
  WITH ATTRIBUTE atmTrailId;
  DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 24};


atmTrailRequest-vcLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS atmTrailRequest AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vcLayerNetworkDomain
  AND SUBCLASSES;
  WITH ATTRIBUTE atmTrailRequestId;
DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 25};



atmTrailRequest-vpLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS atmTrailRequest AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vpLayerNetworkDomain
  AND SUBCLASSES;
  WITH ATTRIBUTE atmTrailRequestId;
DELETE
        DELETES-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 26};



eventForwardingDiscriminator-vcLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS
        "ITU-T Rec. X.721 | ISO/IEC 10165-2":eventForwardingDiscriminator;
  NAMED BY SUPERIOR OBJECT CLASS vcLayerNetworkDomain
  AND SUBCLASSES;
  WITH ATTRIBUTE "ITU-T Rec. X.721 | ISO/IEC 10165-2":discriminatorId;
  CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 27};


eventForwardingDiscriminator-vpLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS
        "ITU-T Rec. X.721 | ISO/IEC 10165-2":eventForwardingDiscriminator;
  NAMED BY SUPERIOR OBJECT CLASS vpLayerNetworkDomain
  AND SUBCLASSES;
  WITH ATTRIBUTE "ITU-T Rec. X.721 | ISO/IEC 10165-2":discriminatorId;
  CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 28};


log-vcLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS
        "ITU-T Rec. X.721 | ISO/IEC 10165-2":log AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vcLayerNetworkDomain
  AND SUBCLASSES;
  WITH ATTRIBUTE "ITU-T Rec. X.721 | ISO/IEC 10165-2":logId;
  CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 29};


log-vpLayerNetworkDomain NAME BINDING
  SUBORDINATE OBJECT CLASS
        "ITU-T Rec. X.721 | ISO/IEC 10165-2":log AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS vpLayerNetworkDomain
  AND SUBCLASSES;
  WITH ATTRIBUTE "ITU-T Rec. X.721 | ISO/IEC 10165-2":logId;
  CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 30};


vcLayerNetworkDomain-networkR1 NAME BINDING
  SUBORDINATE OBJECT CLASS vcLayerNetworkDomain AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS "ITU-T M.3100:1995": networkR1
  AND SUBCLASSES;
  WITH ATTRIBUTE “ITU-T M.3100:1995”: networkId;
  CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 31};


vpLayerNetworkDomain-networkR1 NAME BINDING
  SUBORDINATE OBJECT CLASS vpLayerNetworkDomain AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS "ITU-T M.3100:1995": networkR1
  AND SUBCLASSES;
  WITH ATTRIBUTE “ITU-T M.3100:1995”: networkId;
  CREATE
        WITH-REFERENCE-OBJECT,
        WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {atmfM4NwNameBinding 32};
```

Actions

```
addTpsToSubnetworkConnection ACTION
    BEHAVIOUR addTpsToSubnetworkConnectionBeh;
    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX AtmNwMIBMod.AddTpsInformation;
    WITH REPLY SYNTAX AtmNwMIBMod.AddTpsReply;
REGISTERED AS {atmfM4NwAction 1};

addTpsToSubnetworkConnectionBeh BEHAVIOUR
    DEFINED AS
```

"This ACTION adds atmSubnetworkTPs to an existing multipoint subnetwork connection in the vcLayerNetworkDomain or vpLayerNetworkDomain. The subnetworkTPs are identified either directly or indirectly with a set of descriptors providing VPI/VCI, etc. An error condition is raised if the termination points are incorrect, already used, do not have matching traffic descriptors, or if atmSubnetwork is unable to provide sufficient bandwidth. The request is considered successful if at least one of the legs is added.

Input: Id of endpoint or descriptors of the endpoint: interfaceId (serverTTPId), and VPI/VCI. Optionally a list of identifiers for objects (atmSubnetwork, atmSubnetworkConnection, atmLinkConnection, other) to be used for route selection.

Output: Ids of the atmSubnetworkConnection and added atmSubnetworkTPs and list of end-points not added.

Errors: protocol-specific addressing errors, incorrectTerminationPoints, subnetworkTPConnected, non-matchingDescriptors, operationFails.

Results: Creation or association of atmSubnetworkTPs to the atmSubnetworkConnection.

A response indicating partial addition (indicating both successful and unsuccessful end points) will be generated in cases where addition of endpoints to a multipoint atmSubnetworkConnection neither completely succeeds (all endpoints additions succeed), nor completely fails (no endpoint additions succeed).
";

```
addTpsToTrail ACTION
    BEHAVIOUR addTpsToTrailBeh;
    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX AtmNwMIBMod.AddTpsInformation;
    WITH REPLY SYNTAX AtmNwMIBMod.AddTpsReply;
REGISTERED AS {atmfM4NwAction 2};

addTpsToTrailBeh BEHAVIOUR
    DEFINED AS
```

"This ACTION adds, or schedules the addition of TTPs to an existing multipoint trail connection in the layer network. The TTPs are identified directly or indirectly by an interface within which the vcLayerNetworkDomain or vpLayerNetworkDomain manager selects a point. The indirect case may use an optional descriptor providing VPI/VCI. An error condition is raised if the atmTrail is incorrect, the termination points are incorrect, already used, do not have matching traffic descriptors of the existing trail, or if

```
vcLayerNetworkDomain or vpLayerNetworkDomain is unable to provide sufficient
bandwidth.

Input:  Id of atmTrail, descriptors of the endpoint:  interfaceId (serverTTPId), and VPI/
VCI. .  If deferred request: requestServiceTime.

Output: Ids of the atmTrail and added TTPs.

Errors:  protocol-specific addressing errors, incorrectTerminationPoints, TTPDisabled,
TTPLocked, TTPConnected, non-matchingDescriptors, operationFails.

Results:  Creation or association of TTPs to the atmTrail.

A response indicating partial addition (indicating both successful and unsuccessful end
points) will be generated in cases where addition of endpoints to a multipoint atmTrail
neither completely succeeds (all endpoints additions succeed), nor completely fails (no
endpoint additions succeed).
";


cancelTrailRequest ACTION
    BEHAVIOUR cancelTrailRequestBeh;
    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX AtmNwMIBMod.CancelRequestInformation;
    WITH REPLY SYNTAX AtmNwMIBMod.CancelRequestReply;
REGISTERED AS {atmfM4NwAction 3};

cancelTrailRequestBeh BEHAVIOUR
    DEFINED AS
"This ACTION cancels a deferred trail setup, modification, release, addTps, or removeTps
request.  This ACTION will cause the requestStatus attribute of the atmTrailRequest object
to change to user canceled.

Input:  Identifier of the atmTrailRequest to be canceled.

Output:  Id of the canceled atmTrailRequest.

Errors:  noSuchRequest, requestNotCanceled

Results:  Cancels the atmTrailRequest.
";


modifyLinkConnection ACTION
    BEHAVIOUR modifyLinkConnectionBeh;
    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX AtmNwMIBMod.ModifyInformation;
    WITH REPLY SYNTAX AtmNwMIBMod.ModifyReply;
REGISTERED AS {atmfM4NwAction 4};

modifyLinkConnectionBeh BEHAVIOUR
    DEFINED AS
"This ACTION modifies a point-to-point connection between two connected
subnetworkTPs in the two linked atmSubnetworks.  Modifies the traffic descriptors or
```

```
QOS of the identified atmLinkConnection.  An error condition is raised if the identifier is
incorrect, or if atmLink is unable to provide sufficient bandwidth or is locked.  Use of the
modify action may result in a brief interruption of service.

Input: Identifier of atmLinkConnection, trafficDescriptors and/or QOS.
Output:  Ids atmLinkConnection and confirmation

Errors: incorrectLinkConnectionId, atmLinkLocked, insufficientBandwidth,
operationFails.

Results:  Modification of the atmLinkConnection.";


modifySubnetworkConnection ACTION
    BEHAVIOUR modifySubnetworkConnectionBeh;
    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX AtmNwMIBMod.ModifyInformation;
    WITH REPLY SYNTAX AtmNwMIBMod.ModifyReply;
REGISTERED AS {atmfM4NwAction 5};

modifySubnetworkConnectionBeh BEHAVIOUR
    DEFINED AS
"This ACTION modifies a point-to-point or a multipoint connection between
subnetworkTPs in the atmSubnetwork.  The subnetworkConnection is identified using its
identifier.  This ACTION modifies the traffic descriptors or QOS of the
subnetworkConnection.  An error condition is raised if the identifier is incorrect, or if
atmSubnetwork is unable to provide sufficient bandwidth.  Use of the modify action may
result in a brief interruption of service.

Input: atmSubnetworkConnection ID, trafficDescriptors, and/or QOS.
Output:  atmSubnetworkConnectionId and confirmation.

Errors: incorrectAtmSubnetworkConnectionId, operationFails.

