Telecommunications
Information
Networking
Architecture
Consortium

Network Resource Information Model Specification

Version 3.0

Abstract: This document presents information objects and relationships needed for network resource management functions in the areas of Connection Management, Resource Configuration Management, Fault Management, and Accounting Management. The information model presented is independent of any specific functional architecture of the management functions. The information model is transmission and switching technology independent. It describes a network abstractly in terms of network elements, aggregations of network elements, the topological relationship between the elements, endpoints of connections, and connections. In addition, the model defines management support objects for alarm surveillance, resource configuration, and accounting. The information model is specified using the OMT notation and Quasi-GDMO+GRM.

Keywords: TINA-C, information specification, managed object, network level management, TMN, ODP, GDMO, network.

Author(s): N. Natarajan, Hannu Flinck, R.M. Rosli

Editor: N. Natarajan

Type: TINA-C Baseline

Document Label: NRIM_v3.0_97_12_17

Date: December 17, 1997

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Purpose

This document specifies a *Network Resource Information Model* (NRIM) for the management of networks envisioned in the TINA architecture. Such networks are capable of supporting multimedia and multipoint communication sessions and may consist of network elements based on different technologies. The purpose of this document is to describe such a network from the information viewpoint perspective. The information specification is presented using TINA-C Information Modelling Concepts [2].

The Network Resource Information Model (NRIM) presents common managed object classes relevant to network resource management within the following TINA management functional areas:

- Network Topology Configuration Management
- Connection Management
- Fault Management
- Accounting Management

It should be noted that the first two management areas identified above, i.e., Network Topology Configuration Management and Connection Management, together constitute the Configuration Management area as identified in the TMN standards. As the names suggest, the Network Topology Configuration Management area deals with management of network topology, and the Connection Management area deals with management of connections.

It is important to note that although the purpose of this information model, hereafter simply referred to as the NRIM, is to describe the information entities (information elements and relationships) needed for network resource management functions in the above functional areas, the model is not dependent on any specific computational (or functional) architecture of the management functions. The NRIM is independent of the architecture of the individual management functions. In this sense, the NRIM is a “common information model”. Other TINA-C specification documents that provide detailed specifications for specific management functions, such as the Network Resource Architecture document [1] and the Cons Reference Point document [5], may include information elements that are derived from the information elements defined in the NRIM.

The NRIM is a transmission and switching technology independent information specification of network resources. It describes a network abstractly in terms of network elements, aggregations of network elements, the topological relationship between the elements, endpoints of connections (termination points), and transport entities (such as connections) that transport information between two or more termination points. In addition, the NRIM also defines management support objects for alarm surveillance, resource configuration, and accounting. Further, the NRIM defines objects for representing collections of network resources that are under the control of an individual network administration.

This version of the NRIM specification does not address management of connectionless networks, such as the Internet. It is expected that this limitation will be overcome in the next issue of the NRIM specification.

Following the information specification notations specified in [2], the NRIM is described using the Quasi-GDMO + GRM notation for specification texts and the OMT notation for information model diagrams.

1.2 Audience

The audience for this document is any of the following:

- Architects within the TINA Consortium that are concerned with the development of architectures for specific management functions, such as connection management, resource configuration management, and accounting management.
- Architects within the TINA Consortium that are concerned with the development of detailed specifications for TINA reference points related with network connectivity services.
- Any one interested in the TINA Network Resource Architecture [1]
- System architects and designers of network management applications
- Network operators
- Service designers

1.3 How to Read This Document

This document describes a generic (technology independent) network resource information model. The model has been divided into a number of fragments (e.g., network, fault management). Each fragment contains an overview section. Readers interested only in a brief overview of the TINA Network Resource Information Model (NRIM) should focus on the overview section of each fragment. Readers interested in detailed specifications of network resource management functions should read the detailed information specification portion of each fragment.

The reader is assumed to have some basic knowledge of OMT graphical notation and TINA information modelling concepts [2]. For a detailed description on the usage of the network resource information model, the reader is referred to the architecture and specification documents that deal with specific management functions, such as the Network Resource Architecture document [1] and The Cons Reference Point document [5].

The remainder of this document is organized as follows. *Section 2* describes the goals and requirements for the NRIM.

Section 3 describes the basic concepts used in the NRIM, and identifies the main sources used as inputs to the development of the NRIM. The relationship between the NRIM and the TMN layers is also discussed.

Sections 4 to *10* are the main chapters of this document and these describe the information model in detail. The model is presented in a number of fragments, each fragment describing a certain subject area. This grouping is done only for documentation purposes. The fragments and the sections that describe these fragments are listed below:

- *Section 4*, Network Fragment
- *Section 5*, Connectivity Fragment
- *Section6*, Termination Point Fragment
- *Section 7*, Domain and Management Support Fragment
- *Section 8*, Resource Configuration Fragment
- *Section 9*, Fault Management Fragment
- *Section 10*, Accounting Management Fragment

Each fragment is organized as follows. It contains an overview subsection that describes the main concepts and relationships specified in the fragment, which is followed by a subsection that provides the detailed information specification for the fragment using Quasi-GDMO+GRM (Q-GDMO) and OMT notations, and is presented in the following manner:

- OMT Diagram for the fragment
- Object Types
- Relationship Types

The following appendices are included:

- *Appendix A* describes the changes made to the previous version (1995 version) of the NRIM specification document [4].
- *Appendix B* defines the syntax for the data types used in the network resource information model.
- *Appendix C* identifies the standards from which the different elements of the NRIM originated, and also identifies other standards that use similar concepts.
- *Appendix D* includes relevant network element level aspects.

1.4 Relationships to other TINA-C documents

The NRIM is described using the Information Modelling Concepts specified in [2].

The information model presented in this document is used in different TINA management functional areas (e.g. Connection Management, Network Topology Configuration Management, and Accounting Management). Examples of the usage of the objects defined in the NRIM are found in the Network Resource Architecture (NRA) document [1] and the ConS Reference Point specification document [5].

Currently, there is some duplication of text between Section 2 of the NRA document and Section 3 of this document. It has been recognized that this material logically belongs to this document. In the next version of the NRA document, this duplication will be eliminated.

2. NRIM Specification Overview

This chapter gives an overview of the network resource information model (NRIM) specification presented in this document. It describes the objectives of the NRIM specification, the requirements that the specification has to meet, and the overall structure of the NRIM specification.

2.1 Goals

The objectives of the NRIM specification are to:

- Specify the network resource information model provided to service applications.
- Specify the network resource information model provided to network management applications
- Enable reuse of service and network management application software by describing a generic resource information model which is independent of the underlying technologies.

2.2 Requirements on the NRIM Specification

- The NRIM specification shall describe object classes that are needed to represent resource aspects of complex multimedia and multipoint communication sessions. Such a communication session will be made up of one or more network connections (point-to-point or multipoint connections) with some Quality of Service (QoS) and synchronization constraints associated with them.
- The NRIM specification shall describe object classes that are needed to represent a network that supports complex communication sessions described above. Such a network will be made up of components based on different technologies (e.g., ATM, Wireless, SDH, and POTS).
- The NRIM specification shall be transmission and switch technology independent
- The NRIM specification shall describe object classes needed to represent
    - Network view of the network, containing layering, partitioning and connectivity aspects
    - Management support objects
- The information specification shall be based on existing standards within this area; these are ITU-T (TMN), Network Management Forum (NMF), OSI Systems Management Standards, and ATM Forum. The information specification shall be in alignment with the TMN principles of separating management functionality into several logical layers.

2.3 Overall Structure of the NRIM Specification

The TINA Network Resource Information Model defines information elements used in different resource management functional areas.

The specifications of network resource objects are presented in Section 4 to Section 10. It is structured as shown in Table 2-1:

Table 2-1. Specification of network resource objects

Section	Name	Short Description
Section 4	Network Fragment	Describes the overall structure of a network using basic concepts such as layering and partitioning.
Section 5	Connectivity Fragment	Describes objects representing connectivity across the network.
Section 6	Termination Point Fragment	Describes the end points of connectivity elements and some topological elements.
Section 7	Domain and Management Support Fragment	Describes objects that represent collections of resources under the control of a network administration or management function.
Section 8	Resource Configuration Fragment	Describes support objects used by resource configuration.
Section 9	Fault Management Fragment	Describes support objects used by fault management.
Section 10	Accounting Management Fragment	Describes support objects used by accounting management.

The detailed NRIM specifications are presented using the Quasi-GDMO defined in [2]. Quasi-GDMO is a GDMO based notation tailored for TINA-C information specifications. Major differences from the standard GDMO are:

- An object does not include an attribute that is used for its identification.
- An object does not include attributes that represent relationships with other objects. (Relationships are specified abstractly using relationship types.)
- Packages are always written within object definitions.
- Name bindings are not specified.

Readers are referred to [2] for more details on Quasi-GDMO.

3. Basic Concepts

This section discusses the basic concepts used in the NRIM specification. These concepts have been adopted from several existing and evolving standards in the discipline of network information modelling. In the development of the NRIM specification, a primary goal has been to incorporate concepts and object class definitions from these existing and emerging standards, wherever possible, and to define new concepts and object classes only where necessary. This chapter gives an overview of the basic modelling concepts used in the NRIM, and identifies for each concept, the origin of the concept. A detailed comparison between the NRIM and other standard models is presented in Appendix C.

3.1 TMN Layers and the Scope of NRIM

3.1.1 TMN Layers

The Telecommunications Management Network (TMN) standard, defined in ITU-T Recommendation M.3010 [7], has identified that the functionality of managing a telecommunications network can be divided into a number of functional layers as shown in Figure 3-1.

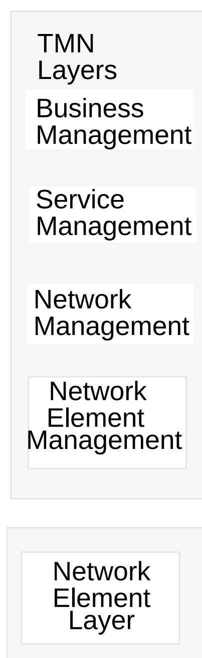


Figure 3-1. TMN Layers

The **network element management** layer manages a subset of network elements contained in a network, either on an individual basis or in aggregation.

The **network management** layer has the responsibility for the management of all the network elements using the management capabilities presented by the network element management layer. It is not concerned with how a particular network element provides services internally. Complete visibility of the whole network is typical and a vendor independent view will need to be maintained. This layer interacts with the service management layer on end-to-end connections, performance, faults etc. across the network.

The **service management** layer is concerned with, and responsible for, the contractual aspects of services that are being provided to customers or available to potential new customers. This layer provides the customer interface, interacts with other services or service providers and it interacts with the network management layer.

The **business management** layer has the responsibility for the total enterprise and is the layer at which agreements between operators are made. This layer is not a TINA-C concern.

3.1.2 Information Specification Aspects

Several different aspects[1] of management information may be defined for management purposes, e.g., network element level aspect, resource management level aspect and the service management level aspect. These aspects are not restrictive but focus on one particular use of the information from the standpoint of the user of that information.

The **network element level** aspect is concerned with the information that is required to manage specific equipment resources that provide network element layer functions.This refers to the information required to manage the physical resources, communication resources, and support functions within one network element.

The **resource management level** aspect is concerned with the information representing the network, both physically and logically. This view is used to represent a global view of the network. It is concerned with how individual network element entities are related, topologically interconnected, and configured to provide and maintain end-to-end connectivity.

The **service management level** aspect is concerned with how the network level aspects are utilized to provide a network service, and as such is concerned with the requirements of a network service (e.g. availability, cost, accounting, etc.) and how these requirements are met through the use of the network, and all related customer information.

1. Readers may recognize similarities with the M.3100 viewpoints of management information. TINA has chosen to use the term aspect to describe the M.3100 viewpoints in order to avoid confusion with the ODP viewpoints.

3.1.3 Relationship TMN Layers and Information Specification Aspects

The relationship between TMN layers and information specification aspects is shown in Figure 3-2.

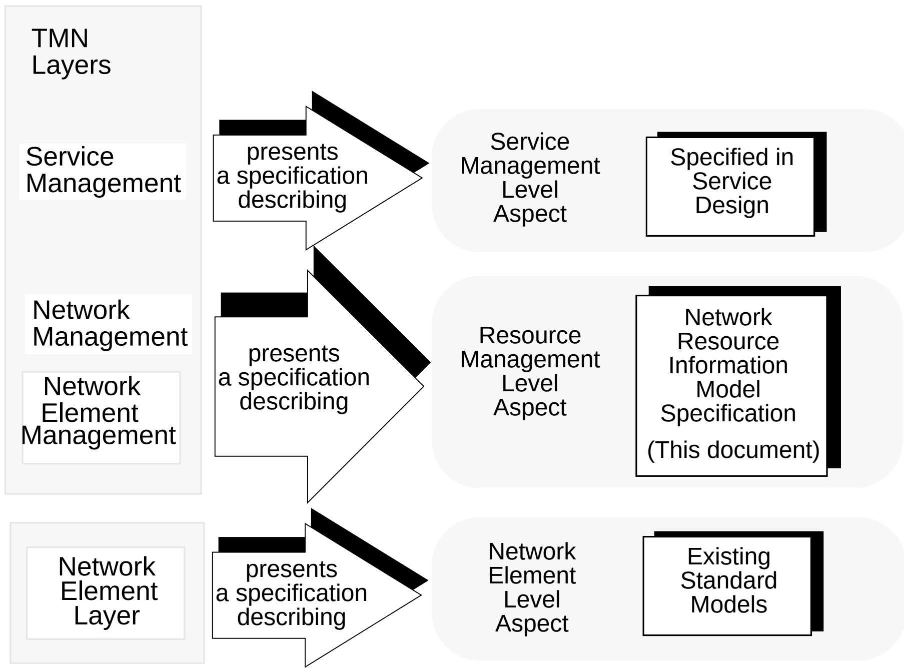


Figure 3-2. Relationship between TMN layers and Information Specification Aspects

The figure shows how the different TMN layers present different aspects of information:

- Service management layer presents a specification describing service management level aspects.
- The two TMN layers network management and network element management layers present a specification describing resource management level aspects.
- Network element layer presents a specification describing network element level aspects.

The figure also shows how the current network resource information model specification (this document) is concerned only with the **resource management level** aspect (also called **network level** aspect). In general, the information model presented by the resource management level to the service management level can be technology specific. Thus, it is possible to define ATM Network Level Model [22], SDH Network Level Model [28], and so on. However, the network level model defined in this document is technology independent. In this respect, the scope of NRIM is similar to that of the *Common Information Viewpoint* defined by ITU-T G.853 [11], *INA Management Information Model* [26], and the *Generic Managed Object Class Library for the Network Level View* defined by European Telecommunication Standards Institute (ETSI) [29].

3.2 Overall Structure of a TINA Network

A network modeled by the NRIM, hereafter referred to as a *TINA Network*, is a transport network that is capable of transporting multimedia information. The information traffic carried by the network will be heterogeneous in terms of data formats, bandwidth requirements, and other Quality of Service (QoS) characteristics. The network traffic in a TINA network can consist of inter-related multimedia streams, and the TINA network transports such traffic ensuring stream synchronization. The application level end points (TINA logical view) in the TINA network model are stream interfaces associated with TINA applications or multimedia devices.[2] See Figure 3-3.

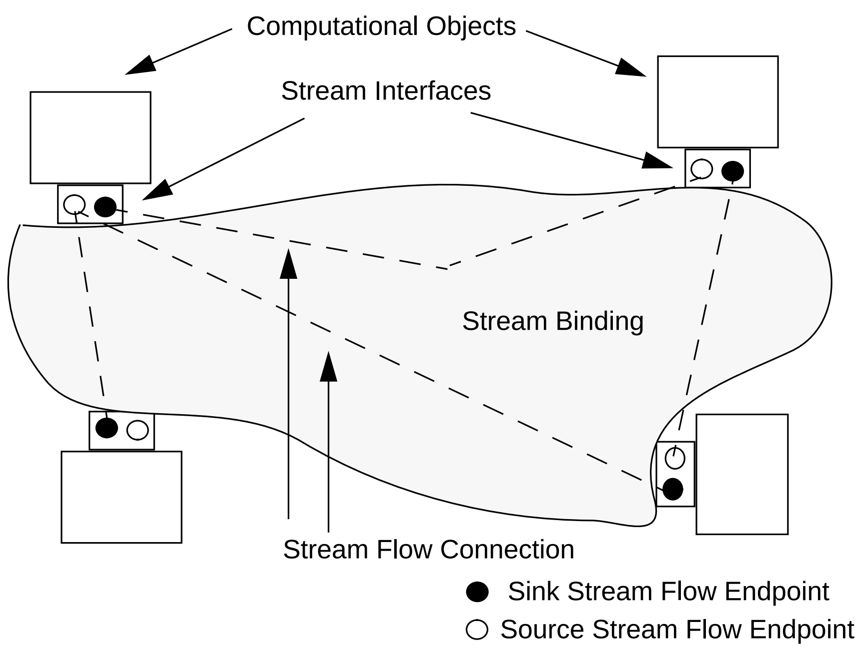


Figure 3-3. TINA Network (Logical View)

Figure 3-3 represents the logical view (application level, end-to-end view) of the TINA network model. Connectivity resources at this level are called *Stream Flow Connections* (SFCs). A stream flow connection is bounded by two or more *Stream Flow End Point*s (SFEPs). An SFEP is either an information source or sink, but not both (i.e., SFEPs are unidirectional). A source SFEP can be bound to one or more sink SFEPs (providing point-to-

2. The Kernel Transport Network that transports request and reply messages for computational objects interaction via operational interfaces is a higher level abstraction that can be modeled using the concepts defined in the NRIM.

multipoint connectivity). *Stream Flow End Point Pool* (SFEP Pool) is a modeling construct that aggregates SFEPs belonging to an application or a multimedia device. (A SFEP Pool is the representation of a stream interface in the NRIM). A *Stream Binding* is a modeling concept that represents a collection of stream flow connections that have been grouped together for some purpose at the application level.

An SFEP can terminate only one stream flow connection. Associated with a stream flow end point is the characteristic information accepted/delivered at that SFEP. These properties include frame structure identification, QoS, etc. The frame structure and QoS of the source and sink stream flow end points bound by a stream flow need not be identical but must be compatible[3].

In the physical view, a TINA network is divided into two main components: one is the *Connectivity Layer Network* (CLNW) and the other is formed by the communication resources contained in *Customer Premises Equipment* (CPE), see Figure 3-4. A CPE may be either a simple terminal device (telephone or multimedia device) or a computing system in which TINA compliant applications are deployed. A connectivity layer network is a transport network consisting of a heterogeneous collection of switching resources, transmission resources, and adapters. A connectivity layer network is made up of components of different technologies, such as ATM, Frame Relay, narrow band ISDN, wireless, SDH, or PDH, and is capable of transporting different types of information (Figure 3-4 shows some of the possible technologies).[4] Note that it is possible that a CPE is attached to several such networks.

A communication endpoint at which a connectivity layer network accepts or delivers information is called a *Network Flow End Point* (NFEP), see Figure 3-6. From the perspective of a connectivity layer network, an NFEP may be either a source, sink, or both (contrast this with an SFEP). Associated with an NFEP is a characteristic information accepted/delivered at the NFEP. These properties include frame structure identification, QoS, etc. A *Network Flow Connection* (NFC) is a connectivity resource that transports information between a group of NFEPs. An NFC has one of the following configurations:

- A point-to-point bidirectional connection between two NFEPs
- A point-to-point unidirectional connection between two NFEPs. One of the end points is designated as the source NFEP, and the other end point is designated as the sink NFEP. Information is transported from the source NFEP to the sink NFEP.
- A point-to-multipoint unidirectional connection between two or more NFEPs. One of them is designated as the source NFEP, and the others are designated as the sink NFEPs. Information is transported from the source NFEP to the sink NFEPs.

3. See [3] for details.
4. As noted in Section 1, the current version of NRIM specification does not address management of connectionless networks, such as the Internet.

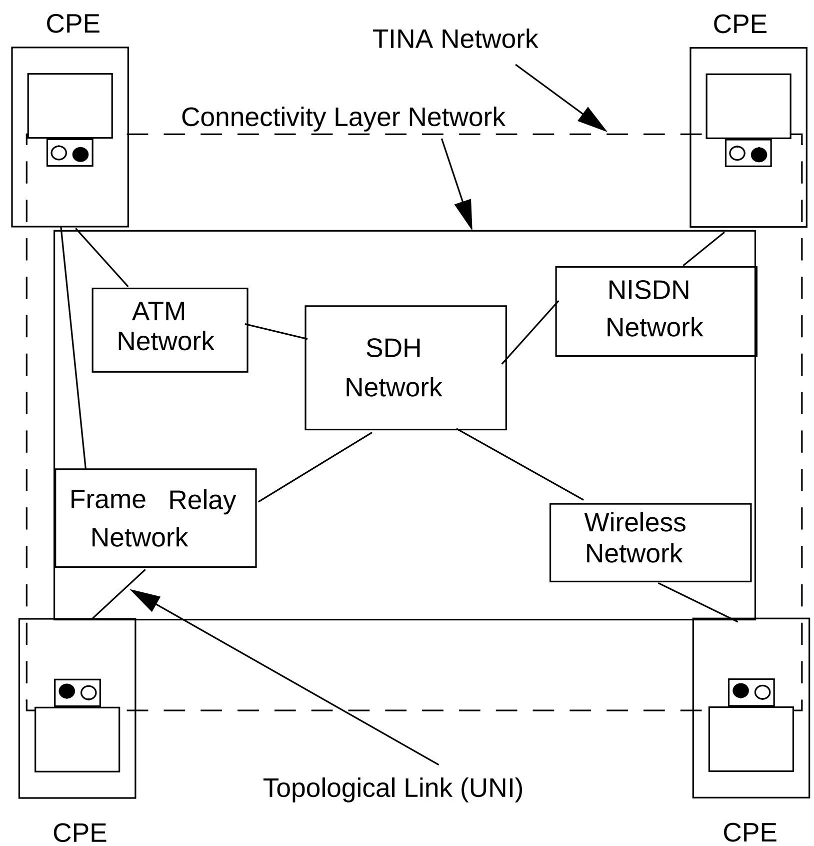


Figure 3-4. TINA Network (Physical View)

From the connectivity perspective, a stream flow connection is composed of one Network Flow Connection and two or more *Terminal Flow Connection*s (TFCs).[5] A terminal flow connection is a connectivity resource that transports information either from an SFEP to an NFEP, or vice versa, or to another SFEP within the same CPE. It is possible that the frame structure and QoS associated with the SFEP and NFEP bound by a terminal flow connection are different, in which case the TFC performs the necessary adaptation. It is possible that multiple TFCs have the same NFEP; i.e., several stream flow connections can be multiplexed over a single network flow connection.

See Figures 3-5 and 3-6. Figure 3-5 illustrates a point-to-multipoint stream flow connection. Figure 3-6 shows the decomposition of this stream flow connection into a point-to-multipoint network flow connection and three terminal flow connections.

5. Using more than one NFC for a SFC, for purposes such as protection switching, is not modelled in the current NRIM.

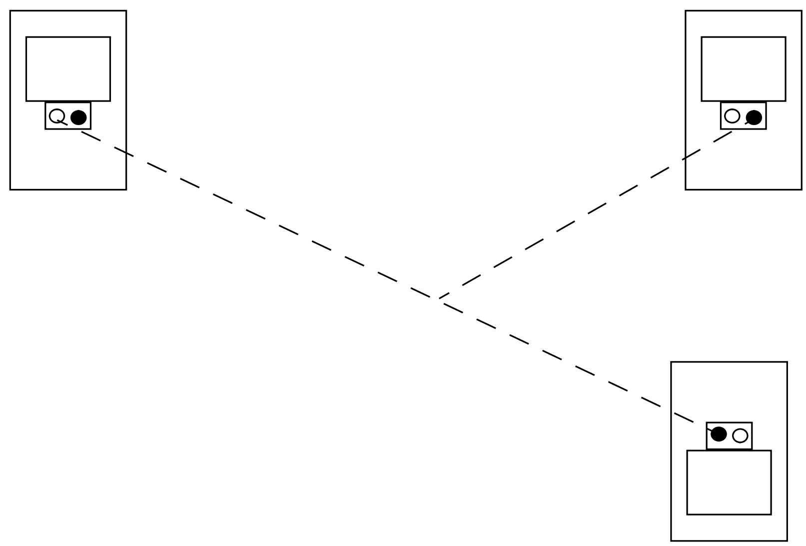


Figure 3-5. A Multipoint Stream Flow Connection

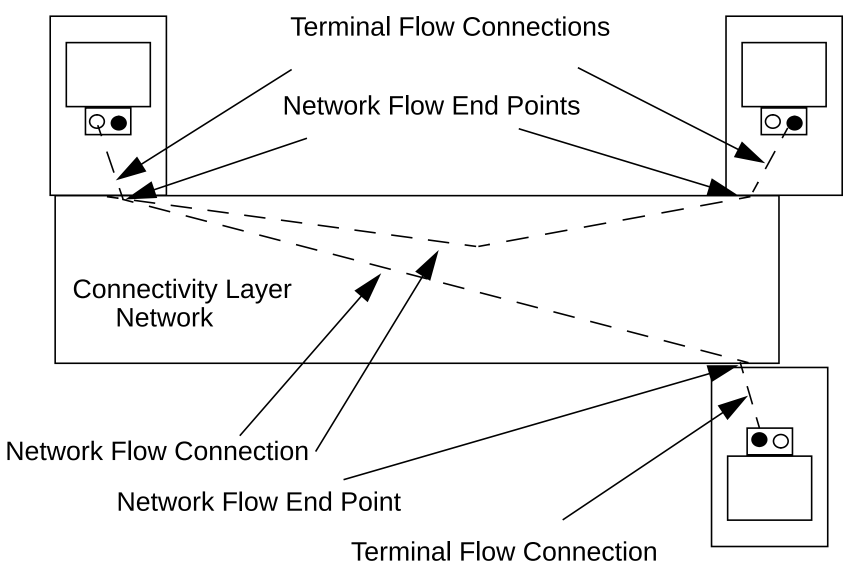


Figure 3-6. Network Flow Connection and Terminal Flow Connections

3.3 Layer Network and Trail

In networking literature, the concept of *Layer Network* (LNW) is used to denote a network that is based on a single technology and that transports information of a specific format, referred to as the characteristic information of the layer network. Examples of layer networks are: ATM Virtual Path (VP) network, ATM Virtual Channel (VC) network, SDH VC4 Path network, and Frame Relay (FR) network.

The concept of a layer network was originally defined in the ITU-T Recommendation G.803 which describes the functional model for SDH transport networks [9]. This concept has been adopted in the subsequent ITU-T Recommendation G.805 [10]. G.805 describes the functional and structural architecture of transport networks in a generic manner. Its concepts and principles are applicable to transport networks based on different technologies, such as SDH networks, ATM networks, and plesiochronous digital hierarchy (PDH) networks.

The layering concept of the network is based on the following ideas:

- Each layer handles one type of characteristic information. Each layer network represents a set of compatible inputs and outputs that may be interconnected and is distinguished by the characteristic information that is transported. The inputs and outputs may be regarded as access points on the layer network. Characteristic information is defined as a signal of characteristic rate, coding, and format. Generally, a layer network is closely tied to a specific type of network transmission and/or switching technology, e.g., SDH/SONET VC-4, ATM virtual channel (ATM VC), or ATM virtual path (ATM VP)
- Management of each layer network can be classified into similar functions (e.g. Fault Management, Security Management Accounting Management can be assigned to different Layer Networks)
- It is simpler to design and operate each layer separately than it is to design and operate the entire transport network as a single entity
- Each layer is able to have its own operations and maintenance capability
- Each network layer may be defined independently of the other layers
- It is possible to add or change the technology and/or structure of a single layer network without affecting other layers.

A connectivity layer network is made up of one or more layer networks. A layer network may be related to another layer network in a connectivity layer network in one of two ways:

- **Peer-to-Peer Relationship:** This is the case when information delivered by one layer network is adapted and given as input to the other layer network, and vice versa. This relationship is symmetric, and is referred to as *layer interworking relationship*. An example of this relationship is adaptation of Frame Relay to ATM (VP/VC), and vice versa.

- **Client-Server Relationship:** This is described later in Section 3.4.2. This is the case when a group of link connections in one layer (the *client layer*) network is served by a trail in the other layer (the *server layer*) network. Such a group of link connections is called a *topological link*.

The network resource (or entity) that transports information across a layer network between two or more endpoints in the layer network is called a *trail*. Thus, a trail is defined relative to a layer network. For example, a Virtual Path (VP) trail is a resource that transports ATM VP cells across a VP layer network. Depending on the nature of the layer network, a trail may or may not be directly related to the physical network. For example, a trail in the SDH path layer gives a logical view of the transport capacity, that is not necessarily related to the physical network. Whereas a trail in the physical media layer is related directly to an actual fibre (SDH physical optical section) or coaxial cable (SDH physical electrical section). A trail may have one of the following configurations: point-to-point bidirectional, point-to-point unidirectional, or point-to-multipoint unidirectional.

3.4 Structure of a Layer Network

3.4.1 Subnetworks and Links

A layer network is decomposed into *subnetworks* that are interconnected by *links* between them, see Figure 3-7. As defined in G.805, a link represents a topological relationship between two subnetworks and the potential for connectivity between the subnetworks. Each subnetwork may be further decomposed into smaller subnetworks interconnected by links until the desired level of detail is revealed. This will generally be when the subnetwork is equivalent to a single network element (switch or digital cross-connect).

A link is configured using one or more trails in an underlying server layer network. To distinguish between the general concept of a trail, and the specific use of a trail for configuring a link, the concept of *topological link* is defined in the NRIM. A topological link is the representation of a server layer trail in a client layer network. The end points of a topological link, called *topological link termination points*, are the points at which adaptation of client layer information to server layer information occur. Thus, a topological link is configured using exactly one trail in the underlying server layer network[6], and a link is configured using one or more topological links. See Figure 3-8. Figure 3-8 shows a layer network configuration consisting of two subnetworks. The subnetworks are interconnected by two topological links. Each termination point of a topological link is called a *Topological Link Termination Point* (TLTP). A link has been configured using the two topological links. This link represents the aggregate capacity for connectivity between the two subnetworks. Each termination point of the link is called a *Link Termination Point* (LTP).

A link can be configured using topological links in any of the following ways:

- 1:1 configuration: A link is configured using one topological link by assigning the entire bandwidth of the topological link (server layer trail) to the link.

6. The prefix “topological” is used to emphasize use of this specific configuration.

- N:1 configuration: A link is configured using one topological link by assigning to the link only a portion of the bandwidth of the topological link. Many such links can be configured using a topological link.
- 1:N configuration: A link is configured using a set of topological links by assigning the entire bandwidth of all topological links in the set. The topological links may be interconnecting different pairs of network elements.

As an illustration of the distinction between the concepts of link and topological link, consider an ATM network where the ATM NEs are interconnected using SDH paths. The provision of a SDH path connecting two ATM NEs and the provision of ATM-SDH adaptation units in the ATM NEs constitutes the provisioning of a topological link connecting the two ATM NEs in the ATM (VP) layer network. A configuration parameter of this topological link is the range of VPIs supported over the topological link (this depends on the capabilities of the adaptation equipment in the two NEs). From this topological link, many links are created in the following manner. Each link is assigned a portion of the bandwidth and a subset of the VPI range of the topological link. Thereafter, routing and bandwidth management for subnetwork connections and trails are done on the basis of links (and not on the topological link basis). A possible use of this multiple link configuration is routing different types of traffic on different links; e.g., Constant Bit Rate traffic is routed on one link and Available Bit Rate traffic is routed on another link. Another possible use is to dedicate a link for carrying traffic that belong to certain customers.

Irrespective of how a link is configured, the network resource that transports information across a link is called a *link connection*. The network resource that transports information across a subnetwork between two or more end points in the subnetwork is called a *subnetwork connection*, see Figure 3-7. Reflecting the partitioning of a layer network into subnetworks and links, a trail is made up of one or more subnetwork connections and link connections. Similarly, reflecting the partitioning of a subnetwork into subnetworks and topological links, a subnetwork connection may also be made up of one or more subnetwork connections and link connections, see Figure 3-7.

The partitioning concept is useful for defining:

- Significant administrative boundaries between network operators jointly providing end-to-end paths within a single layer network.
- Management domain boundaries (i.e., scope of management functions or systems) within the portion of a layer network that is under the control of a single network operator.

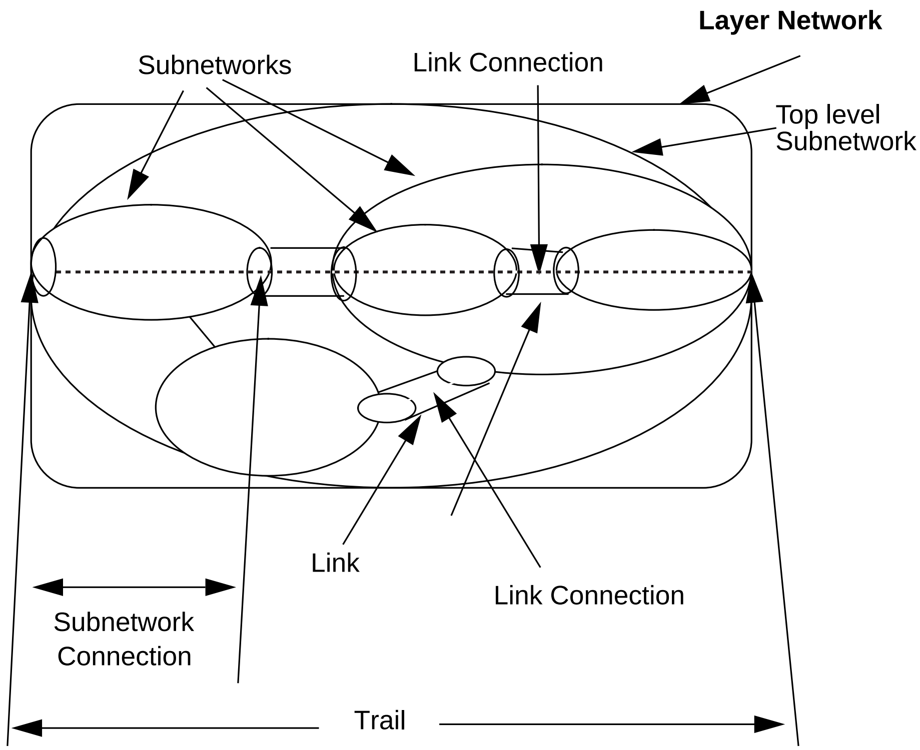


Figure 3-7. Structure of a Layer Network

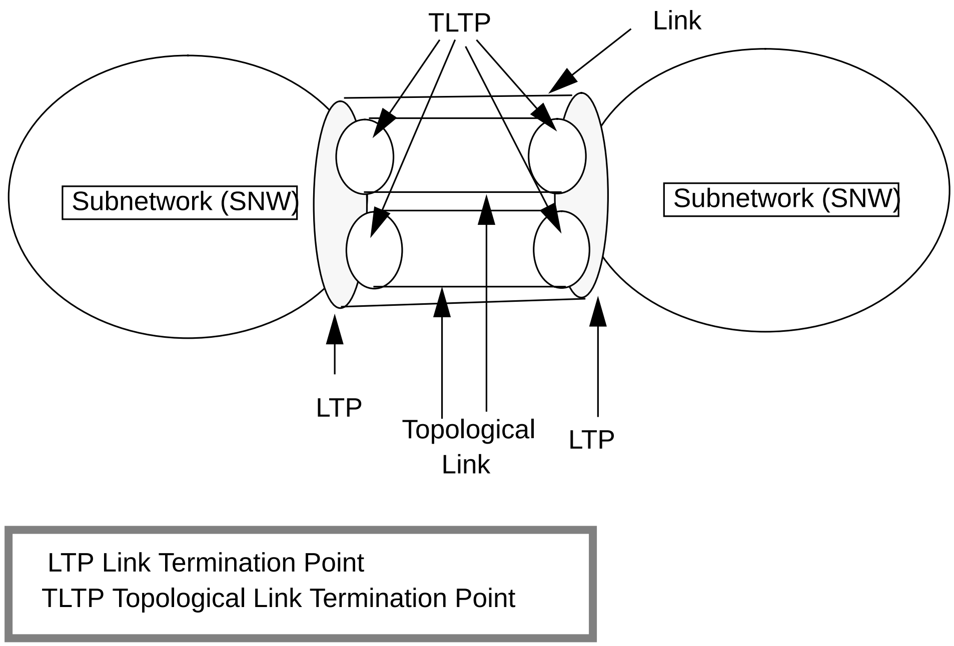


Figure 3-8. Links and Topological Links

3.4.2 Client-Server Relationship

As mentioned in Section 3.3, two layer networks may have a client-server relationship. This relationship exists when the transport capabilities of one layer (the server layer) are used in the other layer (the client layer). More specifically, this relationship is established when link connections in the client layer network are provided by a trail in the server layer network. See Figure 3-9. Some examples of the client-server relationship between layer networks are listed below:

- Link connections in a ATM VP layer network are provided by a path (the name of the trail in the path layer) in the SDH path layer network.
- Link connections in a ATM VC layer network are provided by a trail in a VP layer network.
- Link connections in a SDH path layer network are provided by a section (name of the trail in the transmission media layer network) in the transmission media layer network.

The end points of a link connection are called *Network Connection Termination Points* (NWCTP). It should be noted that while a trail or a subnetwork connection may be either a point-to-point or a point-to-multipoint connection, a link connection is always a point-to-point connection.

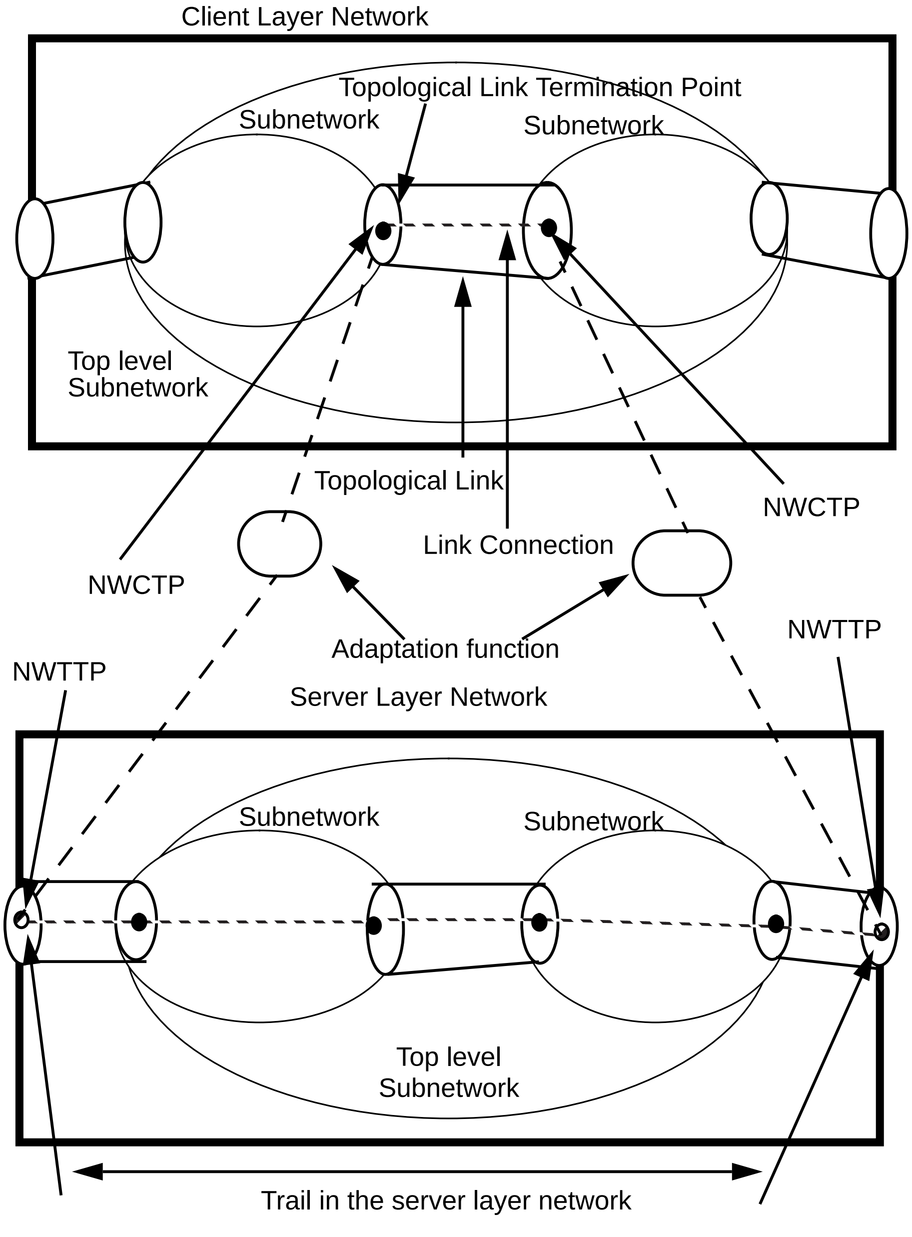


Figure 3-9. Client-Server Relationship

3.4.3 Edges and Network Connection Termination Points

As described in Section 3.4.1, it is possible that a subnetwork is partitioned into two or more subnetworks interconnected by links. Reflecting this partitioning, a subnetwork connection may also be made up of two or more subnetwork connections and link connections. In general, the lifetime of a subnetwork connection and its component subnetwork connections and link connections may be different. That is, when a subnetwork connection is deleted, some of the component subnetwork connections and link connections may continue to exist. Similarly, a subnetwork connection may be set up using existing subnetwork connections and link connections. These capabilities are useful for rerouting trails and subnetwork connections upon failures. To allow for this generality, the NRIM distinguishes between an end point of a subnetwork connection and an end point of a link connection. This distinction is especially important for unidirectional subnetwork connections and link connections, and in configurations where a subnetwork is partitioned into lower level subnetworks. The two kinds of end points are distinguished using the concepts of network connection termination points and edges as described below (see Figure 3-10):

- **Network Connection Termination Point (NWCTP):** A termination of a link connection is called a network connection termination point. A link connection is only a point-to-point connection, and thus a link connection has only two network connection termination points.
- **Edge:** An extremity of a subnetwork connection is called an edge.[7] A point-to-point subnetwork connection has two edges, and a multipoint subnetwork connection has more than two edges. An edge of a subnetwork connection is bound to a network connection termination point. This binding may change during the lifetime of the subnetwork connection.

In the above, an edge was defined as an extremity of a subnetwork connection. Actually, a more precise definition is that an edge is a *potential* extremity of a subnetwork connection. That is to say, it is possible to create an edge first, and subsequently either designate it as an end point for a new subnetwork connection or attach it to an existing multipoint subnetwork connection. It is also possible that an edge that is not bound to a subnetwork connection is used to replace an edge that is bound to a subnetwork connection. This feature is useful in supporting mobility.

Reflecting the partitioning levels of a subnetwork, a subnetwork connection in a composite subnetwork is partitioned into subnetwork connections in the component subnetworks. In such a situation, an edge of the composite subnetwork connection and an edge of the component subnetwork connection will be bound to the same NWCTP. See Figure 3-10.

7. The concept of Edge is identical to the concept of Subnetwork Termination Point defined in ITU-T G.853-01 Recommendation [11] and ATM Forum Network View MIB [22]. The term “Edge” is used here since this term has been prevalent in TINA-C since 1993.

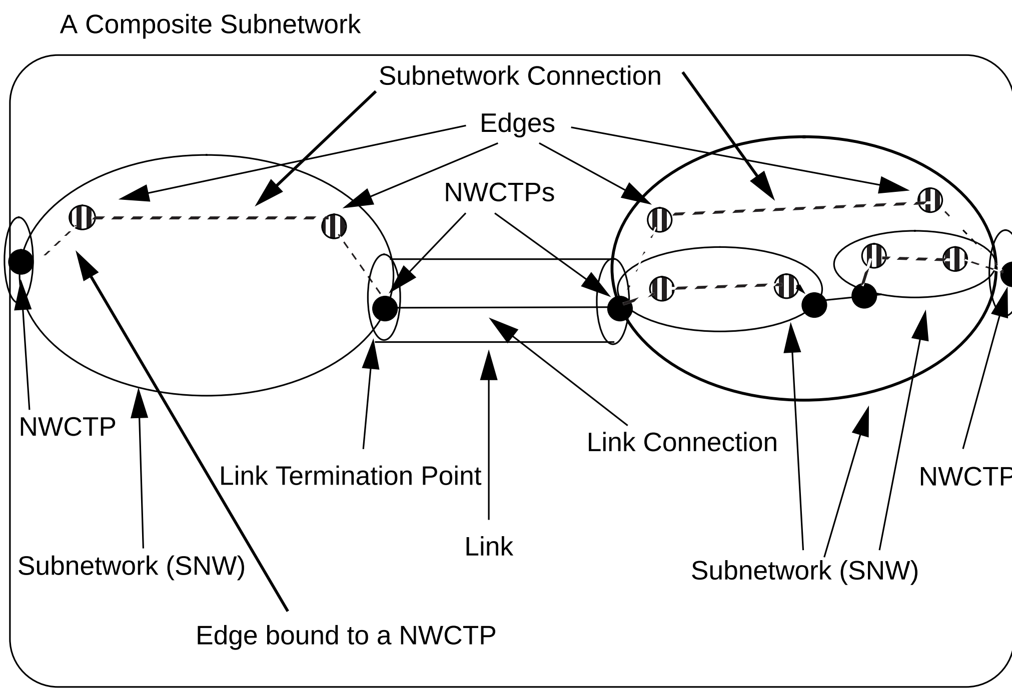


Figure 3-10. Relationship Between Edges and NWCTPs

3.4.4 Tandem Connection

A tandem connection is an arbitrary series of contiguous subnetwork connections and/or link connections and is used to represent an arbitrary segment of a trail. The extremities of a tandem connection are either network connection termination points or trail termination points. Just as in the case of a trail or a subnetwork connection, a tandem connection may be either point-to-point or point-to-multipoint.

Figure 3-11 illustrates tandem connections. As can be seen from the figure, a limiting case of a tandem connection is a subnetwork connection. The concept of tandem connection is very useful in situations where a trail spans multiple subnetworks and the subnetworks are under the control of different administrations. In such situations, a connectivity provider may offer to other connectivity providers a tandem connection service that includes setup, monitoring, and release of tandem connections.

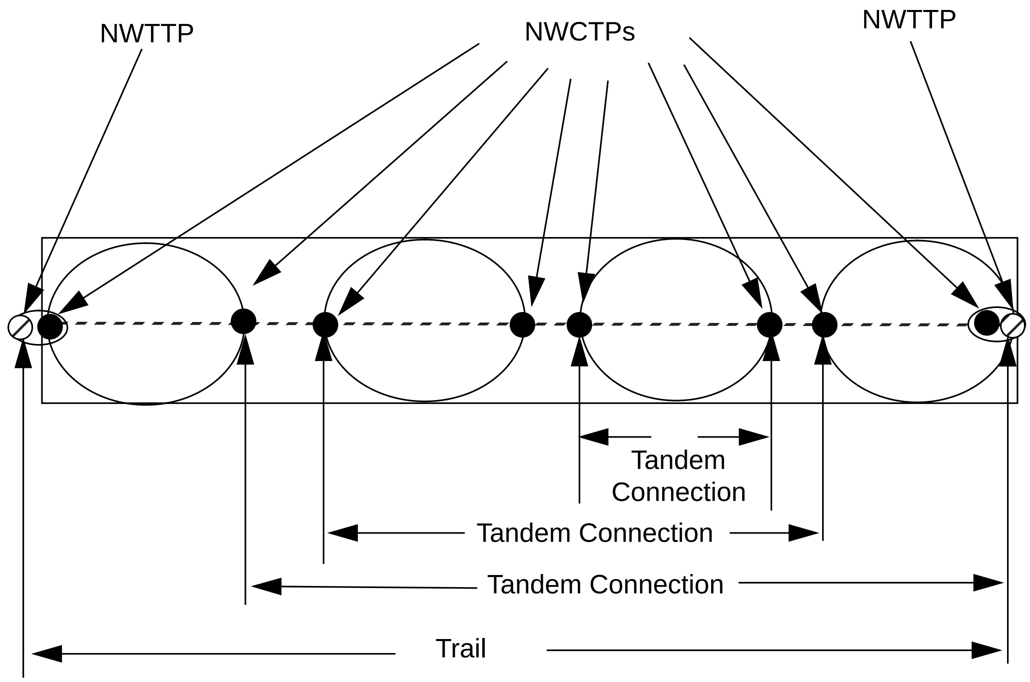


Figure 3-11. Tandem Connection

3.5 Management Functions Modelling

Network management standards have identified five management functional areas (or aspects) [7]. These are: *Fault Management*, *Configuration Management*, *Accounting Management*, *Performance Management*, and *Security Management*. These various functional areas are usually collectively referred to as FCAPS. One objective of NRIM is to specify information elements that are required for management functions in the various functional areas. The NRIM specification uses the following approach for this purpose.

Information elements needed for the FCAPS management functions are specified in a generic manner, i.e., independent of the information elements associated with individual resources. For each management functional area, an object class is defined, and that object class has the information elements needed for management activities in that functional area. For example, the class *Configurable* contains information elements that are required for configuration management functions. This class contains configuration related attributes, such as administrative state and operational state. Similarly, classes that represent resources are defined independent of the classes associated with specific management functions.

When a management application (or service) is designed, the designer determines the management requirements applicable to each resource under the purview of the application. If a resource, e.g., a subnetwork connection, is determined to be a resource that is required to support configuration management functions, the application designer represents such a resource using a class that inherits the Configurable class and the class represents the basic resource. Similar inheritance is used for other functional areas. This is illustrated in Figure 3-12. (The figure should be viewed only as an illustrative example.). It should be noted that NRIM does not determine management requirements on resources since NRIM is independent of specific management applications.

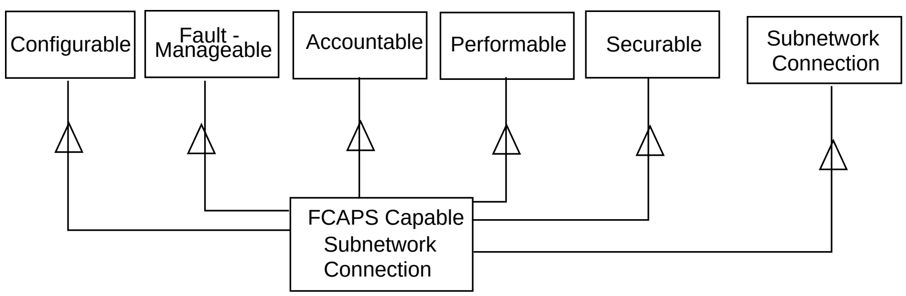


Figure 3-12. Possible manageable inheritance for resource object types

The current version of the NRIM specification defines objects only for the following three functional areas: Configuration management, Fault management, and Accounting management.

3.6 Domain Concepts

In network management, the concept of a *domain* is typically used to refer to a collection of resources that have some common properties with regard to their management. The current NRIM specification defines two types of domains:

- **Administrative Domain**: An administrative domain is a collection of resources that are the under control of a single network administration. (Thus, this notion is associated with ownership of resources.) This notion is fundamental since a global telecommunication network will have parts under the control of different administrations, and these administrations have to cooperate among them (exchange information and offer management capabilities) to provide services to customers. To cope with the complexities (technical and business complexities) of large scale networks, a network administration should have a detailed view of resources in its domain, and only an abstracted (or summarized) view of resources in other administrative domains. In the current NRIM specification, the concept of administrative domain is applied in two levels: at the level of the connectivity layer network, and at the level of layer networks. The portion of the connectivity layer network that is under the control of a single administration is called a *connectivity provider domain*. The portion of a layer network that is under the control of a single administration is called a *layer network domain*. The reason for defining administrative domains in two levels is that it facilitates the modelling of the relationships between layer networks, and between layer networks and a connectivity layer network.

- **Management Domain**: A management domain is a collection of resources within an administration domain that are under the control of a management function, such as connection management function or fault management function. Note that a management domain does not span administrative domains. An administrative domain may consist of one or more management domains. A resource may belong to multiple management domains but always belongs to only one administrative domain.

4. Network Fragment

4.1 Introduction

The Network Fragment of NRIM defines information objects and relationships that are used to model the topological structure of a TINA network from the perspective of one TINA stakeholder (administrative domain) playing the connectivity provider business role. Typically, different portions of a TINA network will be under the control of different network administrations, and the structure of the TINA network seen by each administration may differ.

From the perspective of a connectivity provider, a TINA network is seen as being composed of a *Connectivity Layer Network* (CLNW) to which CPEs are attached. The connectivity layer network that makes up a TINA network contains a number of *Layer Networks* (LNWs) and access points to them. The connectivity layer provides an aggregated view of layer networks that are interconnected by means of interworking units, or bridges and is thus able to support connectivity between network flow end points of different characteristic information. The connectivity layer network concept generalizes the layer network concept of G.803 and it is needed for describing connectivity between termination points of different characteristic information. The motivation for defining the notion of a connectivity layer is to define a model of a transport network that spans multiple communication technologies (ATM, N-ISDN, wireless, etc.). Layer networks in a connectivity layer have a peer to peer relationship or client server relationship.

Each layer network of a connectivity layer network represents a set of compatible inputs and outputs that may be interconnected and is characterized by the characteristic information that is transported across the layer network. The access points of a layer network at which the layer network accepts and/or delivers information are called *Network Trail Termination Point*s (NWTTPs) and they are further described in the Termination Point Fragment (Section 6).

Different portions of a layer network may be under the control of different network administrations. From the perspective of a connectivity provider, a layer network is seen as being made up of one or more *Layer Network Domain*s (LNDs).[1] The portion that is controlled by the connectivity provider is called the *Local Layer Network Domain* (LLND), and each portion that is under the control of another connectivity provider is called a *Foreign Layer Network Domain* (FLND).

The topological structure of a LND is represented using two kinds of topological components: *Subnetworks* and *Links*. A LND consists of a subnetwork (referred to as the top level subnetwork) on which one or more links terminate. Each such link interconnects the top level subnetwork of a LND with either the top level subnetwork of another LND or a CPE. The top level subnetwork of a LND may in turn be composed of two or more subnetworks (referred to as lower level subnetworks) interconnected by links. This subnetwork decomposi-

1. It is important to note that the structure of a layer network in terms of layer network domains seen by one connectivity provider, will in general, be different from the layer network structure seen by other connectivity providers. This “relativistic” view is a necessary consequence of differing business arrangements between the different connectivity providers.

tion may occur in multiple levels until the subnetwork maps directly to a network element (NE). Thus, a link is a topological component used for representing an interconnection between a subnetwork and either another subnetwork or a CPE. A link can be configured in one of the following ways:

- Using one or more topological links: A *Topological Link* is a logical or physical transmission path interconnecting two NEs (more generally two subnetworks) and it is supported by a Trail in the server layer network. A topological link has a specific bandwidth capacity which is determined by the bandwidth of the server layer trail. A link can be configured by assigning either a portion of the bandwidth or the entire bandwidth of one or more topological links. Thus, the relationship between links and topological links is many-to-many. At link configuration time, the link connections associated with the link may or may not be established. This is a technology dependent matter. In ATM, a link represents the potential for link connections, and link connections are not created at link configuration time. In SDH, a link usually represents a bundle of link connections, and in this case, the link connections that make up the link are created at link configuration time.
- Using two or more links: In this case, a link is configured as an aggregation of two or more links. A link that is such an aggregation of links is called a composite link. Composite links are very useful in representing aggregate capacity of the interconnections between two subnetworks, such as for example, between the top level subnetworks in two neighbour LNDs.

The end points of a link are called *Link Termination Points (LTPs)* and the end points of a topological link are called *Topological Link Termination Points (TLTPs).* A LTP is configured using one more TLTPs. A LTP represents either the potential for several link connection terminations or a group of existing link connection terminations. A termination point of a link connection is called a *Network Connection Termination Point* (NWCTP).

The access points of a connectivity layer network at which the connectivity layer network accepts and/or delivers information are called *Network Flow Endpoint*s (NFEPs) and they are further described in Section 6. Unlike a NWTTP or a NWCTP, a NFEP does not represent any transport processing function (such as trail integrity monitoring or adaptation) and is used only to represent in a generic (i.e., technology independent) manner an extremity of a network flow connection that may span multiple layer networks. A NFEP has a one-to-one association with a NWTTP. A topological point on the boundary of a connectivity layer network (i.e., resident in a CPE) that represents the potential for several network flow terminations to occur at the point is called a *Network Flow Endpoint Pool* (NFEPPool). A NFEPPool is a collection of LTPs, and the LTPs grouped under a NFEPPool may span different layer network domains of the same connectivity provider.

4.2 Overview of the Network Fragment

The object types defined in the network fragment are listed and briefly described in Table 4-1.

Table 4-1. Object types defined in the Network Fragment

Object Types	Description
Network	Represents the TINA network from the perspective of a connectivity provider.
Connectivity Layer Network (CLNW)	Represents the connectivity layer network that makes up the TINA network from the perspective of a connectivity provider.
Layer Network (LNW)	Represents a layer network that is a component of the connectivity layer network.
Layer Network Domain (LND)	Represents the part of a layer network that is under the control of one administrative domain. This is a noninstantiable supertype and is used only for inheritance. A layer network domain consists of a top level subnetwork and a set of links.
Local Layer Network Domain (LLND)	Represents the part of a layer network that is under the control of the local administrative domain. Subtype of Layer Network Domain.
Foreign Layer Network Domain (FLND)	Represents from the perspective of a connectivity provider the part of a layer network that is under the control of a foreign network administration.
Subnetwork (SNW)	Represents an interconnected group of network elements or subnetworks that is entirely within one layer network domain. A subnetwork can be partitioned into a number of subnetworks and links.
Topological Link (TL)	Represents a logical or physical transmission path that directly interconnects either two subnetworks. or a subnetwork and a CPE. A topological link is configured from a trail at the server layer.

Table 4-1. Object types defined in the Network Fragment

Object Types	Description
Link (L)	Represents either the potential for link connections between two subnetworks or a bundle of link connections that have been provisioned to transport information between either two subnetworks or a subnetwork and a CPE. A link is configured by either using one or more topological links or aggregating two or more links.
Customer Premises Equipment (CPE)	Represents a terminal equipment, such as a computer, a phone, or an audio/video equipment attached to the connectivity layer network.

A brief explanation is in order regarding the CPE object. This object has been introduced so that end-to-end connectivity, stream flow connections and stream bindings, can be represented in the view of a connectivity provider. It is not suggested that a connectivity provider is aware of all CPEs included in the TINA network. Typically, a connectivity provider is aware of all CPEs attached to its domain, and becomes aware of CPEs attached to other connectivity provider domains during the course of a stream flow setup. Such “foreign” CPEs will dynamically appear and disappear (i.e., transient objects) in the view of the connectivity provider.

The relationship types defined in the network fragment are listed and briefly described in Table 4-2.

Table 4-2. Relationships defined in the Network Fragment

Relationship	Description
ServedByTrail	Relates a topological link that is a part of a local layer network domain with the trail in a server layer network domain (local or foreign) that serves the topological link.
ServedByTL	Relates a link with a topological link that has been used to configure the link.

Apart from the specific relationships defined above, the following generic relationships are also defined in this fragment:

- Aggregation relationship between Network (composite) and Connectivity Layer Network (component)
- Aggregation relationship between Network (composite) and CPE (component)
- Aggregation relationship between Connectivity Layer Network (composite) and Layer Network (component)
- Aggregation relationship between Layer Network (composite) and Local Layer Network Domain (component)
- Aggregation relationship between Layer Network (composite) and Foreign Layer Network Domain (component)
- Aggregation relationship between Layer Network Domain (composite) and Subnetwork (component)
- Aggregation relationship between Layer Network Domain (composite) and Link (component)
- Aggregation relationship between Subnetwork (composite) and Subnetwork (component)
- Aggregation relationship between Subnetwork (composite) and Link (component)
- Aggregation relationship between Link (composite) and Link (component)
- Inheritance relationship between Layer Network Domain (supertype) and Local Layer Network Domain (subtype)
- Inheritance relationship between Layer Network Domain (supertype) and Foreign Layer Network Domain (subtype)
- Inheritance relationship between AdministrativeDomain (supertype) and Layer Network Domain (subtype). See Section 7 for a description of the object type AdministrativeDomain.

4.3 OMT Diagram for the Network Fragment

The OMT diagram for the Network Fragment is shown in Figure 4-1. Object types and relationships that are shown in these figures and that are not defined in this fragment are defined in the Termination Point Fragment (Section 6).

The constraints C1 through C4 labelled in the OMT diagram are described below:

- C1: A SNW object participates in exactly one of the two relationships
- C2: A L object participates in exactly one of the two relationships
- C3: A TLTP object participates in exactly one of the two relationships
- C4: If a L and a TL are related by the ServedByTL relationship, it implies that the TLTP related to the TL via the ExtremityOfTopLink relationship, and the LTP related to the L via the ExtremityOfLink relationship are related by the ServedByTLTP relationship.

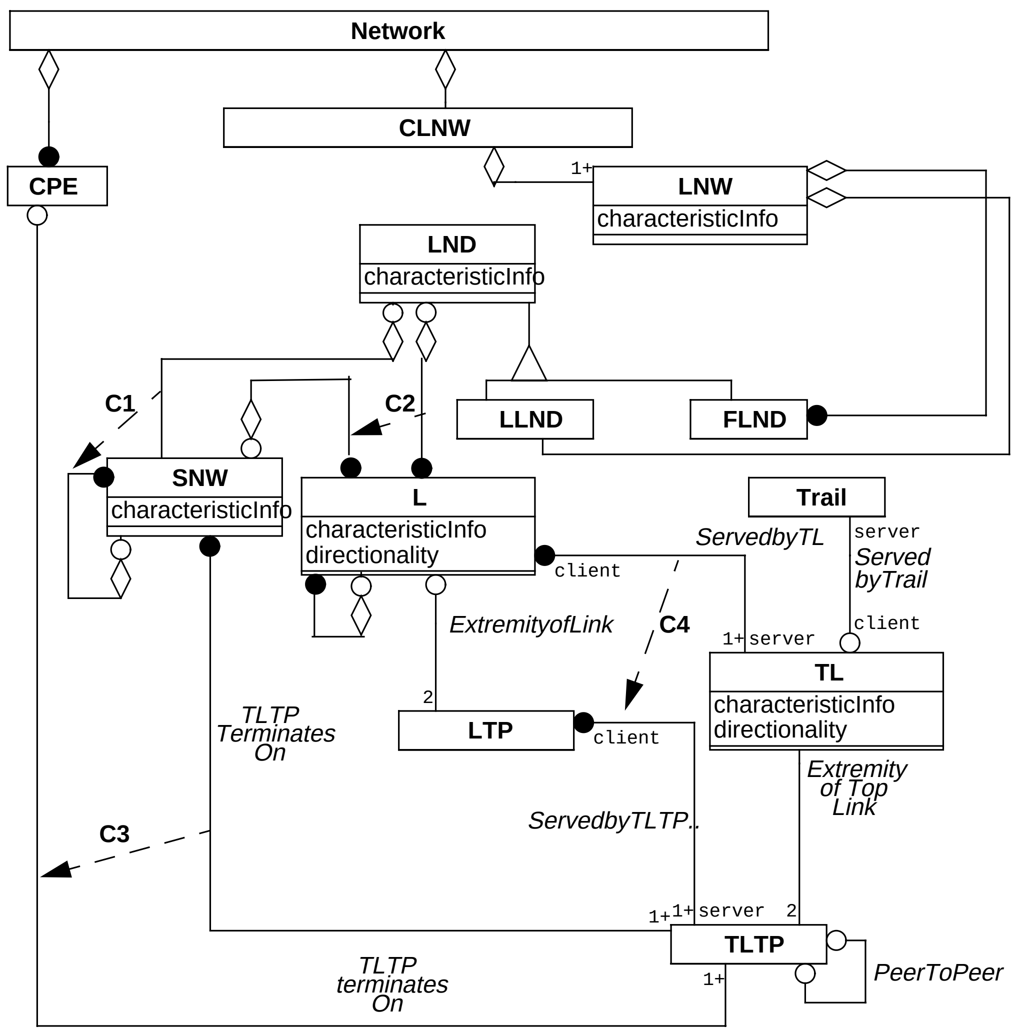


Figure 4-1. OMT Diagram for Network Fragment

4.4 An Illustrative Example

Figure 4-2 illustrates an example TINA network configuration consisting of two connectivity provider domains, referred to as Domain 1 and Domain 2. In the figure, CPEs are named CPE1 to CPE6, switches are named A to H, and topological links are named TL1 to TL15.

Domain 1 has the following network components:

- A frame relay network (LND 1) consisting of two switches, A and B
- An ATM Virtual Path (VP) network (LND 2), consisting of three switches, C, D, and E, and an inter-carrier topological link that connects to an ATM switch in Domain 2
- A CPE (CPE 1) attached to the frame relay network
- A CPE (CPE 2) attached to both the frame relay network and the ATM network
- A CPE (CPE 3) attached to the ATM network
- A CPE (CPE 4) attached to the ATM network
- An interworking unit that performs peer-to-peer adaptation between the frame relay network and the ATM network. The interworking unit is a part of the ATM switch C. Thus, switch C is a part of both the frame relay network and the ATM network.
- An SDH network (LND 3) that serves as the transmission network interconnecting the switches and the CPEs. (The network elements of the SDH network have not been shown in Figure 4-2 to simplify the figure).

Domain 2 has the following network components:

- An ATM Virtual Path (VP) network (LND 4), consisting of three switches, F, G, and H
- Two CPEs (CPE 5 and CPE 6) attached to the ATM network
- An SDH network (LND 5) that serves as the transmission network interconnecting the switches and the CPEs. (The network elements of the SDH network have not been shown in Figure 4-2 to simplify the figure).

The business arrangement between the two connectivity providers is such that Provider 1 does not offer to Provider 2 access to the Frame Relay network contained in Domain 1 (possibly because the ATM-Frame Relay interworking is just for a trial within Domain 1).

Figure 4-3 illustrates the topology view of Connectivity Provider 1. Figures 4-5 to 4-8 illustrate the instance diagrams corresponding to the topological view of Connectivity Provider 1. Figure 4-4 illustrates the topology view of Connectivity Provider 2. Figures 4-9 to 4-11 illustrate the instance diagrams corresponding to the topological view of Connectivity Provider 2. The “relativistic” nature of the topological view is apparent in these figures. Notice that a connectivity provider sees only the top level subnetwork of a layer network domain in a foreign administration, and does not see the decomposition of the top level subnetwork.

In the view of Provider 1, the TINA Network is composed of three layer networks (SDH, Frame Relay, and ATM VP layer networks). The SDH and Frame Relay layer networks are completely contained in Domain 1 while the ATM VP layer network has a portion that belongs to Domain 2. (See Figure 4-3). Each layer network domain contained in Domain 1 (LND 1, LND 2, and LND3) has a two level partitioning of subnetworks. That is, each layer network domain consists of a top level subnetwork that is further decomposed into several lower level subnetworks, each corresponding to a switch. To simplify the figures, the lower level subnetworks are not shown for the SDH layer network. Each topological link, except the inter-domain topological link (TL11), is used to configure one link. To facilitate bandwidth management, two links have been configured using the topological link TL11 that interconnects Domain 1 and Domain 2. In this arrangement, bandwidth of one link is managed by Domain 1 and bandwidth on the other link is managed by Domain 2.

In the view of Provider 2, the TINA Network is composed of two layer networks (SDH and ATM VP layer networks). The SDH layer network is completely contained in Domain 2 while the ATM VP layer network has a portion that belongs to Domain 1. (See Figure 4-4). Each layer network domain contained in Domain 2 (LND4 and LND5) has a two level partitioning of subnetworks. The lower level subnetworks of the SDH layer network are not shown in the figures. Each topological link in Domain 2 is used to configure one link.

The peer-to-peer adaptation between the Frame Relay network (LND1) and the ATM network (LND2) is represented in the topological view in the following manner (see Figure 4-6 and Figure 4-8). The topological link TL5 that interconnects the Frame Relay switch B and the ATM switch C is represented as a component of LND1. TL5 transports frames (not ATM cells). A TLTP object, TLTP181, represents the extremity of TL5 on the ATM side and is associated with the SNW object that represents the ATM switch C. TLTP181 also represents the potential for Frame Relay connection terminations on the ATM switch C. Another TLTP object, TLTP182, represents the potential for ATM VP connection terminations on the ATM switch C. The objects TLTP181 and TLTP182 are related by the PeerToPeer relationship. This relationship represents the potential for the adaptation of Frame Relay trails and ATM VP trails in the ATM switch C. The actual adaptation of trails is represented using a PeerToPeer relationship between Frame Relay and ATM VP Network Trail Termination Points. See Section 6 for further details on this relationship.

The following naming convention is used for objects in the instance diagrams (this naming scheme is used only for illustration). Each LND, CPE, and TL object is assigned a unique integer as a suffix which is the same as the number used in Figure 4-2 to denote the corresponding resource. Other objects are named as follows. Each object is assigned a two part integer as a suffix; the first part is derived from the suffix of the corresponding LND object; and the second part is a unique number that distinguishes the object from other instances of the same object type in the instance diagrams pertaining to the LND.

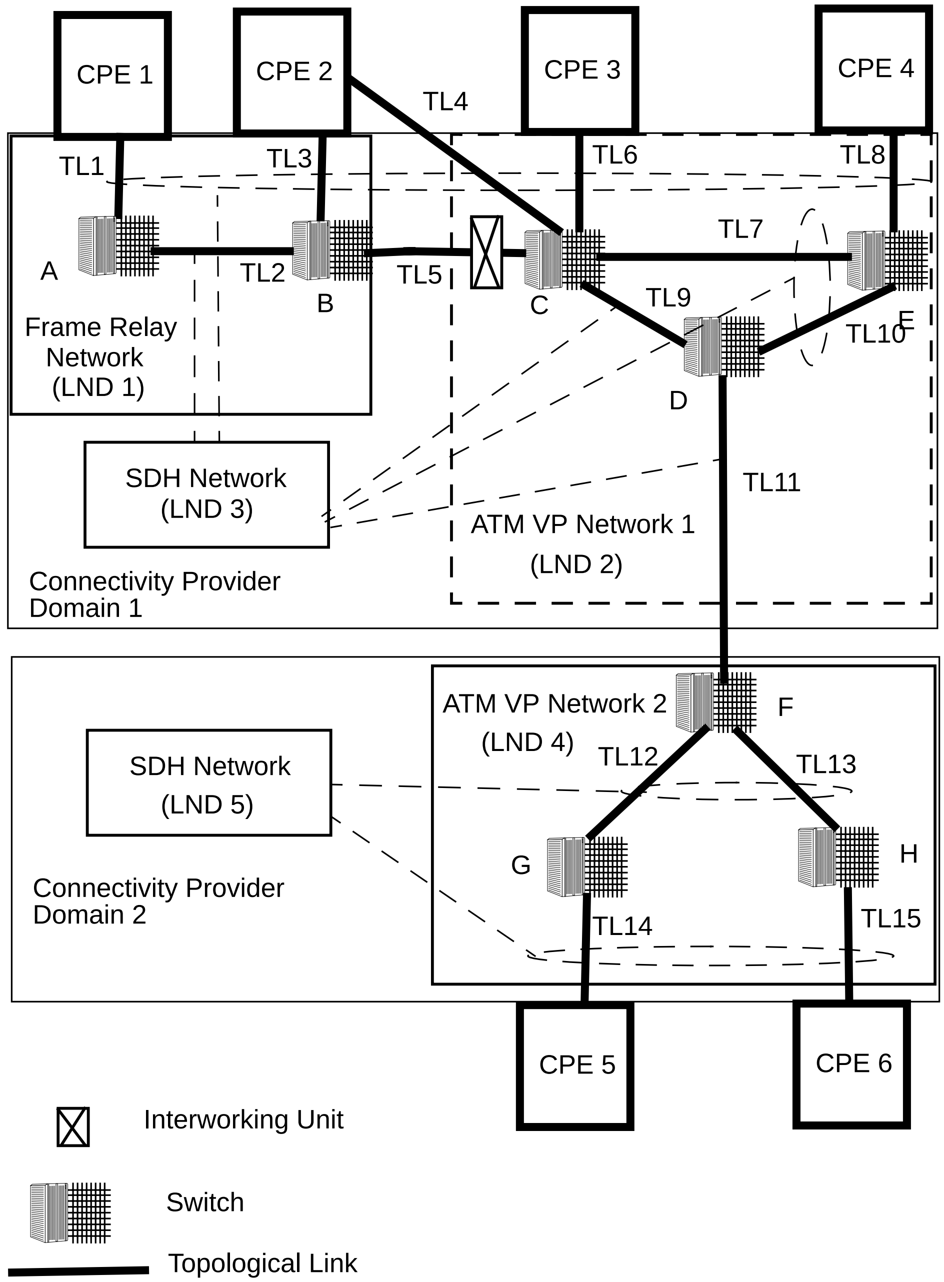


Figure 4-2. An Illustrative TINA Network

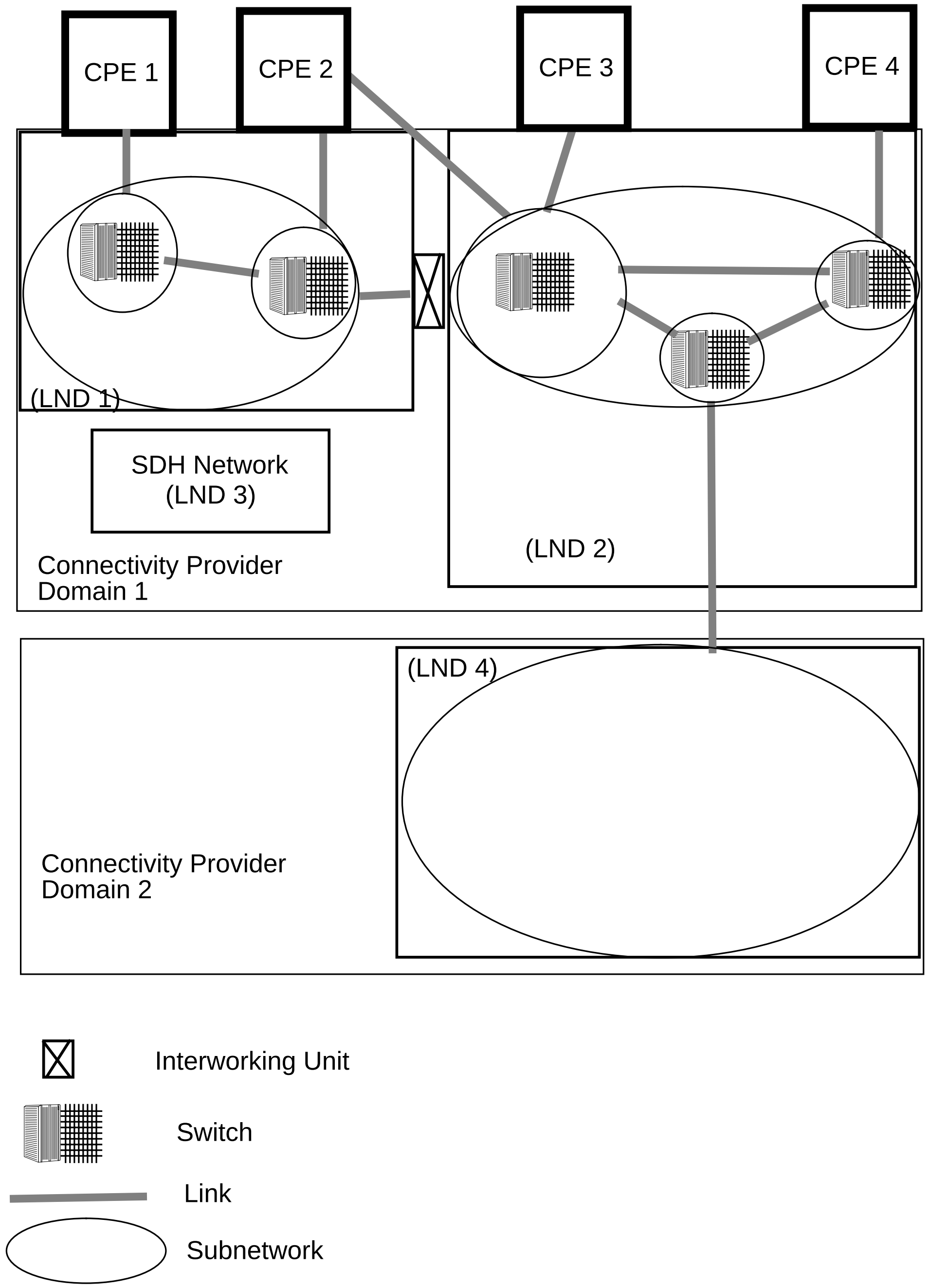


Figure 4-3. Topology View of Provider 1

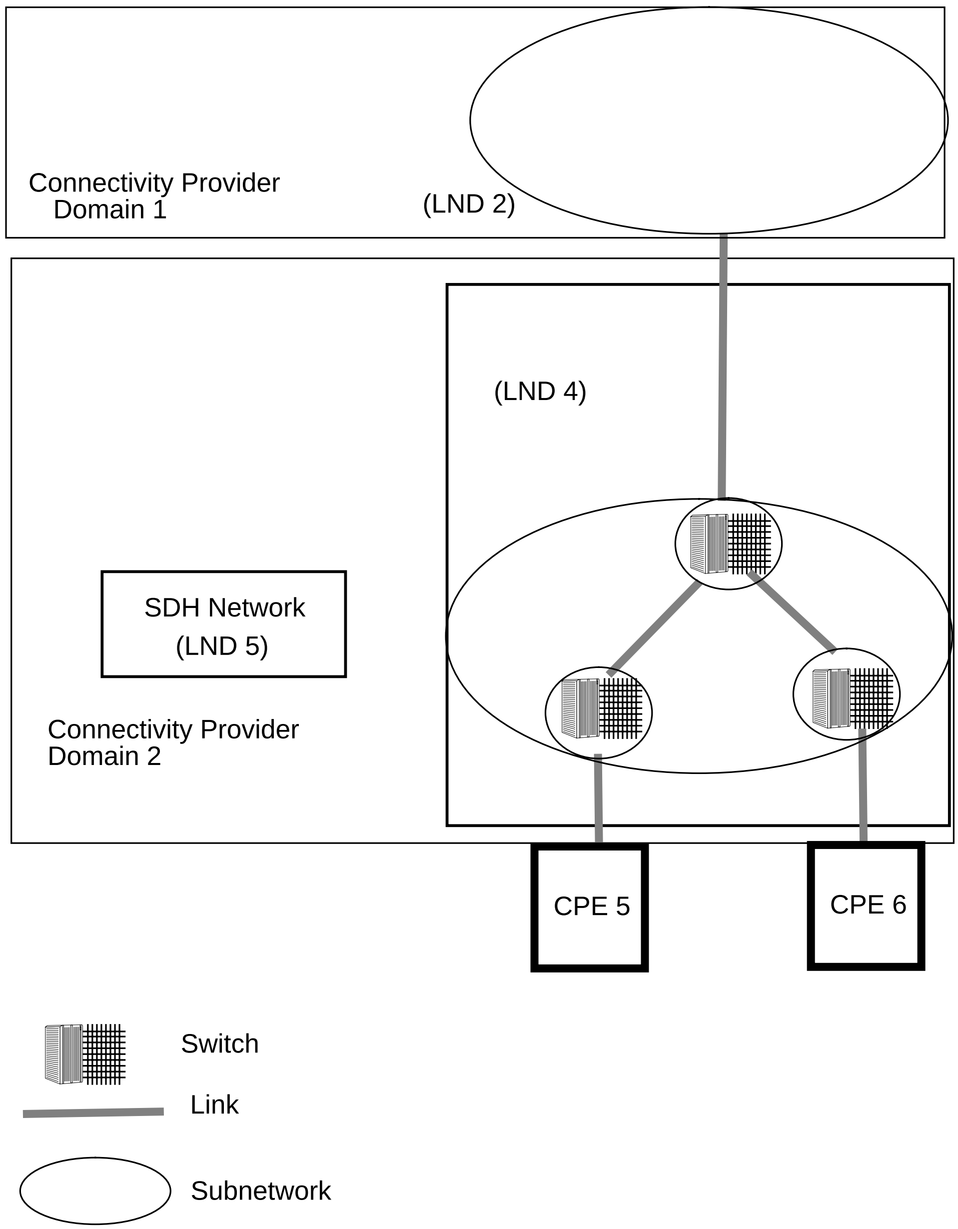


Figure 4-4. Topology View of Provider 2

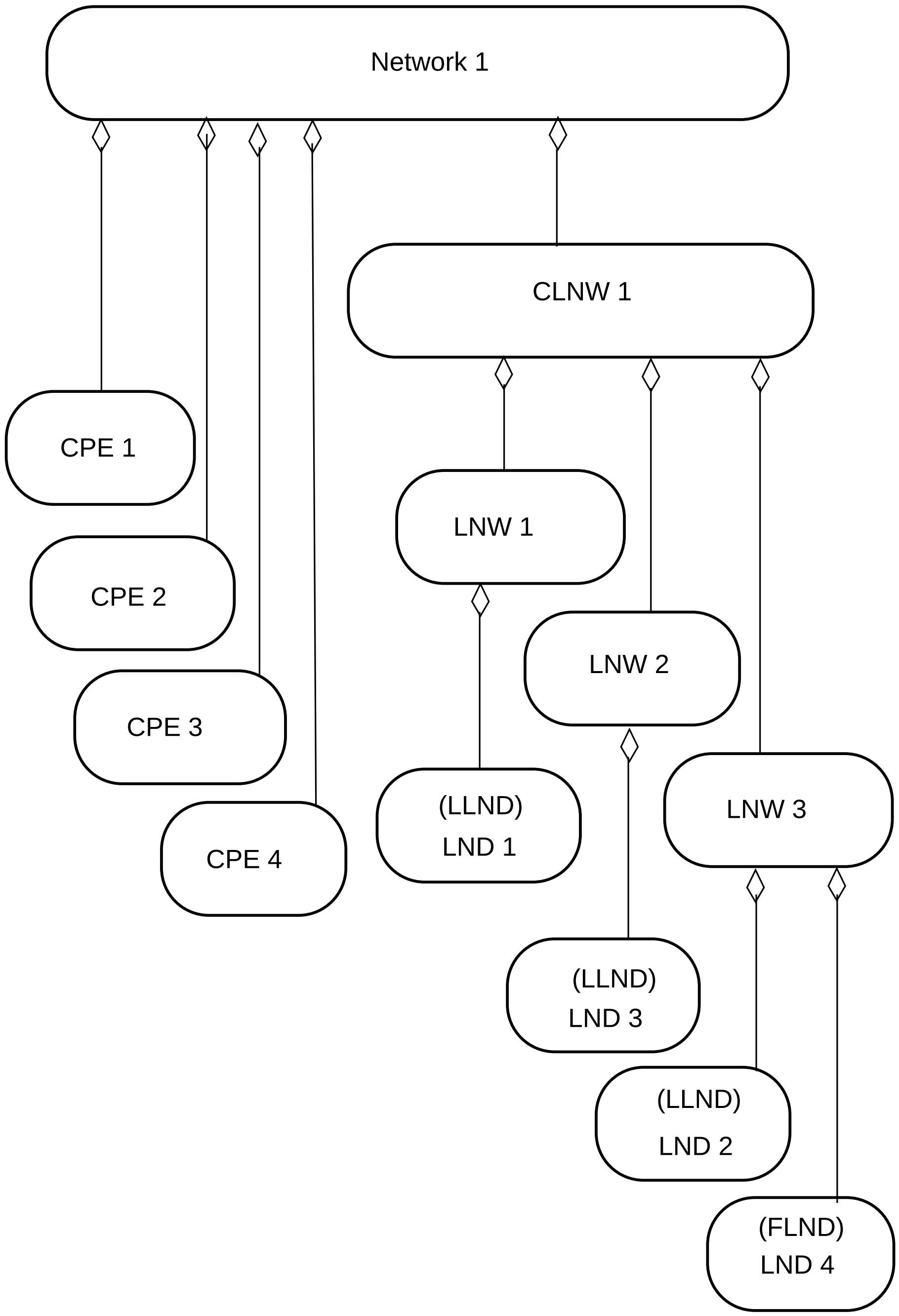


Figure 4-5. Instance Diagram for Topology View of Provider 1 (Part 1)