Results:  Modification of identified atmSubnetworkConnection.
";


modifyTrail ACTION
    BEHAVIOUR modifyTrailBeh;
    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX AtmNwMIBMod.ModifyInformation;
    WITH REPLY SYNTAX AtmNwMIBMod.ModifyReply;
REGISTERED AS {atmfM4NwAction 6};

modifyTrailBeh BEHAVIOUR
    DEFINED AS
"This ACTION modifies, or schedules the modification of the parameters of a point-to-
point or a multipoint connection between connected TTPs in the layer network.  The trail is
identified directly using the trail id.  This ACTION will allow modification of traffic
descriptors, and QOS.  An error condition is raised if the atmTrailId is incorrect or if
vcLayerNetworkDomain or vpLayerNetworkDomain is unable to provide sufficient
bandwidth.  Use of the modify action may result in a brief interruption of service.
```

```
Input:  Trail Id, trafficDescriptors, QOS or TTPs.  If deferred request: requestServiceTime.

Output:  Ids of atmTrail and modify confirmed or Id of the deferred atmTrailRequest for
deferred request.

Errors: incorrectTrailId, incorrectTerminationPoints, TTPDisabled, TTPLocked,
TTPConnected,  operationFails.

Results:  Modification of atmTrail.
";


releaseLinkConnection ACTION
     BEHAVIOUR releaseLinkConnectionBeh;
     MODE    CONFIRMED;
     WITH INFORMATION SYNTAX AtmNwMIBMod.ReleaseInformation;
     WITH REPLY SYNTAX AtmNwMIBMod.ReleaseReply;
REGISTERED AS {atmfM4NwAction 7};

releaseLinkConnectionBeh BEHAVIOUR
     DEFINED AS
" This ACTION releases a point-to-point connection between subnetworkTPs in each of the
linked atmSubnetwork.  The atmLinkConnection or atmSubnetworkTPs are identified
directly.  The matched atmLinkConnection and associated atmSubnetworkTPs are released.
An error condition is raised if a match cannot be made.

Input:  Identifier(s) of the atmLinkConnection or atmSubnetworkTPs.

Output:  Ids of the released atmLinkConnection and atmSubnetworkTPs.

Errors:  protocol-specific addressing errors, incorrectTerminationPoints,
incorrectLinkConnection.

Results:  Release of atmLinkConnection and an associated atmSubnetworkTPs.";


releaseSubnetworkConnection ACTION
     BEHAVIOUR releaseSubnetworkConnectionBeh;
     MODE    CONFIRMED;
     WITH INFORMATION SYNTAX AtmNwMIBMod.ReleaseInformation;
     WITH REPLY SYNTAX AtmNwMIBMod.ReleaseReply;
REGISTERED AS {atmfM4NwAction 8};

releaseSubnetworkConnectionBeh BEHAVIOUR
     DEFINED AS
"This ACTION releases a point-to-point or a multipoint connection between
subnetworkTPs in the atmSubnetwork.  The atmSubnetworkConnection or
atmSubnetworkTPs are identified directly.  The matched atmSubnetworkConnection and
associated atmSubnetworkTPs are released.  An error condition is raised if a match cannot
be made. The retain connections indicator may be used to retain atmLinkConnections and
atmSubnetworkConnections after the atmSubnetworkConnection is released.

Input:  Identifier(s) of the atmSubnetworkConnection or atmSubnetworkTPs, and retain
connections indicator.
```

Output: Ids of the released atmSubnetworkConnection and atmSubnetworkTPs.

Errors: protocol-specific addressing errors, incorrectTerminationPoints, incorrectSubnetworkConnection.

Results: Release of atmSubnetworkConnection and an associated atmSubnetworkTPs.

A response indicating partial release (indicating both successful and unsuccessful end points) will be generated in cases where a multipoint atmSubnetworkConnection release attempt neither completely succeeds (all endpoints released), nor completely fails (zero endpoints released).
";

```
releaseTrail ACTION
    BEHAVIOUR releaseTrailBeh;
    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX AtmNwMIBMod.ReleaseInformation;
    WITH REPLY SYNTAX AtmNwMIBMod.ReleaseReply;
REGISTERED AS {atmfM4NwAction 9};

releaseTrailBeh BEHAVIOUR
    DEFINED AS
```

"This ACTION releases, or schedules the release of a point-to-point or a multipoint trail between TTPs in the layer network. The atmTrail is identified directly. The matched atmTrail and associated TTPs are released. An error condition is raised if a match cannot be made. The retain connections indicator may be used to retain atmLinkConnections and atmSubnetworkConnections after the trail is released.

Input: Identifier of the atmTrail, retain connections indicator. . If deferred request: requestServiceTime.

Output: Ids of the released atmTrail and TTPs. or Id of the deferred atmTrailRequest for deferred request

Errors: protocol-specific addressing errors, incorrectTerminationPoints, TTPNotConnected, incorrectTrails.

Results: Release of atmTrail and an associated TTPs.

A response indicating partial release (indicating both successful and unsuccessful end points) will be generated in cases where a multipoint atmTrail release attempt neither completely succeeds (all endpoints released), nor completely fails (zero endpoints released).
";

```
removeTpsFromSubnetworkConnection ACTION
    BEHAVIOUR removeTpsFromSubnetworkConnectionBeh;
    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX AtmNwMIBMod.RemoveTpsInformation;
    WITH REPLY SYNTAX AtmNwMIBMod.RemoveTpsReply;
REGISTERED AS {atmfM4NwAction 10};
```

```
removeTpsFromSubnetworkConnectionBeh BEHAVIOUR
    DEFINED AS
"This ACTION releases atmSubnetworkTPs from a multipoint atmSubnetworkConnection
in the vcLayerNetworkDomain or vpLayerNetworkDomain.  The
atmSubnetworkConnection and the atmSubnetworkTPs are identified directly.  The
atmSubnetworkConnection and associated atmSubnetworkTPs matched and the
atmSubnetworkTPs are released.  An error condition is raised if a match cannot be made.
The retain connections indicator may be used to retain atmLinkConnections and
atmSubnetworkConnections after atmSubnetworkTPs are removed from the
atmSubnetworkConnection. .  The request is considered successful if at least one of the
legs is removed.

Input:  Identifier(s) of the atmSubnetworkConnection and atmSubnetworkTPs, and retain
connections indicator.

Output:  Ids of the released atmSubnetworkConnection and atmSubnetworkTPs.

Errors:  protocol-specific addressing errors, incorrectTerminationPoints,
incorrectSubnetworkConnection.

Results:  Release of the given atmSubnetworkTPs from the atmSubnetworkConnection.

A response indicating partial removal (indicating both successful and unsuccessful end
points) will be generated in cases where removal of endpoints from a multipoint
atmSubnetworkConnection neither completely succeeds (all endpoints removals succeed),
nor completely fails (no endpoint removals succeed).
";


removeTpsFromTrail ACTION
    BEHAVIOUR removeTpsFromTrailBeh;
    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX AtmNwMIBMod.RemoveTpsInformation;
    WITH REPLY SYNTAX AtmNwMIBMod.RemoveTpsReply;
REGISTERED AS {atmfM4NwAction 11};

removeTpsFromTrailBeh BEHAVIOUR
    DEFINED AS
"This ACTION releases, or schedules the release of TTPs from a multipoint trail in the
layer network.  The atmTrail and the TTPs are identified directly.  The atmTrail and
associated TTPs are matched and the TTPs are released.  An error condition is raised if a
match cannot be made.  The retain connections indicator may be used to retain
atmLinkConnections and atmSubnetworkConnections after TTP is removed from the trail.

Input:  Identifier(s) of the atmTrail and TTPs, and retain connections indicator.  If deferred
request: requestServiceTime.

Output:  Ids of the released atmTrail and TTPs. or Id of the deferred atmTrailRequest for
deferred request

Errors:  protocol-specific addressing errors, incorrectTerminationPoints,
TTPNotConnected, incorrectTrails.
```

Results: Release of the given TTPs from the atmTrail.

A response indicating partial removal (indicating both successful and unsuccessful end points) will be generated in cases where removal of endpoints from a multipoint atmTrail neither completely succeeds (all endpoints removals succeed), nor completely fails (no endpoint removals succeed).
";

```
setupLink ACTION
    BEHAVIOUR setupLinkBeh;
    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX AtmNwMIBMod.SetupLinkInfo;
    WITH REPLY SYNTAX AtmNwMIBMod.SetupReply;
REGISTERED AS {atmfM4NwAction 12};

setupLinkBeh BEHAVIOUR
    DEFINED AS
```

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

Input: atmlinkTPs or descriptors of the endpoints: interfaceId (serverTTPId or atmLinkTPId), available bandwidth, and atmNetworkAccessProfilePointer or VPI range, VCI range, and maximum bandwidth.

Output: Ids of new atmLink and associated atmLinkTPs and atmNetworkAccessProfiles.

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

Results: Creation of atmLink and an associated atmLinkTPs and atmNetworkAccessProfiles";