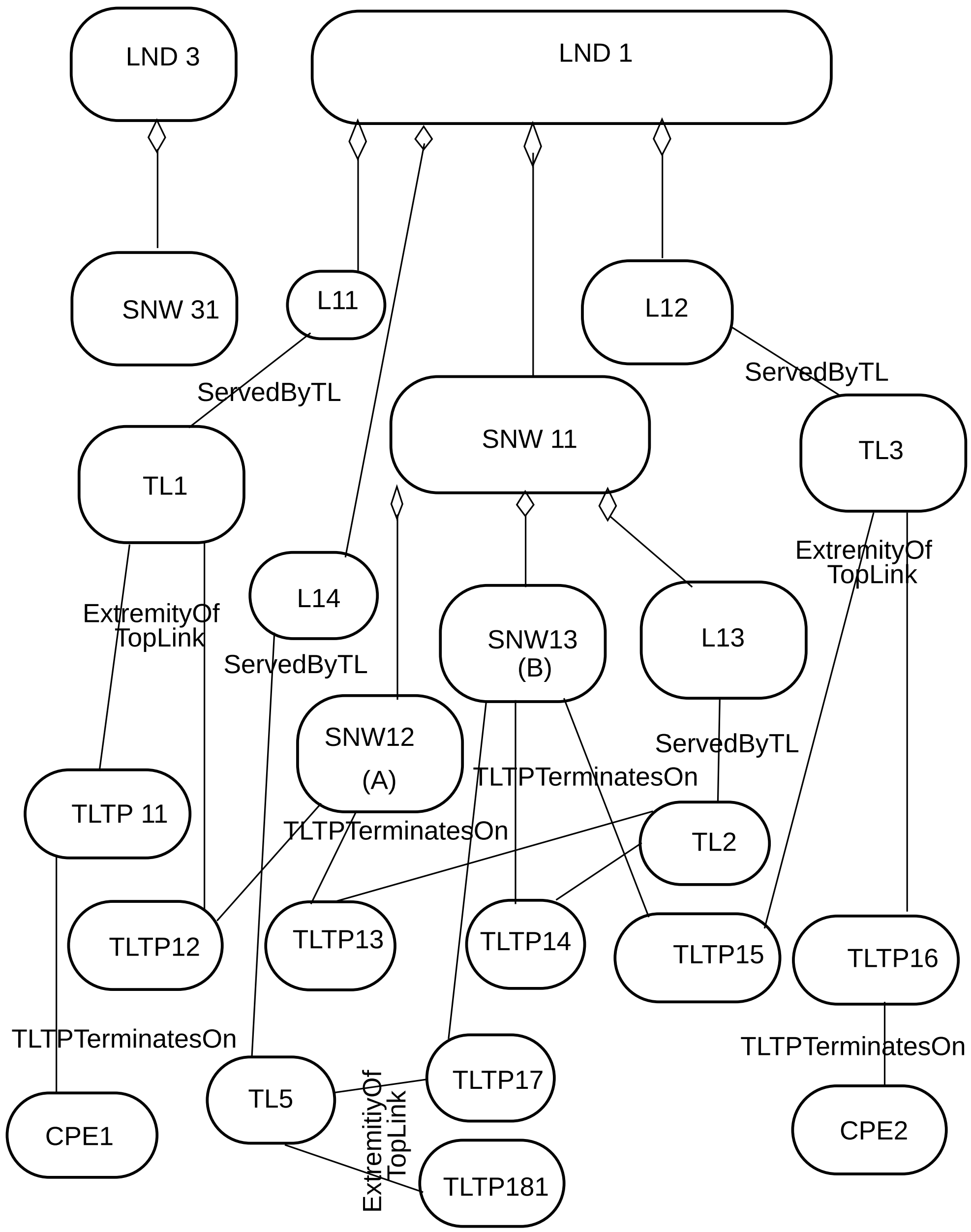


Figure 4-6. Instance Diagram for Topology View of Provider 1 (Part 2)

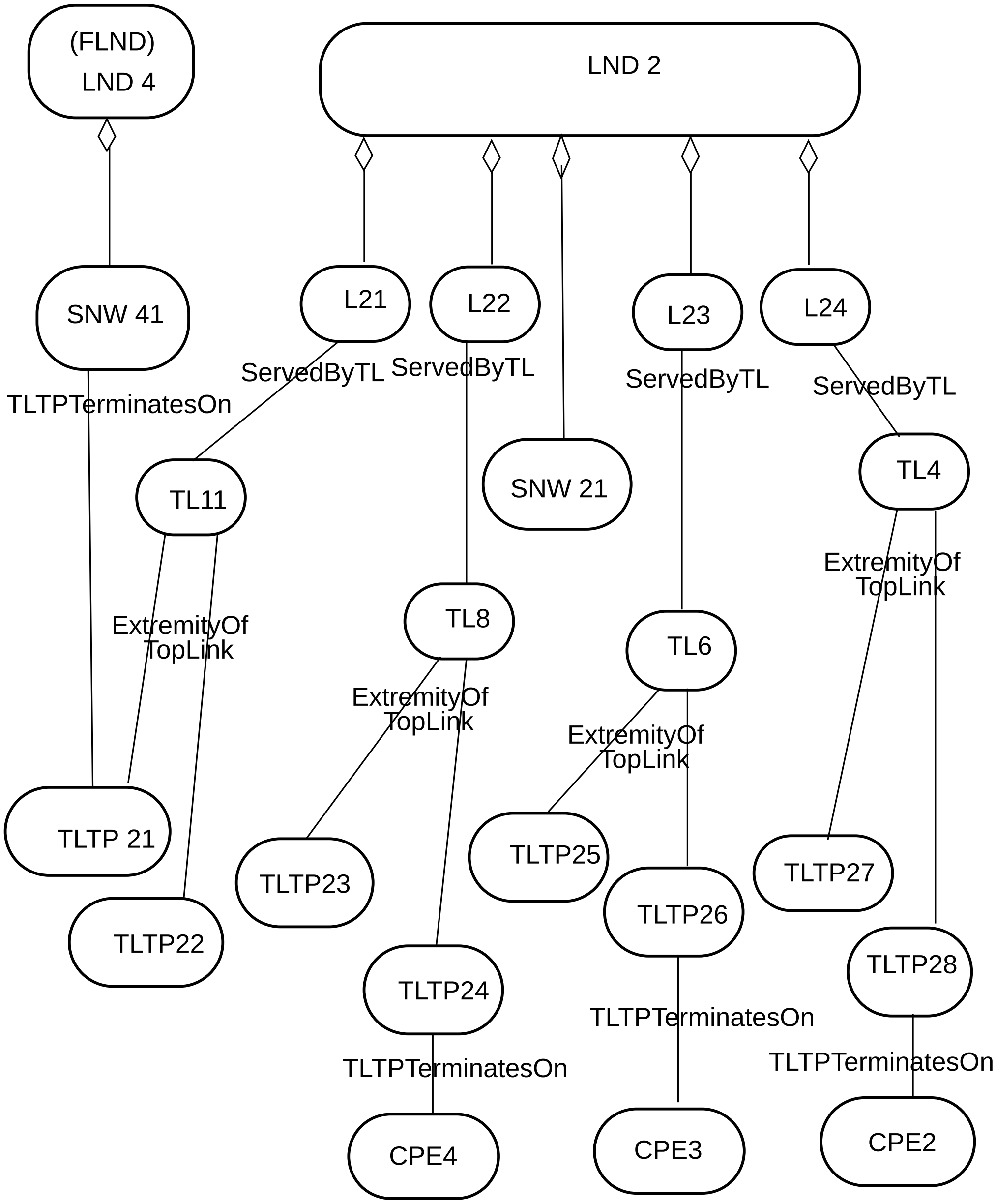


Figure 4-7. Instance Diagram for Topology View of Provider 1 (Part 3)

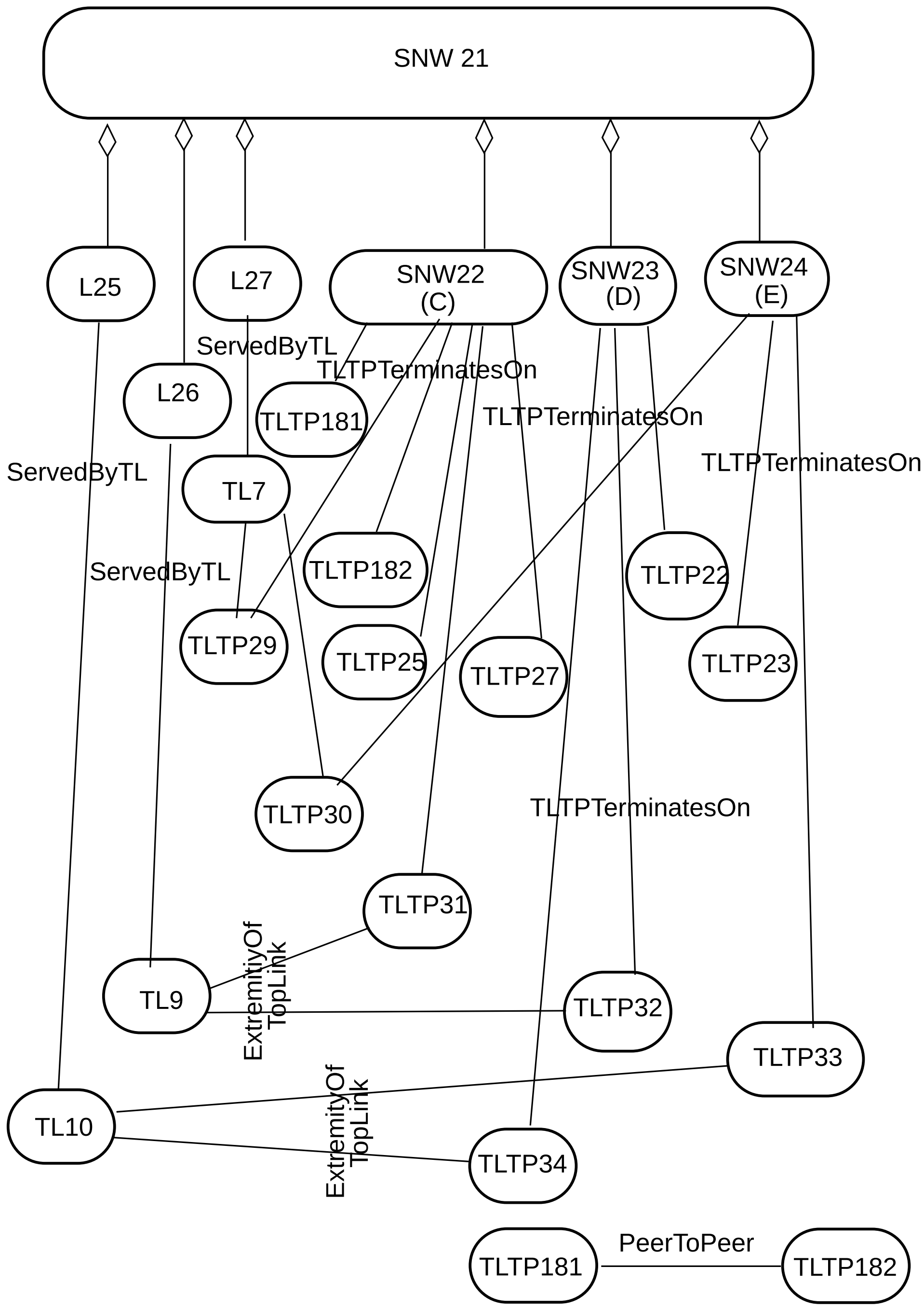


Figure 4-8. Instance Diagram for Topology View of Provider 1 (Part 4)

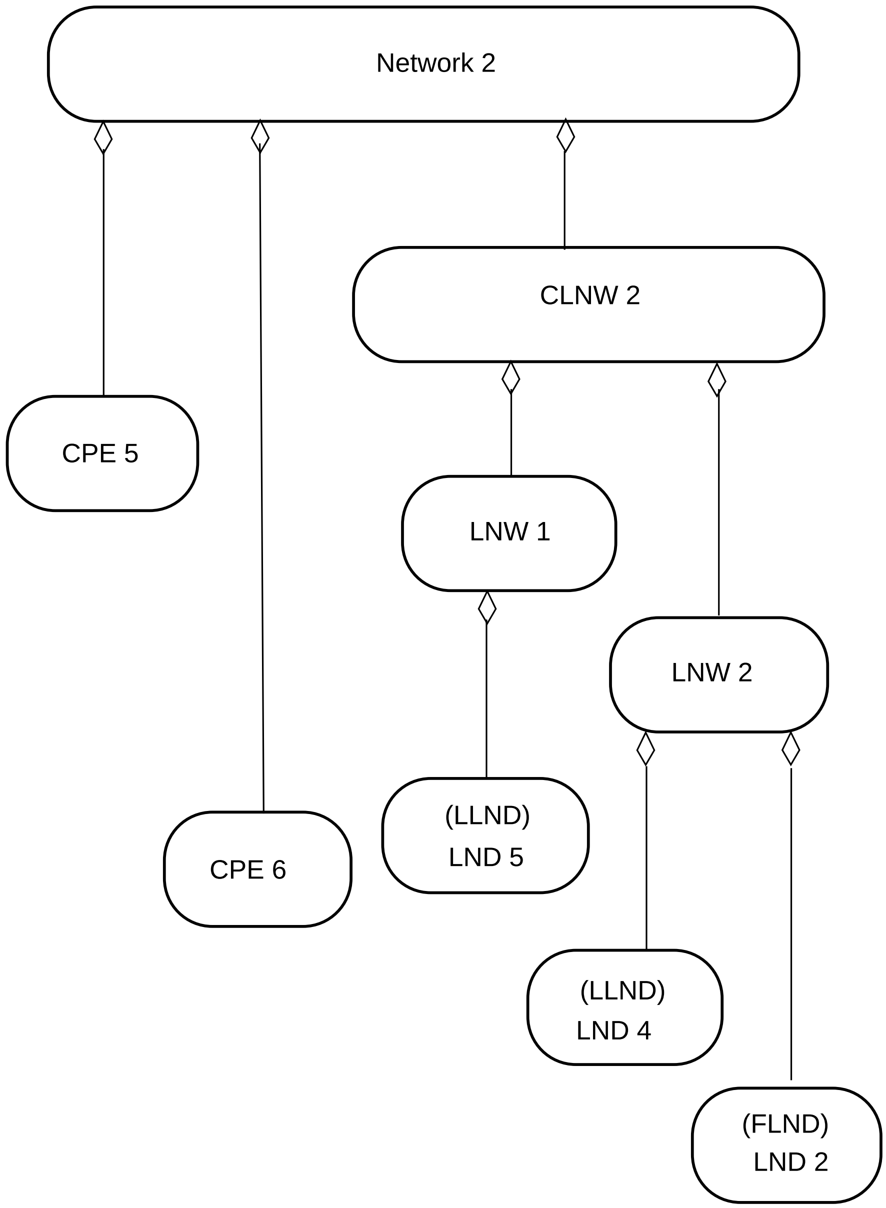


Figure 4-9. Instance Diagram for Topology View of Provider 2 (Part 1)

.

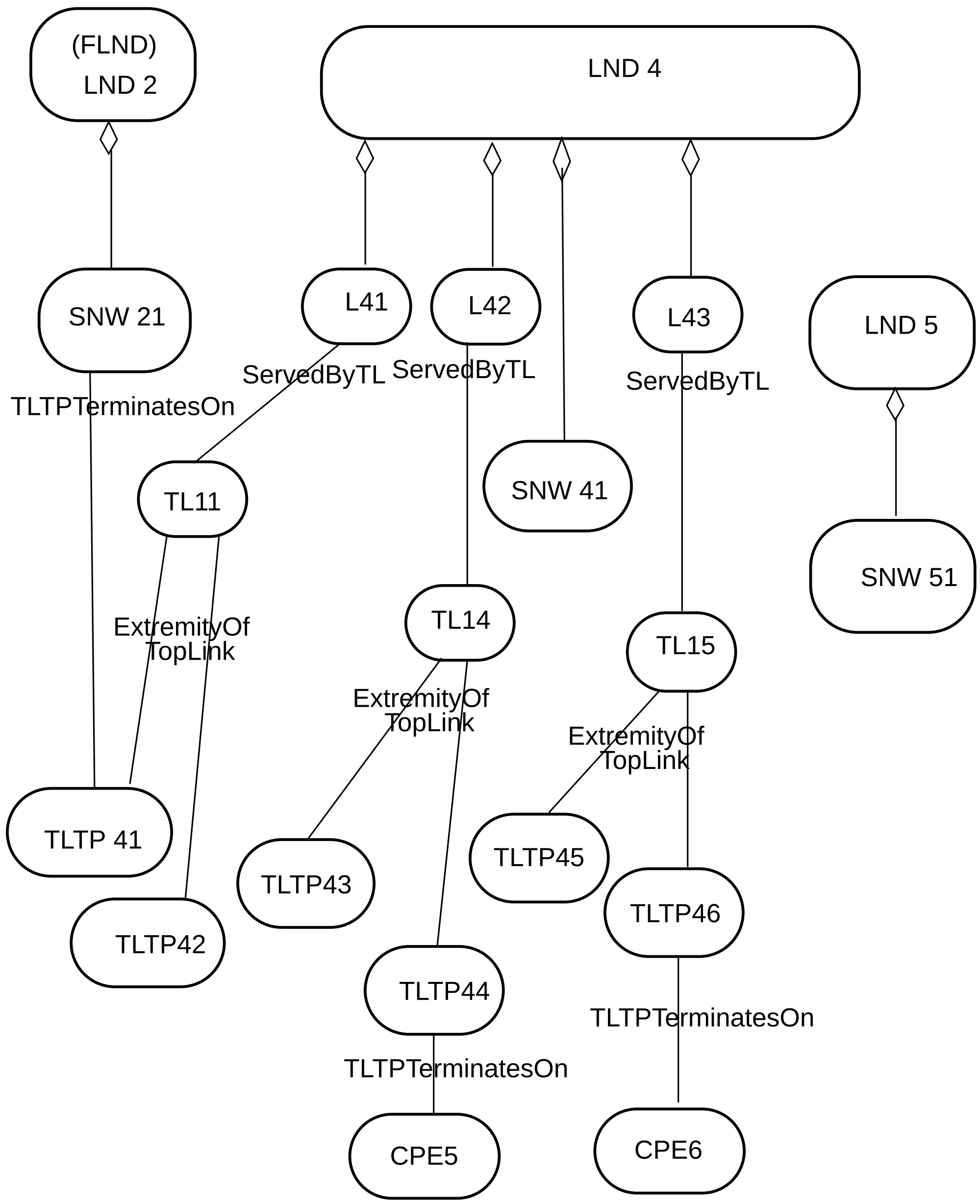


Figure 4-10. Instance Diagram for Topology View of Provider 2 (Part 2)

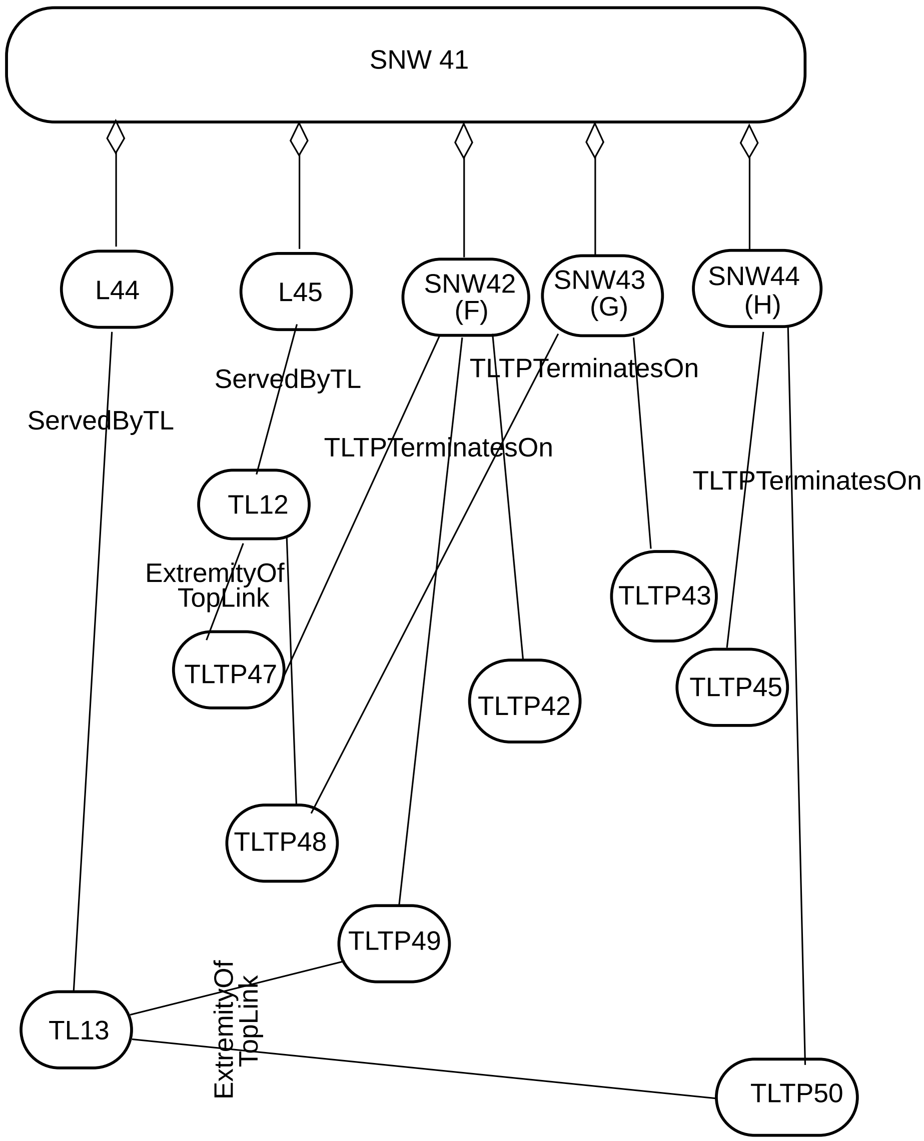


Figure 4-11. Instance Diagram for Topology View of Provider 2 (Part 3)

4.5 Quasi-GDMO+GRM Definition of the Network Fragment

4.5.1 Object Types

4.5.1.1 Network

```
Network OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY Network-package PACKAGE
  BEHAVIOUR Network-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: This object type represents a TINA network from the
        perspective of a connectivity provider.
      ";
  ATTRIBUTES;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

4.5.1.2 ConnectivityLayerNetwork

```
ConnectivityLayerNetwork OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY ConnectivityLayerNetwork-package PACKAGE
  BEHAVIOUR ConnectivityLayerNetwork-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: This object type represents the connectivity layer network
        that makes up the TINA network from the perspective of a connectivity
        provider.
      ";
  ATTRIBUTES;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

4.5.1.3 LayerNetwork

```
LayerNetwork OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY LayerNetwork-package PACKAGE
  BEHAVIOUR LayerNetwork-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: This object type represents a layer network that is a
        component of the connectivity layer network that makes up the TINA
        network.
      The attribute characteristicInfo specifies the characteristic
        information transported by the layer network.
```

```
    ";
  ATTRIBUTES
    characteristicInfo
      PERMITTED VALUES: CharacteristicInfo
      GET;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

4.5.1.4 LayerNetworkDomain

```
LayerNetworkDomain OBJECT TYPE
  DERIVED FROM AdministrativeDomain;
  CHARACTERIZED BY LayerNetworkDomain-package PACKAGE
  BEHAVIOUR LayerNetworkDomain-Behaviour BEHAVIOUR DEFINED AS
    "
    COMMENTS: This object type represents the part of a layer network that
      is under the control of one administrative domain. The information
      regarding the administrative domain is represented in the attributes
      derived from the supertype AdministrativeDomain. This object type is
      noninstantiable is used only for inheritance. The topology of a layer
      network domain consists of a subnetwork (called the top level
      subnetwork) to which zero or more links are attached.
    The attribute characteristicInfo specifies the characteristic
      information transported by the layer network domain.
    ";
  ATTRIBUTES;
    characteristicInfo
      PERMITTED VALUES: CharacteristicInfo
      GET;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

4.5.1.5 LocalLayerNetworkDomain

```
LocalLayerNetworkDomain OBJECT TYPE
  DERIVED FROM LayerNetworkDomain;
  CHARACTERIZED BY LocalLayerNetworkDomain-package PACKAGE
  BEHAVIOUR LocalLayerNetworkDomain-Behaviour BEHAVIOUR DEFINED AS
    "
    COMMENTS: This object type represents (in the view of a connectivity
      provider) the part of a layer network that is under the control of
      the connectivity provider. This object type is a subtype of
      LayerNetworkDomain.
    ";
  ATTRIBUTES;
```

```
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

4.5.1.6 ForeignLayerNetworkDomain

```
ForeignLayerNetworkDomain OBJECT TYPE
  DERIVED FROM LayerNetworkDomain;
  CHARACTERIZED BY ForeignLayerNetworkDomain-package PACKAGE
  BEHAVIOUR ForeignLayerNetworkDomain-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: This object type represents (in the view of a connectivity
        provider) the part of a layer network that is under the control of
        another connectivity provider. This object type is a subtype of
        LayerNetworkDomain.
      ";
  ATTRIBUTES;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

4.5.1.7 Subnetwork

```
Subnetwork OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY Subnetwork-package PACKAGE
  BEHAVIOUR Subnetwork-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: This object type represents an interconnected group of
        network elements or subnetworks that is entirely within one layer
        network domain. A subnetwork can be partitioned into two or more
        subnetworks interconnected by links.
      The attribute characteristicInfo specifies the characteristic
        information transported by the subnetwork.
      ";
  ATTRIBUTES;
      characteristicInfo
        PERMITTED VALUES: CharacteristicInfo
        GET;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

4.5.1.8 TopologicalLink

```
TopologicalLink OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY TopologicalLink-package PACKAGE
  BEHAVIOUR TopologicalLink-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: This object type represents a logical or physical
        transmission path that directly interconnects either two subnetworks
        or a subnetwork and a CPE. A topological link in a (client) layer
        network is configured using a trail in an underlying server layer
        network. A topological link has two ends, called Topological Link
        Termination Points (TLTPs). One of the two TLTPs is designated as
        the a-End and the other as the z-End.
      This object type has two attributes:
      characteristicInfo: This attribute specifies the characteristic
        information transported by the topological link.
      directionality: This attribute specifies the directionality of the
        topological link; the value can be either unidirectional or
        bidirectional. If unidirectional, information is transported only
        from the a-End to the z-End. If bidirectional, information is
        transported in either direction.
      Technology dependent specializations of this type may include
        additional information, such as
            totalIngressBandwidth: the provisioned capacity) for traffic
              from the a-End to the z-End of the topological link.
            totalEgressBandwidth: the provisioned capacity) for traffic
              from the z-End to the a-End of the topological link.
            ingressDelay: average and/or maximum delay in the transport of
              information from the a-End to the z-end.
            egressDelay: average and/or maximum delay in the transport of
              information from the z-End to the a-end.
      ";
  ATTRIBUTES
      characteristicInfo
        PERMITTED VALUES: CharacteristicInfo
        GET;
      directionality
      PERMITTED VALUES: LinkDirectionality
        GET;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

4.5.1.9 Link

```
Link OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY Link-package PACKAGE
  BEHAVIOUR Link-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: This object type represents either the potential for link
        connections between two subnetworks or a bundle of link connections
        that have been provisioned for transporting information between
        either two subnetworks or a subnetwork and a CPE.
      A link can be configured using topological links in any of the
        following ways:
        1:1 configuration: The link is configured by assigning the entire
        bandwidth of the topological link to the link.
        N:1 configuration: The link is configured by assigning only a portion
        of the bandwidth of a topological link to the link.
        1:N configuration: The link is configured by assigning the entire
        bandwidth of a set of topological links. The topological links may
        be terminating on different network elements.
      A link has two ends, called Link Termination Points (LTPs). One of the
      two LTPs is designated as the a-End and the other as the z-End.
      This object type has two attributes:
      characteristicInfo: This attribute specifies the characteristic
        information transported by the link.
      directionality: This attribute specifies the directionality of the
        link; the value can be either unidirectional or bidirectional. If
        unidirectional, information is transported only from the a-End to
        the z-End. If bidirectional, information is transported in either
        direction.
      Technology dependent specializations of this type may include
        additional information, such as
              totalIngressBandwidth: the provisioned capacity) for traffic
                from the a-End to the z-End of the link.
              totalEgressBandwidth: the provisioned capacity) for traffic
                from the z-End to the a-End of the link.
              ingressDelay: average and/or maximum delay in the transport of
                information from the a-End to the z-end.
              egressDelay: average and/or maximum delay in the transport of
                information from the z-End to the a-end.
      ";

  ATTRIBUTES
      characteristicInfo
        PERMITTED VALUES: CharacteristicInfo
        GET;
      directionality
      PERMITTED VALUES: LinkDirectionality
```

```
      GET;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

4.5.1.10 CPE

```
CPE OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY CPE-package PACKAGE
  BEHAVIOUR CPE-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: This object type represents a terminal equipment, such as a
       computer, phone, or an audio-video equipment attached to the
       connectivity layer network.
     ";
  ATTRIBUTES;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

4.5.2 Relationship Types

4.5.2.1 ServedByTrail

```
ServedByTrail RELATIONSHIP TYPE
  CHARACTERIZED BY ServedByTrail-package PACKAGE
  BEHAVIOUR ServedByTrail-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The ServedByTrail relationship type represents (in the view
     of a connectivity provider) the relationship between a topological
     link in a client layer network and a trail in a server layer network.
     The topological link is a part of a layer network domain of the
     connectivity provider while the trail supporting the topological link
     is under the control of either the same connectivity provider or
     another connectivity provider. The following rules govern the
     ServedByTrail relationship:
          1. The relationship has two roles: the role client is played by
             a TopologicalLink object and the role server is played by a
             Trail object.
          2. A TopologicalLink object A is related with a Trail object B
             if and only if the topological link represented by A is
             configured using the trail represented by B.
          3. A TopologicalLink object participates in exactly one
             ServedByTrail relationship.
          4. A Trail object participates in zero or one ServedByTrail
             relationship.";
  ROLE client
     RELATED TYPES TopologicalLink;
```

```
    ROLE CARDINALITY CONSTRAINT (0..1);
  ROLE server
    RELATED TYPES Trail;
    ROLE CARDINALITY CONSTRAINT (1..1);
REGISTERED AS ??;
```

4.5.2.2 ServedByTL

```
ServedByTL RELATIONSHIP TYPE
  CHARACTERIZED BY ServedByTL-package PACKAGE
  BEHAVIOUR ServedByTL-Behaviour BEHAVIOUR DEFINED AS
    "
    COMMENTS: The ServedByTL relationship type represents (in the view of
    a connectivity provider) the relationship between a link in a local
    layer network domain and a topological link in the same layer network
    domain that has been used to configure the link. The following rules
    govern the ServedByTL relationship:
         1. The relationship has two roles: the role client is played by
            a Link object and the role server is played by a
            TopologicalLink object.
         2. A Link object A is related with a TopologicalLink object B if
            and only if the link represented by A is configured using the
            topological link represented by B.
         3. A Link object participates in one or more ServedByTL
            relationships.
         4. A TopologicalLink object participates in zero or more
            ServedByTL relationships.";
  ROLE client
    RELATED TYPES Link;
    ROLE CARDINALITY CONSTRAINT (0..N);
  ROLE server
    RELATED TYPES TopologicalLink;
    ROLE CARDINALITY CONSTRAINT (1..N);
REGISTERED AS ??;
```

5. Connectivity Fragment

5.1 Introduction

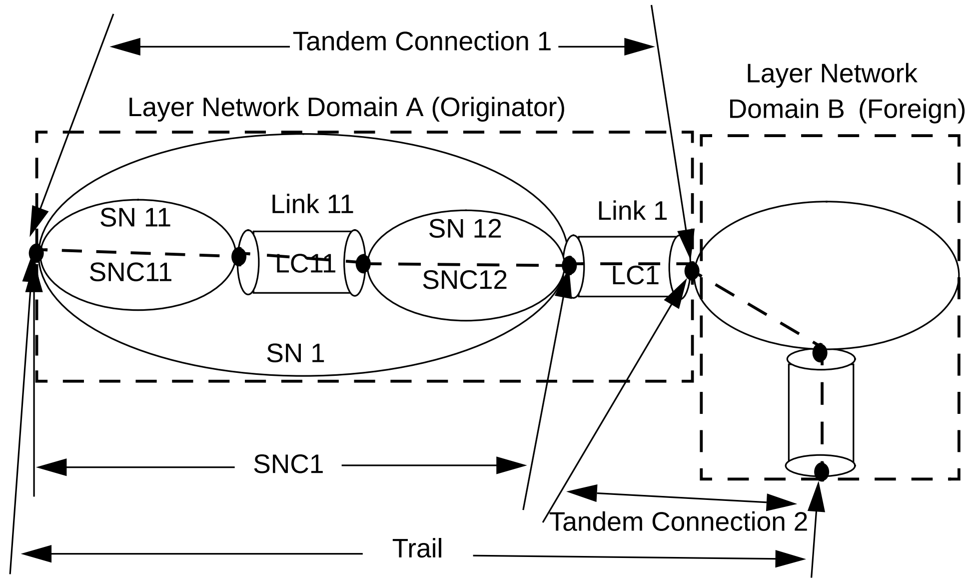
This fragment describes the information objects and relationships used to represent the connectivity resources of a TINA network. Section 3 introduced the different kinds of connectivity resources modelled in NRIM, namely, Stream Flow Connection, Network Flow Connection, Terminal Flow Connection, Trail, Tandem Connection, Subnetwork Connection, and Link Connection. This section presents the detailed definitions of information objects that represent these resources, and the relationships between these objects. As in the case of the Network Fragment, the model defined in the Connectivity Fragment is also from the perspective of one connectivity provider (administrative domain). Thus the model defined here represents the connectivity view as perceived by one connectivity provider, and does not represent a global view of a TINA network. This concept is illustrated in Figure 5-1.

Figure 5-1 illustrates a scenario where a layer network is made up of two layer network domains, and a trail spans the two layer network domains.[1] In NRIM, the view of a trail that spans multiple layer network domains is different in each domain. The view held by a domain depends on whether the domain originated the trail, i.e., initiated the trail setup. In Figure 5-1, LND A is the originator domain. The originator domain has an end-to-end view of the trail; i.e., it sees the trail as being made up of one tandem connection in its domain, and a tandem connection in the foreign layer network domain (LND B). The non-originator domain (LND B in this case) sees only the tandem connection in its domain, and does not have an end-to-end view of the trail.

A similar asymmetry in views held by domains occurs also in a situation where a tandem connection or a network flow connection spans multiple administrative domains.

As illustrated in the figure, a layer network domain has a detailed view of all components of the tandem connection in its domain (i.e., the subnetwork connections and the link connections that make up the tandem connection), but has only the abstracted end-to-end view of the tandem connection in the foreign domain. This is consistent with the topology view concept that was described in Section 4, where a connectivity provider does not see the partitioning structure of foreign layer network domains.

1. Such trails can be formed only after the participant connectivity providers (corresponding to the layer network domains) have established business arrangements to cooperate and provide connectivity services to each other.



.

Figure 5-1. View of a Trail in the Originator Layer Network Domain

5.2 Overview of the Connectivity Fragment

The connectivity fragment defines the object types listed in Table 5-1.

Table 5-1. Object types identified in the Connectivity Fragment

Object types	Description
Stream Flow Connection (SFC)	The resource that transfers information in a unidirectional manner between applications in a TINA network. The information is transported from a Source Stream Flow End Point to one or more Sink Stream Flow End Points.
Logical Connection Graph (LCG)	Represents a group of stream flow connections that have been grouped for some purpose. Representation of a communication session from the resource perspective.

Table 5-1. Object types identified in the Connectivity Fragment

Object types	Description
Network Flow Connection (NFC)	The resource that transfers information across the connectivity layer network of a TINA network. The information is transported between a group of Network Flow End Points. The connection topology may be point-to-point bidirectional, point-to-point unidirectional, or point-to-multipoint unidirectional.
Physical Connection Graph (PCG)	Represents a group of network flow connections that form the network part for a group of stream flow connections. Representation of a connectivity session from the resource perspective.
Terminal Flow Connection (TFC)	The resource that transports information within a CPE either from a Stream Flow End Point to a Network Flow End Point, or vice versa, or between two Stream Flow End Points.
Nodal Connection Graph (NCG)	Represents a group of terminal flow connections in a CPE that form a terminal part for a group of stream flow connections. Representation of a terminal communication session from the resource perspective.
Trail (T)	The resource that transfers information between two or more endpoints of a layer network. The end points are called Network Trail Termination Points (NWTTPs). The connection topology may be point-to-point bidirectional, point-to-point unidirectional, or point-to-multipoint unidirectional.
Link Connection (LC)	The resource that transfers information between two Network Connection Termination Points (NWCTPs) that are either in two adjacent subnetworks or in a subnetwork and a CPE. The connection topology is either point-to-point bidirectional or point-to-point unidirectional.

Table 5-1. Object types identified in the Connectivity Fragment

Object types	Description
Subnetwork Connection (SNC)	The resource that transfers information across a subnetwork. The end points of a subnetwork connection are called Edges. Each Edge is bound to a NWCTP. The connection topology may be point-to-point bidirectional, point-to-point unidirectional, or point-to-multipoint unidirectional.
Tandem Connection (TC)	A tandem connection represents a portion of a trail that either exists in a local layer network domain or spans one or more foreign layer network domains. The endpoints of a tandem connection are NWTTPs or NWCTPs. From the perspective of a LND, a tandem connection consists of a subnetwork connection across the top level subnetwork of the local LND, zero or more link connections that are cross-connected by the subnetwork connection, and zero or more tandem connections in one or more foreign LNDs.

The relationships between connectivity objects defined in this fragment are listed in Table 5-2. Some of the relationships are between connectivity objects and objects defined in other fragments (network or termination point fragment). The reason for including relationships to objects in other fragments is to show a complete picture of the connectivity to the reader. All relationships described in Table 5-2 are binary relationships.

Table 5-2. Relationships defined within the Connectivity Fragment

Relationship	Relationship Description
Has	This relationship represents “a container” relationship between a topological object (such as layer network domain and subnetwork) and a connectivity object (such as trail and subnetwork connection).
Maps To NFC	This relationship represents the mapping of a stream flow connection to a network flow connection.
Maps ToTFC	This relationship represents the mapping of a stream flow connection to a terminal flow connection.

Table 5-2. Relationships defined within the Connectivity Fragment

Relationship	Relationship Description
Root	Relationship between a connectivity object other than a TFC and LC (i.e., SFC, NFC, Trail, TC, or SNC) and the object that represents the source end point of the connectivity (SFEP, NFEP, NWTTP, NWCTP, or Edge). If the topology of the connectivity is point-to-point bidirectional, one of the two end points is arbitrarily chosen to participate in the association represented by this relationship.
SFCBranch	Relationship between a SFC and a SFEP object that represents a sink of the information flow. If the SFC topology is point-to-multipoint unidirectional, this relationship represents a point-to-point branch of the SFC. In such a topology, the attribute, Operational State, of the relationship represents the operational state of the branch.
NFCBranch	Relationship between a NFC and a NFEP that is a sink of the NFC. If the topology of the NFC is point-to-point bidirectional, one of the two end points is arbitrarily is chosen to participate in the association represented by this relationship. If the NFC topology is point-to-multipoint unidirectional, this relationship represents a point-to-point branch of the NFC. In such a topology, the attribute, Operational State, of the relationship represents the operational state of the branch.

Table 5-2. Relationships defined within the Connectivity Fragment

Relationship	Relationship Description
TrailBranch	Relationship between a Trail and a NWTTP. If the trail is point-to-point bidirectional, one of the two end points is arbitrarily is chosen to participate in the association represented by this relationship. Otherwise, i.e., the topology is point-to-point unidirectional or point-to-multipoint unidirectional, only the end point that is a sink of the information flow participates in the association represented by this relationship. Thus, if the trail topology is point-to-multipoint unidirectional, this relationship represents a point-to-point branch of the multipoint trail. In such a topology, the attribute, Operational State, of the relationship represents the operational state of the branch.
TCBranch	Relationship between a TC and a NWTP that generically represents either a NWTTP or a NWCTP. If the topology of the connectivity is point-to-point bidirectional, one of the two end points is arbitrarily chosen to participate in the association represented by this relationship. Otherwise, i.e., the topology is point-to-point unidirectional or point-to-multipoint unidirectional, only the end point that is a sink of the information flow participates in the association represented by this relationship. Thus, if the connectivity topology is point-to-multipoint unidirectional, this relationship represents a point-to-point branch of the multipoint connectivity. In such a topology, the attribute, Operational State, of the relationship represents the operational state of the branch.

Table 5-2. Relationships defined within the Connectivity Fragment

Relationship	Relationship Description
SNCBranch	Relationship between a SNC and an Edge. If the SNC is point-to-point bidirectional, one of the two end points is arbitrarily is chosen to participate in the association represented by this relationship. Otherwise, i.e., the topology is point-to-point unidirectional or point-to-multipoint unidirectional, only the end point that is a sink of the information flow participates in the association represented by this relationship. Thus, if the SNC topology is point-to-multipoint unidirectional, this relationship represents a point-to-point branch of the multipoint subnetwork connection. In such a topology, the attribute, Operational State, of the relationship represents the operational state of the branch.
SExtremity	Relationship between a TFC object and an SFEP object that represents an end point of the terminal flow connection represented by the TFC.
NExtremity	Relationship between a TFC object and an NFEP object that represents an end point of the terminal flow connection represented by the TFC.
ExtremityOfLC	Relationship between a LC object and a NWCTP object that represents an end point of the link connection represented by the LC.
Requestor	Relationship between a tandem connection and a foreign layer network domain. This represents the fact that the tandem connection was requested by the foreign layer network domain.

Apart from the specific relationships defined above, the following generic relationships are also defined in this fragment:

- Aggregation relationship between Logical Connection Graph (composite) and Stream Flow Connection (component)
- Aggregation relationship between Physical Connection Graph (composite) and Network Flow Connection (component)
- Aggregation relationship between Nodal Connection Graph (composite) and Terminal Flow Connection (component)

- Aggregation relationship between Network Flow Connection (composite) and Trail (component)
- Aggregation relationship between Trail (composite) and Tandem Connection (component)
- Aggregation relationship between Tandem Connection (composite) and Subnetwork Connection (component)
- Aggregation relationship between Tandem Connection (composite) and Link Connection (component)
- Aggregation relationship between Tandem Connection (composite) and Tandem Connection (component)
- Aggregation relationship between Subnetwork Connection (composite) and Subnetwork Connection (component)
- Aggregation relationship between Subnetwork Connection (composite) and Link Connection (component)

5.3 OMT Diagram for Connectivity Fragment

The OMT diagram for the Connectivity Fragment is shown in two parts in Figures 5-2 and 5-3.

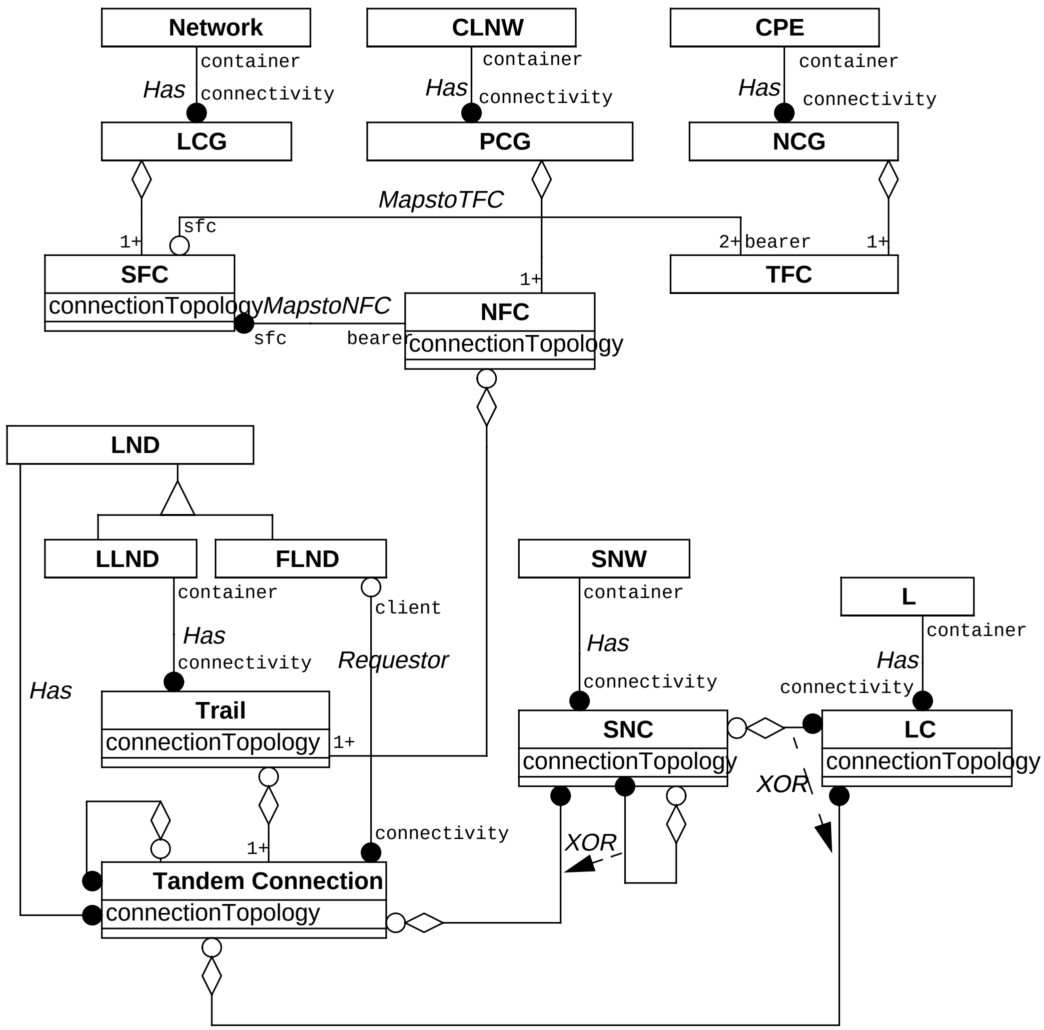


Figure 5-2. OMT Diagram for Connectivity Fragment (Part 1)

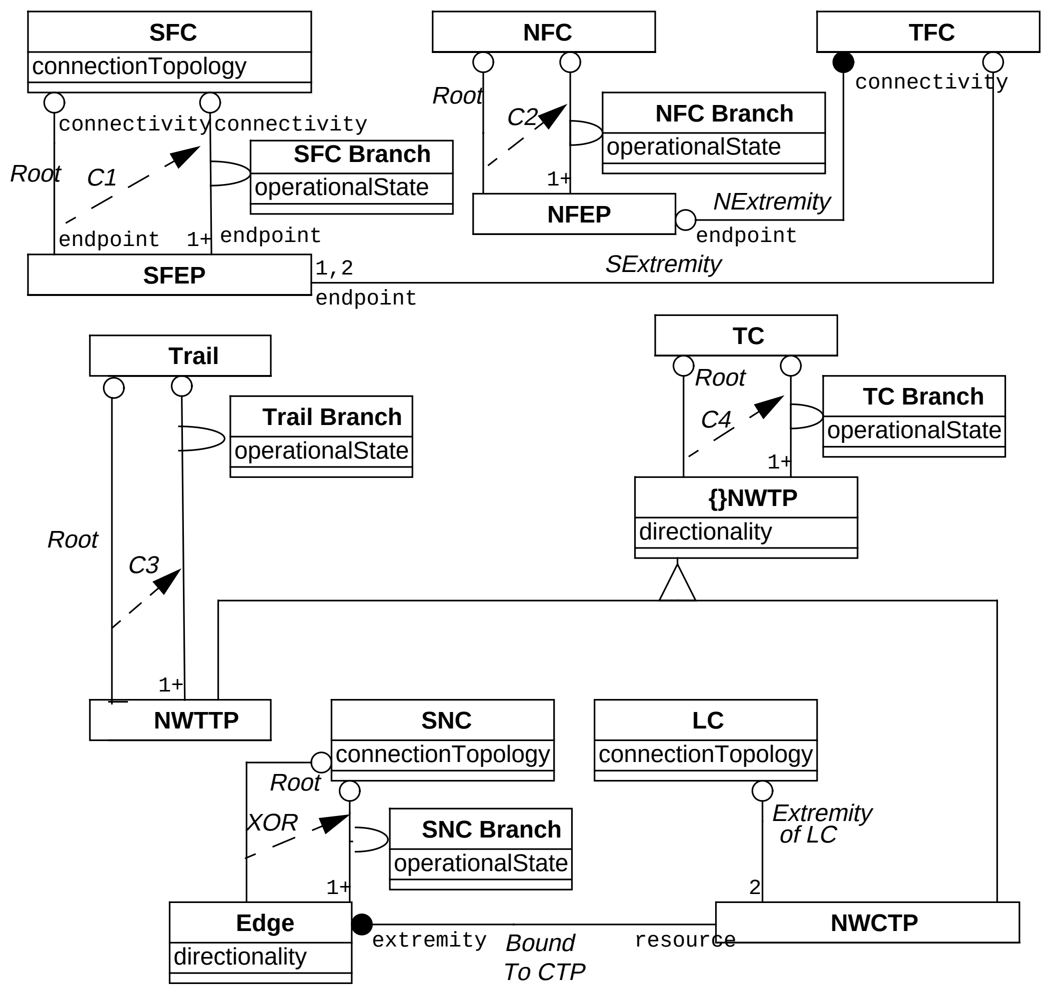


Figure 5-3. OMT Diagram for Connectivity Fragment (Part 2)

The constraints C1 through C5 labelled in the above OMT diagram are described below:

- C1: A SFEP object participates in exactly one of the two relationships
- C2: A NFEP object participates in exactly one of the two relationships
- C3: A NWTTP object participates in exactly one of the two relationships
- C4: A NWTP object participates in exactly one of the two relationships

5.4 Illustrative Examples

5.4.1 Has Relationship

Figure 5-4 illustrates the use of the *Has* relationship through an object instance diagram. (See Section 5.5.2.1 for the detailed definition of this relationship.) Figure 5-4 is related to Figure 5-1. Figure 5-4 illustrates the connectivity fragment of the information base in Layer Network Domain A.

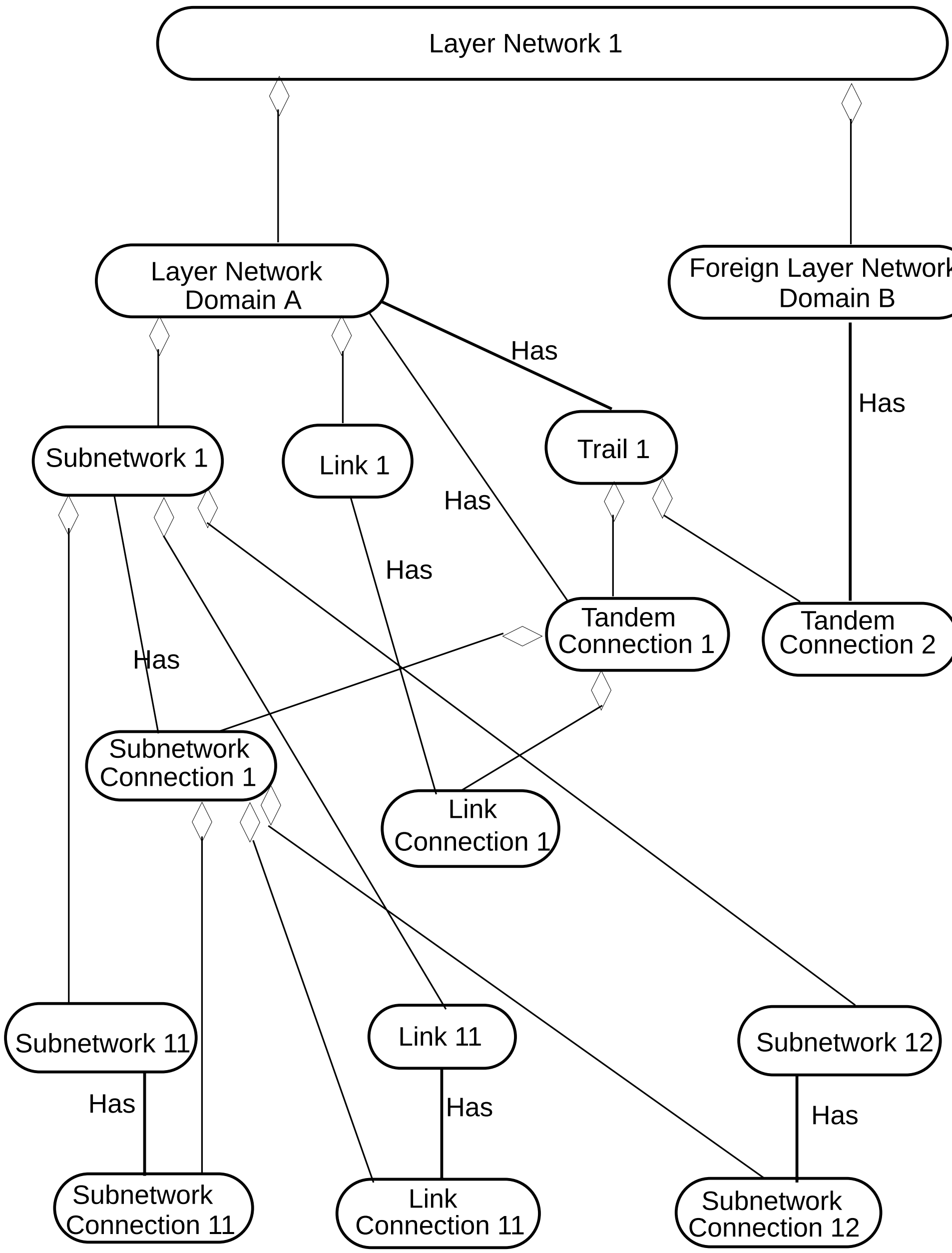


Figure 5-4. The Connectivity View in Provider Domain A

5.4.2 Requestor Relationship

Figure 5-5 is the object instance diagram that depicts the connectivity in Connectivity Provider Domain B for the trail scenario illustrated in Figure 5-1. This view complements the view in Domain A. Notice that the detailed view of Trail 1 is not visible in Domain B. Instead, it has a detailed view of Tandem Connection 2, although this is not shown in Figure 5-5 for simplicity.

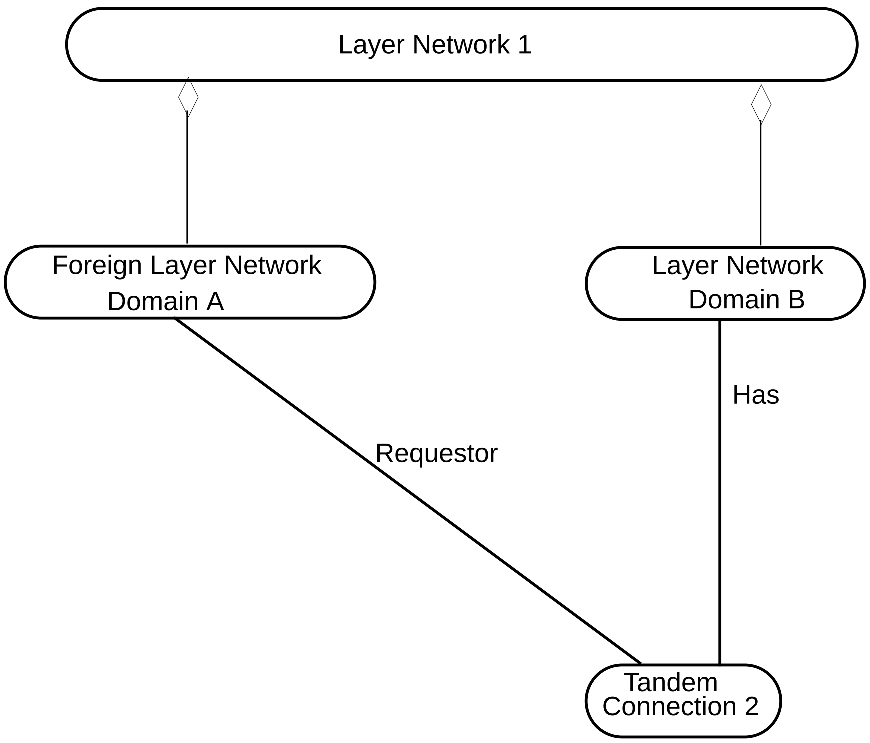


Figure 5-5. The Connectivity View in Provider Domain B

5.4.3 Root, Branch, and Bound To Relationships

Figure 5-6 is an object instance diagram that illustrates the use of Root, Branch, Bound To TTP, and Bound To CTP relationships. This diagram is related to Figure 5-1, and shows the relationships between the different connectivity objects that are components of Tandem Connection 1. (This is the view in Domain A).

Note that Edge 1 and Edge 3 are bound to the same NWCTP (so are Edge 2 and Edge 6) reflecting the partitioning of the subnetwork connection SNC1. The TTP Bound To CTP relationship between NWTTP and NWCTP shown in the figure is discussed in Section 6.

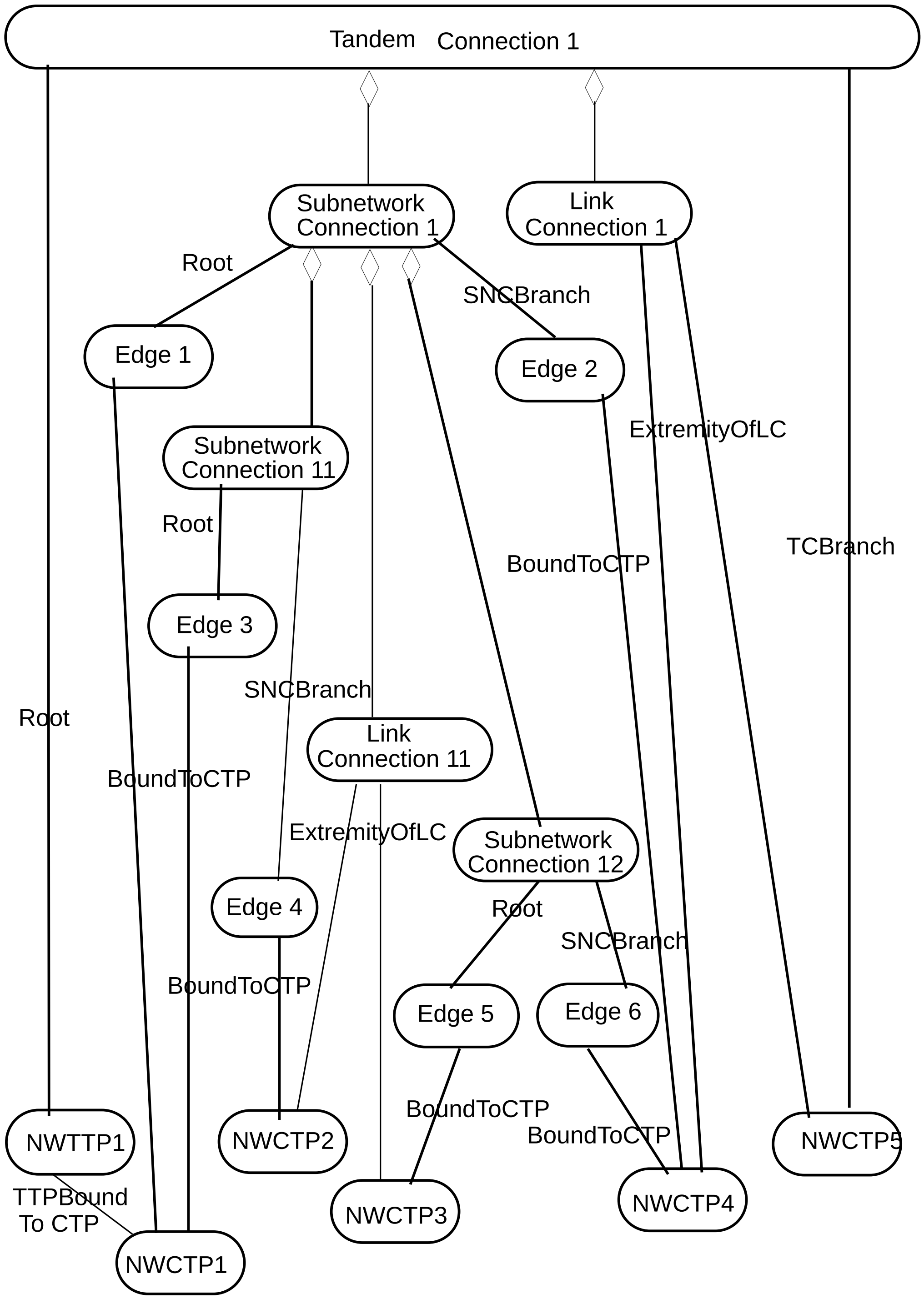


Figure 5-6. Use of Root, Branch, and Bound To Relationships

5.4.4 Multipoint Connections and the Branch Relationship

When a subnetwork connection (or trail or network flow connection) has a point-to-multipoint topology, it is very useful to have an explicit representation of the operational state of each branch of the point-to-multipoint connection. In such a topology, sometimes, it is possible that connectivity from the source end point to some sink end points is not operational, while connectivity to other sink end points is operational. To represent such information, an explicit representation of the operational state of each branch of a multipoint connection is needed.

In NRIM, this is accomplished using the class representation of the SNCBranch relationship.[2] A SNCBranch relationship instance relates a SNC object with an object that represents a sink edge. It represents the point-to-point branch connectivity between the source edge and the corresponding sink edge. The operational state of the branch is represented using the attribute Operational State of the SNCBranch relationship instance. (See the detailed Quasi-GDMO + GRM definition given later in this section.) Similar relationships are used for representing other multipoint connectivities, i.e., multipoint SFCs, multipoint NFCs, multipoint trails, and multipoint tandem connections.

Figure 5-7 illustrates a point-to-multipoint subnetwork connection where connectivity to the sink end points Edge 2 and Edge 4 is operational and connectivity to the sink end point Edge 3 is not operational. Figure 5-8 shows the corresponding object instance diagram.

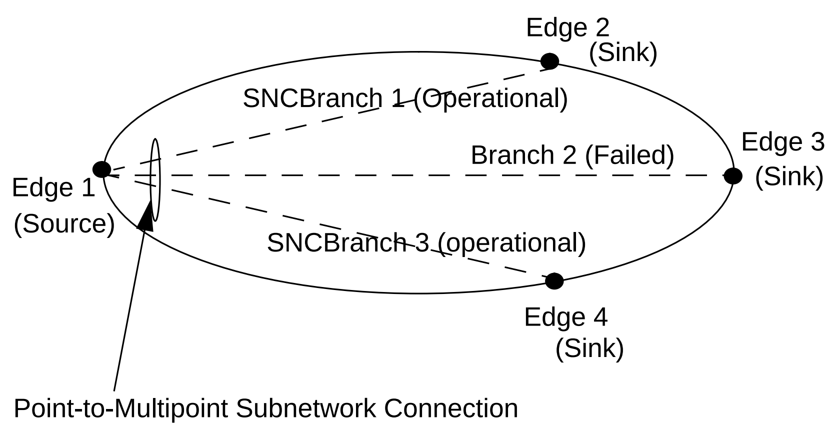


Figure 5-7. A Point-to-Multipoint Subnetwork Connection

2. These class instances were not shown in Figure 5-5 to simplify the diagram.

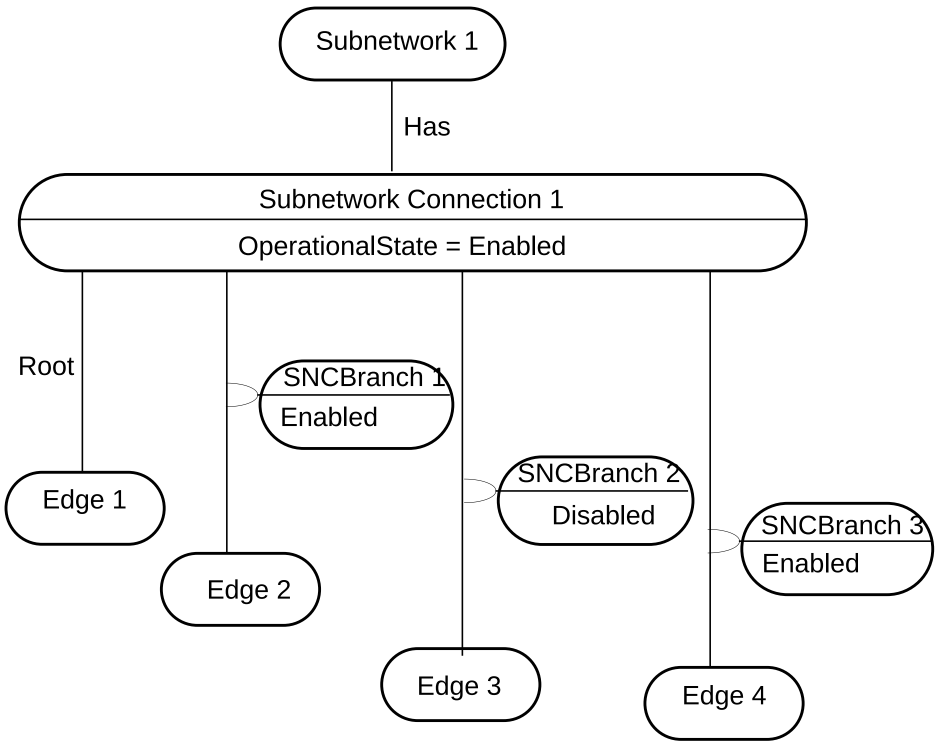


Figure 5-8. Use of the Branch Relationship in a Multipoint Connection

5.4.5 Representation of Flow Connections and Trails

To illustrate the representation of stream flow connections, network flow connections, and trails, consider the following connectivity scenario in the network whose topology is shown in Figure 4-2: (See Figure 5-9.)

- A SFEP Pool, SFEPPool1, with one SFEP, SFEP1 is in CPE4.
- A SFEP Pool, SFEPPool2, with one SFEP, SFEP2, is in CPE 1.
- A SFEP Pool, SFEPPool3, with one SFEP, SFEP3, is in CPE 6.
- A multipoint stream flow connection, SFC1, with SFEP1 as the source end point and SFEP2 and SFEP3 as the sink end points is established in the network. Note that this stream flow connection spans two administrative domains and three layer network domains.
- SFC1 is mapped to three terminal flow connections, TFC1, TFC2, and TFC3, one within each CPE, and a network flow connection, NFC1.

- NFC1 is composed of two trails; a point-to-multipoint ATM VP trail, Trail1, and a point-to-point Frame Relay trail, Trail2.

.

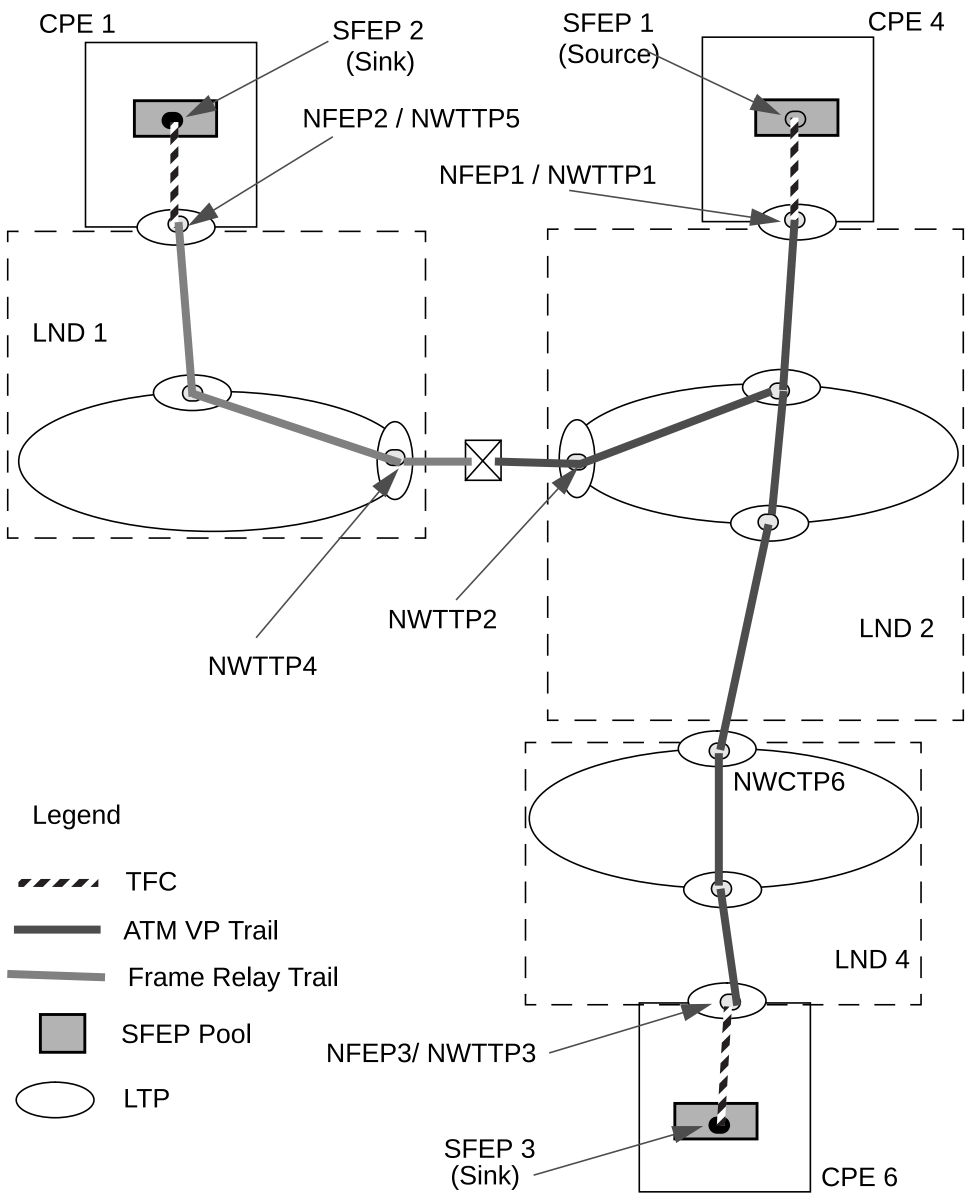


Figure 5-9. A Stream Flow Connection Scenario

Figures 5-10, 5-11, and 5-12 show the information base in Provider Domain 1 (to which LND1 and LND 2 belong) for the connectivity scenario shown in Figure 5-9. The decomposition of tandem connections is not shown in these figures.

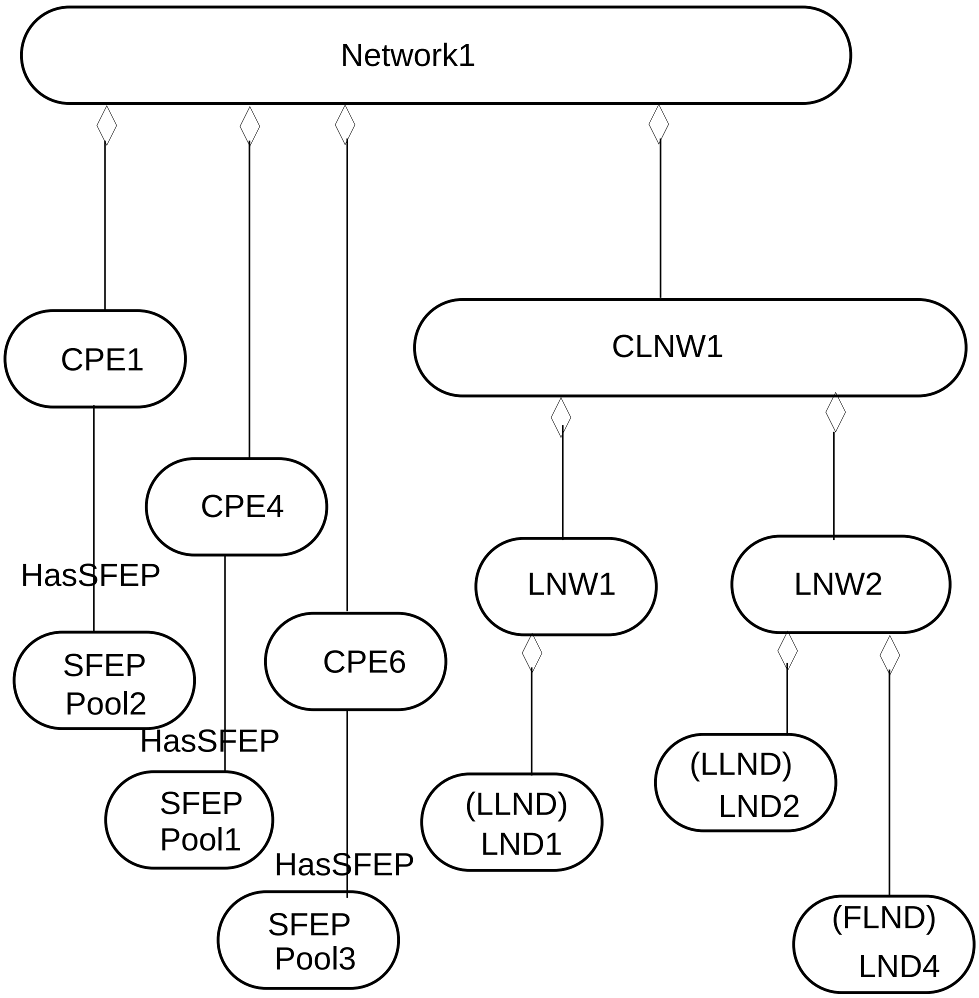


Figure 5-10. The Connectivity View in Provider Domain 1 (Part 1)

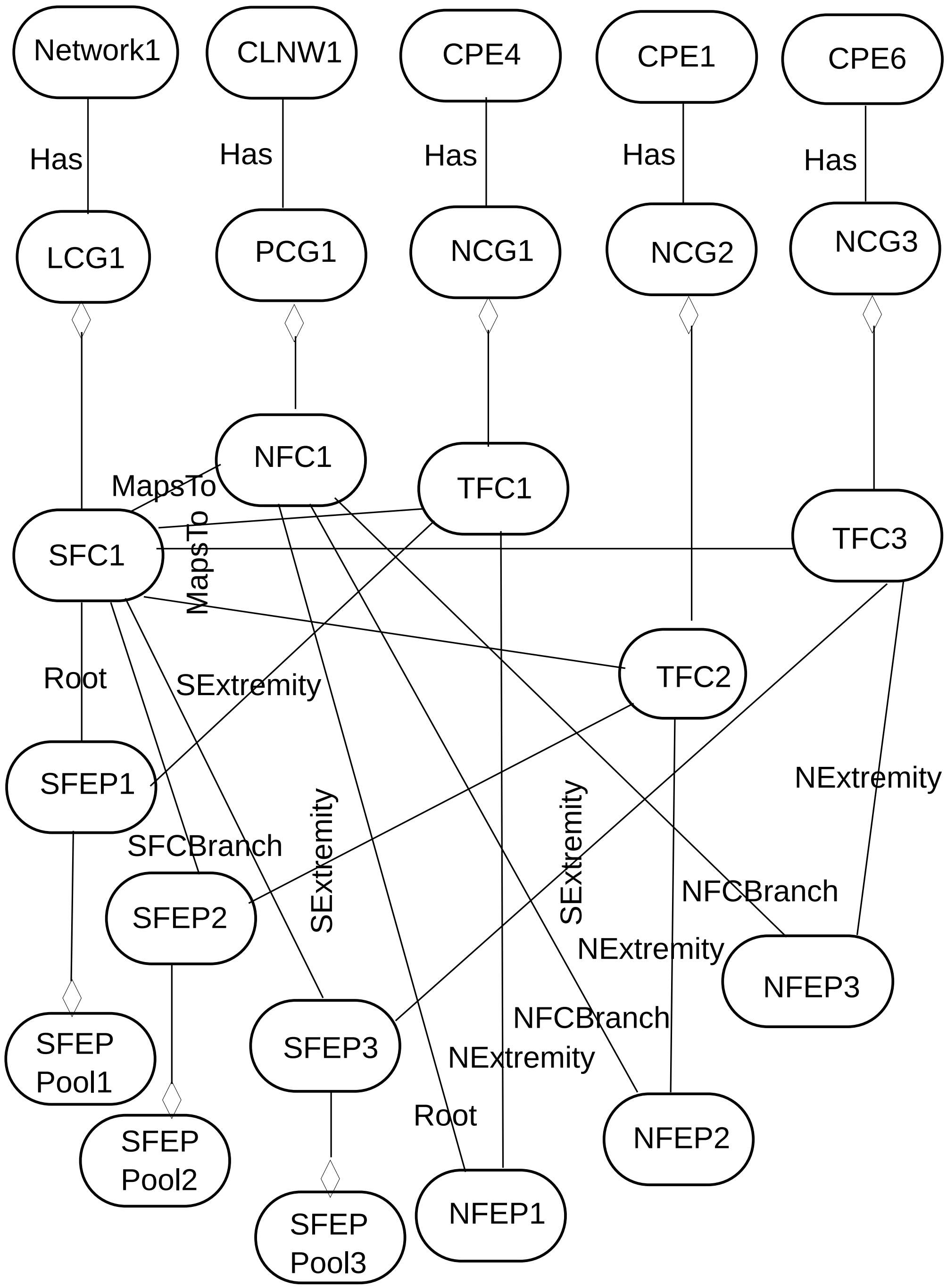


Figure 5-11. The Connectivity View in Provider Domain 1 (Part 2)

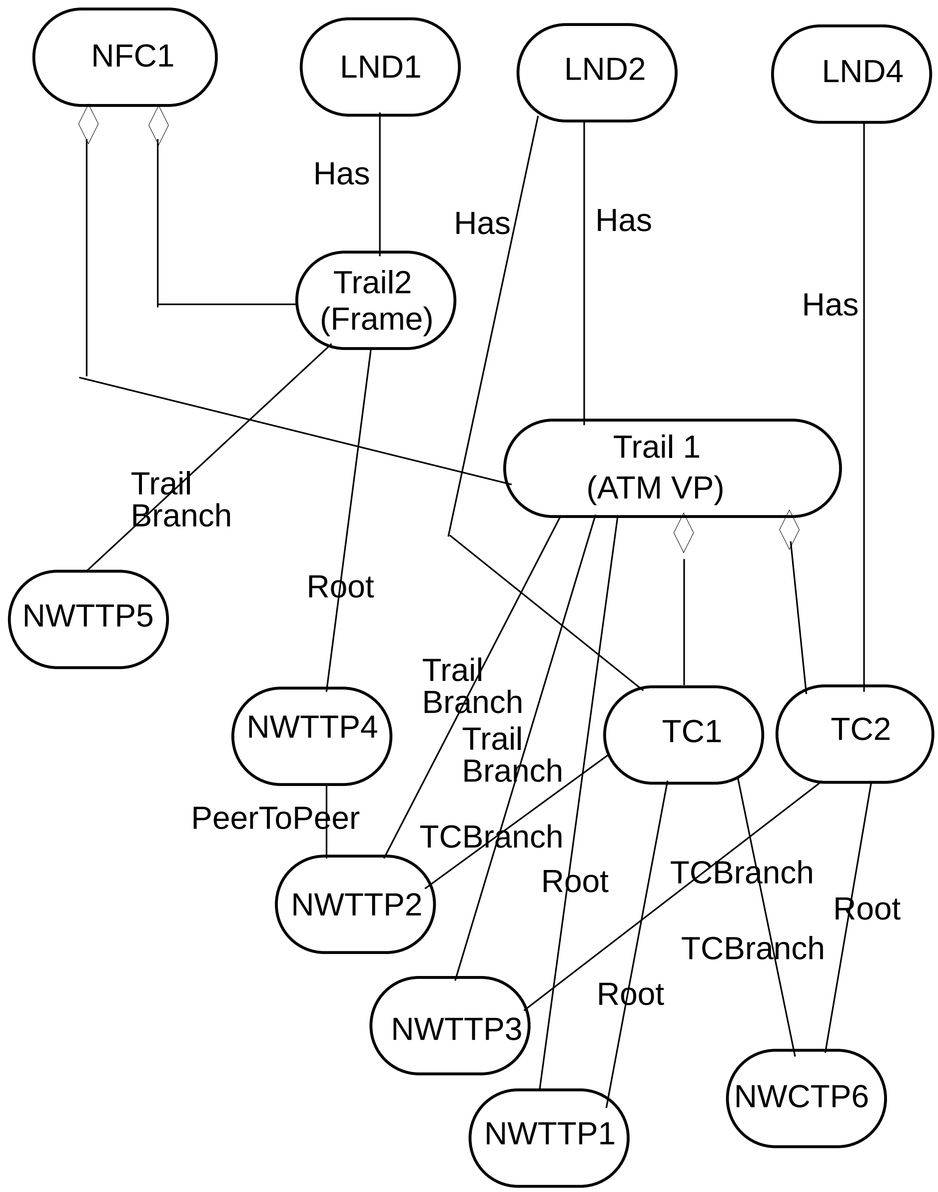


Figure 5-12. The Connectivity View in Provider Domain 1 (Part 3)

5.4.6 Representation of Tandem Connections Spanning Multiple LNDs

When a LND, say LND1 receives a request from another LND, say LND2, to set up a tandem connection, the former may determine that the tandem connection to be set up spans one or more foreign layer network domains (other than LND2). In such a case LND1 sets up the portion of the tandem connection in the local LND, and requests one or more foreign LNDs to setup the remainder of the tandem connection. The foreign LNDs with which a LND interacts for this purpose is determined by the connection management architecture and is thus outside the scope of NRIM. A cascade model and an hierarchical model are some of the possible architecture alternatives. Irrespective of the control architecture, the notions of originator domain, destination domain, and transit domain are defined for a tandem connection, and the view of a tandem connection in a LND is relative to the role of the LND with respect to the tandem connection, just as in the case of trails.

As an example, consider the configuration shown in Figure 5-13. In this configuration, a trail, called Trail1, has been set up that spans three LNDs, with LND A as the originator domain, LND B as a transit domain, and LND C as the destination domain. From the perspective of LND A, Trail1 is composed of a tandem connection TC1 in LND A and a tandem connection TC2 that spans LND B and LND C. (Depending on the business arrangement between LND A and LND B, LND C may or may not be visible to LND A). From the perspective of LND B, it is the originator domain of TC2 and LND C is the destination domain of TC2. From LND B’s perspective, the tandem connection TC2 is composed of a tandem connection TC21 in LND B and a tandem connection TC31 in LND C. Note that LND A is not aware of TC21 and TC31, LND B is not aware of Trail1 and TC1, and LND C is not aware of Trail1, TC1, TC2, and TC21.