```
setupLinkConnection ACTION
    BEHAVIOUR setupLinkConnectionBeh;
    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX AtmNwMIBMod.SetupInformation;
    WITH REPLY SYNTAX AtmNwMIBMod.SetupReply;
REGISTERED AS {atmfM4NwAction 13};

setupLinkConnectionBeh BEHAVIOUR
    DEFINED AS
```

"This ACTION sets up a point-to-point connection between two non-connected subnetworkTPs in the each of the linked atmSubnetworks. The subnetworkTPs are identified either directly or indirectly with a set of descriptors providing VPI/VCI, traffic descriptors, QOS, etc. An error condition is raised if the termination points are incorrect, already used, do not have matching traffic descriptors, or if atmLink is unable to provide sufficient bandwidth or is locked.

Objects (such as atmSubnetwork, atmLink, or managedElement, etc) may be referenced by the routeDescriptionList as being excluded, mandatory, or preferred. If an object is described as mandatory it must be used in setting up a new connection. An attempt must be made during setup to include an object described as preferred. An excluded object must not be used in a connection.

Connection type objects (such as atmTrail, atmSubnetworkConnection, etc) may be referenced by the routeDescriptionList as same route or diverse route. A new connection being created shall follow the same route as a sameRoute referenced object. A new connection must follow a different route than a referenced object referred to as diverseRoute.

Input: Id of endpoints or descriptors of the endpoints: interfaceId (serverTTPId, atmLink, atmLinkTP, or atmSubnetwork), VPI, VCI, trafficDescriptors, and QOS. The connectionType for an atmLinkConnection must be point-to-point.

Output: Ids of new atmLinkConnection and associated atmSubnetworkTPs. (+ atmLinkId)

Errors: protocol-specific addressing errors, incorrectTerminationPoints, atmLinkLocked, atmSubnetworkTPConnected, non-matchingDescriptors, operationFails.

Results: Creation of atmSubnetworkTPs and an associated atmLinkConnection.
";

```
setupSubnetworkConnection ACTION
    BEHAVIOUR setupSubnetworkConnectionBeh;
    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX AtmNwMIBMod.SetupInformation;
    WITH REPLY SYNTAX AtmNwMIBMod.SetupReply;
REGISTERED AS {atmfM4NwAction 14};

setupSubnetworkConnectionBeh BEHAVIOUR
    DEFINED AS
```

"This ACTION sets up a point-to-point or a multipoint connection between non-connected subnetworkTPs in the atmSubnetwork. The subnetworkTPs are identified either directly or indirectly with an interface identifier or a set of descriptors providing VPI/VCI, traffic descriptors, QOS, etc. An error condition is raised if the termination points are incorrect, already used, do not have matching traffic descriptors, or if atmSubnetwork is unable to provide sufficient bandwidth. For multipoint, the request is successful if at least one of the legs is established.

The routing information may be either explicitly stated in the ACTION or the ACTION can point to an existing instance of the atmRoutingProfile object class. The routingDetails are a set of topological routing constraints that can be applied to a connection during setup. The routingDetails are a list of objects (such as Links, Subnetworks, existing connections) and their use in routing (exclude, mandatory, preferred, same route, diverse route) and the maximum number of hops. The connection shall not be established if the routing criteria cannot be met. If maxHops is specified, the connection shall not be established if the maximum number of hops is exceeded.

Objects (such as atmSubnetwork, atmLink, or managedElement, etc.) may be referenced by the routeDescriptionList as being excluded, mandatory, or preferred. If an object is described as mandatory it must be used in setting up a new connection. An attempt must be made during setup to include an object described as preferred. An excluded object must not be used in a connection.

Connection type objects (such as atmTrail, atmSubnetworkConnection, etc) may be referenced by the routeDescriptionList as same route or diverse route. A new connection being created shall follow the same route as a sameRoute referenced object. A new connection must follow a different route than a referenced object referred to as diverseRoute.

Input: Connection Type (point-to-point, multicast, merge, multicast/merge, full multipoint), subnetworkTPs or descriptors of the endpoints: interfaceId (serverTTPId or atmLinkTPId), VPI, VCI, trafficDescriptors, and QOS. Optionally a list of identifiers for objects (atmSubnetwork, atmSubnetworkConnection, atmLinkConnection, other) to be used for route selection.

Output: Ids of new atmSubnetworkConnection and associated atmSubnetworkTPs or related CTPs.

Errors: protocol-specific addressing errors, incorrectTerminationPoints, subnetworkTPConnected, non-matchingDescriptors, operationFails.

Results: Creation of atmSubnetworkTPs and an associated atmSubnetworkConnection

A response indicating partial setup (indicating both successful and unsuccessful end points) will be generated in cases where a multipoint atmSubnetworkConnection set up attempt neither completely succeeds (all endpoints succeed), nor completely fails (zero or one endpoints succeed).
";

```
setupTrail ACTION
    BEHAVIOUR setupTrailBeh;
    MODE    CONFIRMED;
    WITH INFORMATION SYNTAX AtmNwMIBMod.SetupInformation;
    WITH REPLY SYNTAX AtmNwMIBMod.SetupReply;
REGISTERED AS {atmfM4NwAction 15};

setupTrailBeh BEHAVIOUR
    DEFINED AS
```

"This ACTION sets up, or schedules the setup of a point-to-point or a multipoint connection between non-connected TTPs in the layer network. The TTPs are identified directly or indirectly by an interface within which the vcLayerNetworkDomain or vpLayerNetworkDomain selects a point. The indirect case may use a set of optional descriptors providing VPI/VCI, traffic descriptors, QOS, etc. An error condition is raised if the termination points are incorrect, already used, do not have matching traffic descriptors, or if vcLayerNetworkDomain or vpLayerNetworkDomain is unable to provide sufficient bandwidth.

The routing information may be either explicitly stated in the ACTION or the ACTION can point to an existing instance of the atmRoutingProfile object class. The routingDetails are a set of topological routing constraints that can be applied to a trail during setup. The routingDetails are a list of objects (such as Links, Subnetworks, existing connections) and their use in routing (exclude, mandatory, preferred, same route, diverse route) and the maximum number of hops. The connection shall not be established if the routing criteria cannot be met. If maxHops is specified, the connection shall not be established if the maximum number of hops is exceeded.

Objects (such as atmSubnetwork, atmLink, or managedElement, etc) may be referenced by the routeDescriptionList as being excluded, mandatory, or preferred. If an object is described as mandatory it must be used in setting up a new connection. An attempt must be made during setup to include an object described as preferred. An excluded object must not be used in a connection.

Connection type objects (such as atmTrail, atmSubnetworkConnection, etc) may be referenced by the routeDescriptionList as same route or diverse route. A new connection being created shall follow the same route as a sameRoute referenced object. A new connection must follow a different route than a referenced object referred to as diverseRoute.

Input: Connection Type (point-to-point, multicast, merge, multicast/merge, full multipoint), Descriptors of the endpoints: interfaceId (serverTTPId), VPI, VCI, trafficDescriptors, and QOS. Optionally a list of identifiers for objects (atmSubnetwork, atmSubnetworkConnection, atmLinkConnection, other) to be used for route selection. If deferred request: requestServiceTime.

Output: Ids of new atmTrail and associated TTPs or Id of the deferred atmTrailRequest for deferred request.

Errors: protocol-specific addressing errors, incorrectTerminationPoints, TTPDisabled, TTPLocked, TTPConnected, non-matchingDescriptors, operationFails.

Results: Creation or association of TTPs and an associated atmTrail.

A response indicating partial setup (indicating both successful and unsuccessful end points) will be generated in cases where a multipoint atmTrail set up attempt neither completely succeeds (all endpoints succeed), nor completely fails (zero or one endpoints succeed).
";

ASN.1 Productions

```
AtmNwMIBMod -- {1 3 6 1 4 1 353 atmForumNetworkManagement(5) atmfM4(1)
atmfCmipNwView(2) informationModel(0) asn1Module(2) AtmNwMIBMod(0)}

DEFINITIONS IMPLICIT TAGS ::= BEGIN

  -- exports everything
  IMPORTS
  Boolean,
  NameType,
  ObjectList,
  PointerOrNull,
  UserLabel
  FROM
  ASN1DefinedTypesModule { itu-t(0) recommendation(0) m(13) gnm(3100)
    informationModel(0) asn1Modules(2) asn1DefinedTypesModule(0)}

  CDVTolerance,
  MaxBurstSize,
  PeakCellRate,
  QosClass,
  SustainableCellRate
  FROM
  AtmMIBMod { 1 3 6 1 4 1 353 atmForumNetworkManagement(5) atmfM4(1)
atmfCmipNEView(1) informationModel(0) asn1Module(2) atmMIBMod(0)}

  atmCharacteristicInfo,
  vcCI,
  vpCI
  FROM
  AtmMIBMod {itu-t(0) recommendation(0) i(9) atmm(751)
    informationModel(0) asn1Module(2) atm(0)}


  ObjectInstance
  FROM
  CMIP-1 {joint-iso-itu-t(2) ms(9) cmip(1) modules(0) protocol(3)}

  AdministrativeState
  FROM
  Attribute-ASN1Module { joint-iso-itu-t(2) ms(9) smi(3) part2(2)
asn1Module(2) 1};

  atmfM4NwObjectClass OBJECT IDENTIFIER ::= {1 3 6 1 4 1 353
atmForumNetworkManagement(5) atmfM4(1) atmfCmipNwView(2) informationModel(0)
managedObjectClass(3)}

  atmfM4NwPackage OBJECT IDENTIFIER ::= {1 3 6 1 4 1 353
atmForumNetworkManagement(5) atmfM4(1) atmfCmipNwView(2) informationModel(0)
package(4)}
```