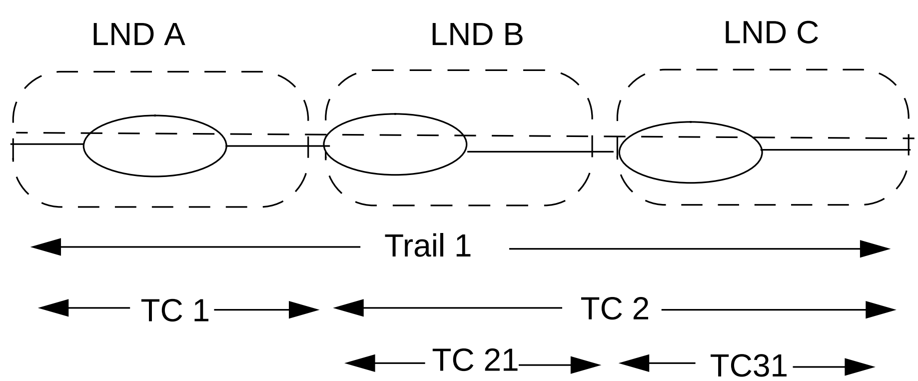


Figure 5-13. A Tandem Connection Spanning Multiple LNDs

5.5 Quasi-GDMO+GRM Definition of the Connectivity Fragment

5.5.1 Object Types

5.5.1.1 StreamFlowConnection

```
StreamFlowConnection OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY StreamFlowConnection-package PACKAGE
  BEHAVIOUR StreamFlowConnection-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: This object type represents the connectivity resource that
       transports information in a unidirectional manner across a TINA
       network from one source stream flow end point to one or more sink
       stream flow end points.The connection topology may be one of the
       following:
       Point-to-point unidirectional: in this topology, the stream flow
       connection has two SFEPs. Information is transported in only one
       direction, i.e., from the source SFEP to the sink SFEP. information
       regarding this traffic is represented in the object representing the
       source SFEP. Since there is no traffic from the sink to the source,
       the bandwidth and QoS attributes of the sink SFEP object shall have
       Null values.
       Point-to-multipoint unidirectional: in this topology, the stream
       flow connection has two or more SFEPs. Information is transported
       in only one direction, i.e., from SFEP designated as the source to
       the SFEPs designated as the sinks. QoS (including bandwidth)
       information represented in the source SFEP object represent the
       default QoS parameters for the traffic to each sink SFEP. QoS
       attributes of a sink SFEP object may have either a Null value or
       non-Null value. If the value is non-Null, the information
       represented in the sink SFEP object specify the information for the
       traffic from the source SFEP to the specific sink SFEP, overriding
       the default traffic information specified in the source SFEP.
     In addition to the attributes inherited from its supertypes, this
       object type has the following attribute:
     connectionTopology: the possible values are point-to-point
       unidirectional, and point-to-multipoint unidirectional.
     ";
  ATTRIBUTES
     connectionTopology
       PERMITTED VALUES: ConnectionTopology
       GET;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

5.5.1.2 Logical Connection Graph

```
LogicalConnectionGraph OBJECT TYPE
  DERIVED FROM Entity;
```

```
  CHARACTERIZED BY LogicalConnectionGraph-package PACKAGE
  BEHAVIOUR LogicalConnectionGraph-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: This object type represents a group of stream flow
        connections that have been grouped for some purpose.It represents a
        communication session from the resource perspective.
      ";
  ATTRIBUTES;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

5.5.1.3 NetworkFlowConnection

```
NetworkFlowConnection OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY NetworkFlowConnection-package PACKAGE
  BEHAVIOUR NetworkFlowConnection-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: This object type represents the connectivity resource that
        transports information across the connectivity layer network of a
        TINA network. The information is transported between two or more
        network flow end points (NFEPs) as described below. The connection
        topology may be one of the following:
        Point-to-point bidirectional: in this topology, the network flow
        connection has two NFEPs. One of them is arbitrarily designated as
        the source NFEP and the other as the sink NFEP. Information is
        transported from the source NFEP to the sink NFEP, and vice versa.
        Bandwidth and QoS may be different for each direction of traffic.
        Information regarding each direction of the traffic is represented
        in the NFEP objects.
        Point-to-point unidirectional: in this topology, the network flow
        connection has two NFEPs. Information is transported in only one
        direction, i.e., from the NFEP designated as the source NFEP to the
        other NFEP designated as the sink NFEP. Information regarding each
        direction of the traffic is represented in the NFEP objects.
        Point-to-multipoint unidirectional: in this topology, the network
        flow connection has two or more NFEPs. Information is transported
        from one of the NFEPs, designated as the source NFEP, to the other
        NFEPs, designated as the sink NFEPs. QoS information (including
        bandwidth) represented in the source NFEP object represent the
        default QoS parameters for the traffic to each sink NFEP. QoS
        attributes of a sink NFEP object may have either a Null value or non-
        Null value. If the value is non-Null, the information represented in
        the sink NFEP object specify the information for the traffic from
        the source NFEP to the specific sink NFEP, overriding the default
        traffic information specified in the source NFEP.
        For all topologies, the source and sink NFEP designation is uniformly
        represented using the Root and NFCBranch relationship respectively
      In addition to the attributes inherited from its supertypes, this
        object type has the following attribute:
```

```
    connectionTopology: the possible values are point-to-point
      bidirectional, point-to-point unidirectional, and point-to-
      multipoint unidirectional.
    ";
  ATTRIBUTES
    connectionTopology
      PERMITTED VALUES: ConnectionTopology
      GET;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

5.5.1.4 PhysicalConnectionGraph

```
PhysicalConnectionGraph OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY PhysicalConnectionGraph-package PACKAGE
  BEHAVIOUR PhysicalConnectionGraph-Behaviour BEHAVIOUR DEFINED AS
    "
    COMMENTS: This object type represents a group of network flow
      connections that form the network part of a group of stream flow
      connections included in a LogicalConnectionGraph.Represents a
      connectivity session from the resource perspective.
    ";
  ATTRIBUTES;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

5.5.1.5 TerminalFlowConnection

```
TerminalFlowConnection OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY TerminalFlowConnection-package PACKAGE
  BEHAVIOUR TerminalFlowConnection-Behaviour BEHAVIOUR DEFINED AS
    "
    COMMENTS: This object type represents the connectivity resource that
      transports information in a unidirectional manner either from a SFEP
      to a NFEP, from a NFEP to a SFEP, or from a SFEP to another SFEP. In
      all cases, the two end points shall be resident in the same CPE. A
      TFC and an SFEP end point of the TFC are related by the relationship
      SExtremity. A TFC and an NFEP end point of the TFC are related by
      the relationship NExtremity. The bandwidth and QoS information
      associated with each end point of a terminal flow connection may be
      either identical or different. If different, the TFC performs the
      necessary adaptation.
    ";
  ATTRIBUTES;
  ACTIONS;
```

```
  NOTIFICATIONS;
REGISTERED AS ??;
```

5.5.1.6 NodalConnectionGraph

```
NodalConnectionGraph OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY NodalConnectionGraph-package PACKAGE
  BEHAVIOUR NodalConnectionGraph-Behaviour BEHAVIOUR DEFINED AS
    "
    COMMENTS: This object type represents a group of terminal flow
      connections resident in the same CPE and that form a terminal part
      of a group of stream flow connections included in a
      LogicalConnectionGraph. Represents a terminal communication session
      from the resource perspective.
    ";
  ATTRIBUTES;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

5.5.1.7 Trail

```
Trail OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY Trail-package PACKAGE
  BEHAVIOUR Trail-Behaviour BEHAVIOUR DEFINED AS
    "
    COMMENTS: This object type represents the connectivity resource that
      transports information across a layer network. The information is
      transported between two or more network trail termination points
      (NWTTPs) as described below. The connection topology may be one of
      the following:
      Point-to-point bidirectional: in this topology, the trail has two
      NWTTPs. One of them is arbitrarily designated as the source NWTTP
      and the other as the sink NWTTP. Information is transported from the
      root NWTTP to the sink NWTTP, and vice versa. Bandwidth and QoS may
      be different for each direction of traffic and are represented in
      the NWCTP objects that are related to the NWTTP objects via the
      TTPBoundToCTP relationship.
      Point-to-point unidirectional: in this topology, the trail has two
      NWTTPs. Information is transported in only one direction, i.e., from
      the NWTTP designated as the source NWTTP to the other NWTTP
      designated as the sink NWTTP. Information regarding this traffic is
      represented in the NWCTP objects that are related to the NWTTP
      objects via the TTPBoundToCTP relationship.
      Point-to-multipoint unidirectional: in this topology, the trail has
      two or more NWTTPs. Information is transported from one of the
      NWTTPs, designated as the source NWTTP, to the other NWTTPs,
      designated as the sink NWTTPs. Bandwidth and QoS information
      represented in the NWCTP related to the root NWTTP object represent
      the default bandwidth and QoS parameters for the traffic to each sink
      NWTTP. Bandwidth and QoS attributes of the NWCTP object related to
```

```
      a sink NWTTP object may have either a Null value or non-Null value.
      If the value is non-Null, the information represented in the NWCTP
      object specify the information for the traffic from the source NWTTP
      to the specific sink NWTTP, overriding the default traffic
      information specified in the NWCTP related to the source NWTTP.

    From the perspective of a connectivity provider, a trail that is set
      up by it is composed of one or more tandem connections; one tandem
      connection within the local layer network domain, and zero or more
      tandem connections, each set up by a foreign layer network domain
      traversed by the trail. Each tandem connection is further decomposed
      into subnetwork connections and link connection. In the view of a
      connectivity provider, only the decomposition of the tandem
      connection in the local layer network domain is visible.

    The source and sink NWTTP designation is represented using the Root
      and TrailBranch relationships respectively.

    In addition to the attributes inherited from its supertypes, this
      object type has the following attribute:

    connectionTopology: the possible values are point-to-point
      bidirectional, point-to-point unidirectional, and point-to-
      multipoint unidirectional.

    ";

  ATTRIBUTES

    connectionTopology

      PERMITTED VALUES: ConnectionTopology

      GET;

  ACTIONS;

  NOTIFICATIONS;

REGISTERED AS ??;
```

5.5.1.8 LinkConnection

```
LinkConnection OBJECT TYPE

  DERIVED FROM Entity;

  CHARACTERIZED BY LinkConnection-package PACKAGE

  BEHAVIOUR LinkConnection-Behaviour BEHAVIOUR DEFINED AS

    "

    COMMENTS: This object type represents the connectivity resource that
      transports information across a link between either two adjacent
      subnetworks or a subnetwork and a CPE. The information is transported
      between two network connection termination points (NWCTPs) as
      described below. The connection topology may be one of the following:

      Point-to-point bidirectional: in this topology, the link connection
      has two NWCTPs. Information is transported between the two NWCTPs
      in both directions. Bandwidth and QoS may be different for each
      direction of traffic. Information regarding the traffic is
      represented in both NWCTPs.

      Point-to-point unidirectional: in this topology, the link connection
      has two NWCTPs. The NWCTP contained in the source LTP of the link
      is designated as the source NWCTP. The NWCTP contained in the sink
      LTP of the link is designated as the sink NWCTP. Information is
      transported in only one direction, i.e., from the source NWCTP to
      the sink NWCTP. Information regarding this traffic is represented
      in both NWCTPs.
```

```
    In addition to the attributes inherited from its supertypes, this
      object type has the following attribute:
    connectionTopology: the possible values are point-to-point
      bidirectional and point-to-point unidirectional.
    ";
  ATTRIBUTES
    connectionTopology
      PERMITTED VALUES: ConnectionTopology
      GET;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

5.5.1.9 SubnetworkConnection

```
SubnetworkConnection OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY SubnetworkConnection-package PACKAGE
  BEHAVIOUR SubnetworkConnection-Behaviour BEHAVIOUR DEFINED AS
    "
    COMMENTS: This object type represents the connectivity resource that
      transports information across a subnetwork. The information is
      transported between two or more edges as described below. The
      connection topology may be one of the following:
      Point-to-point bidirectional: in this topology, the subnetwork
      connection has two edges. One of them is arbitrarily designated as
      the source edge and the other as the sink edge. Information is
      transported from the source edge to the sink edge, and vice versa.
      Bandwidth and QoS may be different for each direction of traffic.
      Information regarding the traffic is represented in the NWCTP
      objects that are related with the Edge objects via the BoundToCTP
      relationship.
      Point-to-point unidirectional: in this topology, the subnetwork
      connection has two edges. Information is transported in only one
      direction, i.e., from the edge designated as the source edge to the
      other edge designated as the sink edge. Information regarding the
      traffic is represented in the NWCTP objects that are related with
      the Edge objects via the BoundToCTP relationship.
      Point-to-multipoint unidirectional: in this topology, the subnetwork
      connection has two or more edges. Information is transported from
      one of the edges, designated as the source edge, to the other edges,
      designated as the sink edges. Bandwidth and QoS information
      represented in the NWCTP related to the source Edge object represent
      the default bandwidth and QoS parameters for the traffic to each sink
      edge. Bandwidth and QoS attributes of the NWCTP object related to a
      sink Edge object may have either a Null value or non-Null value. If
      the value is non-Null, the information represented in the NWCTP
      object specify the information for the traffic from the source edge
      to the specific sink edge, overriding the default traffic
      information specified in the NWCTP related to the source edge.
      The source and sink edge designation is represented using the
      relationships Root and SNCBranch respectively.
```

```
    A subnetwork connection may further be decomposed into two or more
      subnetwork connections and one or more link connections. In the view
      of a connectivity provider, only the decomposition of subnetwork
      connections in the local layer network domains is visible.
    In addition to the attributes inherited from its supertypes, this
      object type has the following attribute:
    connectionTopology: the possible values are point-to-point
      bidirectional, point-to-point unidirectional, and point-to-
      multipoint unidirectional.
    ";
  ATTRIBUTES
    connectionTopology
      PERMITTED VALUES: ConnectionTopology
      GET;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

5.5.1.10 TandemConnection

```
TandemConnection OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY TandemConnection-package PACKAGE
  BEHAVIOUR TandemConnection-Behaviour BEHAVIOUR DEFINED AS
    "
    COMMENTS: This object type represents (in the view of a connectivity
      provider) a contiguous trail portion that has two components; a
      component that exists in a local layer network domain and an optional
      component that spans one or more foreign layer network domains. The
      component that is in the local LND is seen by the connectivity
      provider as being composed of a subnetwork connection across the top
      level subnetwork of the local LND and zero more link connections that
      are cross-connected by the subnetwork connection. The end points of
      a tandem connection are NWTTPs or NWCTPs.NWTTPs and NWCTPs are
      generically represented as Network Termination Points (NWTPs).The
      topology of a tandem connection may be one of the following:
      Point-to-point bidirectional: in this topology, the tandem
      connection has two NWTPs. One of them is arbitrarily designated as
      the source NWTP and the other as the sink NWTP. Information is
      transported from the source NWTP to the sink NWTP, and vice versa.
      Bandwidth and QoS may be different for each direction of traffic.
      Information regarding the traffic is represented in the NWCTP
      objects.
      Point-to-point unidirectional: in this topology, the tandem
      connection has two NWTPs. Information is transported in only one
      direction, i.e., from the NWTP designated as the source NWTP to the
      other NWTP designated as the sink NWTP. Information regarding the
      traffic is represented in the NWCTP objects.
      Point-to-multipoint unidirectional: in this topology, the tandem
      connection has two or more NWTPs. Information is transported from
      one of the NWTPs, designated as the source NWTP, to the other NWTPs,
      designated as the sink NWTPs. Bandwidth and QoS information
      represented in the root NWCTP object represent the default bandwidth
```

```
        and QoS parameters for the traffic to each sink NWCTP. Bandwidth and
        QoS attributes of a sink NWCTP object may have either a Null value
        or non-Null value. If the value is non-Null, the information
        represented in the sink NWCTP object specify the information for the
        traffic from the source NWCTP to the specific sink NWCTP, overriding
        the default traffic information specified in the source NWCTP.
      In addition to the attributes inherited from its supertypes, this
        object type has the following attribute:
      connectionTopology: the possible values are point-to-point
        bidirectional, point-to-point unidirectional, and point-to-
        multipoint unidirectional.
      ";
  ATTRIBUTES
      connectionTopology
        PERMITTED VALUES: ConnectionTopology
        GET;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

5.5.2 Relationship Types

5.5.2.1 Has

```
Has RELATIONSHIP TYPE
  CHARACTERIZED BY Has-package PACKAGE
  BEHAVIOUR Has-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: The Has relationship type represents a container
      relationship between a topological object and a connectivity object.
      The following rules govern the Has relationship:
           1. The relationship has two roles: the role container is played
              by a Network, ConnectivityLayerNetwork, CPE,
              LayerNetworkDomain, Subnetwork, or Link object; the role
              connectivity is played by a LogicalConnectionGraph,
              PhysicalConnectionGraph, NodalConnectionGraph, Trail,
              TandemConnection, SubnetworkConnection, or LinkConnection
              object.
           2. Only the following pairs of object types can be related:
              <Network, LCG>, <CLN, PCG>, <CPE, NCG>, <LND, Trail>,
              <LND,TC>, <SN, SNC>, and <L, LC>.
           3. An object can play the container role in zero or more Has
              relationships.
           4. An object can play the connectivity role in exactly one Has
              relationship.";
  ROLE container
      RELATED TYPES Network, ConnectivityLayerNetwork, CPE,
                LayerNetworkDomain, Subnetwork, Link;
      ROLE CARDINALITY CONSTRAINT (1..1);
```

```
  ROLE connectivity
     RELATED TYPES LogicalConnectionGraph, PhysicalConnectionGraph,
                 NodalConnectionGraph, Trail, TandemConnection,
                 SubnetworkConnection, LinkConnection;
     ROLE CARDINALITY CONSTRAINT (0..N);
REGISTERED AS ??;
```

5.5.2.2 MapsToNFC

```
MapsToNFC RELATIONSHIP TYPE
  CHARACTERIZED BY MapsToNFC-package PACKAGE
  BEHAVIOUR MapsToNFC-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The MapsToNFC relationship type represents the mapping of a
     stream flow connection to a network flow connection. The following
     rules govern the MapsToNFC relationship:
          1. The relationship has two roles: the role sfc is played by a
             SFC object; the role bearer is played by a NFC object.
          2. A NFC object is related with a SFC object if and only if the
             NFC represented by the former supports the SFC represented by
             the latter.
          3. A SFC object is related with exactly one NFC object.
          4. A NFC object is related with zero or more SFC objects.";
  ROLE sfc
     RELATED TYPES StreamFlowConnection;
     ROLE CARDINALITY CONSTRAINT (0..N);
  ROLE bearer
     RELATED TYPES NetworkFlowConnection;
     ROLE CARDINALITY CONSTRAINT (1..1);
REGISTERED AS ??;
```

5.5.2.3 MapsToTFC

```
MapsToTFC RELATIONSHIP TYPE
  CHARACTERIZED BY MapsToTFC-package PACKAGE
  BEHAVIOUR MapsToTFC-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The MapsToTFC relationship type represents the mapping of a
     stream flow connection to a terminal flow connection. The following
     rules govern the MapsToNFC relationship:
          1. The relationship has two roles: the role sfc is played by a
             SFC object; the role bearer is played by a TFC object.
          2. A TFC object is related with a SFC object if and only if the
             TFC represented by the former supports the SFC represented by
             the latter.
          3. A SFC object is related with two or more TFC objects.
          4. A TFC object is related with zero or one SFC object.";
```

```
  ROLE sfc
     RELATED TYPES StreamFlowConnection;
     ROLE CARDINALITY CONSTRAINT (0..1);
  ROLE bearer
     RELATED TYPES TerminalFlowConnection;
     ROLE CARDINALITY CONSTRAINT (2..N);
REGISTERED AS ??;
```

5.5.2.4 Root

```
Root RELATIONSHIP TYPE
  CHARACTERIZED BY Root-package PACKAGE
  BEHAVIOUR Root-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The Root relationship type represents the relationship
             between a connectivity object other than a TFC and LC (i.e.,
             SFC, NFC, Trail, TC, or SNC) and an object that represents the
             source end point of the connectivity (SFEP, NFEP, NWTTP,
             NWCTP, or Edge). If the topology of the connectivity is point-
             to-point bidirectional, one of the two end points is
             arbitrarily chosen to participate in this relationship.
             Otherwise, i.e., if the topology is point-to-point
             unidirectional or point-to-multipoint unidirectional, only
             the end point that is the source of the information flow
             participates in this relationship.The following rules govern
             the Root relationship:
          1. The relationship has two roles: the role connectivity is played
             by a StreamFlowConnection, NetworkFlowConnection, Trail,
             TandemConnection, or SubnetworkConnection object; the role
             endpoint is played by a SFEP, NFEP, NWTTP, NWCTP, or Edge
             object.
          2. Only the following pairs of object types can be related: <SFC,
             SFEP>, <NFC, NFEP>, <Trail, NWTTP>, <TC, NWTTP>, <TC, NWCTP>,
             and <SNC, Edge>.
          3. The value of the directionality attribute of the endpoint
             object shall be either source or bidirectional.
          4. A connectivity object is related with exactly one endpoint
             object.
          5. An endpoint object is related with at most one connectivity
             object.";
  ROLE connectivity
     RELATED TYPES StreamFlowConnection, NetworkFlowConnection, Trail,
                TandemConnection, SubnetworkConnection;
     ROLE CARDINALITY CONSTRAINT (0..1);
  ROLE endpoint
     RELATED TYPES StreamFlowEndPoint, NetworkFlowEndPoint,
                NetworkTrailTerminationPoint,
                NetworkConnectionTerminationPoint, Edge;
     ROLE CARDINALITY CONSTRAINT (1..1);
```

```
REGISTERED AS ??;
```

5.5.2.5 SFCBranch

```
SFCBranch RELATIONSHIP TYPE
  CHARACTERIZED BY SFCBranch-package PACKAGE
  BEHAVIOUR SFCBranch-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The SFCBranch relationship type represents the relationship
             between a SFC and an SFEP object that represents a sink end
             point of the SFC. In the case of a point-to-multipoint
             connection, this relationship represents a point-to-point
             branch of the multipoint connection. In such a case, the
             attribute, operationalState, represents the operational state
             of the branch. The following rules govern the SFCBranch
             relationship:
          1. The relationship has two roles: the role connectivity is
             played by a StreamFlowConnection object; the role endpoint is
             played by a SFEP object.
          2. The value of the directionality attribute of the endpoint
             object shall be sink.
          3. A SFC object is related with one or more SFEP objects.
          4. An SFEP object is related with at most one SFC object.";
  ATTRIBUTES
     operationalState
       PERMITTED VALUES: OperationalState
       GET;
  ROLE connectivity
     RELATED TYPES StreamFlowConnection;
     ROLE CARDINALITY CONSTRAINT (0..1);
  ROLE endpoint
     RELATED TYPES StreamFlowEndPoint;
     ROLE CARDINALITY CONSTRAINT (1..N);
REGISTERED AS ??;
```

5.5.2.6 NFCBranch

```
NFCBranch RELATIONSHIP TYPE
  CHARACTERIZED BY NFCBranch-package PACKAGE
  BEHAVIOUR NFCBranch-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The NFCBranch relationship type represents the relationship
             between a NFC and an NFEP object that represents a sink end
             point of the NFC. If the topology of the connectivity is
             point-to-point bidirectional, one of the two end points is
             arbitrarily chosen to participate in this relationship.
             Otherwise, i.e., if the topology is point-to-point
             unidirectional or point-to-multipoint unidirectional, only
             the end point that is a sink of the information flow
             participates in this relationship.Thus, in the case of a
```

```
                point-to-multipoint connection, this relationship represents
                a point-to-point branch of the multipoint connection. In such
                a case, the attribute, operationalState, represents the
                operational state of the branch. The following rules govern
                the NFCBranch relationship:
            1. The relationship has two roles: the role connectivity is played
                by a NetworkFlowConnection object; the role endpoint is played
                by a NFEP object.
            2. The value of the directionality attribute of the endpoint
                object shall be either sink or bidirectional.
            3. A NFC object is related with one or more NFEP objects.
            4. An NFEP object is related with at most one NFC object.";
  ATTRIBUTES
      operationalState
        PERMITTED VALUES: OperationalState
        GET;
  ROLE connectivity
      RELATED TYPES NetworkFlowConnection;
      ROLE CARDINALITY CONSTRAINT (0..1);
  ROLE endpoint
      RELATED TYPES NetworkFlowEndPoint;
      ROLE CARDINALITY CONSTRAINT (1..N);
REGISTERED AS ??;
```

5.5.2.7 TrailBranch

```
TrailBranch RELATIONSHIP TYPE
  CHARACTERIZED BY TrailBranch-package PACKAGE
  BEHAVIOUR TrailBranch-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: The TrailBranch relationship type represents the
                relationship between a Trail and a NWTTP object that
                represents a sink end point of the trail. If the topology of
                the trail is point-to-point bidirectional, one of the two end
                points is arbitrarily chosen to participate in this
                relationship. Otherwise, i.e., if the topology is point-to-
                point unidirectional or point-to-multipoint unidirectional,
                only the end point that is a sink of the information flow
                participates in this relationship.Thus, in the case of a
                point-to-multipoint connection, this relationship represents
                a point-to-point branch of the multipoint connection. In such
                a case, the attribute, operationalState, represents the
                operational state of the branch. The following rules govern
                the TrailBranch relationship:
            1. The relationship has two roles: the role connectivity is played
                by a Trail object; the role endpoint is played by a NWTTP
                object.
            2. The value of the directionality attribute of the endpoint
                object shall be either sink or bidirectional.
            3. A Trail object is related with one or more NWTTP objects.
            4. A NWTTP object is related with at most one Trail object.";
```

```
  ATTRIBUTES
      operationalState
        PERMITTED VALUES: OperationalState
        GET;
  ROLE connectivity
      RELATED TYPES Trail;
      ROLE CARDINALITY CONSTRAINT (0..1);
  ROLE endpoint
      RELATED TYPES NetworkTrailTerminationPoint;
      ROLE CARDINALITY CONSTRAINT (1..N);
REGISTERED AS ??;
```

5.5.2.8 TCBranch

```
TCBranch RELATIONSHIP TYPE
  CHARACTERIZED BY TCBranch-package PACKAGE
  BEHAVIOUR TCBranch-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: The TCBranch relationship type represents the relationship
                between a TandemConnection and a NWTP object that represents
                a sink end point (NWCTP or NWTTP) of the tandem connection.
                If the topology of the trail is point-to-point bidirectional,
                one of the two end points is arbitrarily chosen to participate
                in this relationship. Otherwise, i.e., if the topology is
                point-to-point unidirectional or point-to-multipoint
                unidirectional, only the end point that is a sink of the
                information flow participates in this relationship.Thus, in
                the case of a point-to-multipoint connection, this
                relationship represents a point-to-point branch of the
                multipoint connection. In such a case, the attribute,
                operationalState, represents the operational state of the
                branch. The following rules govern the TCBranch relationship:
           1. The relationship has two roles: the role connectivity is
              played by a TandemConnection object; the role endpoint is
              played by a NWTP object.
           2. The value of the directionality attribute of the endpoint
              object shall be either sink or bidirectional.
           3. A TandemConnection object is related with one or more NWTP
              objects.
           4. A NWTP object is related with at most one TandemConnection
              object.";
  ATTRIBUTES
      operationalState
        PERMITTED VALUES: OperationalState
        GET;
  ROLE connectivity
      RELATED TYPES TandemConnection;
      ROLE CARDINALITY CONSTRAINT (0..1);
  ROLE endpoint
```

```
    RELATED TYPES NetworkTerminationPoint;
    ROLE CARDINALITY CONSTRAINT (1..N);
REGISTERED AS ??;
```

5.5.2.9 SNCBranch

```
SNCBranch RELATIONSHIP TYPE
  CHARACTERIZED BY SNCBranch-package PACKAGE
  BEHAVIOUR SNCBranch-Behaviour BEHAVIOUR DEFINED AS
    "
    COMMENTS: The SNCBranch relationship type represents the relationship
            between a SubnetworkConnection and an Edge object that
            represents a sink end point of the subnetwork connection. If
            the topology of the trail is point-to-point bidirectional, one
            of the two end points is arbitrarily chosen to participate in
            this relationship. Otherwise, i.e., if the topology is point-
            to-point unidirectional or point-to-multipoint
            unidirectional, only the end point that is a sink of the
            information flow participates in this relationship.Thus, in
            the case of a point-to-multipoint connection, this
            relationship represents a point-to-point branch of the
            multipoint connection. In such a case, the attribute,
            operationalState, represents the operational state of the
            branch. The following rules govern the SNCBranch relationship:
         1. The relationship has two roles: the role connectivity is played
            by a SubnetworkConnection object; the role endpoint is played
            by an Edge object.
         2. The value of the directionality attribute of the endpoint
            object shall be either sink or bidirectional.
         3. A SubnetworkConnection object is related with one or more Edge
            objects.
         4. An Edge object is related with at most one SubnetworkConnection
            object.";
  ATTRIBUTES
    operationalState
      PERMITTED VALUES: OperationalState
      GET;
  ROLE connectivity
    RELATED TYPES SubnetworkConnection;
    ROLE CARDINALITY CONSTRAINT (0..1);
  ROLE endpoint
    RELATED TYPES Edge;
    ROLE CARDINALITY CONSTRAINT (1..N);
REGISTERED AS ??;
```

5.5.2.10 SExtremity

```
SExtremity RELATIONSHIP TYPE
  CHARACTERIZED BY SExtremity-package PACKAGE
  BEHAVIOUR SExtremity-Behaviour BEHAVIOUR DEFINED AS
```

```
    "
    COMMENTS: The SExtremity relationship type represents the
            relationship between a TerminalFlowConnection object that
            represents a TFC and an SFEP object that represents an end
            point (source or sink) of the TFC.The following rules govern
            the SExtremity relationship:
         1. The relationship has two roles: the role connectivity is
            played by a TerminalFlowConnection object; the role endpoint
            is played by a SFEP object.
         2. A TFC object is related with one or two SFEP objects.
         3. An SFEP object is related with at most one TFC object.";
  ROLE connectivity
    RELATED TYPES TerminalFlowConnection;
    ROLE CARDINALITY CONSTRAINT (0..1);
  ROLE endpoint
    RELATED TYPES StreamFlowEndPoint;
    ROLE CARDINALITY CONSTRAINT (1..2);
REGISTERED AS ??;
```

5.5.2.11 NExtremity

```
NExtremity RELATIONSHIP TYPE
  CHARACTERIZED BY NExtremity-package PACKAGE
  BEHAVIOUR NExtremity-Behaviour BEHAVIOUR DEFINED AS
    "
    COMMENTS: The NExtremity relationship type represents the
            relationship between a TerminalFlowConnection object that
            represents a TFC and an NFEP object that represents an end
            point (source or sink) of the TFC.The following rules govern
            the NExtremity relationship:
         1. The relationship has two roles: the role connectivity is
            played by a TerminalFlowConnection object; the role endpoint
            is played by an NFEP object.
         2. A TFC object is related with at most one NFEP object.
         3. An NFEP object is related with zero or more TFC objects.";
  ROLE connectivity
    RELATED TYPES TerminalFlowConnection;
    ROLE CARDINALITY CONSTRAINT (0..N);
  ROLE endpoint
    RELATED TYPES NetworkFlowEndPoint;
    ROLE CARDINALITY CONSTRAINT (0..1);
REGISTERED AS ??;
```

5.5.2.12 ExtremityOfLC

```
ExtremityOfLC RELATIONSHIP TYPE
  CHARACTERIZED BY ExtremityOfLC-package PACKAGE
```

```
  BEHAVIOUR ExtremityOfLC-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: The ExtremityOfLC relationship type represents the
              relationship between a LinkConnection object that represents
              a LC and an NWCTP object that represents an end point (source
              or sink) of the LC.The following rules govern the
              ExtremityOfLC relationship:
           1. The relationship has two roles: the role connectivity is played
              by a LinkConnection object; the role endpoint is played by an
              NWCTP object.
           2. A LC object is related with exactly two NWCTP objects.
           3. An NWCTP object is related with at most one LC object.";
  ROLE connectivity
      RELATED TYPES LinkConnection;
      ROLE CARDINALITY CONSTRAINT (0..1);
  ROLE endpoint
      RELATED TYPES NetworkConnectionTerminationPoint;
      ROLE CARDINALITY CONSTRAINT (2..2);
REGISTERED AS ??;
```

5.5.2.13 Requestor

```
Requestor RELATIONSHIP TYPE
  CHARACTERIZED BY Requestor-package PACKAGE
  BEHAVIOUR Requestor-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: The Requestor relationship type represents the relationship
              between a TandemConnection object that represents a TC and a
              Foreign LND object that represents the foreign layer network
              domain that requested the tandem connection setup.The
              following rules govern the Requestor relationship:
           1. The relationship has two roles: the role connectivity is played
              by a TandemConnection object; the role client is played by a
              ForeignLayerNetworkDomain object.
           2. This relationship is established when the tandem connection is
              established in a local LND upon a request from a foreign LND
              and exists as long as the tandem connection exists.
           3. A TC object is related with at most one Foreign LND object.
           4. A Foreign LND object is related with zero or more TC objects.";
  ROLE connectivity
      RELATED TYPES TandemConnection;
      ROLE CARDINALITY CONSTRAINT (0..N);
  ROLE client
      RELATED TYPES ForeignLayerNetworkDomain;
      ROLE CARDINALITY CONSTRAINT (0..1);
REGISTERED AS ??;
```

6. Termination Point Fragment

6.1 Introduction

This fragment describes the end points of the topological and connectivity resources defined in the previous two chapters. Such end points are called termination points.

Figure 6-1 shows the termination points related to topological components of a layer network domain.

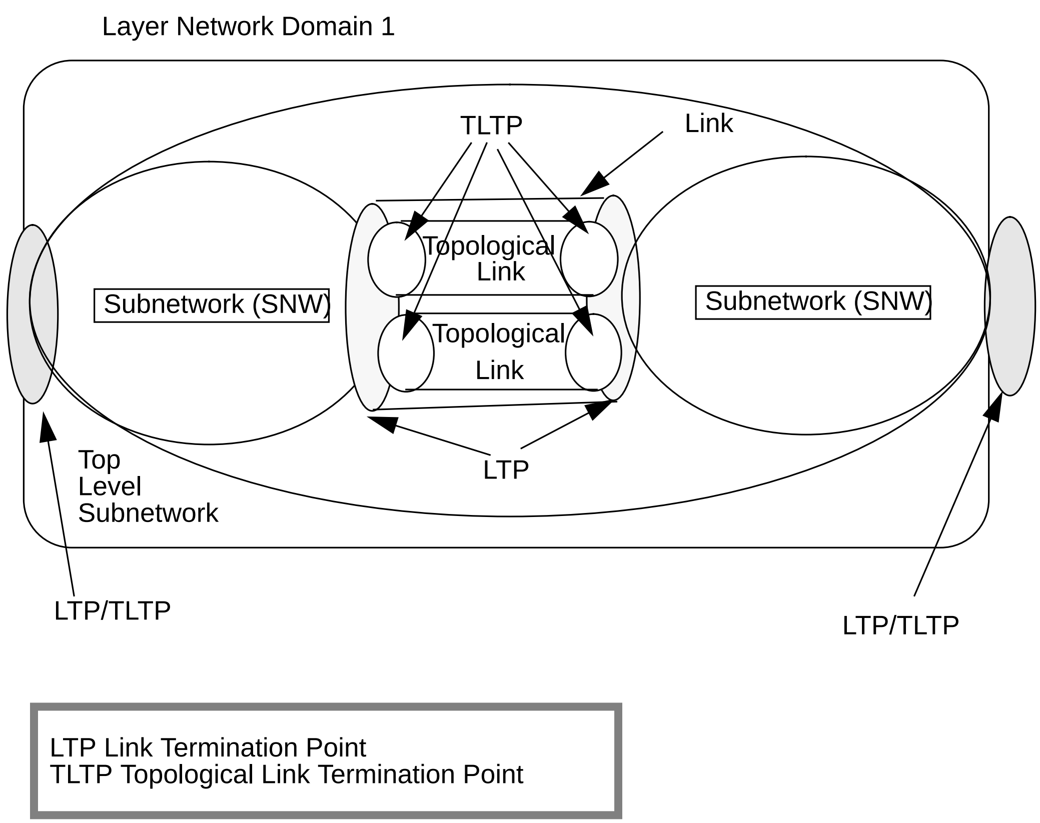
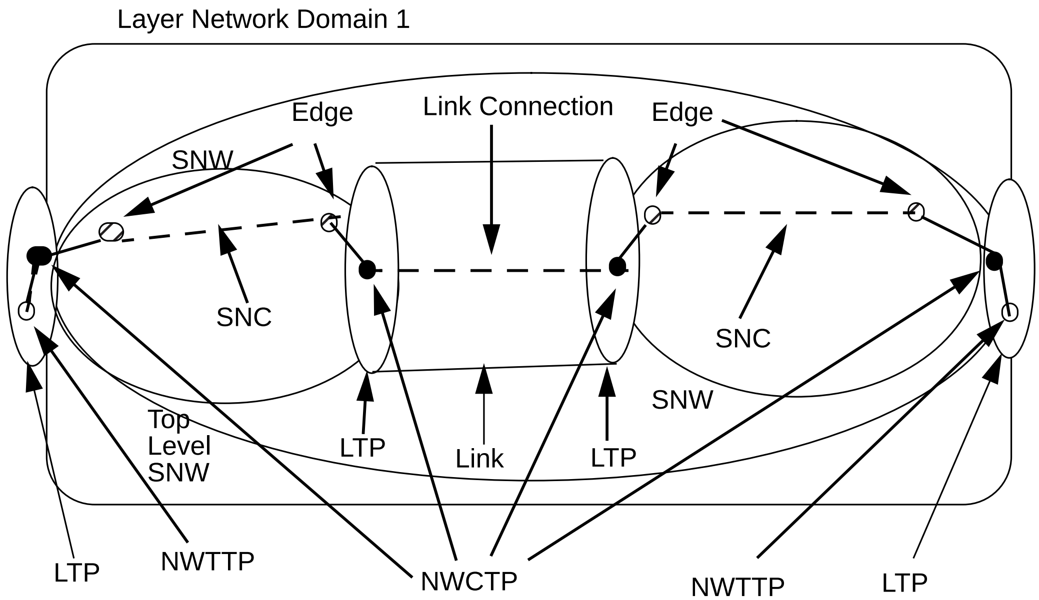


Figure 6-1. Topology Related Termination Points in a Layer Network Domain

Figure 6-1 shows a layer network domain configuration where the LND is composed of two subnetworks. The subnetworks are interconnected by two topological links. Each termination point of a topological link is called a *Topological Link Termination Point* (TLTP). A link has been configured using the two topological links. This link represents the aggregate capacity for connectivity between the two subnetworks. Each termination point of the link is called a *Link Termination Point* (LTP). At the boundary of the LND, there are two LTPs each configured using a TLTP. The topological links terminated by these TLTPs are not under the control of this LND, and hence are not included in the LND in the figure.

Figure 6-2 shows the termination points for connectivity related components of a layer network domain using a scenario where a trail has been set up in the layer network domain.



NWTTP Network Trail Termination Point
NWCTP Network Connection Termination Point
LTP Link Termination Point
SNW Subnetwork
SNC Subnetwork Connection

Figure 6-2. Connectivity Related Termination Points in a Layer Network Domain

As shown in Figure 6-2, the end points of a trail are called Network Trail Termination Points (NWTTPs). The end points of a link connection are called Network Connection Termination Points (NWCTPs). The end points of a subnetwork connection are called Edges (defined in Section 5). In the scenario shown in Figure 6-2, the trail is composed of two subnetwork connections and a link connection. As shown in the figure, each NWTTP is bound to a NWCTP and each edge is bound to a NWCTP.

Figure 6-3 shows the termination points for topology related components of a connectivity layer network that is entirely within one connectivity provider domain.

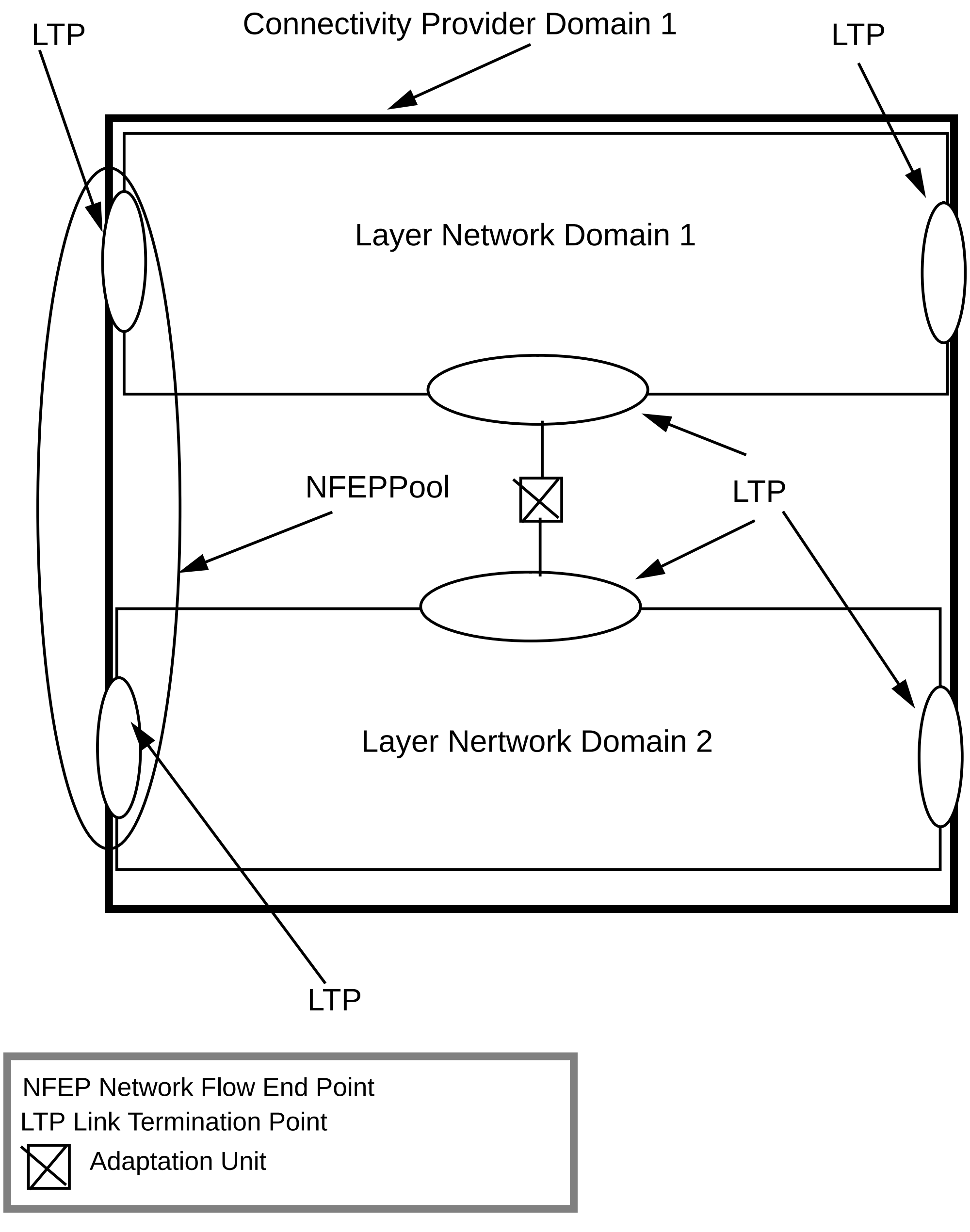


Figure 6-3. Topology Related Termination Points in a Connectivity Layer Network

The connectivity layer network shown in Figure 6-3 consists of two LNDs interconnected by an adaptation unit. A NFEP Pool has been configured using two LTPs, one belonging to each LND. This NFEP Pool terminates on a CPE (not shown in the figure). This aggregation of LTPs into a NFEP Pool has been done to facilitate flexible selection of NFEPs from the NFEP Pool during a network flow connection set up. The NFEP may be selected either by the entity that requests the network flow connection or by the entity that sets up the network flow connection that takes into account the layer network domains to which the NFEPs to be bound belong to and the topology of the connectivity layer network in terms of layer network domains and adaptation units.

Figure 6-4 illustrates the termination points related to a point to multipoint network flow connection (NFC) set up in the connectivity layer network shown in Figure 6-3. The end points of a network flow connection are called Network Flow End Points (NFEPs). The NFC is composed of two trails, one in LND 1 and another in LND 2. The trail in LND 1 is a point to multipoint trail while the trail in LND 2 is point to point trail. Each NFEP is bound to a NWTTP as shown in the figure.