```
   atmfM4NwAttribute OBJECT IDENTIFIER ::= {1 3 6 1 4 1 353
atmForumNetworkManagement(5) atmfM4(1) atmfCmipNwView(2) informationModel(0)
attribute(7)}

   atmfM4NwNameBinding OBJECT IDENTIFIER ::= {1 3 6 1 4 1 353
atmForumNetworkManagement(5) atmfM4(1) atmfCmipNwView(2) informationModel(0)
nameBinding(6)}

   atmfM4NwAction OBJECT IDENTIFIER ::= {1 3 6 1 4 1 353
atmForumNetworkManagement(5) atmfM4(1) atmfCmipNwView(2) informationModel(0)
action(9)}


restorationDefault RestorationMode ::= availRoutingAndReRouting


AddTpsDetails ::= SEQUENCE OF NewZ-EndpointDetails


AddTpsInformation ::= SEQUENCE {
        connectionId [0]                ObjectInstance,
        requestedAddTime [1]GeneralizedTime         OPTIONAL,
        details [2]                     AddTpsDetails}

AddTpsReply ::= CHOICE {
        failed [0]                      FailureProblem,
        rescheduled [1]                 RescheduledInfo,
        added [2]                       ConnectInfo,
        scheduled [3]                   ScheduledInfo,
        partial [4]                     PartialInfo }

CancelRequestInformation ::= ObjectInstance

CancelRequestReply ::= SEQUENCE {
        cancelResponse [0]              CancelResponse,
        requestId [1]                   ObjectInstance }

CancelResponse ::= ENUMERATED {
        requestCanceled (0),
        noSuchRequest (1),
        cannotCancel (2) }

ConnectInfo ::= SEQUENCE {
        connectionId                    ObjectInstance,
        terminationPoints               SEQUENCE OF ObjectInstance }

ConnectionType ::= ENUMERATED {
        broadcast (0), -- point-to-multipoint
        merge (1), -- multipoint-to-point
        composite (2), -- root-to-leaves & leaves-to-root
        multipoint (3), -- multipoint-to-multipoint
        pointToPoint (4) } -- point-to-point
```

```
ConnectionTypesSupported ::= SET OF ConnectionType

DescriptorsChangeDetails ::= SEQUENCE {
        a-to-zTrafficDescriptor [0]      TrafficDescriptor OPTIONAL,
        a-IngressTrafficDescriptor [1] TrafficDescriptor OPTIONAL,
        z-EgressDescriptors [2] SEQUENCE OF EndPointTrafficDescriptor OPTIONAL,
        z-IngressDescriptors[3] SEQUENCE OF EndPointTrafficDescriptor OPTIONAL,
        z-to-aTrafficDescriptors[4] SEQUENCE OF EndPointTrafficDescriptor
OPTIONAL }

EndPoint ::= SEQUENCE {

        tpOrDescriptor [0]              TPorDescriptor,         trailEndPoint [1]
        Boolean                 OPTIONAL }

EndPointDescriptor ::= SEQUENCE {
        interfaceId [0]                 ObjectInstance, -- an uni, bissi, bici, or linkTP
        vpi [1]                         INTEGER OPTIONAL, -- requested VPI
        vci [2]                         INTEGER OPTIONAL} -- requested VCI

EndPointDetails ::= CHOICE {
        endPtsBroadcast [0]             EndPtsBroadcast,
        endPtsMerge [1]                 EndPtsMerge,
        endPtsComposite [2]             EndPtsComposite,
        endPtsMultipoint [3]            EndPtsMultipoint,
        endPtsPointToPoint [4]          EndPtsPointToPoint }

EndPointTrafficDescriptor ::= SEQUENCE {
        endPoint [0]                    EndPoint,
        trafficDescriptors[1]           TrafficDescriptor }

EndPtsBroadcast ::= SEQUENCE {
        a-EndPoint [0]                  EndPoint,
        a-to-zTrafficDescriptor [1]     TrafficDescriptor,
        z-Endpoints [2]                 SEQUENCE OF EndPoint }

EndPtsComposite ::= SEQUENCE {
        a-EndPoint [0]                  EndPoint,
        a-to-zTrafficDescriptor [1]     TrafficDescriptor,
        a-IngressTrafficDescriptor [2]TrafficDescriptor,
        z-EndPoints [3]                 SEQUENCE OF EndPoint,
        z-EgressDescriptors[4]          SEQUENCE OF TrafficDescriptor }

EndPtsMerge ::= SEQUENCE {
        a-EndPoint [0]                  EndPoint,
        a-IngressTrafficDescriptor [1]TrafficDescriptor,
        z-EndPoints [2]                 SEQUENCE OF EndPoint,
        z-EgressDescriptors[3]          SEQUENCE OF TrafficDescriptor }

EndPtsMultipoint ::= SEQUENCE {
        z-EndPoints [0]                 SEQUENCE OF EndPoint,
        z-EgressDescriptors[1]          SEQUENCE OF TrafficDescriptor,
        z-IngressDescriptors[2]         SEQUENCE OF TrafficDescriptor }
```

```
EndPtsPointToPoint ::= SEQUENCE {
        a-EndPoint [0]                  EndPoint,
        a-to-zTrafficDescriptor [1]     TrafficDescriptor,
        z-EndPoint [2]          EndPoint,
        z-to-aTrafficDescriptor [3]     TrafficDescriptor }

FailureCode ::= ENUMERATED {
        addressingError (0),
        noSuchTpInstance (1),
        disabledTpInstance (2),
        lockedTpInstance (3),
        lockedFabric (4),
        alreadyConnected (5),
        notConnected (6),
        mismatchingTpInstance (7),
        invalidTime (8),
        noSuchConnection (9),
        cannotUseGivenVPIorVCI (10),
        noSuchResource (11),
        notMultipoint (12),
        cannotRemovePrimary (13),
        insufficientBandwidth (14),
        multipointNotSupported (15),
        routingCriteriaNotMet (16),
        operationFails (17),
        underlyingSystemFailure (18) }

FailureProblem ::= SEQUENCE {
        failureCode [0]                         FailureCode,
        incorrectInstances [1]          SET OF ObjectInstance           OPTIONAL }

Integer ::= INTEGER

IntegerOrNull ::=CHOICE {
        integer INTEGER,
        null    NULL }

LinkDetails ::= SEQUENCE {
        availableIngressBandwidth [0]           Integer,
        availableEgressBandwidth[1]             Integer,
        maxAssignableEgressBandwidth [2]        Integer,
        maxAssignableIngressBandwidth [3]       Integer,
        fromTermination [4]             ObjectInstance,
        toTermination [5]               ObjectInstance,
        fromDetails [6]                 ObjectInstanceOrProfileDetails,
        toDetails [7]                   ObjectInstanceOrProfileDetails}

LinkTerminationPoints ::= SEQUENCE {
        a-LinkTP [0]                    ObjectInstance,
        z-LinkTP [1]                    ObjectInstance }

ModifyInformation ::= SEQUENCE {
        connectionId [0]                ObjectInstance,
        requestedModifyTime [1]         GeneralizedTime         OPTIONAL,
        details [2]                     DescriptorsChangeDetails}

ModifyReply ::= CHOICE {
        failed [0]                      FailureProblem,
        rescheduled [1]                 RescheduledInfo,
        modified [2]                    ConnectInfo,
        scheduled [3]                   ScheduledInfo }

NewZ-EndpointDetails ::= SEQUENCE {
        z-Endpoint [0]                  EndPoint,
        z-IngressDescriptor [1]         TrafficDescriptor       OPTIONAL,
        z-EgressDescriptor [2]TrafficDescriptor         OPTIONAL,
        routeSelection [3]              RouteDescription        OPTIONAL }

ObjectInstanceOrProfileDetails ::= CHOICE {
        objectInstance [0]              ObjectInstance,
        profileDetails [1]              ProfileDetails }

PartialInfo ::= SEQUENCE {
        activated [0]           ConnectInfo,
        failed [1]              SEQUENCE OF FailureProblem }

ProfileDetails ::= SEQUENCE {
        totalEgressBandwidth [0]                Integer,
        totalIngressBandwidth[1]                Integer,
        maxNumActiveConnectionsAllowed [2]      Integer,
        vpiOrVciRange [3]                       VpiOrVciRange }

ProvisionType ::= ENUMERATED {
        manual (0),
        automatic (1) }

ReleaseInformation ::= SEQUENCE {
        connectionId [0]                ObjectInstance,
        requestedRemovalTime [1]        GeneralizedTime         OPTIONAL,
        retainConnectionsInd [2]                Boolean }

ReleaseReply ::= CHOICE {
        failed [0]                      FailureProblem,
        rescheduled [1]                 RescheduledInfo,
        removed [2]                     ConnectInfo,
        scheduled [3]                   ScheduledInfo,
        partial [4]                     PartialInfo }

RemoveTpsDetails ::= SEQUENCE {
        removeZ-Endpoints               SEQUENCE OF ObjectInstance,
        retainConnectionsInd            Boolean }

RemoveTpsInformation ::= SEQUENCE {
        connectionId [0]                ObjectInstance,
        requestedRemovalTime [1]        GeneralizedTime         OPTIONAL,
        details [2]                     RemoveTpsDetails}

RemoveTpsReply ::= CHOICE {
        failed [0]                  FailureProblem,
        rescheduled [1]             RescheduledInfo,
        removed [2]                 ConnectInfo,
        scheduled [3]               ScheduledInfo,
        partial [4]                 PartialInfo }

RequestActionInfo ::= CHOICE {
        setup [0]                   SetupInformation,
        modify [1]                  ModifyInformation,
        release [2]                 ReleaseInformation,
        addTps [3]                  AddTpsInformation,
        removeTps [4]               RemoveTpsInformation }

RequestStatus ::= ENUMERATED {
        notScheduled (0),
        scheduled (1),
        suspended (2),
        userCanceled (3),
        beingHandled (4),
        completed (5) }

RequestType ::= ENUMERATED {
        setup (0),
        modify (1),
        release (2),
        addTps (3),
        removeTps (4) }

RescheduledInfo ::= SEQUENCE {
        requestId [0]               ObjectInstance,
        committedTime [1]           GeneralizedTime}

RestorableType ::= ENUMERATED {
        restorable (0),
        nonrestorable (1)}

RestorationMode ::= ENUMERATED {
        unavailable (0),
        availRoutingOnly (1),
        availReRoutingOnly (2),
        availRoutingAndReRouting (3) }

RouteDescription ::= SEQUENCE {
        referenceObject             ObjectInstance,
        option                      RoutingOption}

RouteDescriptionList ::= SET OF RouteDescription

RoutingDetails ::= SEQUENCE {
        routeSelection [0]          RouteDescriptionList  OPTIONAL,
        maxHops [1]                 IntegerOrNull  OPTIONAL }

RoutingInfo ::= CHOICE {
        routingDetails [0]          RoutingDetails,
        atmRoutingProfilePointer [1]  ObjectInstance }

RoutingOption ::= ENUMERATED {
        mandatory (0),  -- must use the object in establishing the connection
        preferred (1),  -- attempt to use the object in establishing the connection
        exclude (3),  -- do not use the object in establishing connection
        sameRoute (4),  -- use same route as referenced object
        diverseRoute (5) }  -- use different route than referenced object

ScheduledInfo ::= SEQUENCE {
        requestId                     ObjectInstance }

SetupDetails ::= SEQUENCE {
        connectionType [0]            ConnectionType,
        restorableType [1]            RestorableType          OPTIONAL,
        administrativeState      [2]  AdministrativeState     OPTIONAL,
        endPointDetails [3]           EndPointDetails,
        routeingInfo [4]              RoutingInfo    OPTIONAL }

SetupInformation ::= SEQUENCE {
        label [0]                     UserLabel,
        requestedActivationTime [1]   GeneralizedTime         OPTIONAL,
        details [2]                   SetupDetails}

SetupLinkInfo::= CHOICE {

        linkDetails [0]               LinkDetails,
        linkTerminationPoints [1]     LinkTerminationPoints }
SetupReply ::= CHOICE {
        failed [0]                    FailureProblem,
        rescheduled [1]               RescheduledInfo,
        activated [2]                 ConnectInfo,
        scheduled [3]                 ScheduledInfo,
        partial [4]                   PartialInfo }

Time ::= GeneralizedTime

TPorDescriptor ::= CHOICE {
        terminationPoint [0]          ObjectInstance,
        descriptor [1]                EndPointDescriptor }

TrafficDescriptor ::= SEQUENCE {
        delayVariation [0]            CDVToleranceOPTIONAL,
        burstSize [1]                 MaxBurstSize  OPTIONAL,
        peakRate [2]                  PeakCellRate   OPTIONAL,
        sustainableRate [3]           SustainableCellRate,
        class [4]                     QosClass }

VirtualId ::= SEQUENCE {
        vpi [0]                       INTEGER, -- VPI value
        vci [1]                       INTEGER OPTIONAL} -- VCI value

VpiOrVciRange ::= SEQUENCE {
        lowVID [0]                    INTEGER,  -- low end of VPI or VCI range
```

```
    highVID [1]                    INTEGER }  -- high end of VPI or VCI range