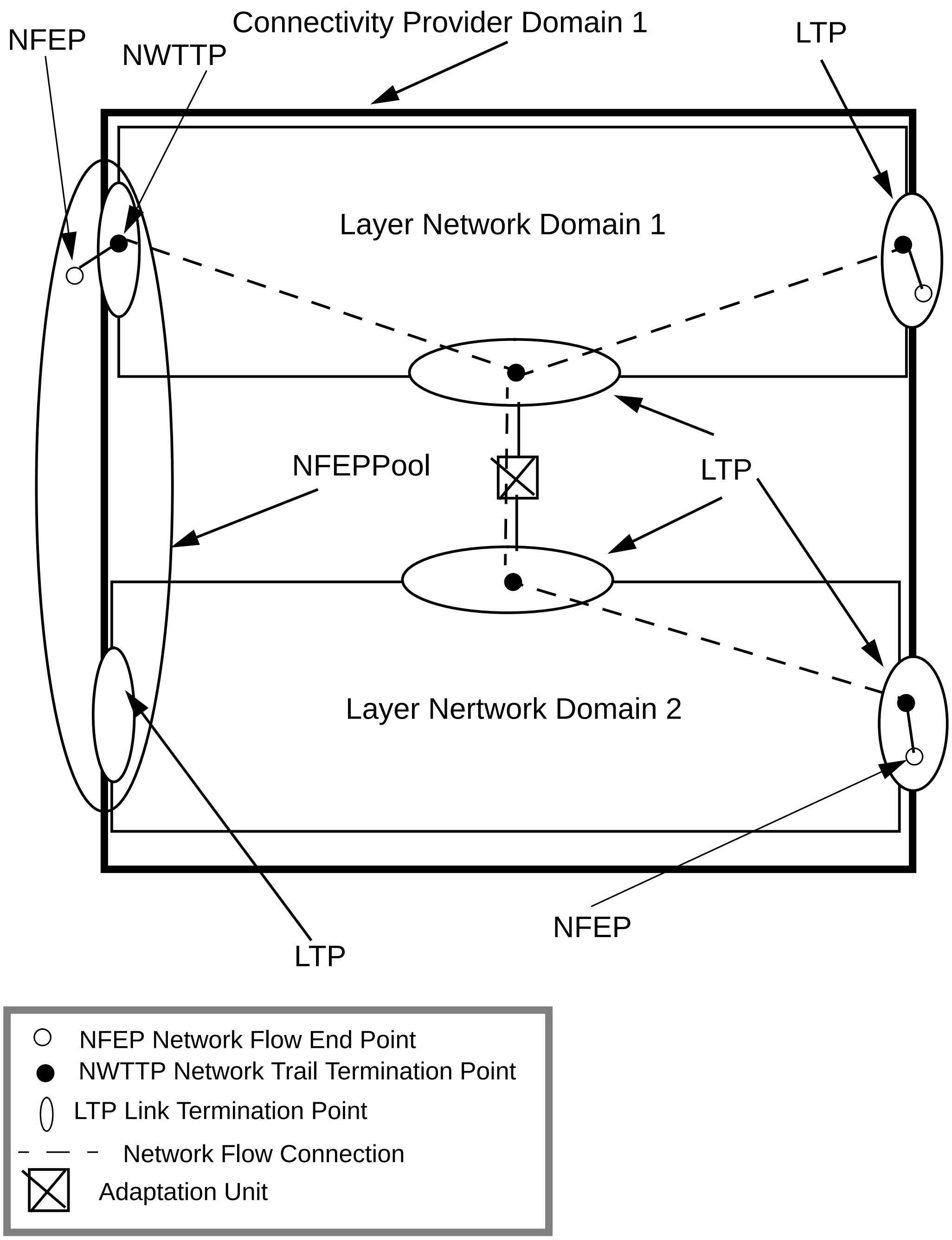


Figure 6-4. Connectivity Related Termination Points in a Connectivity Layer Network

6.2 Overview of the Termination Point Fragment

The termination point fragment defines the object types listed in Table 6-1.

Table 6-1. Object types in the Termination Point Fragment

Object types	Description
Stream Flow End Point (SFEP)	Represents an end point of a Stream Flow Connection. Models the point at which information is accepted or delivered to applications in a TINA Network.
Stream Flow End Point Pool (SFEP Pool)	Represents a collection of stream flow end points
Abstract Stream Flow End Point (ASFEP)	Represents generically a SFEP or a SFEP Pool. Defined only for inheritance. SFEP and SFEP Pool object types are subtypes of this type.
Network Flow End Point (NFEP)	Represents an end point of a Network Flow Connection. Models the point at which information is accepted or delivered in a Connectivity Layer Network.
Network Flow End Point Pool (NFEP Pool)	Represents a collection of network flow end points. A NFEP Pool is configured using one or more LTPs.
Abstract Network Flow End Point (ANFEP)[a]	Represents generically a NFEP or a NFEP Pool. Defined only for inheritance. NFEP and NFEP Pool object types are subtypes of this type
Network Termination Point (NWTP)	Represents generically a NWTTP or a NWCTP. Defined only for inheritance. NWTTP and NWCTP object types are subtypes of this type
Network Trail Termination Point (NWTTP)	Represents an end point of trail. Models the point at which information is accepted or delivered in a layer network.
Network Connection Termination Point (NWCTP)	Represents an endpoint of a link connection.
Topological Link Termination Point (TLTP)	Represents an end point of a topological link. Represents the point at which the client link to server layer trail adaptation occurs.

Table 6-1. Object types in the Termination Point Fragment

Object types	Description
Link Termination Point (LTP)	Represents an end point of a link. A LTP is configured using one or more TLTPs. When a LTP is configured, one or more NWCTPs may also be created.
Edge (E)	Represents an end point of a subnetwork connection. Each Edge is bound to a NWCTP.

a. Abstract Network Flow End Point (ANFEP) is called Resource Flow End Point (RFEP) in the Network Resource Architecture document [1].

The relationships defined within the termination point fragment are shown in Table 6-2.

Table 6-2. Relationships defined within the Termination Point Fragment

Relationships	Relationship Descriptions
Bound To CTP	Relationship between an Edge object and a NWCTP object. This relationship is established when the edge is bound to the connection termination point.
Bound To TTP	Relates a NFEP with a NWTTP. This relationship is established when the NFEP is bound to the NWTTP.
TTP Bound To CTP	Relates a NWTTP with a NWCTP. This relationship is established when the NWTTP is bound to the NWCTP.
Has SFEP	Relates a CPE with an ASFEP. This relationship is established when an SFEP or SFEP Pool is created in the CPE.
Has NFEP	Relates a CLNW with an ANFEP. This relationship is established when an NFEP or NFEP Pool is created in the boundary of a connectivity layer network.
Has TTP	Relates a LND (supertype of LLND and FLND) with a NWTTP. This relationship is established when a NWTTP is created in a layer network domain.

Table 6-2. Relationships defined within the Termination Point Fragment

Relationships	Relationship Descriptions
Has CTP	Relates a TLTP object with a NWCTP. This relationship is established when a NWCTP is created in a topological link termination point.
Extremity of Top Link	Relates a Topological Link object representing a topological link with a TLTP object representing an end point of the topological link.
Extremity of Link	Relates a Link object representing a link with a LTP object representing an end point of the link.
Served By TLTP	Relates a LTP object representing a termination of a link with a TLTP object representing a termination of a topological link. This relationship is established when the link is configured using the topological link.
LND Bounded by LTP	Relates a LND object representing a layer network domain with a LTP object representing a link termination point that is on the boundary of the layer network domain.
SNW Bounded by LTP	Relates a Subnetwork object representing a subnetwork with a LTP object representing a link termination point that is on the boundary of the subnetwork.
CPE Bounded by LTP	Relates a CPE object representing a CPE with a LTP object representing a link termination point that is on the CPE.
TLTP Terminates On	Relates a TLTP object representing an end point of a topological link with either a CPE or a SNW object representing the topological component in which the topological link termination occurs.
TTP Adapts TLTP	Relates a TLTP object representing a topological link termination with a NWTTP representing a termination of the server layer trail that supports the topological link.

Table 6-2. Relationships defined within the Termination Point Fragment

Relationships	Relationship Descriptions
PeerToPeer	Relates either two TLTPs or two NWTTPs of different characteristic information in adjacent layer network domains. A relationship between two TLTPs represents in an abstract manner an interworking unit that adapts the two different characteristic information. A relationship between two NWTTPs models the chaining of trails in the two layer networks that make up a network flow connection.
Supported by LTP	Relates a NFEP Pool object representing a NFEP pool with a LTP object representing a LTP that supports the NFEP pool.

Apart from the specific relationships defined above, the following generic relationships are also defined in this fragment:

- Aggregation relationship between SFEP Pool (composite) and ASFEP (component)
- Aggregation relationship between NFEP Pool (composite) and ANFEP (component)
- Aggregation relationship between LTP (composite) and NWCTP (component)
- Aggregation relationship between LTP (composite) and LTP (component)
- Inheritance relationship between ASFEP (supertype) and SFEP and SFEP Pool (subtypes)
- Inheritance relationship between ANFEP (supertype) and NFEP and NFEP Pool (subtypes)
- Inheritance relationship between NWTP (supertype) and NWTTP and NWCTP (subtypes)

6.3 OMT Diagram for Termination Point Fragment

The OMT Diagram for the Termination Point Fragment is presented in two parts in Figures 6-5 and 6-6.

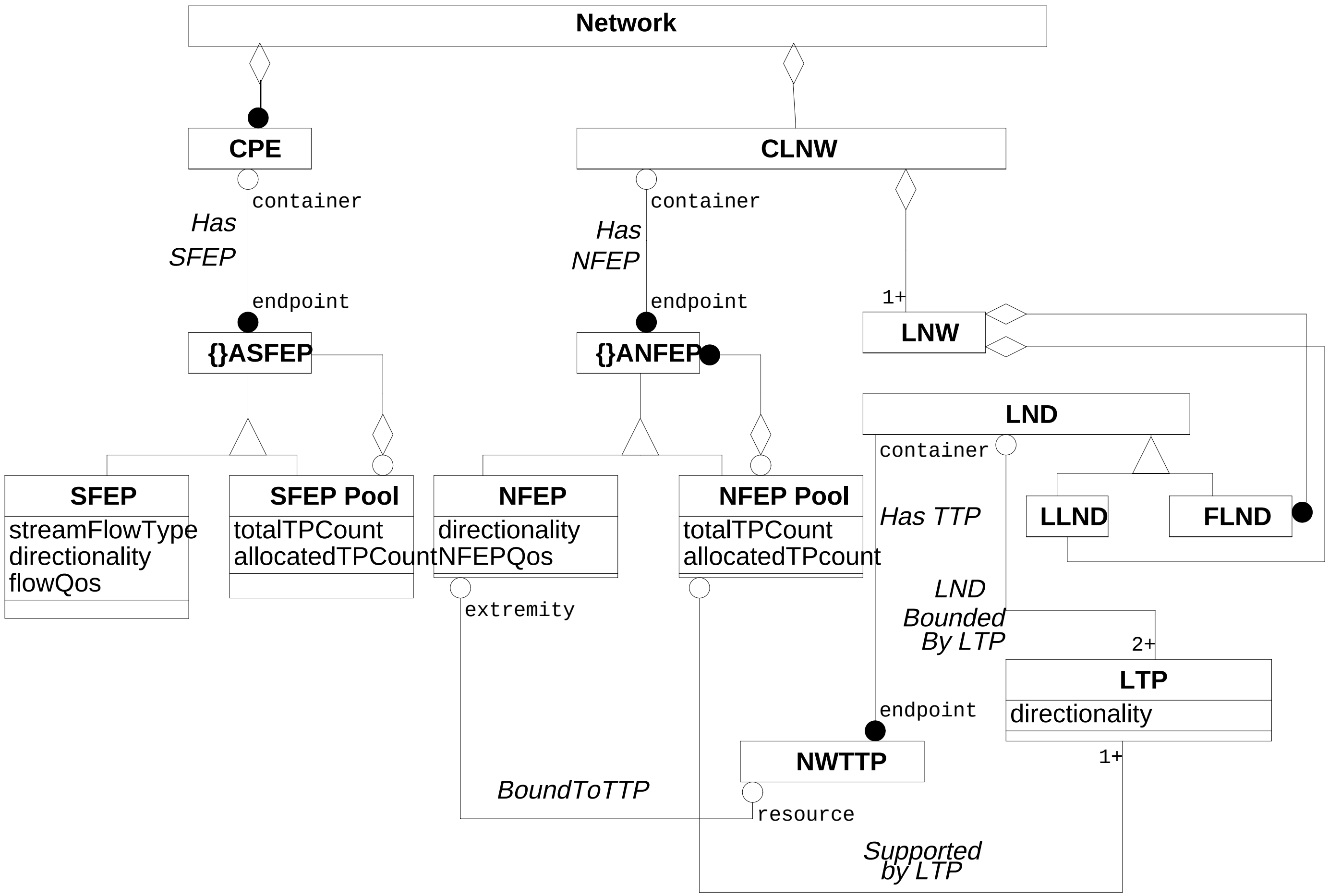


Figure 6-5. OMT Diagram for Termination Point Fragment (Part 1)

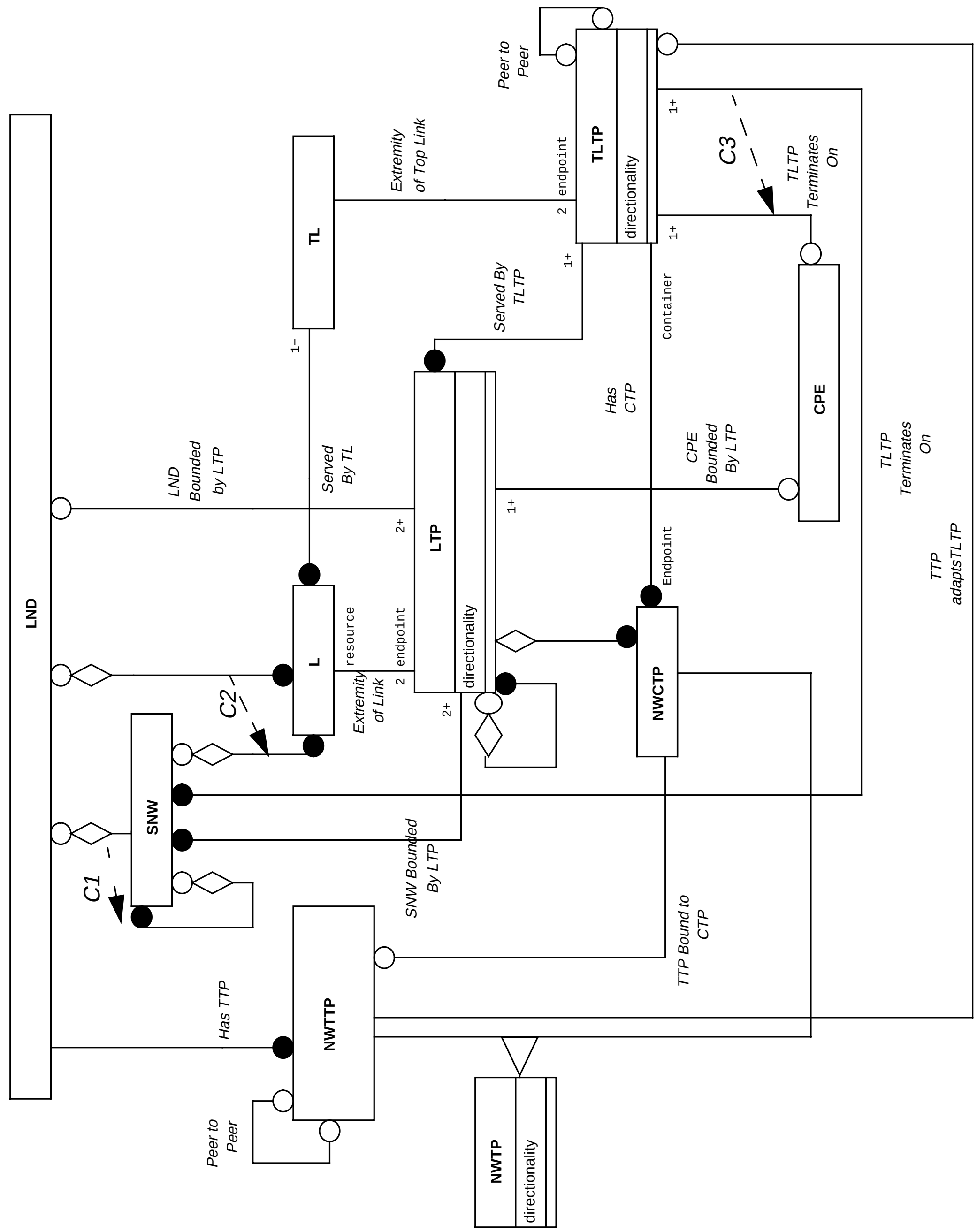


Figure 6-6. OMT Diagram for Termination Point Fragment (Part 2)

The constraints labelled C1, C2, and C3 in the above figure are described below:

- C1: A SNW object participates in exactly one of the two relationships
- C2: A L object participates in exactly one of the two relationships
- C3: A TLTP object participates in exactly one of the two relationships

6.4 Illustrative Examples

6.4.1 Termination Point Pool

Termination points are typically grouped into pools. Three kinds of termination point pools are defined in NRIM:

- Link Termination Point (LTP): This is a collection of NWCTPs.
- Network Flow End Point Pool: This is a collection of NFEPs. A NFEP Pool is configured using one or more LTPs.
- Stream Flow End Point Pool: This is a collection of SFEPs. A SFEP Pool is the representation of a stream interface in the NRIM.

Termination point pools are useful in the specification of connections where the choice of a particular termination point is unimportant and could be any termination point within a certain collection. Termination point pools can be configured for any of the following purposes:

- **addressing**: the termination points may be associated with some common entity such as a terminal or a link termination point.
- **resource management**: this includes resource aggregation, where the resource may be either bandwidth or channel number, and resource partitioning for handling different types of traffic.

Termination points within a termination point pool may themselves be termination point pools thus giving rise to a hierarchy.

Some example of termination point pools are listed below:

- A workstation is connected to an ATM network. The workstation has two ATM adapter cards fitted on it which allows it to access two ATM switches. The potential for ATM VC link connections between the terminal and each individual switch (element subnetwork) is represented as a link that maps 1:1 to the underlying server layer (SDH trail). The aggregate connectivity, i.e., the potential for ATM VC link connections between the terminal and the ATM network (including both switches), is represented as a link (composite link) that is configured using the two element links. The complete range of the VCIs available for use over both the cards is represented using an attribute of the LTP (termination point pool) of the composite link. The VCIs available over each card is represented using an attribute of the LTP (termination point pool) of the element link. This is an example of link aggregation within a layer network.

- Consider the TINA network configuration illustrated in Figure 4-2. In this configuration, CPE 2 is attached to an ATM switch and a Frame Relay switch. The connectivity to each switch is represented as a link. A NFEP Pool is configured using the two LTPs. When an application in CPE2 requests a stream flow connection with an application in another CPE (such as CPE1 or CPE4), it specifies only the NFEP Pool leaving the selection of NFEP (and thereby the choice of transport technology) to the management function responsible for setting up the stream flow connection. This is an example of link aggregation in a connectivity layer network spanning multiple layer networks.

6.4.2 TTP Bound To CTP relationship

The TTP Bound To CTP relationship is defined between a NWTTP and a NWCTP. This relationship associates a trail termination point with a network connection termination point within the same LTP. Figure 6-7 illustrates an example scenario involving this relationship between a NWTTP and a NWCTP.

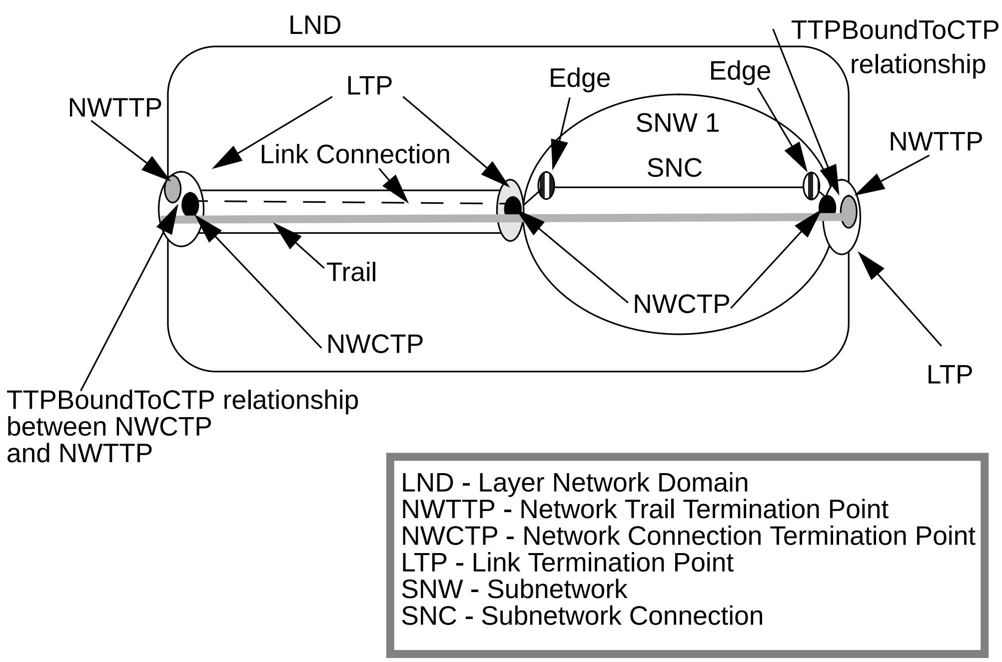


Figure 6-7. An Illustration of the TTP Bound To CTP Relationship

6.4.3 Extremity of Link and Bounded By LTP relationships

To illustrate the use of the relationships Extremity of Link, LND Bounded By LTP, and SNW Bounded by LTP, consider the network configuration shown in Figure 5-1. In this configuration, LND A is composed of a subnetwork SN1 and a link Link1. SN1 in turn is composed of two subnetworks SN11 and SN12 and a link Link11. Figure 6-8 illustrates the use of the Bounded By LTP relationship in the representation this network configuration.

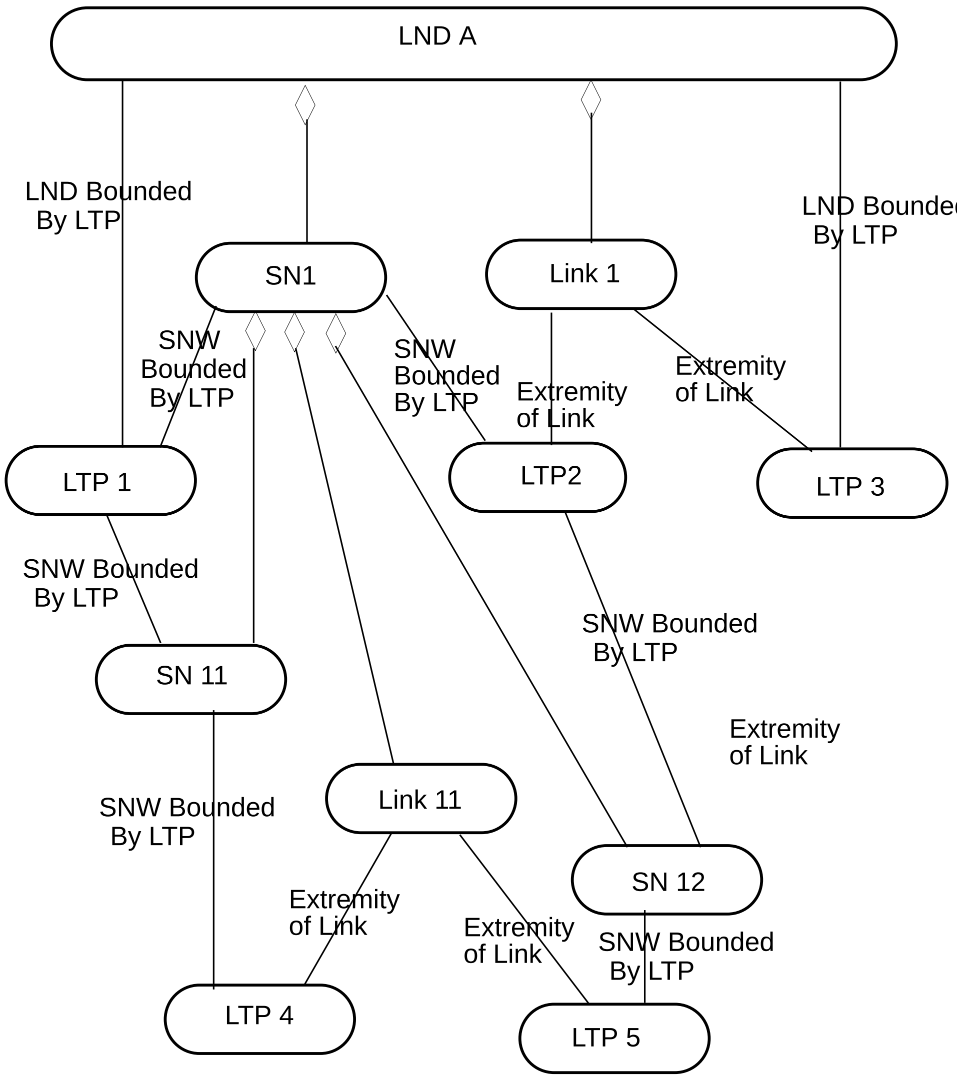


Figure 6-8. Use of Extremity of Link and Bounded By LTP Relationships

Notice that in the information base illustrated in Figure 6-8, LTP 1 participates in both LND Bounded By LTP and SNW Bounded By LTP relationships. Further, LTP1 participates in two instances of the SNW Bounded By LTP relationship. This is a consequence of the composition of the subnetwork SN1 that has subnetwork SN11 as a component. A similar observation can be made for LTP2.

6.4.4 TLTPTerminatesOn relationship

This relationship exists between either a subnetwork and a TLTP or a CPE and a TLTP depending on whether the topological termination occurs on a subnetwork or a CPE. If a subnetwork is composed of lower level subnetworks, a topological link that terminates on a composite subnetwork terminates also on the component subnetworks that share the same boundary. This is illustrated below.

Consider the network topology shown in Figure 6-9. In this topology, Topological Link 1 interconnects the subnetworks SN1 and SN2. One end of this topological link, TLTP1, is in SN1 and the other end is in SN2. SN1 is a composite subnetwork made up of subnetworks SN11 and SN12. The topological link termination represented by TLTP1 is also in the subnetwork SN12 as a consequence of the decomposition of SN1. Although TLTP1 is in both SN12 and SN1, it is important to note that Topological Link 1 interconnects only SN1 and SN2, and not SN12 and SN2. In general, the two subnetworks that are interconnected by a topological link are the highest level subnetworks on which the topological link terminations occur.

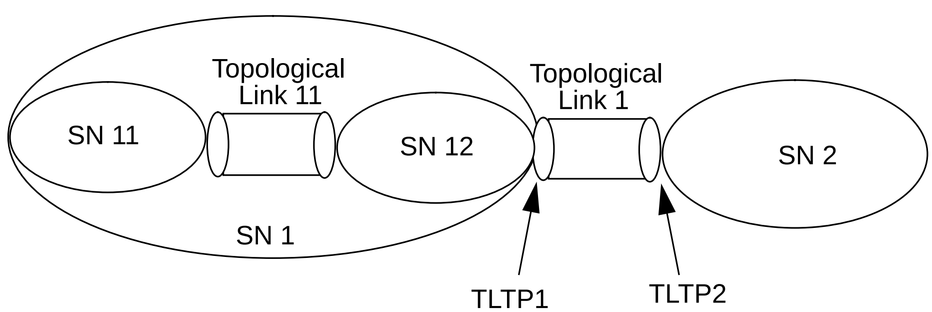


Figure 6-9. Topological Link Termination on Multiple Subnetworks

6.4.5 TTPadaptsTLTP relationship

As mentioned in Section 3 and in the network fragment there can be a client/server relationship between layer networks (or domains). This happens when a topological link is served by a trail at the server layer. TTPadaptsTLTP relationship is needed to relate the trail termination point at the server layer to the corresponding topological link termination point in the client layer.

6.4.6 PeerToPeer relationship

This relationship exists between either two trail termination points or two topological link termination points of adjacent layer network domains of different characteristic information. A PeerToPeer relationship between two TLTPs represents in an abstract manner a real resource, like adapter or interworking unit that adapts the two different characteristic information. This relationship is used in the topology representation of a connectivity layer network. A PeerToPeer relationship between two NWTTPs models the chaining of trails in different layer networks that may occur in setting up a network flow connection. This relationship is used in the connectivity representation of a network flow connection.

Figure 6-10 shows an example of this relationship between TLTPs. Figure 6-11 shows an example of this relationship between NWTTPs.

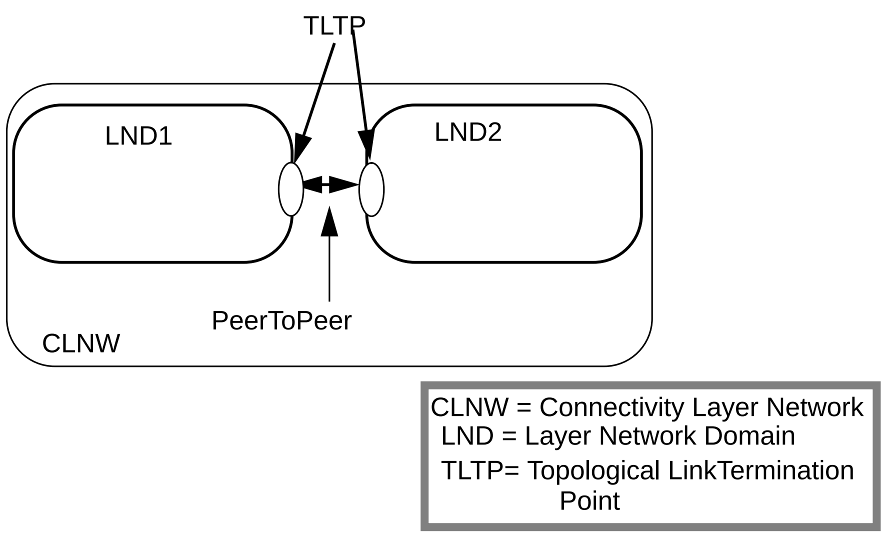


Figure 6-10. PeerToPeer Relationship Between TLTPs

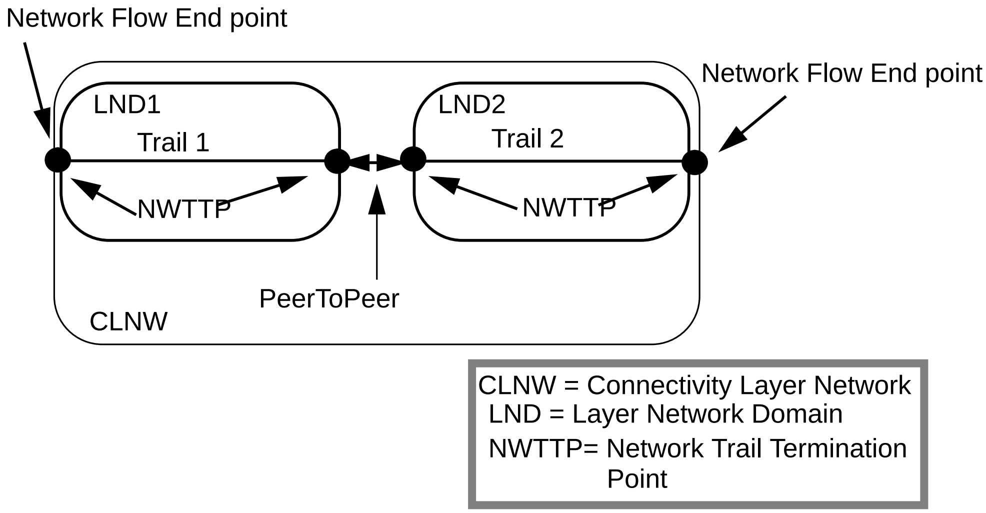


Figure 6-11. PeerToPeer Relationship Between NWTTPs

6.5 Quasi-GDMO Definition of the Termination Point Fragment

6.5.1 Object Types

6.5.1.1 AbstractStreamFlowEndPoint

```
AbstractStreamFlowEndPoint OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY AbstractStreamFlowEndPoint-package PACKAGE
  BEHAVIOUR AbstractStreamFlowEndPoint-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: This object type represents generically an SFEP or SFEP
        Pool. This type is defined only for inheritance and is not
        instantiable. StreamFlowEndPoint and StreamFlowEndPointPool object
        types are subtypes of this type.
      ";
  ATTRIBUTES;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

6.5.1.2 StreamFlowEndPoint

```
StreamFlowEndPoint OBJECT TYPE
  DERIVED FROM AbstractStreamFlowEndPoint;
  CHARACTERIZED BY StreamFlowEndPoint-package PACKAGE
  BEHAVIOUR StreamFlowEndPoint-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: This object type represents an end point of a stream flow
        connection; i.e., a point at which information is accepted or
        delivered to applications in a TINA network. An SFEP is associated
        with an application or a multi-media device in a CPE.
      In addition to the attributes derived from its supertype, an SFEP
        object has the following attributes:
        streamFlowType: this attribute specifies the type of stream flow
            associated with the SFEP.
        directionality: this attribute specifies the directionality of the
            SFEP; the value can be either source or sink.
        flowQoS: this attribute specifies the QoS (including bandwidth)
            associated with the SFEP. The QoS is specified using a sequence
            of tag-value pairs, where each tag denotes a QoS attribute. The
            QoS attributes applicable to a SFEP depend on the stream flow
            type.
      ";

  ATTRIBUTES
      streamFlowType
        PERMITTED VALUES: FlowType
```

```
      GET-REPLACE;
    directionality
      PERMITTED VALUES: TPDirectionality
      GET-REPLACE;
    flowQoS
      PERMITTED VALUES: QoSAttributeList
      GET-REPLACE;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

6.5.1.3 StreamFlowEndPointPool

```
StreamFlowEndPointPool OBJECT TYPE
  DERIVED FROM AbstractStreamFlowEndPoint;
  CHARACTERIZED BY StreamFlowEndPointPool-package PACKAGE
  BEHAVIOUR StreamFlowEndPointPool-Behaviour BEHAVIOUR DEFINED AS
    "
    COMMENTS: This object type represents a collection of SFEPs or SFEP
      Pools. A SFEP Pool object represents the resource aspects of a stream
      interface associated with a TINA application.
    In addition to the attributes derived from its supertype, an SFEP Pool
      object has the following attributes:
      totalTPCount: this attribute specifies the capacity of the SFEP Pool
          in terms of the total number of SFEPs or SFEP Pools that can be
          contained in the SFEP Pool.
      allocatedTPCount: this attribute specifies the number of SFEPs or
          SFEP Pools that are currently contained in the SFEP Pool.
    ";

  ATTRIBUTES
    totalTPCount
      PERMITTED VALUES: INTEGER
      GET;
    allocatedTPCount
      PERMITTED VALUES: INTEGER
      GET-REPLACE;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

6.5.1.4 AbstractNetworkFlowEndPoint

```
AbstractNetworkFlowEndPoint OBJECT TYPE
  DERIVED FROM Entity;
```

```
  CHARACTERIZED BY AbstractNetworkFlowEndPoint-package PACKAGE
  BEHAVIOUR AbstractNetworkFlowEndPoint-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: This object type represents generically an NFEP or NFEP
       Pool. This type is defined only for inheritance and is not
       instantiable. NetworkFlowEndPoint and NetworkFlowEndPointPool
       object types are subtypes of this type.
     ";
  ATTRIBUTES;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

6.5.1.5 NetworkFlowEndPoint

```
NetworkFlowEndPoint OBJECT TYPE
  DERIVED FROM AbstractNetworkFlowEndPoint;
  CHARACTERIZED BY NetworkFlowEndPoint-package PACKAGE
  BEHAVIOUR NetworkFlowEndPoint-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: This object type represents an end point of a network flow
       connection; i.e., a point at which information is accepted or
       delivered in a connectivity layer network. An NFEP is a technology
       independent representation of the Network Trail Termination Point
       associated with the NFEP.
     In addition to the attributes derived from its supertype, an NFEP
       object has the following attributes:
       directionality: this attribute specifies the directionality of the
           NFEP; the value can be one of the following: source, sink, or
           bidirectional.
       NFEPQoS: this attribute specifies the QoS (including bandwidth) for
           both the traffic originating from the NFEP and received at the
           NFEP.The QoS is specified using a sequence of tag-value pairs,
           where each tag denotes a QoS attribute. The QoS attributes
           applicable to a NFEP are technology dependent, and includes
           information such as traffic type (CBR, VBR, ABR, etc.), bandwidth
           guarantees, delay guarantees, and information loss rate.
     ";

  ATTRIBUTES
     directionality
       PERMITTED VALUES: TPDirectionality
       GET-REPLACE;
     NFEPQoS
       PERMITTED VALUES: QoSAttributeList
       GET-REPLACE;
  ACTIONS;
  NOTIFICATIONS;
```

```
REGISTERED AS ??;
```

6.5.1.6 NetworkFlowEndPointPool

```
NetworkFlowEndPointPool OBJECT TYPE
  DERIVED FROM AbstractNetworkFlowEndPoint;
  CHARACTERIZED BY NetworkFlowEndPointPool-package PACKAGE
  BEHAVIOUR NetworkFlowEndPointPool-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: This object type represents a collection of NFEPs or NFEP
        Pools. A NFEP Pool is configured using either one or more LTPs (that
        may span layer networks) or NFEP Pools.
      In addition to the attributes derived from its supertype, an NFEP Pool
        object has the following attributes:
        totalTPCount: this attribute specifies the capacity of the NFEP Pool
            in terms of the total number of NFEPs or NFEP Pools that can be
            contained in the NFEP Pool.
        allocatedTPCount: this attribute specifies the number of NFEPs or
            NFEP Pools that are currently contained in the NFEP Pool.
      ";

  ATTRIBUTES
      totalTPCount
        PERMITTED VALUES: INTEGER
        GET;
      allocatedTPCount
        PERMITTED VALUES: INTEGER
        GET-REPLACE;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

6.5.1.7 NetworkTerminationPoint

```
NetworkTerminationPoint OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY NetworkTerminationPoint-package PACKAGE
  BEHAVIOUR NetworkTerminationPoint-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: This object type represents generically either a Network
        Trail Termination Point (NWTTP) or a Network Connection Termination
        Point (NWCTP). This is a noninstantiable type and is used only for
        inheritance.
      In addition to the attributes derived from its supertype, a
        NetworkTerminationPoint object has the following attributes:
        directionality: this attribute specifies the directionality of the
            NWTP; the value can be one of the following: source, sink, or
            bidirectional.
```

```
      ";

  ATTRIBUTES
      directionality
        PERMITTED VALUES: TPDirectionality
        GET-REPLACE;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

6.5.1.8 NetworkTrailTerminationPoint

```
NetworkTrailTerminationPoint OBJECT TYPE
  DERIVED FROM NetworkTerminationPoint;
  CHARACTERIZED BY NetworkTrailTerminationPoint-package PACKAGE
  BEHAVIOUR NetworkTrailTerminationPoint-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: This object type represents an end point of a trail in a
        layer network. It represents the point at which a layer network
        receives or delivers traffic (its characteristic information).
        Traffic and bandwidth information of the traffic are represented in
        the NWCTP with which the NWTTP is associated. Only technology
        independent aspects are represented in this object type. Technology
        specific trail terminations (e.g., ATM VP layer TTP) should be
        derived from this type to represent additional information.
      ";

  ATTRIBUTES
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

6.5.1.9 NetworkConnectionTerminationPoint

```
NetworkConnectionTerminationPoint OBJECT TYPE
  DERIVED FROM NetworkTerminationPoint;
  CHARACTERIZED BY NetworkConnectionTerminationPoint-package PACKAGE
  BEHAVIOUR NetworkConnectionTerminationPoint-Behaviour BEHAVIOUR DEFINED
  AS
      "
      COMMENTS: This object type represents an end point of a link
        connection.Technology specific link connection terminations (e.g.,
        ATM VP layer CTP) should be derived from this type to represent
        additional information (such as VPI value and traffic descriptors).
      ";

  ATTRIBUTES
```

```
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

6.5.1.10 TopologicalLinkTerminationPoint

```
TopologicalLinkTerminationPoint OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY TopologicalLinkTerminationPoint-package PACKAGE
  BEHAVIOUR TopologicalLinkTerminationPoint-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: This object type represents an end point of a topological
        link. It represents the point at which the client link to server
        layer trail adaptation occurs.
      In addition to the attributes derived from its supertypes, a TLTP
        object has the following attribute:
        directionality: this attribute specifies the directionality of the
            TLTP; the value can be one of the following: source, sink, or
            bidirectional.
      Technology dependent specializations of this type may include
        additional information; e.g., ATM VP layer TLTP object may represent
        VPI range and bandwidth information. The bandwidth information
        represents the provisioned capacity.
      ";

  ATTRIBUTES
      directionality
        PERMITTED VALUES: TPDirectionality
        GET-REPLACE;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

6.5.1.11 LinkTerminationPoint

```
LinkTerminationPoint OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY LinkTerminationPoint-package PACKAGE
  BEHAVIOUR LinkTerminationPoint-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: This object type represents an end point of a link. A LTP
        is configured using one or more TLTPs. A LTP serves as a container
        for NWCTPs. When a LTP is configured, one or more NWCTPs may also be
        created.
      In addition to the attributes derived from its supertype, a LTP object
        has the following attributes:
```

```
      directionality: this attribute specifies the directionality of the
         LTP; the value can be one of the following: source, sink, or
         bidirectional.
    Technology dependent specializations of this type may include
      additional information; e.g., ATM VP layer LTP object may represent
      VPI range, total bandwidth, and available bandwidth information.
    ";

  ATTRIBUTES
    directionality
      PERMITTED VALUES: TPDirectionality
      GET-REPLACE;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

6.5.1.12 Edge

```
Edge OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY Edge-package PACKAGE
  BEHAVIOUR Edge-Behaviour BEHAVIOUR DEFINED AS
    "
    COMMENTS: This object type represents an end point of a subnetwork
      connection. An edge can be dynamically bound to a NWCTP and
      subsequently unbound.
    This object type has the following attribute:
    directionality: this specifies whether the edge is a source, sink, or
      bidirectional.
    ";
  ATTRIBUTES
    directionality
      PERMITTED VALUES:TPDirectionality
      GET;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

6.5.2 Relationship Types

6.5.2.1 BoundToCTP

```
BoundToCTP RELATIONSHIP TYPE
  CHARACTERIZED BY BoundToCTP-package PACKAGE
  BEHAVIOUR BoundToCTP-Behaviour BEHAVIOUR DEFINED AS
```

```
"
COMMENTS: The BoundToCTP relationship type represents the relationship
between an Edge object and a NWCTP object. This relationship is
established when the edge is bound to a network connection termination
point. The following rules govern the BoundToCTP relationship:
     1. The relationship has two roles: the role extremity is played
        by an Edge object; the role resource is played by a NWCTP
        object.
     2. The directionalities of the edge and NWCTP should be compatible
        as defined below:
.
```

Table 6-3. Edge and NWCTP Compatibility

Edge Directionality	NWCTP Directionality
Bidirectional	Bidirectional
Source	Sink, Bidirectional
Sink	Source, Bidirectional