END
```

References

[1] ITU-T Recommendation I.610, " B-ISDN Operation and Maintenance Principles and Functions," November 1995.

[2] ITU-T Recommendation I.751, “ATM Management of NE View," March 1996.

[3] ITU-T Recommendation G.805, "Generic Functional Architecture of Transport Network," November 1995.

[4] ITU-T Recommendation M.3100, "Generic Network Information Model," July 1995.

[5] ISO/IEC IS 10165-1 | ITU-T Recommendation X.720, "Information Processing Systems - Open Systems Interconnection - Structure of Management Information (SMI) - Part 1: Management Information Model," January 1992.

[6] ISO/IEC IS 10165-2 | ITU-T Recommendation X.721, "Information Technology - Open Systems Interconnection - Structure of Management Information (SMI) - Part 2: Definition of Management Information," February 1992; plus Technical Corrigendum 1.

[7] ISO/IEC IS 10165-4 | ITU-T Recommendation X.722, "Information Processing Systems - Open Systems Interconnection - Structure of Management Information (SMI) - Part 4: Guidelines for the Definition of Managed Objects," January 1992.

[8] ATM-Forum af-nm-0020.000, "M4 Interface Requirements and Logical MIB: ATM Network Element-view," October 1994.

[9] ATM-Forum af-nm-0027.001, "CMIP Specification for the M4 Interface, Version 1.0" March 1995.

[10] ATM Forum af-nm-0058.000, “M4 Network View Requirements and Logical MIB”, June 1996.

[11] ATM Forum af-nm-0074.000, “M4 Network View Requirements and Logical MIB Addendum”, December 1996.

[12] ITU-T Recommendation G.852-01, "Management of the Transport Network - Enterprise Viewpoint for Simple Subnetwork Connection Management,” Draft Recommendation, 1996.

[13] ITU-T Recommendation G.853-01, "Common Elements of the Information Viewpoint for the Management of a Transport Network,” Draft Recommendation, 1996.

[14] ITU-T Recommendation G.854-01, "Computational Interfaces for Basic Transport Network Model,” Draft Recommendation, 1996.

Appendix A: ER Model

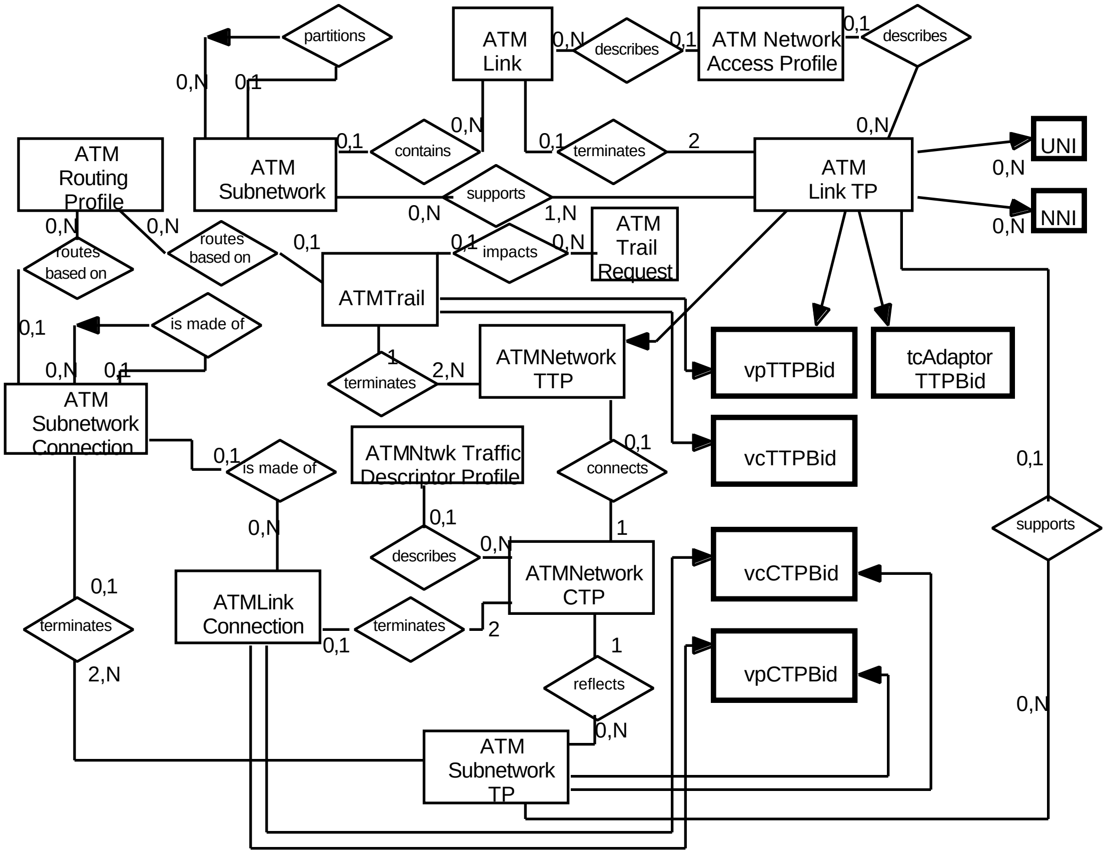

Figure A.1 Network View Entity Relationship Diagram

E-R diagrams graphically represent entity types and relationships. An entity is a unique thing that has properties (attributes, behavior, etc.) of its own.

E-R diagrams represent entity sets that can be instantiated. Relationships show associations between entities. The inheritance (Figure 2) and containment (Figure 1) diagrams are used along with the above E-R diagram to represent abstractions and containment (dependency) relationships respectively. Abstractions are used to show super class to subclass relationships. A dependency relationship is used to infer that the existence of dependent entity is conditioned upon the existence of a related superior entity.

On the E-R diagrams: a box denotes an entity and a diamond denotes a general relationship. The cardinality of the relationships is also represented on the diagrams. The E-R diagram above focuses on the ATM Network View. Here the entities, except for the ones in bold boxes, correspond to the CMIP objects defined in this document. The bold boxes represent objects in the M4 NE View, and the arrowed lines to them indicate possible pointer relationships. It is not required that both the M4 Network View and these NE View objects be provided across the same interface.

Appendix B: APPLICATION SCENARIOS

The purpose of this document it to show how the CMIP object classes defined for the M4 Network-Level View can be used depending on the operations architecture. Three application scenarios are covered:

- The first one shows the case of a **subnetwork connection management** providing **both a network-level view** of the subnetwork **and an NE-view** of the NEs involved. It is a combination of Ensembles described in Section 6.3.2 of the "Network-View Requirements and Logical MIB", with in addition, control of the NE-view, as described in Section 6.3.3.
- The second one shows the case of an **"opaque" subnetwork connection/trail management** system interfacing with a network management system which controls several subnetworks, each being "opaque," that is providing no visibility into the components of the subnetwork. It corresponds to the Ensemble described in Section 6.3.1 of the "Network-View Requirements and Logical MIB".
- The third one shows the case of a **Trail management** across two heterogeneous subnetworks managed by the same carrier. It is another version of the Ensemble described in Section 6.3.2 of the "Network-View Requirements and Logical MIB, without visibility at the NE-level.

Each of these scenarios uses the setup of a connection or a trail as an example to describe which object classes are used. These scenarios are kept simple. More complex scenarios can be built from the object model, when needed by specific applications and network architectures. Also, the three cases may be fully or partially combined. As an example, in the case of subnetwork management, the subnetwork, instead of having an NE-view, may have a blackbox view, using only the network-view.

Subnetwork Management/ Subnetwork with NE-view Scenario:

Network Management Context:

In this scenario, a network management system controls one or more subnetwork management systems which provide both the network-level view and the network-element level view (Figure B.1). A request to set-up a VP subnetwork connection is sent to the network management system, which, in turn, decomposes this request into three, one for each of the NEs.

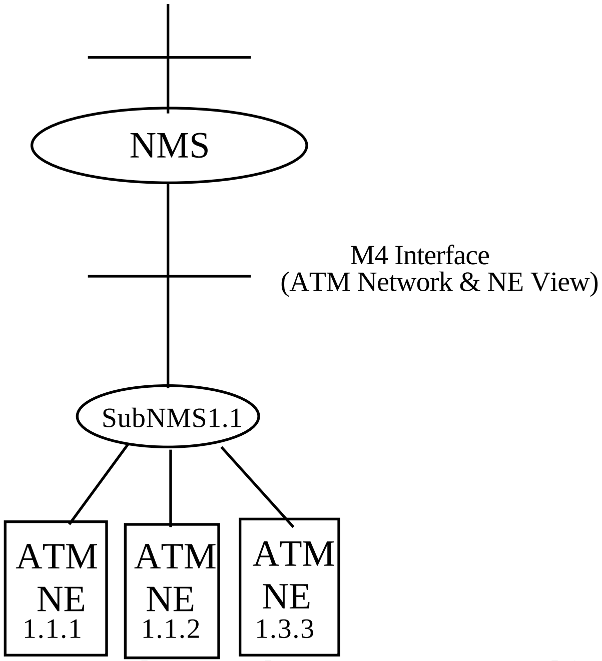

Figure B.1 Subnetwork Management Architecture

Object Classes Used:

In this scenario, the subnetwork management system controls a vcLayerNetworkDomain, a vpLayerNetworkDomain, and also the three NEs. Therefore, both the NE-view and most of the network-view are used. In the network-view, the nwCTPs and nwTTPs object classes are not used, since the snTPs can point directly to the underlying CTPs in the NEs. Trails do not need to be supported in the subNMS to NMS view, but the subNMS supports its part of trail setup by setting up SNCs and associating TTPs if the SNC terminates a Trail. NMS is responsible for the management of trails.. In this case, the layerNetworkDomain is used mainly for containment and for supporting the case of multiple subnetworks easily. Figure B.2 shows the containment tree of the object classes used in this scenario.

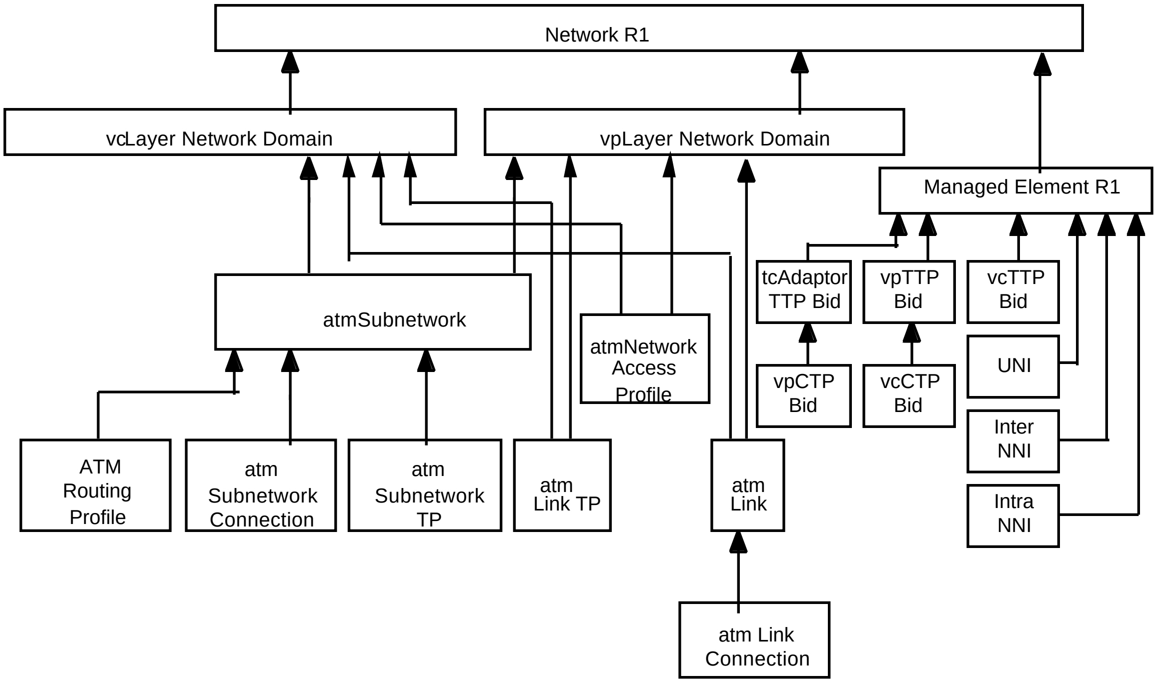

Figure B.2 Containment Relationship of the Object Classes Used

Topology:

The topology of the subnetwork is described to the management system as follows:

- a set of network elements
- a set of links between the different network elements
- a subnetwork, within the VP layer, which is made of the links and the NE for their VP part**.**

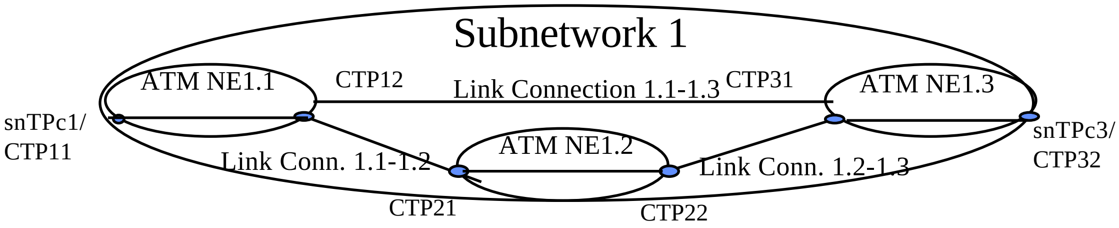

Figure B.3 Topology and Subnetwork Connection Set-Up

Connection Setup (the sequencing of the steps is not normative):

Step 1: The network management system sends to the subnetwork a setup request (M-ACTION) with identifiers for snTP1 and snTP3 (or the related interface) and the associated routing constraints, expressed in terms of links or nodes (note that either links or nodes can be used, but not both mixed up). Whatever objects are needed to service the request (such as: CTPs, profiles, etc) are created automatically.

Step 2: The subnetwork management system decides of the best route, selects the links and nodes (NEs) according to the constraints and the traffic descriptors, and send the corresponding connect requests to each of the NEs (M-ACTION on the atmFabric)

Step 3: The NEs execute each of the requests for the subnetwork management system, uses or creates the necessary CTPs (and TTPs if ever needed), and the corresponding atmCrossconnections, and acknowledge the creation of the entities and the completion of the requests.

Step 4: The subnetwork management system, upon reception of the acknowledgment, creates;

- atmSnTP1 and atmSnTP3, if needed, along with the pointers to the corresponding CTPs,
- the atmLinkConnections between NE1.1 and NE1.2 and between NE1.2. and NE1.3, with the atmLinkConnection pointers pointing directly to the NE CTPs (no atmSnTP at the NE-level are needed).
- an atmSubnetworkConnection from atmSnTP1 to atmSnTP3, which lists the linkConnections used by the subnetwork and the atmCrossconnections used by the NEs,

Step 5: The subnetwork management system acknowledge the creation of the atmSubnetworkConnection to the network management system.

Subnetwork Connection and Trail Management/Opaque Subnetwork Scenario:

Network Management Context:

In this scenario, a network management system controls one or more subnetwork management systems which provide only an opaque (no decomposition into NEs or component subnetworks) network-level view (Figure B.4). A request to set-up a VC trail is sent to the network management system.

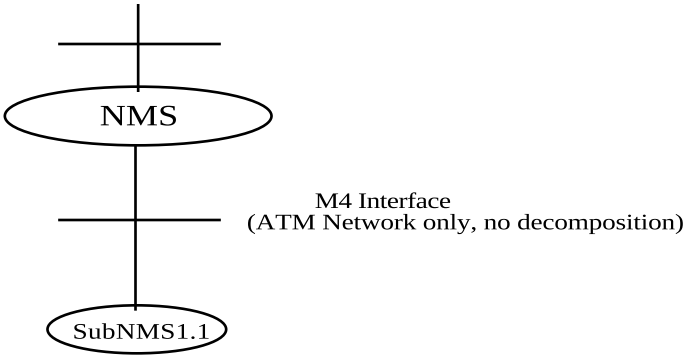

Figure B.4: Service Management Architecture

Object Classes Used:

In this scenario, the service management system is to control network management system, but no NEs. Therefore, nwCTPS and nwTTPs are to be used.

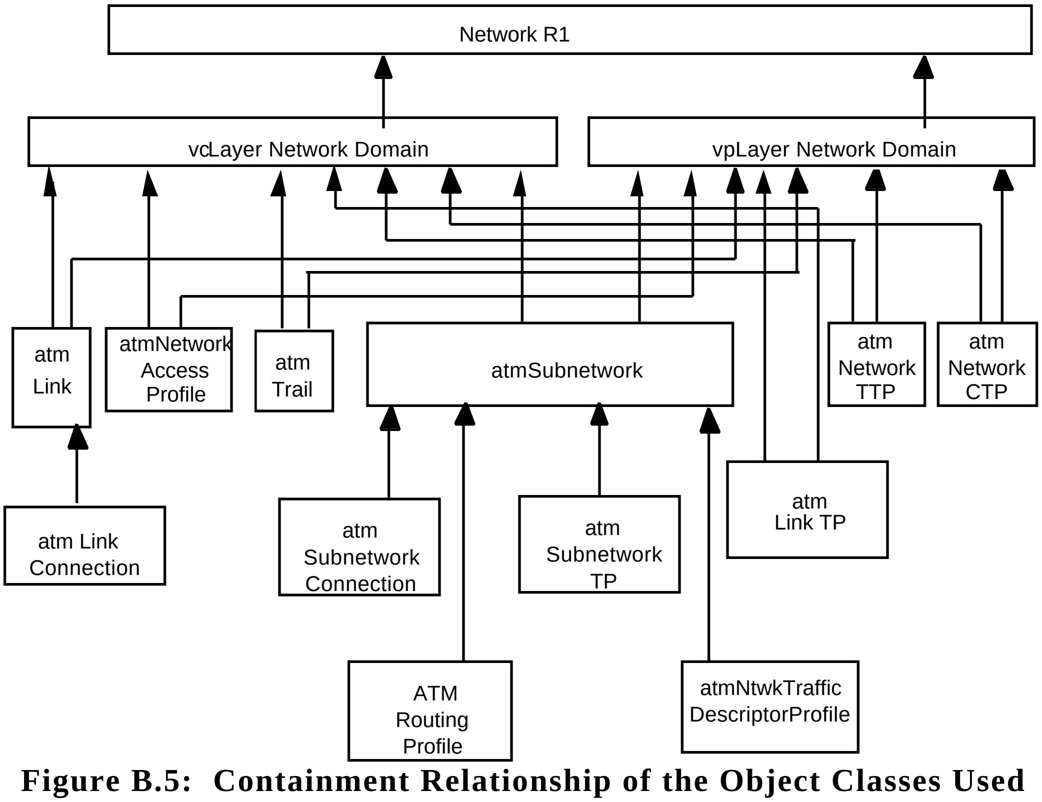

Figure B.5: Containment Relationship of the Object Classes Used

Topology:

The topology of the subnetwork is described to the management system as follows:

- a single subnetwork (note that the network management system may have visibility into multiple subnetwork management systems)

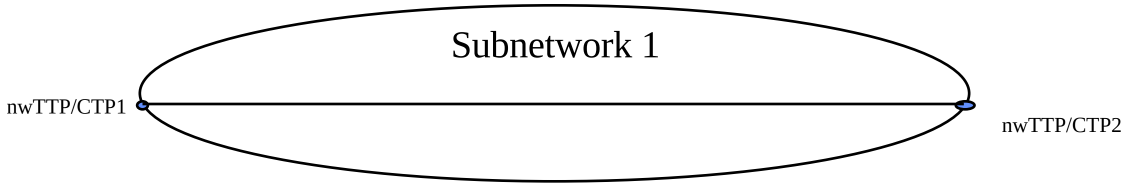

Figure B.6 Topology and Subnetwork Connection Set-Up

Trail Setup (the sequencing of the steps is not normative):

Step 1: The network management system sent to the subnetwork management system a trail connection setup request (M-ACTION) with identifiers for nwTTP/CTP1 and nwTTP/CTP2 (or the related interfaces), and the associated traffic information.

Step 2: The subnetwork management system creates if necessary nwTTPs, if any, and creates the corresponding nwCTPs, and subnetwork connection. The CTPs are bound to the TTPs. The subnetworkConnection points directly to the nwCTPs.

Step 3: The subnetwork management system acknowledges creation of the corresponding atmSubnetworkConnection and associated termination points, including TTPs if the SNC terminates a Trail.

Subnetwork Connection and Trail Management/Heterogeneous Subnetwork Scenario:

Network Management Context:

The Figure B.7 below illustrates a scenario in which two vendor specific subnetworks are managed by a Subnetwork 1 NMS. A Service Management System communicates with the Subnetwork 1 NMS to setup a vcTrail. The Subnetwork 1 NMS determines the subnetworks it must route the vcTrail through. If the vcTrail must traverse subnetwork 1.1 and subnetwork 1.2, the Subnetwork 1 NMS interacts with vendor A NMS to setup the portion of the trail supported by a subnetwork connection though subnetwork 1.1. Likewise the Subnetwork 1 NMS interacts with vendor B NMS to setup the portion of the trail that traverses subnetwork 1.2. Each vendor specific NMS manages its subtending Network Elements to configure cross connections, termination points, etc. to support the subnetwork connection request. This interaction with the Network Elements is transparent to the Subnetwork 1 NMS.