```
     3. An Edge object is related with exactly one NWCTP object.
     4. A NWCTP object is related with zero or more Edge objects.";
  ROLE extremity
     RELATED TYPES Edge;
     ROLE CARDINALITY CONSTRAINT (0..N);
  ROLE resource
     RELATED TYPES NetworkConnectionTerminationPoint;
     ROLE CARDINALITY CONSTRAINT (1..1);
REGISTERED AS ??;
```

6.5.2.2 BoundToTTP

```
BoundToTTP RELATIONSHIP TYPE
  CHARACTERIZED BY BoundToTTP-package PACKAGE
  BEHAVIOUR BoundToTTP-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The BoundToTTP relationship type represents the relationship
     between a NFEP object and a NWTTP object. This relationship is
     established when the NFEP is bound to the NWTTP. The following rules
     govern the BoundToTTP relationship:
          1. The relationship has two roles: the role extremity is played
             by an NFEP object and the role resource is played by a NWTTP
             object.
          2. The directionalities of the NFEP and NWTTP should be compatible
             as defined below:
```

.

Table 6-4. NFEP and NWTTP Compatibility

NFEP Directionality	NWTTP Directionality
Bidirectional	Bidirectional
Source	Sink, Bidirectional
Sink	Source, Bidirectional

```
        3. An NFEP object participates in zero or one BoundToTTP
           relationship.
        4. A NWTTP object participates in zero or one BoundToTTP
           relationship.";

  ROLE extremity
     RELATED TYPES NetworkFlowEndPoint;
     ROLE CARDINALITY CONSTRAINT (0..1);
  ROLE resource
     RELATED TYPES NetworkTrailTerminationPoint;
     ROLE CARDINALITY CONSTRAINT (0..1);
REGISTERED AS ??;
```

6.5.2.3 TTPBoundToCTP

```
TTPBoundToCTP RELATIONSHIP TYPE
  CHARACTERIZED BY TTPBoundToCTP-package PACKAGE
  BEHAVIOUR TTPBoundToCTP-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The TTPBoundToCTP relationship type represents the
     relationship between a NWTTP object and a NWCTP object. This
     relationship is established when the NWTTP is bound to the NWCTP. The
     following rules govern the TTPBoundToCTP relationship:
        1. The relationship has two roles: the role ttp is played by an
           NWTTP object and the role ctp is played by a NWCTP object.
        2. The directionalities of the NWTTP and NWCTP should be
           identical.
        3. A NWTTP object participates in exactly one TTPBoundToCTP
           relationship.
        4. A NWCTP object participates in zero or one TTPBoundToCTP
           relationship.";
```

```
  ROLE ttp
     RELATED TYPES NetworkTrailTerminationPoint;
     ROLE CARDINALITY CONSTRAINT (0..1);
  ROLE ctp
     RELATED TYPES NetworkConnectionTerminationPoint;
     ROLE CARDINALITY CONSTRAINT (1..1);
REGISTERED AS ??;
```

6.5.2.4 HasSFEP

```
HasSFEP RELATIONSHIP TYPE
  CHARACTERIZED BY HasSFEP-package PACKAGE
  BEHAVIOUR HasSFEP-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The HasSFEP relationship type represents the relationship
     between a CPE object and an ASFEP object (that generically represents
     an SFEP or SFEP Pool). This relationship is established when the SFEP
     or SFEP Pool is created in the CPE. The following rules govern the
     HasSFEP relationship:
          1. The relationship has two roles: the role container is played
             by a CPE object and the role endpoint is played by an ASFEP
             object.
          2. A CPE object participates in zero or more HasSFEP
             relationships.
          3. An ASFEP object participates in zero or one HasSFEP
             relationship.";

  ROLE container
     RELATED TYPES CPE;
     ROLE CARDINALITY CONSTRAINT (0..1);
  ROLE endpoint
     RELATED TYPES AbstractStreamFlowEndPoint;
     ROLE CARDINALITY CONSTRAINT (0..N);
REGISTERED AS ??;
```

6.5.2.5 HasNFEP

```
HasNFEP RELATIONSHIP TYPE
  CHARACTERIZED BY HasNFEP-package PACKAGE
  BEHAVIOUR HasNFEP-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The HasNFEP relationship type represents the relationship
     between a ConnectivityLayerNetwork object and an ANFEP object (that
     generically represents an NFEP or NFEP Pool). This relationship is
     established when the NFEP or NFEP Pool is created in the boundary of
     the connectivity layer network. The following rules govern the HasNFEP
     relationship:
```

```
          1. The relationship has two roles: the role container is played
             by a CLNW object and the role endpoint is played by an ANFEP
             object.
          2. A CLNW object participates in zero or more HasNFEP
             relationships.
          3. An ANFEP object participates in zero or one HasNFEP
             relationship.";

  ROLE container
     RELATED TYPES ConnectivityLayerNetwork;
     ROLE CARDINALITY CONSTRAINT (0..1);
  ROLE endpoint
     RELATED TYPES AbstractNetworkFlowEndPoint;
     ROLE CARDINALITY CONSTRAINT (0..N);
REGISTERED AS ??;
```

6.5.2.6 HasTTP

```
HasTTP RELATIONSHIP TYPE
  CHARACTERIZED BY HasTTP-package PACKAGE
  BEHAVIOUR HasTTP-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The HasTTP relationship type represents the relationship
     between a LayerNetworkDomain object and a NWTTP object. This
     relationship is established when the NWTTP is created in the layer
     network domain (local or foreign). The following rules govern the
     HasTTP relationship:
          1. The relationship has two roles: the role container is played
             by a LND object and the role endpoint is played by an NWTTP
             object.
          2. A LND object participates in zero or more HasTTP
             relationships.
          3. A NWTTP object participates in exactly one HasTTP
             relationship.";

  ROLE container
     RELATED TYPES LayerNetworkDomain;
     ROLE CARDINALITY CONSTRAINT (1..1);
  ROLE endpoint
     RELATED TYPES NetworkTrailTerminationPoint;
     ROLE CARDINALITY CONSTRAINT (0..N);
REGISTERED AS ??;
```

6.5.2.7 HasCTP

```
HasCTP RELATIONSHIP TYPE
  CHARACTERIZED BY HasCTP-package PACKAGE
  BEHAVIOUR HasCTP-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: The HasCTP relationship type represents the relationship
      between a TLTP object and a NWCTP object. This relationship is
      established when the NWCTP is created in the topological link
      termination point. The following rules govern the HasCTP relationship:
           1. The relationship has two roles: the role container is played
              by a TLTP object and the role endpoint is played by an NWCTP
              object.
           2. A TLTP object participates in zero or more HasCTP
              relationships.
           3. A NWCTP object participates in exactly one HasCTP
              relationship.";

  ROLE container
      RELATED TYPES TopologicalLinkTerminationPoint;
      ROLE CARDINALITY CONSTRAINT (1..1);
  ROLE endpoint
      RELATED TYPES NetworkConnectionTerminationPoint;
      ROLE CARDINALITY CONSTRAINT (0..N);
REGISTERED AS ??;
```

6.5.2.8 ExtremityOfTopLink

```
ExtremityOfTopLink RELATIONSHIP TYPE
  CHARACTERIZED BY ExtremityOfTopLink-package PACKAGE
  BEHAVIOUR ExtremityOfTopLink-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: The ExtremityOfTopLink relationship type represents the
      relationship between a TopologicalLink object and a TLTP object
      representing an end point of the topological link. The following rules
      govern the ExtremityOfTopLink relationship:
           1. The relationship has two roles: the role tl is played by a
              TopologicalLink object and the role endpoint is played by a
              TLTP object.
           2. A TL object participates in exactly two ExtremityOfTopLink
              relationships.
           3. A TLTP object participates in exactly one ExtremityOfTopLink
              relationship.";

  ROLE tl
      RELATED TYPES TopologicalLink;
      ROLE CARDINALITY CONSTRAINT (1..1);
```

```
  ROLE endpoint
     RELATED TYPES TopologicalLinkTerminationPoint;
     ROLE CARDINALITY CONSTRAINT (2..2);
REGISTERED AS ??;
```

6.5.2.9 ExtremityOfLink

```
ExtremityOfLink RELATIONSHIP TYPE
  CHARACTERIZED BY ExtremityOfLink-package PACKAGE
  BEHAVIOUR ExtremityOfLink-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The ExtremityOfLink relationship type represents the
     relationship between a Link object and a LTP object representing an
     end point of the link. The following rules govern the ExtremityOfLink
     relationship:
          1. The relationship has two roles: the role resource is played
             by a Link object and the role endpoint is played by a LTP
             object.
          2. A Link object participates in exactly two ExtremityOfLink
             relationships.
          3. A LTP object participates in exactly one ExtremityOfLink
             relationship.";

  ROLE resource
     RELATED TYPES Link;
     ROLE CARDINALITY CONSTRAINT (1..1);
  ROLE endpoint
     RELATED TYPES LinkTerminationPoint;
     ROLE CARDINALITY CONSTRAINT (2..2);
REGISTERED AS ??;
```

6.5.2.10 ServedByTLTP

```
ServedByTLTP RELATIONSHIP TYPE
  CHARACTERIZED BY ServedByTLTP-package PACKAGE
  BEHAVIOUR ServedByTLTP-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The ServedByTLTP relationship type represents the
     relationship between an LTP object and a TLTP object that has been
     used to configure the LTP. The following rules govern the ServedByTLTP
     relationship:
          1. The relationship has two roles: the role ltp is played by a
             LTP object and the role tltp is played by a TLTP object.
          2. An LTP object participates in one or more ServedByTLTP
             relationships.
          3. A TLTP object participates in zero or more ServedByTLTP
             relationships.";
```

```
  ROLE ltp
     RELATED TYPES LinkTerminationPoint;
     ROLE CARDINALITY CONSTRAINT (0..N);
  ROLE tltp
     RELATED TYPES TopologicalLinkTerminationPoint;
     ROLE CARDINALITY CONSTRAINT (1..N);
REGISTERED AS ??;
```

6.5.2.11 LNDBoundedByLTP

```
LNDBoundedByLTP RELATIONSHIP TYPE
  CHARACTERIZED BY LNDBoundedByLTP-package PACKAGE
  BEHAVIOUR LNDBoundedByLTP-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The LNDBoundedByLTP relationship type represents the
     relationship between an LND object and an LTP object that represents
     an LTP that is on the boundary of the LND. The following rules govern
     the LNDBoundedByLTP relationship:
          1. The relationship has two roles: the role lnd is played by a
             LND object and the role ltp is played by a LTP object.
          2. An LND object participates in two or more LNDBoundedByLTP
             relationships.
          3. An LTP object participates in zero or one LNDBoundedByLTP
             relationships.";

  ROLE lnd
     RELATED TYPES LayerNetworkDomain;
     ROLE CARDINALITY CONSTRAINT (0..1);
  ROLE ltp
     RELATED TYPES LinkTerminationPoint;
     ROLE CARDINALITY CONSTRAINT (2..N);
REGISTERED AS ??;
```

6.5.2.12 SNWBoundedByLTP

```
SNWBoundedByLTP RELATIONSHIP TYPE
  CHARACTERIZED BY SNWBoundedByLTP-package PACKAGE
  BEHAVIOUR SNWBoundedByLTP-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The SNWBoundedByLTP relationship type represents the
     relationship between a Subnetwork object and an LTP object that
     represents an LTP that is on the boundary of the subnetwork. The
     following rules govern the SNWBoundedByLTP relationship:
```

```
        1. The relationship has two roles: the snw is played by a
           Subnetwork object and the role ltp is played by a LTP object.
        2. A Subnetwork object participates in two or more
           SNWBoundedByLTP relationships.
        3. An LTP object participates in zero or more SNWBoundedByLTP
           relationships.";
  ROLE snw
     RELATED TYPES Subnetwork;
     ROLE CARDINALITY CONSTRAINT (0..N);
  ROLE ltp
     RELATED TYPES LinkTerminationPoint;
     ROLE CARDINALITY CONSTRAINT (2..N);
REGISTERED AS ??;
```

6.5.2.13 CPEBoundedByLTP

```
CPEBoundedByLTP RELATIONSHIP TYPE
  CHARACTERIZED BY CPEBoundedByLTP-package PACKAGE
  BEHAVIOUR CPEBoundedByLTP-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The CPEBoundedByLTP relationship type represents the
     relationship between a CPE object and an LTP object that represents
     an LTP that is on the CPE. The following rules govern the
     CPEBoundedByLTP relationship:
        1. The relationship has two roles: the cpe is played by a CPE
           object and the role ltp is played by a LTP object.
        2. A CPE object participates in one or more CPEBoundedByLTP
           relationships.
        3. An LTP object participates in zero or one CPEBoundedByLTP
           relationship.";
  ROLE cpe
     RELATED TYPES CPE;
     ROLE CARDINALITY CONSTRAINT (0..1);
  ROLE ltp
     RELATED TYPES LinkTerminationPoint;
     ROLE CARDINALITY CONSTRAINT (1..N);
REGISTERED AS ??;
```

6.5.2.14 TLTPTerminatesOn

```
TLTPTerminatesOn RELATIONSHIP TYPE
  CHARACTERIZED BY TLTPTerminatesOn-package PACKAGE
  BEHAVIOUR TLTPTerminatesOn-Behaviour BEHAVIOUR DEFINED AS
     "
```

```
    COMMENTS: The TLTPTerminatesOn relationship type represents the
    relationship between a TLTP object and a Subnetwork or CPE object that
    represents the subnetwork or CPE in which the topological link
    termination occurs. The following rules govern the TLTPTerminatesOn
    relationship:

         1. The relationship has two roles: the topologicalComponent is
            played by a CPE object or Subnetwork and the role tltp is
            played by a TLTP object.

         2. A TLTP is related with at most one CPE object.

         3. A TLTP is related with zero or more Subnetwork objects.

         4. A Subnetwork or CPE object is related with one or more TLTP
            objects.";

  ROLE topologicalComponent

    RELATED TYPES CPE, Subnetwork;

    ROLE CARDINALITY CONSTRAINT (0..N);

  ROLE tltp

    RELATED TYPES TopologicalLinkTerminationPoint;

    ROLE CARDINALITY CONSTRAINT (1..N);

REGISTERED AS ??;
```

6.5.2.15 TTPAdaptsTLTP

```
TTPAdaptsTLTP RELATIONSHIP TYPE

  CHARACTERIZED BY TTPAdaptsTLTP-package PACKAGE

  BEHAVIOUR TTPAdaptsTLTP-Behaviour BEHAVIOUR DEFINED AS

    "

    COMMENTS: The TTPAdaptsTLTP relationship type represents the client-
    server adaptation that occurs between a client layer topological
    termination (represented by a TLTP object) and the server layer trail
    termination (represented by a NWTTP object). The following rules
    govern the TTPAdaptsTLTP relationship:

         1. The relationship has two roles: the tltp is played by a TLTP
            object and the role ttp is played by a NWTTP object.

         2. A TLTP is related with exactly one NWTTP object.

         3. A NWTTP is related with zero or one TLTP object.";

  ROLE tltp

    RELATED TYPES TopologicalLinkTerminationPoint;

    ROLE CARDINALITY CONSTRAINT (0..1);

  ROLE ttp

    RELATED TYPES NetworkTrailTerminationPoint;

    ROLE CARDINALITY CONSTRAINT (1..1);

REGISTERED AS ??;
```

6.5.2.16 PeerToPeer

```
PeerToPeer RELATIONSHIP TYPE

  CHARACTERIZED BY PeerToPeer-package PACKAGE
```

```
  BEHAVIOUR PeerToPeer-Behaviour BEHAVIOUR DEFINED AS
    "
    COMMENTS: The PeerToPeer relationship type represents the
    relationship between either two NWTTPs or two TLTPs of different
    characteristic information and that are in two adjacent layer network
    domains (in the same or different connectivity provider domains). A
    PeerToPeer relationship between two TLTPs represent the inter-layer
    adaptation capability that exists between the two adjacent layer
    network domains (network topology configuration view). A PeerToPeer
    relationship between two NWTTPs represent the adaptation of
    information carried by one trail to the information carried by the
    other trail (connectivity view). The following rules govern the
    PeerToPeer relationship:
         1. The relationship has two roles: the role peer1 is played by a
            NWTTP or TLTP object and the role peer2 is played by another
            distinct NWTTP or TLTP object.
         2. A NWTTP object can be related only with another distinct NWTTP
            object, and a TLTP object can be related only with another
            distinct TLTP object.
         2. A NWTTP or TLTP object can participate in at most one
            relationship.";
  ROLE peer1
    RELATED TYPES NetworkTrailTerminationPoint,
    TopologicalLinkTerminationPoint;
    ROLE CARDINALITY CONSTRAINT (0..1);
  ROLE peer2
    RELATED TYPES NetworkTrailTerminationPoint,
    TopologicalLinkTerminationPoint;
    ROLE CARDINALITY CONSTRAINT (0..1);
REGISTERED AS ??;
```

6.5.2.17 SupportedByLTP

```
SupportedByLTP RELATIONSHIP TYPE
  CHARACTERIZED BY SupportedByLTP-package PACKAGE
  BEHAVIOUR SupportedByLTP-Behaviour BEHAVIOUR DEFINED AS
    "
    COMMENTS: The SupportedByLTP relationship type represents the
    relationship between a NFEP Pool (represented by a NFEP Pool object)
    and a LTP (represented by a LTP object) that has been used to configure
    the NFEP Pool. The following rules govern the SupportedByLTP
    relationship:
         1. The relationship has two roles: the pool is played by a NFEP
            Pool object and the role ltp is played by a LTP object.
         2. A NFEP Pool is related with one or more LTP objects.The
            different LTP objects may be in different layer network
            domains of the connectivity provider.
         3. A LTP object is related with zero or one NFEP Pool object.";
  ROLE pool
    RELATED TYPES NetworkFlowEndPointPool;
    ROLE CARDINALITY CONSTRAINT (0..1);
```

```
  ROLE ltp
     RELATED TYPES LinkTerminationPoint;
     ROLE CARDINALITY CONSTRAINT (1..N);
REGISTERED AS ??;
```

7. Domain and Management Support Fragment

7.1 Introduction

The domain and management support fragment specifies object types and relationship types that are needed to support the various network resource management functions. The domain object types represent sets of objects which are either under the control of a network administration (i.e., owned by the administration) or a management function (i.e., subject to the same management policy). Management support objects support the management of domains as well as the management of resources.

7.2 Overview

The object types specified in the Domain fragment are listed in Table 7-1.

Table 7-1. Classes defined within the Domain fragment

Object types	Description
AdministrativeDomain	Represents a set of network resources and management support objects that are under the control of a single network administration. The object type LayerNetworkDomain is a subtype of this object type. An administrative domain is composed of one or more management domains.
Entity	The top class in the NRIM specification
EventForwardingDiscriminator	Represents the conditions that shall be satisfied by potential event reports.
Log	Represents the stored information of incoming event reports and local notifications. Also defines the criteria for controlling the logging of the information in a domain.
LogRecord	Generic representation of information stored in a log entry. Specific types of log records, such as Alarm Record, are subtypes of this object type.
Manageable	Represents a resource that can be subject to management.

Table 7-1. Classes defined within the Domain fragment

Object types	Description
ManagementDomain	Represents a set of resources (Manageable objects) that is administered by a management function. Associated with a management domain is a set of policies governing the management of resources under the purview of the management domain. A management domain may be composed of other management domains. A management domain has management support objects (logs, discriminators) as its components. Object types representing management domains for specific management areas (such as configuration, fault, and accounting) are subtypes of this type.

The relationship type specified in the domain fragment is shown in Table 7-2.

Table 7-2. Relationships defined in the domain fragment

Relationship types	Description
isAssignedTo	Represents the association between a management domain object and a resource object assigned to the domain
pertainsTo	Represents the association between a LogRecord object and the Manageable object that represents the resource to which the information contained in the log record pertains.

Apart from the specific relationship type defined above, the following generic relationships exist in this fragment:

- Aggregation relationship between AdministrativeDomain (composite) and Management Domain (component)
- Aggregation relationship between ManagementDomain (composite) and Management Domain (component)
- Aggregation relationship between ManagementDomain (composite) and Log (component)
- Aggregation relationship between ManagementDomain (composite) and EventForwardingDiscriminator (component)

- Inheritance relationship between Entity (supertype) and Manageable (subtype)
- Inheritance relationship between AdministrativeDomain (supertype) and LayerNetworkDomain (subtype)

7.2.1 Administrative Domain and Management Domain

In network management, the concept of a *domain* is used to denote a collection of resources that have been grouped for management purposes. The NRIM identifies two such groupings or domains:

- *Administrative Domain*: An administrative domain represents a set of resources to which an stakeholder's administrative policies are applied. Thus, an administrative domain represents a collection of resources that are owned by a single administration. The object type Layer Network Domain is a subtype of this type.
- *Management Domain*: A management domain represents a set of resources controlled by a management function. Resources are assigned to a management function to which a management policy is applied. Resources in an administrative domain are further categorized into several management domains. A resource can be assigned to more than one management domain, but it belongs to one and only one administrative domain. Further, a management domain does not span administrative domains. Figure 7-1 illustrates these concepts.

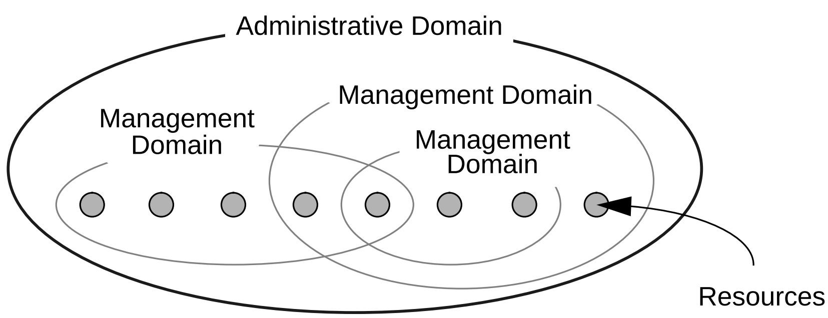


Figure 7-1. Administrative Domain and Management Domains

As shown in the figure, an administrative domain contains zero or more management domains, e.g., Resource Configuration Management (RCM) domains, Fault Management (FM) domains, and so forth.

The object type Management Domain is a base type and is further classified for specific management purposes such as resource configuration management (RCM), fault management (FM), and so forth. Each management domain, i.e., either RCM domain or FM domain, is associated with one or more resources to be managed. A resource may be assigned to one or more different types of domains, but is not assigned to more than one management domain of the same type. A management domain may contain other management domains. Assignment of resources to management domains should be consistent with the network resource topology; a domain boundary should coincide with a subnetwork boundary.

7.2.2 Manageable Resource

The Manageable Resource is a type of information object that represents the requirements for resources to be manageable within a domain. Manageable resources are further classified into specific types of manageable resources: FaultManageable, Configurable, and so on. An information object that represents a network resource (such as the SubnetworkConnection object) is a subtype of one or more of these generic types depending on the management characteristics of the resource. A manageable resource has the isAssignedTo relationship with one or more management domains. Objects that are subtypes of Manageable may participate in relationships with objects that are subtypes of ManagementDomain. For example, a Configurable object is related with a ConfigurationManagementDomain through the relationship ConfigurationManagedBy. Such relationships representing resource grouping for specific management functional areas are defined in the corresponding fragment section of this document (Sections 8,9, and 10).

7.2.3 Management Support Objects

Recall that a ManagementDomain represents a management function characterized by a set of policies governing the management of resources assigned to the management domain. In order to perform its management activities (including its interactions with other management functions), a management function may use several management support objects: logs and event forwarding discriminators. Logs are used to record management information needed by the management function (such as resource creation, resource deletion, resource modification, resource state change, alarms, and so on). See Figure 7-2. Event forwarding discriminators are used to control the communication of events by the management function to other management functions. These interactions are determined by the functional architecture of the management functions (computational viewpoint concern). In the information viewpoint, only the information aspects of these management support objects and their relationships with ManagementDomain objects are represented. In this fragment, management support objects and relationships common to all management functions are defined. Sections 8 to 10 define additional management supports objects and relationships that are specific to the individual management functional areas.

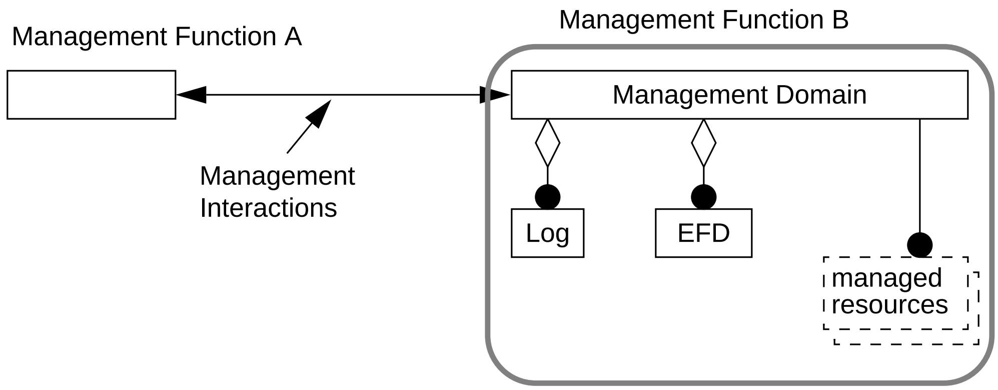


Figure 7-2. Management Support Objects

7.3 OMT Diagram for Domain and Management Support Fragment

The OMT diagram for the Domain and Management Support Fragment is shown in Figure 7-3.

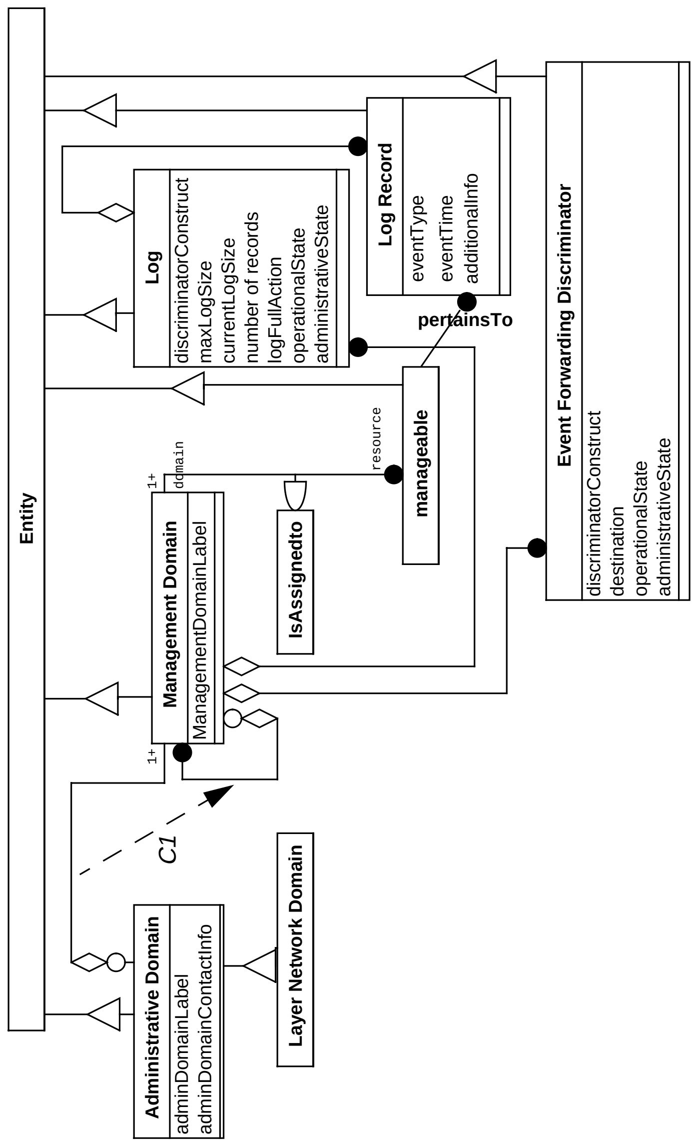


Figure 7-3. OMT Diagram for Domain and Management Support Fragment

The constraint labelled C1 is as follows:

- A ManagementDomain object participates in exactly one of the two relationships.

7.4 Quasi-GDMO Definition of the Domain and Management Support Fragment

7.4.1 Object Types

7.4.1.1 AdministrativeDomain

```
AdministrativeDomain OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY AdministrativeDomain-package PACKAGE
  BEHAVIOUR AdministrativeDomain-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: The Administrative domain object type is a type of
      information object that represents a set of resource information
      objects to which a set of administrative policies is applied. It
      represents a collection of resources under the control of an
      administration. An administrative domain may contain one or more
      management domains.;
      ";
  ATTRIBUTES
      adminDomainLabel
        PERMITTED VALUES: GraphicString
        GET-REPLACE,
      adminDomainContactInfo
        PERMITTED VALUES: GraphicString
        GET-REPLACE;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

7.4.1.2 Entity

```
Entity OBJECT TYPE
  CHARACTERIZED BY Entity-package PACKAGE
  BEHAVIOUR Entity-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: This is the top object type in NRIM. All other object types
      defined in NRIM are subtypes of this type (derived directly or
      indirectly from this type) This object type has no information element
      and is defined only for inheritance purposes.
      ";
  ATTRIBUTES;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

7.4.1.3 EventForwardingDiscriminator

```
EventForwardingDiscriminator OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY EventForwardingDiscriminator-package PACKAGE
  BEHAVIOUR EventForwardingDiscriminator-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: The event forwarding discriminator (EFD) object type is a
      type of information object that is used to define the conditions that
      shall be satisfied by potential event reports before the event report
      is forwarded to a particular destination. The definition of this
      object is based on the definition of the eventForwardingDiscriminator
      managed object class defined in X.721. The EFD function may be
      considered as a generic object management function of an underlying
      environment. However, representing this functionality in the form of
      EFD object may be necessary if a domain supports a TMN-like managed
      system view, and a managed-role entity needs to have a control over
      the event forwarding discrimination function.;
      ";
  ATTRIBUTES
      discriminatorConstruct
        PERMITTED VALUES: DiscriminatorConstruct
        GET-REPLACE,
      destination
        PERMITTED VALUES: Destination
        GET-REPLACE,
      operationalState
        PERMITTED VALUES: OperationalState
        GET,
      administrativeState
        PERMITTED VALUES: AdministrativeState
        GET-REPLACE;
  ACTIONS;
  NOTIFICATIONS
      objectCreation (resourceId: ResourceId, eventTime: GeneralizedTime,
                      creationInfo, additionalInfo: GraphicString),
      objectDeletion (resourceId: ResourceId, eventTime: GeneralizedTime,
                      deletionInfo, additionalInfo: GraphicString),
      stateChange (resourceId: ResourceId, eventTime: GeneralizedTime,
                      stateChangeInfo, additionalInfo: GraphicString);
REGISTERED AS ??;
```

7.4.1.4 Log

```
Log OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY Log-package PACKAGE
  BEHAVIOUR Log-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: The log object type is a type of information object that
      stores within a management domain incoming event reports from other
      management domains as well as notifications generated by the resources
      within the management domain. Associated with a Log object is a
      discriminator that specifies the filtering condition that must be
      satisfied by an event report or notification for it to be recorded in
      the log. The definition of this object is based on the definition of
      the log managed object class defined in X.721.;
      ";
  ATTRIBUTES
      discriminatorConstruct
        PERMITTED VALUES: DiscriminatorConstruct
        GET-REPLACE,
      maxLogSize
        PERMITTED VALUES: INTEGER
        GET-REPLACE,
      currentLogSize
        PERMITTED VALUES: INTEGER
        GET,
      numberOfRecords
        PERMITTED VALUES: INTEGER
        GET,
      logFullAction
        PERMITTED VALUES: LogFullAction
        GET-REPLACE,
      operationalState
        PERMITTED VALUES: OperationalState
        GET,
      administrativeState
        PERMITTED VALUES: AdministrativeState
        GET-REPLACE;
  ACTIONS;
  NOTIFICATIONS
      objectCreation (resourceId: ResourceId, eventTime: GeneralizedTime,
                      creationInfo, additionalInfo: GraphicString),
      objectDeletion (resourceId: ResourceId, eventTime: GeneralizedTime,
                      deletionInfo, additionalInfo: GraphicString),
```

```
     stateChange (resourceId: ResourceId, eventTime: GeneralizedTime,
                     stateChangeInfo, additionalInfo: GraphicString);
REGISTERED AS ??;
```

7.4.1.5 LogRecord

```
LogRecord OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY LogRecord-package PACKAGE
  BEHAVIOUR LogRecord-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: This object type is a type of information object that
     defines information stored in an entry in a log. The definition of
     this object type is based on the definition of the EventLogRecord
     managed object class in X.721.The attribute resourceId identifies the
     network resource to which the log entry pertains;
     ";

  ATTRIBUTES
     eventType
       PERMITTED VALUES: EventType
       GET,
     eventTime
       PERMITTED VALUES: GeneralizedTime
       GET,
     additionalInfo
       PERMITTED VALUES: GraphicString
       GET,
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

7.4.1.6 Manageable

```
Manageable OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY Manageable-package PACKAGE
  BEHAVIOUR Manageable-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The Manageable is a type of information object that
     represents a network resource that can be assigned to a management
     domain. This type is not instantiable. All object types defined in
     NRIM, except management support objects and domain related objects,
     are subtypes of this type. A manageable resource is further
     categorized into specific type of manageable resources: Fault
     Manageable, Configurable, and so on. A Manageable object mandatorily
     participates in a isAssignedTo relationship with a ManagementDomain
     object.
```

```
      ";
  ATTRIBUTES;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

7.4.1.7 ManagementDomain

```
ManagementDomain OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY ManagementDomain-package PACKAGE
  BEHAVIOUR ManagementDomain-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: The management domain object type is a type of information
      object that represents a set of resource information objects to which
      a set of management policies is applied. This object type is the base
      type for other specialized management domain types such as fault
      management domain or resource configuration domain.
      ";
  ATTRIBUTES
      managementDomainLabel
        PERMITTED VALUES: GraphicString
        GET-REPLACE;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

7.4.2 Relationship Types

7.4.2.1 isAssignedTo

```
IsAssignedTo RELATIONSHIP TYPE
  CHARACTERIZED BY IsAssignedTo-package PACKAGE
  BEHAVIOUR IsAssignedTo-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: The IsAssignedTo relationship type represents the
      relationship between a manageable resource and a management domain to
      which the resource is assigned. This relationship type is used only
      for inheritance, and only subtypes of this type are instantiated. The
      following rules govern the IsAssignedTo relationship:
           1. The relationship has two roles: the role resource is played by
              a Manageable object and the role domain is played by a
              ManagementDomain object.
           2. A Manageable object R is related with a ManagementDomain object
              D if and only if the resource represented by R is under the
              purview of the management domain represented by D.
           3. A Manageable object participates in one or more IsAssignedTo
              relationships.
```

```
        4. A ManagementDomain object participates in zero or more
           IsAssignedTo relationships.
    ";
  ROLE resource
    RELATED TYPES Manageable;
    ROLE CARDINALITY CONSTRAINT (0..N);
  ROLE domain
    RELATED TYPES ManagementDomain;
    ROLE CARDINALITY CONSTRAINT (1..N);
REGISTERED AS ??;
```

7.4.2.2 pertainsTo

```
PertainsTo RELATIONSHIP TYPE
  CHARACTERIZED BY PertainsTo-package PACKAGE
  BEHAVIOUR PertainsTo-Behaviour BEHAVIOUR DEFINED AS
    "
    COMMENTS: The PertainsTo relationship type represents the
    relationship between a log record and the manageable resource to which
    the information in the log record pertains. The following rules govern
    the PertainsTo relationship:
        1. The relationship has two roles: the role resource is played
           by a Manageable object and the role record is played by a
           LogRecord object.
        2. A Manageable object R is related with a LogRecord object L if
           and only if the information contained in L pertains to the
           resource represented by R.
        3. A Manageable object participates in zero or more PertainsTo
           relationships.
        4. A LogRecord object participates in exactly one PertainsTo
           relationship.
    ";
  ROLE resource
    RELATED TYPES Manageable;
    ROLE CARDINALITY CONSTRAINT (1..1);
  ROLE record
    RELATED TYPES LogRecord;
    ROLE CARDINALITY CONSTRAINT (0..N);
REGISTERED AS ??;
```

8. Resource Configuration Fragment

8.1 Introduction

This fragment of the NRIM specification describes the information elements (objects and relationships) needed for network resource configuration management functions. In line with the scope of this NRIM specification, this fragment is not dependent on any specific functional architecture for network resource configuration management. This fragment identifies the information that network resources ought to provide so that they can be subject to configuration management. The specification of a functional architecture for network resource configuration management, such as one that is specified in [1]. may define additional information elements that are derived from the information elements defined in this fragment.

Although the focus is on the configuration management of network resources, the information elements can be applicable to other types of resources such as service resources and computing resources.

8.2 Overview of the Resource Configuration Fragment

The object types identified in the resource configuration fragment are described briefly in this section and are listed in Table 8-1.

Table 8-1. Object types specified in the resource configuration fragment

Object types	Description
Configurable	Represents a network resource that can be subject to configuration management. This is a subtype of Manageable.
ConfigurationManagementDomain	Represents a set of Configurable objects that is controlled by a configuration management function. Associated with a configuration management function is a set of policies that govern the configuration management of all Configurable objects under the purview of the function. This is a subtype of ManagementDomain.
ObjectCreationRecord	Represents information contained in a log entry pertaining to the creation of a Configurable object. This is a subtype of LogRecord.

Table 8-1. Object types specified in the resource configuration fragment

Object types	Description
ObjectDeletionRecord	Represents information contained in a log entry pertaining to the deletion of a Configurable object. This is a subtype of LogRecord.
StateChangeRecord	Represents information contained in a log entry pertaining to a change in either the administrative state or the operational state of a Configurable object. This is a subtype of LogRecord.
AttributeValueChangeRecord	Represents information contained in a log entry pertaining to a change in the value of an attribute of a Configurable object. This is a subtype of LogRecord

Table 8-2. Relationship types specified in the resource configuration fragment

Relationships types	Description
ConfigurationManagedBy	This relationship exists between a Configurable object and a ConfigurationManagementDomain object when the resource represented by the Configurable object is managed by the configuration management function represented by the ConfigurationManagement domain object. This is a subtype of IsAssignedTo.
ReportsConfigurationEventsTo	This relationship exists between a Configurable object and a ManagementDomain object if and only if notifications related to the configuration aspects of the resource represented by the Configurable object are sent to the management function represented by the Management domain object

It should be noted that a connection management function is one kind of Configuration Management Domain. It configures different types of connectivity resources: trails, subnetwork connections, and so on.

The following restrictions apply to the generic relationships defined in the Domain and Management Support Fragment (Section 7):

- A ConfigurationManagementDomain object can be related via aggregation only with ConfigurationManagementDomain objects.
- A Log object aggregated under a ConfigurationManagementDomain object is an aggregate of only the following types of log records: CreationRecord, DeletionRecord, StateChangeRecord, and AttributeValueChangeRecord.

Figure 8-1 illustrates the OMT diagram related to the Resource Configuration Fragment. The object type *Configurable* is derived from the object type *Manageable*. The *Configurable* object represents in a generic manner any resource that can be subject to configuration management. Attributes of *Configurable* are *administrativeState* (X.721) and *operationalState* (X.721). The following notifications (X.721) relevant to configuration management are defined for Configurable objects: objectCreation, objectDeletion, stateChange, and attributeValueChange.