As an example, Vendor B NMS plays the role of manager to its subtending NEs, but also plays the role of an agent providing a subnetwork view to the NMS that manages the Subnetwork 1. It possible for Vendor B NMS to provide the NE View objects across the same interface to the NMS that manages Subnetwork 1. Alternately, the NMS that manages Subnetwork 1 may use a different interface to access the M4 NE View objects.

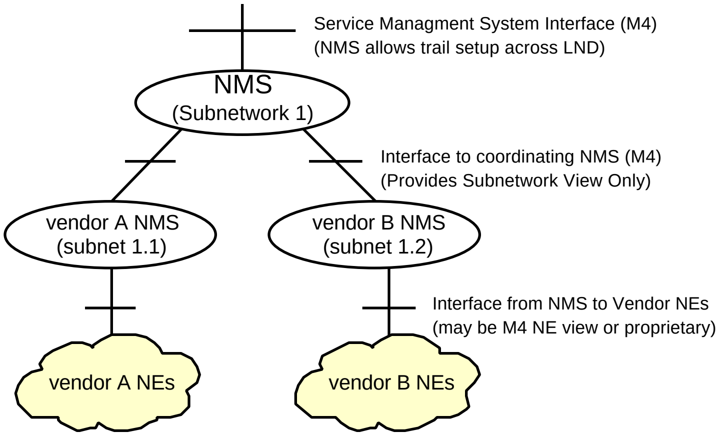
Service Managment System Interface (M4)
(NMS allows trail setup across LND)
NMS
(Subnetwork 1)
Interface to coordinating NMS (M4)
(Provides Subnetwork View Only)
vendor A NMS
(subnet 1.1)
vendor B NMS
(subnet 1.2)
Interface from NMS to Vendor NEs
(may be M4 NE view or proprietary)
vendor A NEs
vendor B NEs

Figure B.7 - Management Systems Architecture

Object Classes Used:

In this scenario, the service management system is to control network management system, but no NEs. Therefore, nwCTPs and nwTTPs are to be used.

Managed Entities involved in this scenario are:

- vcLayerNetworkDomain
- atmTrail
- atmSubnetwork
- atmSubnetworkTP
- atmNetworkTTP and atmNetworkCTP
- atmNetworkTrafficDescriptorProfile
- atmNetworkAccessProfile
- atmRoutingProfile
- atmSubnetworkConnection
- atmLink
- atmLinkTP
- atmLinkConnection

In NMS 1 the vcLayerNetworkDomain object includes actions for setting up trails. NMS 1.1 and NMS 1.2 include instances of the vcLayerNetworkDomain object that do not allow trail actions. The atmTrail object is present only in NMS 1.

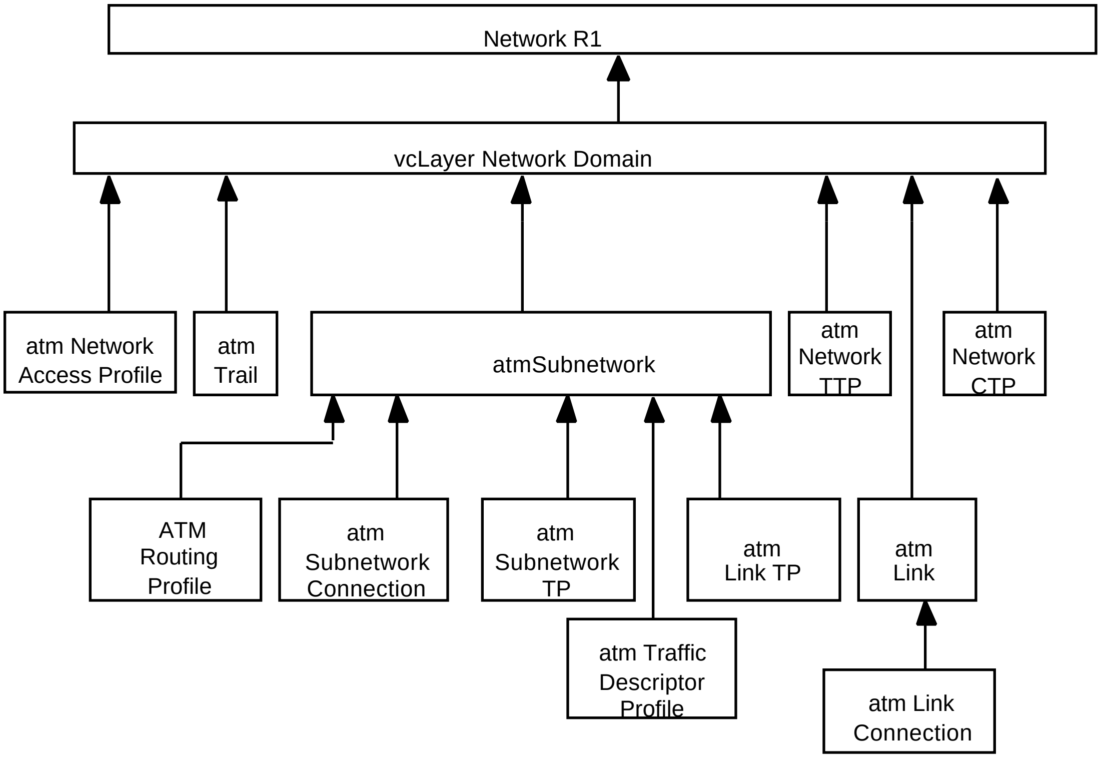

Figure B.8: Containment Relationship of the Object Classes Used

Topology:

Creating a vcTrail connection (VCC) between two subnetwork termination points A and Z on different subnetworks using M4 Network Level MIB proposed in the ATM Forum. Each subnetwork management system corresponds to a vendor specific subnetwork, within the VC Layer Network Domain, managed by a vendor supplied subnetwork management system (EML), i.e. subNet1.1 and subNet1.2.

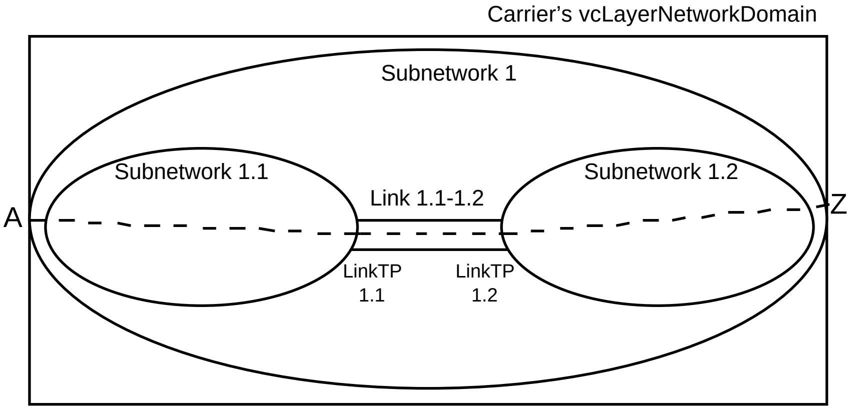

Figure B.9 - Subnetwork Structure

It is assumed that the object instances representing subnetworks (subnetwork 1, subnetwork 1.1 and subnetwork 1.2) and links were previously created through a subnetwork provisioning process.

The subnetwork termination points used to create the subnetwork connection are not pre-existing. subNMS1 manages subNet1 and implements M4 Network Level MIB View. subNet1 comprises of two subnetworks subNet1.1 (vendor A) and subNet1.2 (vendor B). subNMS 1.1 manage subNet1.1 (vendor A switches) and implement M4 NE level MIB or proprietary interface. subNMS 1.2 manages subNet1.2 (vendor B switches) and implement M4 NE level MIB or proprietary interface.

Trail Setup (the sequencing of the steps is not normative):

Step 1: The Subnetwork 1 NMS receives a request (setupTrail ACTION of vcLayerNetworkDomain) from a service layer management system to set-up a VC trail between points A and Z. The termination points A and Z may be identified by a server layer interface and may identify a VPI/VCI within the transport path or by a specific name, and are qualified by their traffic descriptors and QoS. An instance of the atmTrail object is created, but waits to bind the atmNetworkTTPs to the trail.

Step 2: The Subnetwork 1 NMS builds a subnetwork connection through Subnetwork 1 (from A to Z) and determines that the vcTrail must traverse subnetwork 1.1 and subnetwork 1.2 (and Link 1.1-1.2). The Subnetwork 1 NMS instantiates atmSubnetworkTPs for A and Z, but waits to bind them to atmNetworkCTPs.

Step 3: The Subnetwork 1 NMS builds a link connection from Subnetwork 1.1 to Subnetwork 1.2. atmSubnetworkTPs are instantiated for the terminations of the atmLinkConnection. These will be bound when the subnetwork management systems identify atmNetworkCTPs.

Step 4: The Subnetwork 1 NMS interacts with vendor A NMS (setupSubnetworkConnection ACTION) to setup a subnetwork connection though subnetwork 1.1 to support the trail. Endpoints are identified corresponding to A and atmLinkTP 1.1. Likewise the Subnetwork 1 NMS interacts with vendor B NMS to setup the portion of the trail that traverses subnetwork 1.2 (from atmLinkTP 1.2 to Z).

Step 5: Vendor A NMS instantiates two subnetworkTPs, A and one within atmLinkTP1.1 and identifies the route to use for this subnetwork 1.1 connection.

Step 6: Vendor A NMS creates corresponding Connection Termination Points (and TTPs if the SNTP terminates the Trail), which are qualified by their traffic descriptors and QoS, and relates them to the NE view objects if available.

Step 7: Vendor A NMS establishes a VC subnetwork connection between the two subnetwork termination points within the subNet1.1 though interactions with the Network Elements.

Step 8: Vendor A acknowledges the vcSubnetworkConnection (completion of the setupSubnetworkConnection ACTION) within subNet1.1 and terminated by atmSubnetworkTPs within atmSubnetwork 1.1 and indicates the associated atmNetworkCTPs (and TTPs).

Additional steps for Vendor B NMS: steps 5 to 8 are repeated by Vendor B NMS for creating vcSubnetworkConnection within subnetwork 1.2 terminated at Z and a point associated with atmLinkTP 1.2.

Step 9: After the Subnetwork 1 NMS receives acknowledgment from Vendor A NMS and Vendor B NMS for the creation subnetwork connections from A to a TP within atmLinkTP 1.1, and from a TP within atmLinkTP 1.2 and Z:
The Subnetwork 1 NMS binds the atmNetworkCTPs and TTPs to the previously created (subnetwork 1) atmSubnetworkTPs and the atmTrail instance.

Step 10: The Subnetwork 1 NMS acknowledges the creation (completion of the setupTrail ACTION) of vcTrail between A and Z to it’s client, the service management system.