8.3 OMT Diagram for Resource Configuration Fragment

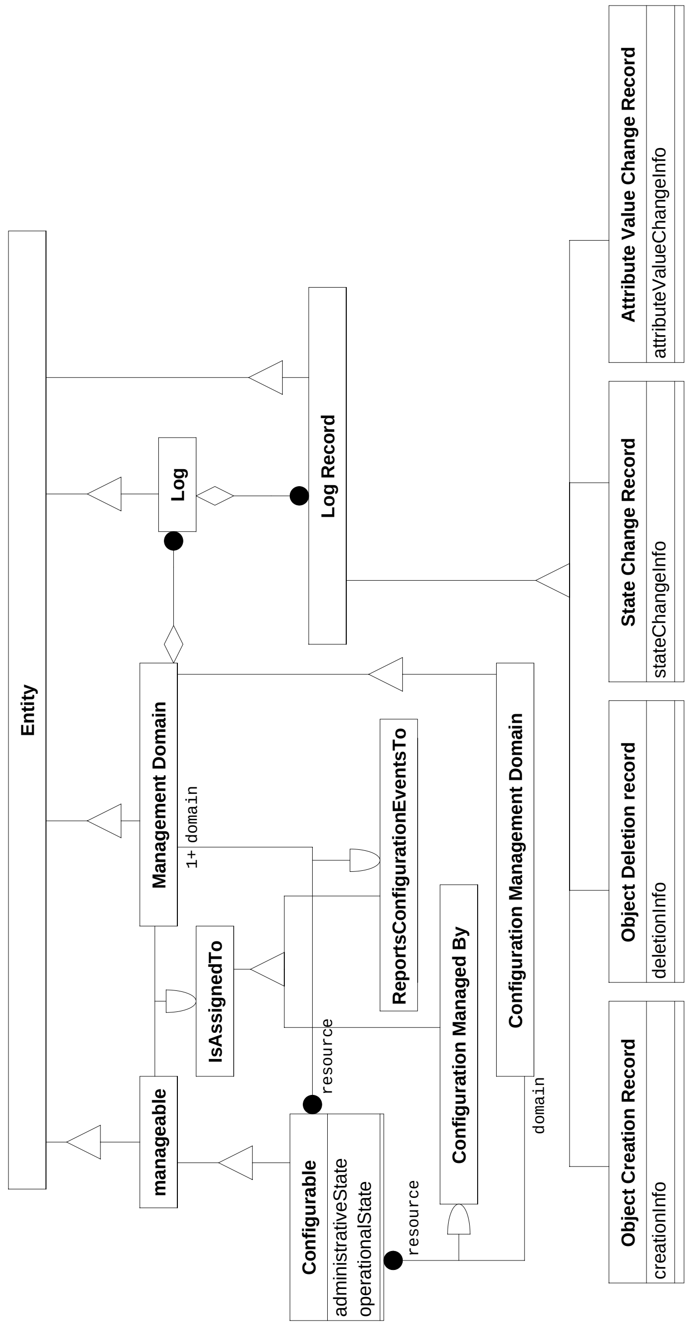


Figure 8-1. OMT Diagram for Resource Configuration Fragment

8.4 Quasi-GDMO Definition of the Resource Configuration Fragment

8.4.1 Object Types

8.4.1.1 Configurable

```
Configurable OBJECT TYPE
  DERIVED FROM Manageable;
  CHARACTERIZED BY Configurable-package PACKAGE
  BEHAVIOUR Configurable-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: This object type represents a network resource that can be
        subject to configuration management; i.e., the availability of the
        resource for use can be administratively controlled; changes in the
        operational state of the resource can be monitored; changes in the
        values of the attributes representing resource properties can be
        monitored; creation and deletion of the resource can be monitored.
        This object type has two attributes:
        administrativeState: possible values are “locked”, “shutting down”,
          and “unlocked”; the value “locked” denotes that the resource is not
          available for use; “unlocked” denotes that the resource is
          available for use; “shutting down” denotes that the use of the
          resource is permitted only to users that are currently using the
          resource.
        operationalState: possible values are “disabled” and “enabled”;
          “disabled” denotes that the resource is totally inoperable;
          “enabled” denotes that the resource is fully or partially operable.
        The following notifications are defined for this object type:
        objectCreation: the triggering condition is the creation of the
          object.
        objectDeletion: the triggering condition is the deletion of the
          object.
        stateChange: the triggering condition is a change in the
          administrative state or the operational state of the resource.
        attributeValueChange: the triggering condition is a change in the
          value of an attribute (including those defined in the subtypes) of
          the object.
      ";
  ATTRIBUTES
      administrativeState
        PERMITTED VALUES: AdministativeState
        GET-REPLACE,
      operationalState
        PERMITTED VALUES: OperationalState
        GET;
  ACTIONS;
  NOTIFICATIONS
      objectCreation (resourceId: ResourceId, eventTime: GeneralizedTime,
                      creationInfo, additionalInfo: GraphicString),
```

```
     objectDeletion (resourceId: ResourceId, eventTime: GeneralizedTime,
                      deletionInfo, additionalInfo: GraphicString),
     stateChange (resourceId: ResourceId, eventTime: GeneralizedTime,
                      stateChangeInfo, additionalInfo: GraphicString),
     attributeValueChange (resourceId: ResourceId, eventTime:
                      GeneralizedTime, attrValChangeInfo,additionalInfo:
                      GraphicString);
REGISTERED AS ??;
```

8.4.1.2 ConfigurationManagementDomain

```
ConfigurationManagementDomain OBJECT TYPE
  DERIVED FROM ManagementDomain;
  CHARACTERIZED BY ConfigurationManagementDomain-package PACKAGE
  BEHAVIOUR ConfigurationManagementDomain-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: This object type represents a set of Configurable resources
       that are under the purview of a configuration management function.
       All network resources that are under the purview of a configuration
       management function are subject to the policies associated with the
       configuration management function.
     ";
  ATTRIBUTES;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

8.4.1.3 ObjectCreationRecord

```
ObjectCreationRecord OBJECT TYPE
  DERIVED FROM LogRecord;
  CHARACTERIZED BY ObjectCreationRecord-package PACKAGE
  BEHAVIOUR ObjectCreationRecord-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: This object type is a type of information object that
     defines information stored in an entry in a log that is created based
     on the information contained in an objectCreation notification. This
     object type is a subtype of LogRecord. The attribute creationInfo
     contains information specific to the object creation, such as the
     initial state of the object.
     ";
  ATTRIBUTES
     creationInfo
       PERMITTED VALUES: GraphicString
       GET,
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

8.4.1.4 ObjectDeletionRecord

```
ObjectDeletionRecord OBJECT TYPE
  DERIVED FROM LogRecord;
  CHARACTERIZED BY ObjectDeletionRecord-package PACKAGE
  BEHAVIOUR ObjectDeletionRecord-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: This object type is a type of information object that defines
     information stored in an entry in a log that is created based on the
     information contained in an objectDeletion notification. This object
     type is a subtype of LogRecord. The attribute deletionInfo contains
     information specific to the object deletion, such as the final state
     of the object.
     ";
  ATTRIBUTES
     deletionInfo
       PERMITTED VALUES: GraphicString
       GET,
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

8.4.1.5 StateChangeRecord

```
StateChangeRecord OBJECT TYPE
  DERIVED FROM LogRecord;
  CHARACTERIZED BY StateChangeRecord-package PACKAGE
  BEHAVIOUR StateChangeRecord-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: This object type is a type of information object that defines
     information stored in an entry in a log that is created based on the
     information contained in a stateChange notification. This object type
     is a subtype of LogRecord. The attribute stateChangeInfo contains
     information specific to the state change event, such as the name of
     the state attribute whose value has changed, its previous value, and
     the new value.
     ";
  ATTRIBUTES
     stateChangeInfo
       PERMITTED VALUES: GraphicString
       GET,
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

8.4.1.6 AttributeValueChangeRecord

```
AttributeValueChangeRecord OBJECT TYPE
```

```
  DERIVED FROM LogRecord;
  CHARACTERIZED BY AttributeValueChangeRecord-package PACKAGE
  BEHAVIOUR AttributeValueChangeRecord-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: This object type is a type of information object that
     defines information stored in an entry in a log that is created based
     on the information contained in an attributeValueChange notification.
     This object type is a subtype of LogRecord. The attribute
     attributeValueChangeInfo contains information specific to the
     attribute value change event, such as names of the attributes that
     have changed, their previous values, and the new values.
     ";
  ATTRIBUTES
     attributeValueChangeInfo
       PERMITTED VALUES: GraphicString
       GET,
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

8.4.2 Relationship Types

8.4.2.1 ConfigurationManagedBy

```
ConfigurationManagedBy RELATIONSHIP TYPE
  DERIVED FROM IsAssignedTo;
  CHARACTERIZED BY ConfigurationManagedBy-package PACKAGE
  BEHAVIOUR ConfigurationManagedBy-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The ConfigurationManagedBy relationship type represents the
     relationship between a configurable resource and a configuration
     management domain to which the resource is assigned. This relationship
     type is a subtype of IsAssignedTo.Thus, the existence of this
     relationship between two objects implies the existence of an
     IsAssignedTo relationship between the same objects. The following
     rules govern the ConfigurationManagedBy relationship:
          1. The relationship has two roles: the role resource is played
             by a Configurable object and the role domain is played by a
             ConfigurationManagementDomain object.
          2. A Configurable object R is related with a
             ConfigurationManagementDomain object D if and only if the
             configuration aspects of the resource represented by R are
             managed by the configuration management function represented
             by D.
          3. A Configurable object participates in exactly one
             ConfigurationManagedBy relationship.
          4. A ConfigurationManagementDomain object participates in zero
             or more ConfigurationManagedBy relationships.
          5. The ConfigurationManagementDomain object to which a
             Configurable object is related may change over time.";
```

```
  ROLE resource
     RELATED TYPES Configurable;
     ROLE CARDINALITY CONSTRAINT (0..N);
  ROLE domain
     RELATED TYPES ConfigurationManagementDomain;
     ROLE CARDINALITY CONSTRAINT (1..1);
REGISTERED AS ??;
```

8.4.2.2 ReportsConfigurationEventsTo

```
ReportsConfigurationEventsTo RELATIONSHIP TYPE
  DERIVED FROM IsAssignedTo;
  CHARACTERIZED BY ReportsConfigurationEventsTo-package PACKAGE
  BEHAVIOUR ReportsConfigurationEventsTo-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The ReportsConfigurationEventsTo relationship type
     represents the relationship between a configurable resource and a
     management domain to which the configuration events related to the
     resource are reported. This relationship type is a subtype of
     IsAssignedTo.Thus, the existence of this relationship between two
     objects implies the existence of an IsAssignedTo relationship between
     the same objects.The following rules govern the
     ReportsConfigurationEventsTo relationship:

          1. The relationship has two roles: the role resource is played by
             a Configurable object and the role domain is played by a
             ManagementDomain object.

          2. A Configurable object R is related with a ManagementDomain
             object D if and only if the configuration events related to
             the resource represented by R are reported to the management
             function represented by D.

          3. A Configurable object participates in one or more
             ReportsConfigurationEventsTo relationships.

          4. A ManagementDomain object participates in zero or more
             ReportsConfigurationEventsTo relationships.

          5. The ManagementDomain objects to which a Configurable object is
             related may change over time.
     ";
  ROLE resource
     RELATED TYPES Configurable;
     ROLE CARDINALITY CONSTRAINT (0..N);
  ROLE domain
     RELATED TYPES ManagementDomain;
     ROLE CARDINALITY CONSTRAINT (1..N);
REGISTERED AS ??;
```

9. Fault Management Fragment

9.1 Introduction

Fault management fragment specifies the management support information objects for fault management. Although the focus is on the fault management of network resources, the information elements can be applicable to other types of resources such as service resources and computing resources. TINA fault management functional area addresses the following five activities: alarm surveillance, fault localization, fault correction, testing/diagnostics, and trouble administration. This version of the document covers information for alarm surveillance and fault localization.

9.2 Overview

Table 9-1 shows the object types specified in the fault management fragment.

Table 9-1. Object types specified in the Fault Management Fragment

Object types	Description
AlarmRecord	Represents the alarm information stored in a Log. This is a subtype of LogRecord.
AlarmSeverityAssignmentProfile	Specifies the assignment of alarm severity to different types of alarms. Each profile object may specify different severity assignments.
CurrentAlarmSummaryControl	Specifies criteria for the generation of a current alarm summary report.
FaultManageable	Represents the management information that a network resource has to provide so that it can be subject to fault management. This is a subtype of Manageable.
FaultManagementDomain	Represents a set of FaultManageable objects that is controlled by a fault management function. Associated with a fault management domain is a set of policies that govern the fault management of all objects in the domain. This is a subtype of ManagementDomain.

Table 9-2 shows the relationship types specified in the fault management fragment.

Table 9-2. Relationships specified in the fault management fragment

Relationship types	Description
SeverityAssignment	Represents the association of a FaultManageable resource with an Alarm Severity Assignment Profile object.

Relationship types	Description
AlarmSurveyedBy	Represents the association between a FaultManageable resource and a Current Alarm Summary Control object. An object is included in a current alarm summary report only if the alarms associated with the resource satisfy the criteria specified in the associated Current Alarm Summary Control object.
FaultManagedBy	Represents the association between a FaultManageable object and a FaultManagementDomain object corresponding to the fault management function that governs the resource represented by the FaultManageable object. This is subtype of IsAssignedTo.
ReportsAlarmsTo	Represents the association between a FaultManageable resource and a Management Domain to which the alarms from the resource are reported. This is a subtype of IsAssignedTo.

The following restrictions apply to the generic relationships defined in the Domain and Management Support Fragment (Section 7):

- A FaultManagementDomain object can be related via aggregation only with FaultManagementDomain objects.
- A Log object aggregated under a FaultManagementDomain object is an aggregate of only AlarmRecord objects.

9.2.1 FaultManageable

The object type FaultManageable defines management information that a network resource has to provide so that it can be subject to fault management. The following attributes are defined for the object type FaultManageable:

- alarmStatus: this is a set valued attribute that describes the status of alarm conditions currently present in the resource. The value of this attribute is any combination of the following values (See X.731 for details):
  - UnderRepair: the resource is currently being repaired.
  - Critical: one or more critical alarms have been detected in the resource, and have not been cleared.
  - Major: one or more major alarms have been detected in the resource, and have not been cleared.
  - Minor: one or more minor alarms have been detected in the resource, and have not been cleared.
  - AlarmOutstanding: one or more alarms have been detected in the resource

- currentProblemList: this attribute identifies the current existing problems in the resource including their severity (See M.3100 for details).

The following notifications (based on X.721) are defined for the object type FaultManageable:

- communicationsAlarm: the triggering condition for this notification is detection of a communications error in the resource.
- qualityofserviceAlarm: the triggering condition for this notification is detection of degradation of a quality of service characteristic associated with the resource.

Other types of alarm notifications defined in M.3100, such as environmentalAlarm and equipmentAlarm are network element level alarms, and thus are not specified in NRIM.

An alarm notification contains the following information:

- Probable Cause: The probable cause qualifies the alarm. The following probable cause values are specified by OSI Alarm Reporting Function and may also be used in a TINA-C environment.

  Communication Alarms may be caused by the following failures:.

  - Loss of Signal
  - Loss of frame
  - Framing error
  - Local node transmission error
  - Remote node transmission error
  - Call establishment error
  - Degraded signal
  - Communications subsystem failure
  - Communication protocol error
  - LAN error
  - DTE-DCE interface error

  QoS Alarms may be caused by the following failures:

  - Response time exceeded
  - Queue size exceeded
  - Bandwidth reduced
  - Retransmission rate exceeded
  - Threshold crossed
  - Performance degraded
  - Congestion

- Resource at or nearing capacity
- Perceived Severity: This attribute indicates the severity of the alarm. The severity levels can be assigned and changed by a fault management function using an alarm severity assignment profile associated with the resource. The following five severity levels are defined in the OSI Alarm Reporting Function:
  - Indeterminate. The severity level cannot be determined
  - Critical. The condition is service affecting and immediate corrective action is required. (e.g., the resource is out of service).
  - Major. The condition is service affecting and urgent corrective action is required. (e.g., the resource capability is degraded).
  - Minor. The condition is not service affecting but corrective action is required to prevent more serious fault.
  - Warning. Potential or impending service affecting fault is detected before any significant effects have been felt.

A FaultManageable object is related with

- a FaultManagementDomain object via a FaultManagedBy relationship,
- one or more ManagementDomain objects via ReportsAlarmsTo relationships, and
- an Alarm Severity Assignment Profile object via a Severity Assignment relationship.
- a CurrentAlarmSummaryControl object via an AlarmSurveyedBy relationship.

9.2.2 Fault Management Domain

The Fault Management Domain is a type of information object which represents a set of resource objects that are under the purview of a fault management function and thus are governed by the same fault management policy. A FaultMangeable resource is related with exactly one FaultManagementDomain object. A fault management domain may have a recursive structure, i.e., a fault management domain may contain other fault management domains. The assignment of resources to fault management domains should be consistent with the network resource topology, e.g., a domain boundary should coincide with a subnetwork boundary.

A fault management domain may contain one or more logs for logging alarm records and one or more alarm severity assignment profiles. The alarm severity assignment can be different within different fault management domains, e.g. the alarm severity of an alarm type might be critical in one domain and minor in another, and so forth. In addition, a fault management domain may support information for the current alarm summary reporting service which is defined in ITU-T Recommendation Q.821 [20]; i.e., a fault management domain may provide to other management domains current alarm summary reports. The criteria for generation of such alarm summary reports are specified in a management support object, called CurrentAlarmSummaryControl, contained in the fault management domain.

9.2.3 Alarm Severity Assignment Profile

The alarm severity assignment profile object type specifies the alarm severity assignment for objects. Instances of this object type are associated with fault-manageable resource objects via SeverityAssignment relationships.

The semantics of a AlarmSeverityAssignmentProfile object is described in ITU-T M.3100 Generic Network Information Model [8].

The alarm severity values may be assigned to each alarm type and cause of alarm. Table 9-3 lists a possible assignment of the severity to alarm types and probable causes.

Table 9-3. Example assignment of Alarm Severity

Alarm Type	Probable Cause	Alarm Severity	Detection
Communications Alarm	loss of Signal	Critical	NE level Network level
	loss of frame	Critical	NE level Network level
	framing error	Critical	NE level Network level
	local node transmission error	Major	NE level
	remote node trans mission error	Minor	NE level
	call establishment error	Minor	NE level
	degraded signal	Critical	NE level
	communications subsystem failure	Critical	NE level
	communication protocol error	Minor	NE level
	LAN error	Major	NE level
	DTE-DCE interface error	Major	NE level

Table 9-3. Example assignment of Alarm Severity

Alarm Type	Probable Cause	Alarm Severity	Detection
QOS Alarm	Response time exceeded	Major	NE level, Network level
	queue size exceeded	Minor	NE level, Network level
	bandwidth reduced	Minor	NE level
	retransmission rate exceeded	Major	NE level, Network level
	threshold crossed	Indeterminate	NE level, Network level
	performance degraded	Major	NE level, Network level
	congestion	Major	NE level
	resource at or nearing capacity	Major	NE level, Network level

9.3 OMT Diagram for Fault Management Fragment

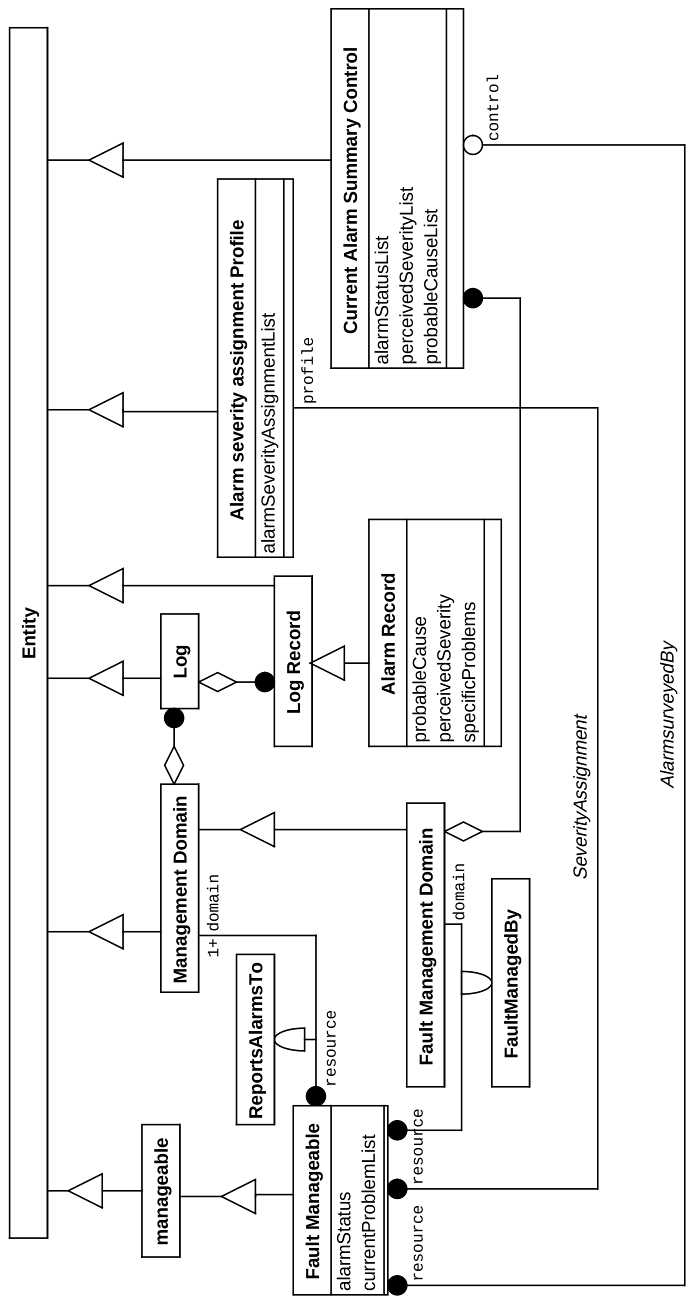


Figure 9-1. OMT Diagram for Fault Management Fragment

9.4 Quasi - GDMO Definition of the Fault Management Fragment

9.4.1 Object Types

9.4.1.1 AlarmRecord

```
AlarmRecord OBJECT TYPE
  DERIVED FROM LogRecord;
  CHARACTERIZED BY AlarmRecord-package PACKAGE
  BEHAVIOUR AlarmRecord-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: The AlarmRecord object type is a type of information object
      that represents the information stored in the log as a result of
      receiving alarm notifications or alarm reports. The definition of this
      object is based on the definition in X.721.;
      ";
  ATTRIBUTES
      probableCause
        PERMITTED VALUES: ProbableCause
        GET,
      perceivedSeverity
        PERMITTED VALUES: Severity
        GET,
      specificProblems
        PERMITTED VALUES: GraphicString
        GET,
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

9.4.1.2 AlarmSeverityAssignmentProfile

```
AlarmSeverityAssignmentProfile OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY AlarmSeverityAssignmentProfile-package PACKAGE
  BEHAVIOUR AlarmSeverityAssignmentProfile-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: The alarm severity assignment profile object type is a type
      of information object that specifies the alarm severity assignment for
      objects. The definition of this object is based on the definition in
      ITU-T M.3100. An object that refers to the alarm severity has the
      SeverityAssignment relationship with an alarm severity assignment
      profile object.;
      ";
  ATTRIBUTES;
      alarmSeverityAssignmentList
```

```
      PERMITTED VALUES: AlarmSeverityAssignmentList
      GET-REPLACE
      ADD-REMOVE,
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

9.4.1.3 CurrentAlarmSummaryControl

```
CurrentAlarmSummaryControl OBJECT TYPE
  DERIVED FROM Entity;
  CHARACTERIZED BY CurrentAlarmSummaryControl-package PACKAGE
  BEHAVIOUR CurrentAlarmSummaryControl-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The current alarm summary control is a type of management
     support object that provides the criteria for the generation of
     current alarm summary reports. A FaultMangeable object R is included
     in an alarm summary report defined by a CurrentAlarmSummaryControl
     object C if and only if all the following conditions are satisfied.
     See ITU-T Recommendation Q.821 for details:
       R and C are related via an AlarmSurveyedBy relationship.
       The values of the alarmStatus attribute of R matches one of the
       values included in the alarmStatusList attribute of C.
       R has an outstanding alarm with a Perceived Severity that matches
       one of the values included in the perceivedSeverityList attribute of
       C
       R has an outstanding alarm with a Probable Cause that matches one of
       the values included in the probableCauseList attribute of C.
     This object type has three attributes:
       alarmStatusList: this is a set of possible Alarm Status.
       perceivedSeverityList: this is a set of possible Perceived
       Severities.
       probableCauseList: this is a set of possible Probable Causes.
     ";
  ATTRIBUTES
     alarmStatusList
       PERMITTED VALUES: AlarmStatusList
       GET-REPLACE
       ADD-REMOVE,
     perceivedSeverityList
       PERMITTED VALUES: SeverityList
       GET-REPLACE
       ADD-REMOVE,
     probableCauseList
       PERMITTED VALUES: ProbableCauseList
       GET-REPLACE
```

```
      ADD-REMOVE;
  ACTIONS,
  NOTIFICATIONS;
REGISTERED AS ??;
```

9.4.1.4 FaultManageable

```
FaultManageable OBJECT TYPE
  DERIVED FROM Manageable;
  CHARACTERIZED BY FaultManageable-package PACKAGE
  BEHAVIOUR FaultManageable-Behaviour BEHAVIOUR DEFINED AS
    "
    COMMENTS: This object type represents a network resource that can be
    subject to fault management. Such a resource has to provide management
    information defined in this object type.
    This object type has two attributes:
      alarmStatus: this is a set valued attribute that describes the
      status of alarm conditions currently present in the resource. The
      value of this attribute is any combination of the following values:
        UnderRepair: the resource is currently being repaired.
        Critical: one or more critical alarms have been detected in the
          resource, and have not been cleared.
        Major: one or more major alarms have been detected in the resource,
          and have not been cleared.
        Minor: one or more minor alarms have been detected in the resource,
          and have not been cleared.
        AlarmOutstanding: one or more alarms have been detected in the
          resource
      currentProblemList: this attribute identifies the current existing
      problems in the resource including their severity.
    The following notifications (based on X.721) are defined for the
    object type FaultManageable:
      communicationsAlarm: the triggering condition for this notification
        is detection of a communications error in the resource.
      qualityofserviceAlarm: the triggering condition for this
        notification is detection of degradation of a quality of service
        characteristic associated with the resource.
    ";
  ATTRIBUTES
    alarmStatus
      PERMITTED VALUES: AlarmStatus
      GET,
    currentProblemList
      PERMITTED VALUES: GraphicString
      GET;
  ACTIONS;
  NOTIFICATIONS
```

```
     communicationsAlarm (probableCause: ProbableCause, perceivedSeverity:
        Severity, specificProblems: GraphicString),
     qualityofServiceAlarm (probableCause: ProbableCause,
        perceivedSeverity: Severity, specificProblems: GraphicString);
REGISTERED AS ??;
```

9.4.1.5 FaultManagementDomain

```
FaultManagementDomain OBJECT TYPE
  DERIVED FROM ManagementDomain;
  CHARACTERIZED BY FaultManagementDomain-package PACKAGE
  BEHAVIOUR FaultManagementDomain-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: This object type represents a set of FaultManageable
       resources that are under the purview of a fault management function.
       All network resources that are under the purview of a fault
       management function are subject to the policies associated with the
       fault management function.
     ";
  ATTRIBUTES;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

9.4.2 Relationship Types

9.4.2.1 FaultManagedBy

```
FaultManagedBy RELATIONSHIP TYPE
  DERIVED FROM IsAssignedTo;
  CHARACTERIZED BY FaultManagedBy-package PACKAGE
  BEHAVIOUR FaultManagedBy-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The FaultManagedBy relationship type represents the
     relationship between a fault manageable resource and a fault
     management domain to which the resource is assigned. This relationship
     type is a subtype of IsAssignedTo. Thus, the existence of this
     relationship between two objects implies the existence of an
     IsAssignedTo relationship between the same objects. The following
     rules govern the FaultManagedBy relationship:
          1. The relationship has two roles: the role resource is played by
             a FaultManageable object and the role domain is played by a
             FaultManagementDomain object.
          2. A FaultManageable object R is related with a
             FaultManagementDomain object D if and only if the fault
             aspects of the resource represented by R are managed by the
             fault management function represented by D.
          3. A FaultManageable object participates in exactly one
             FaultManagedBy relationship.
          4. A FaultManagementDomain object participates in zero or more
             FaultManagedBy relationships.
```

```
          5. The FaultManagementDomain object to which a FaultManageable
             object is related may change over time.";
  ROLE resource
     RELATED TYPES FaultManageable;
     ROLE CARDINALITY CONSTRAINT (0..N);
  ROLE domain
     RELATED TYPES FaultManagementDomain;
     ROLE CARDINALITY CONSTRAINT (1..1);
REGISTERED AS ??;
```

9.4.2.2 ReportsAlarmsTo

```
ReportsAlarmsTo RELATIONSHIP TYPE
  DERIVED FROM IsAssignedTo;
  CHARACTERIZED BY ReportsAlarmsTo-package PACKAGE
  BEHAVIOUR ReportsAlarmsTo-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The ReportsAlarmsTo relationship type represents the
     relationship between a fault manageable resource and a management
     domain to which alarms related to the resource are reported. This
     relationship type is a subtype of IsAssignedTo. Thus, the existence
     of this relationship between two objects implies the existence of an
     IsAssignedTo relationship between the same objects. The following
     rules govern the ReportsAlarmsTo relationship:
          1. The relationship has two roles: the role resource is played
             by a FaultManageable object and the role domain is played by
             a ManagementDomain object.
          2. A FaultManageable object R is related with a ManagementDomain
             object D if and only if alarms related to the resource
             represented by R are reported to the management function
             represented by D.
          3. A FaultManageable object participates in one or more
             ReportsAlarmsTo relationships.
          4. A ManagementDomain object participates in zero or more
             ReportsAlarmsTo relationships.
          5. The ManagementDomain objects to which a FaultManageable object
             is related may change over time.
     ";
  ROLE resource
     RELATED TYPES FaultManageable;
     ROLE CARDINALITY CONSTRAINT (0..N);
  ROLE domain
     RELATED TYPES ManagementDomain;
     ROLE CARDINALITY CONSTRAINT (1..N);
REGISTERED AS ??;
```

9.4.2.3 SeverityAssignment

```
SeverityAssignment RELATIONSHIP TYPE
  CHARACTERIZED BY SeverityAssignment-package PACKAGE
  BEHAVIOUR SeverityAssignment-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: The SeverityAssignment relationship type represents the
      relationship between a FaultManageable object and an
      AlarmSeverityAssignmentProfile object that specifies the severity
      assignment for alarms emitted by the resource represented by the
      FaultManageable object. The following rules govern the
      SeverityAssignment relationship:
           1. The relationship has two roles: the role resource is played by
              a FaultManageable object and the role profile is played by a
              AlarmSeverityAssignmentProfile object.
           2. A FaultManageable object R is related with a
              AlarmSeverityAssignmentProfile object P if and only if the
              severity assignment for the alarms emitted by the resource
              represented by R is defined by the profile represented by P.
           3. A FaultManageable object participates in exactly one
              SeverityAssignment relationship.
           4. A AlarmSeverityAssignmentProfile object participates in zero
              or more SeverityAssignment relationships.
           5. The AlarmSeverityAssignmentProfile object to which a
              FaultManageable object is related may change over time.
      ";
  ROLE resource
      RELATED TYPES FaultManageable;
      ROLE CARDINALITY CONSTRAINT (0..N);
  ROLE profile
      RELATED TYPES AlarmSeverityAssignmentProfile;
      ROLE CARDINALITY CONSTRAINT (1..1);
REGISTERED AS ??;
```

9.4.2.4 AlarmSurveyedBy

```
AlarmSurveyedBy RELATIONSHIP TYPE
  CHARACTERIZED BY AlarmSurveyedBy-package PACKAGE
  BEHAVIOUR AlarmSurveyedBy-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: The AlarmSurveyedBy relationship type represents the
      relationship between a FaultManageable object and a
      CurrentAlarmSummaryControl object that specifies the criteria for the
      inclusion of alarms emitted by the resource represented by the
      FaultManageable object in alarm summary reports. The following rules
      govern the AlarmSurveyedBy relationship:
           1. The relationship has two roles: the role resource is played by
              a FaultManageable object and the role control is played by a
              CurrentAlarmSummaryControl object.
```

```
        2. A FaultManageable object R is related with a
           CurrentAlarmSummaryControl object P if and only if the
           criteria for the inclusion of alarms emitted by the resource
           represented by R in alarm summary reports is defined by the
           alarm summary control object represented by P.
        3. A FaultManageable object participates in zero or one
           AlarmSurveyedBy relationship.
        4. A CurrentAlarmSummaryControl object participates in zero or
           more AlarmSurveyedBy relationships.
        5. The CurrentAlarmSummaryControl object to which a
           FaultManageable object is related may change over time.
     ";
  ROLE resource
     RELATED TYPES FaultManageable;
     ROLE CARDINALITY CONSTRAINT (0..N);
  ROLE control
     RELATED TYPES CurrentAlarmSummaryControl;
     ROLE CARDINALITY CONSTRAINT (0..1);
REGISTERED AS ??;
```

10.Accounting Management Fragment

10.1 Introduction

The Accounting Management fragment specifies the management support information objects for accounting management. The focus is on the accounting management of network resources, but the information elements defined here are applicable also to service resources and computing resources. The specification of a functional architecture for network resource accounting management, such as one that is specified in [1]. may define additional information elements that are derived from the information elements defined in this fragment

10.2 Overview

Table 10-1 shows the object types specified in the accounting management fragment.

Table 10-1. Object types specified in the Accounting Management Fragment

Object types	Description
Accountable	Represents the information that a network resource has to provide so that it can be subject to accounting management. This is a subtype of Manageable.
AccountingManagementDomain	Represents a set of Accountable objects that is controlled by an accounting management function. Associated with an accounting management function is a set of policies that govern the accounting management of all objects in the domain. This is a subtype of Management Domain.
AccountingRecord	Represents the accounting information stored in a log entry. This is a subtype of LogRecord.

Table 10-2 shows the relationship types specified in the accounting management fragment.

Table 10-2. Relationships specified in the Accounting Management fragment

Relationship types	Description
AccountingManagedBy	Represents the association between an Accountable object and an AccountingManagementDomain object corresponding to the accounting management function that governs the resource represented by the Accountable object. This is a subtype of IsAssignedTo.
ReportsAccountingDataTo	Represents the association between an Accountable resource and a Management Domain to which the accounting data on the resource is reported. This is a subtype of IsAssignedTo.

The following restrictions apply to the generic relationships defined in the Domain and Management Support Fragment (Section 7):

- An AccountingManagementDomain object can be related via aggregation only with AccountingManagementDomain objects.
- A Log object aggregated under an AccountingManagementDomain object is an aggregate of only AccountingRecord objects.

10.2.1 Accountable

The object type Accountable defines the management information that a network resource has to provide so that it can be subject to accounting management. An accountable resource gathers accounting statistics on an interval basis. The length of this accounting interval as well as the enabling and disabling of the accounting statistics gathering capability of the resource. are controlled by the accounting management domain to which the resource is assigned. The assignment of an Accountable resource to an accounting management domain is represented using a AccountingManagedBy relationship. When the accounting statistics gathering capability of a resource is enabled, the resource gathers the statistics for each accounting interval, and reports the accounting data for each interval by sending a AccountingReport notification. Such notifications can be received by any management domain that subscribes to these notifications. This subscription relationship is represented using a relationship type called ReportsAccountingDataTo.

Since NRIM is technology independent, it specifies only one kind of accounting statistics: traffic, measured in number of bits, transported during an accounting interval. Information models for specific technologies may specify additional accounting statistics. See [1] for examples of accounting statistics that are applicable to ATM technology.

10.2.2 Accounting Management Domain

The object type Accounting Management Domain represents a set of Accountable objects that are under the purview of an accounting management function and thus are governed by the same accounting management policy. Accountable objects have a AccountingManagedBy relationship with an accounting management domain. An accounting management domain may have a recursive structure, i.e., an accounting management domain may contain other accounting management domains. The assignment of resources to domains should be consistent with the network resource topology, e.g., a domain boundary should coincide with a subnetwork boundary.

An accounting management domain may contain one or more logs for logging information contained in AccountingReport notifications received from Accountable resources in the domain. The information contained in these notifications are stored in AccountingRecord objects.

10.3 OMT Diagram for Accounting Management Fragment

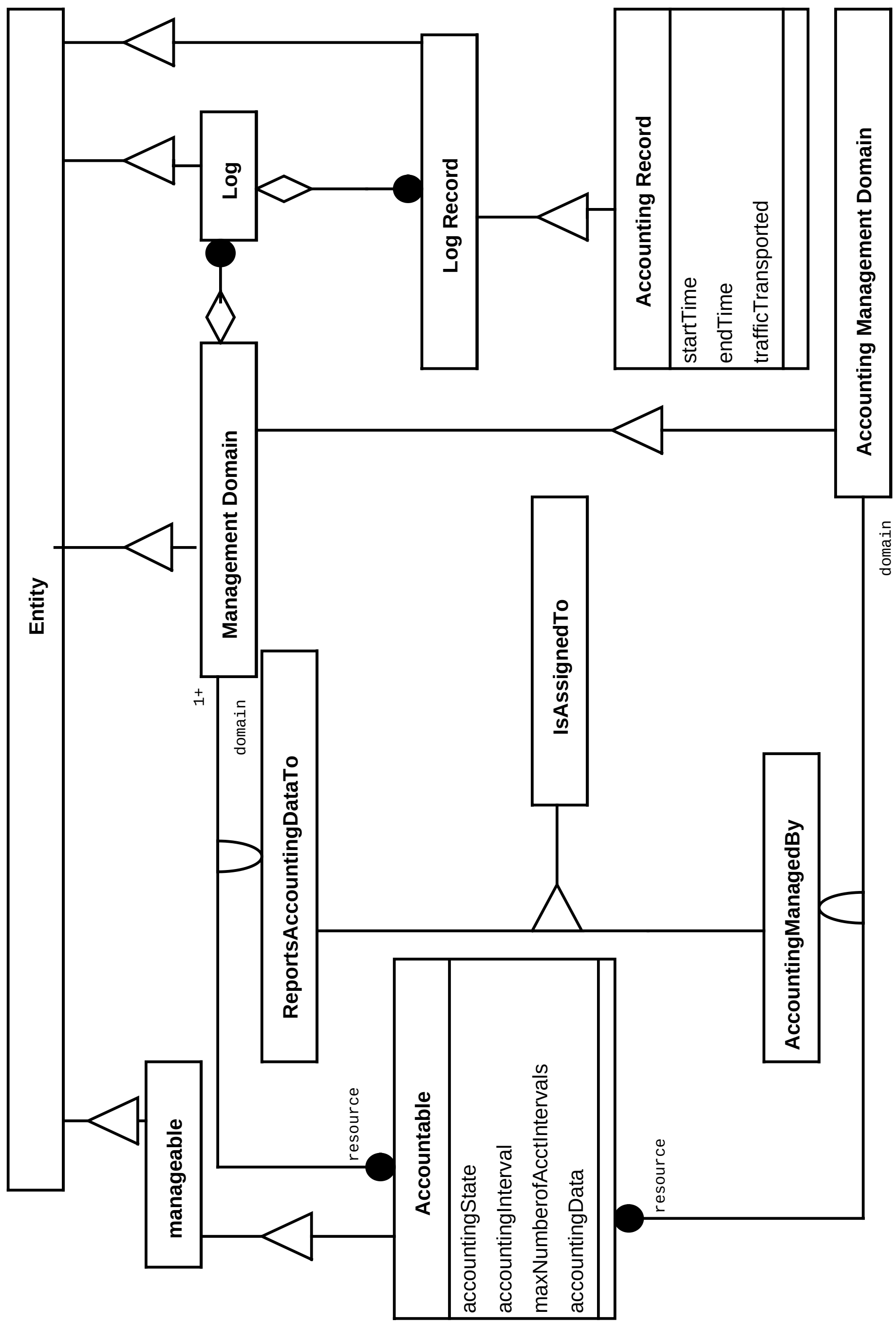


Figure 10-1. OMT Diagram for Accounting Management Fragment

10.4 Quasi - GDMO Definition of the Accounting Management Fragment

10.4.1 Object Types

10.4.1.1 Accountable

```
Accountable OBJECT TYPE
  DERIVED FROM Manageable;
  CHARACTERIZED BY Accountable-package PACKAGE
  BEHAVIOUR Accountable-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: This object type represents a network resource that can be
       subject to accounting management; i.e., the accounting parameters
       regarding the resource can be monitored.
       This object type has four attributes:
       accountingState: possible values are “enabled” and “disabled”; the
         value “enabled” denotes that gathering of accounting parameters by
         the resource is enabled; the value “disabled” denotes that
         gathering of accounting parameters by the resource is disabled. If
         the accountingState is enabled, the resource gathers accounting
         statistics and periodically reports the statistics. The period is
         determined by the value of the attribute accountingInterval.
       accountingInterval: this attribute specifies the length of the time
         interval that constitutes an accounting interval; accounting
         statistics are gathered for each accounting interval.
       The following notification is defined for this object type:
       accountingReport: the triggering condition is the elapse of an
         accounting interval. The following information is included in the
         notification: the start time of the interval, the end time for the
         interval, and the amount of traffic, in bits, transported during
         the interval.
     ";
  ATTRIBUTES
     accountingState
       PERMITTED VALUES: AccountingState
       GET-REPLACE,
     accountingInterval
       PERMITTED VALUES: AccountingInterval
       GET-REPLACE;
  ACTIONS;
  NOTIFICATIONS
     AccountingReport (startTime: GeneralizedTime, endTime:
                       GeneralizedTime, trafficVolume: REAL);
REGISTERED AS ??;
```

10.4.1.2 AccountingManagementDomain

```
AccountingManagementDomain OBJECT TYPE
  DERIVED FROM ManagementDomain;
  CHARACTERIZED BY AccountingManagementDomain-package PACKAGE
  BEHAVIOUR AccountingManagementDomain-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: This object type represents a set of Accountable resources
       that are under the purview of an accounting management function. All
       network resources that are under the purview of an accounting
       management function are subject to the policies associated with the
       accounting management function.
     ";
  ATTRIBUTES;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

10.4.1.3 AccountingRecord

```
AccountingRecord OBJECT TYPE
  DERIVED FROM LogRecord;
  CHARACTERIZED BY AccountingRecord-package PACKAGE
  BEHAVIOUR AccountingRecord-Behaviour BEHAVIOUR DEFINED AS
     "
     COMMENTS: The AccountingRecord object type is a type of information
     object that represents the information stored in a log as a result of
     receiving AccountingReport notifications.
     ";
  ATTRIBUTES
     startTime
       PERMITTED VALUES: GeneralizedTime
       GET,
     endTime
       PERMITTED VALUES: GeneralizedTime
       GET,
     trafficVolume
       PERMITTED VALUES: REAL
       GET;
  ACTIONS;
  NOTIFICATIONS;
REGISTERED AS ??;
```

10.4.2 Relationship Types

10.4.2.1 AccountingManagedBy

```
AccountingManagedBy RELATIONSHIP TYPE
  DERIVED FROM IsAssignedTo;
  CHARACTERIZED BY AccountingManagedBy-package PACKAGE
  BEHAVIOUR AccountingManagedBy-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: The AccountingManagedBy relationship type represents the
      relationship between an Accountable resource and an accounting
      management domain to which the resource is assigned. This relationship
      type is a subtype of IsAssignedTo. Thus, the existence of this
      relationship between two objects implies the existence of an
      IsAssignedTo relationship between the same objects. The following
      rules govern the AccountingManagedBy relationship:
           1. The relationship has two roles: the role resource is played
              by an Accountable object and the role domain is played by an
              AccountingManagementDomain object.
           2. An Accountable object R is related with an
              AccountingManagementDomain object D if and only if the
              accounting aspects of the resource represented by R are
              managed by the accounting management function represented by
              D.
           3. An Accountable object participates in exactly one
              AccountingManagedBy relationships.
           4. An AcountingManagementDomain object participates in zero or
              more AccountingManagedBy relationships.
           5. The AccountingManagementDomain object to which an Accountable
              object is related may change over time.";
  ROLE resource
      RELATED TYPES Accountable;
      ROLE CARDINALITY CONSTRAINT (0..N);
  ROLE domain
      RELATED TYPES AccountingManagementDomain;
      ROLE CARDINALITY CONSTRAINT (1..1);
REGISTERED AS ??;
```

10.4.2.2 ReportsAccountingDataTo

```
ReportsAccountingDataTo RELATIONSHIP TYPE
  DERIVED FROM IsAssignedTo;
  CHARACTERIZED BY ReportsAccountingDataTo-package PACKAGE
  BEHAVIOUR ReportsAccountingDataTo-Behaviour BEHAVIOUR DEFINED AS
      "
      COMMENTS: The ReportsAccountingDataTo relationship type represents
      the relationship between an accountable resource and a management
      domain to which the accounting statistics related to the resource are
      reported. This relationship type is a subtype of IsAssignedTo. Thus,
      the existence of this relationship between two objects implies the
      existence of an IsAssignedTo relationship between the same objects.
      The following rules govern the ReportsAccountingDataTo relationship:
           1. The relationship has two roles: the role resource is played by
              an Accountable object and the role domain is played by a
              ManagementDomain object.
           2. An Accountable object R is related with a ManagementDomain
              object D if and only if the accounting statistics related to
              the resource represented by R are reported to the management
              function represented by D.
           3. An Accountable object participates in one or more
              ReportsAccountingDataTo relationships.
           4. A ManagementDomain object participates in zero or more
              ReportsAccountingDataTo relationships.
           5. The ManagementDomain objects to which an Accountable object is
              related may change over time.
      ";
  ROLE resource
      RELATED TYPES Accountable;
      ROLE CARDINALITY CONSTRAINT (0..N);
  ROLE domain
      RELATED TYPES ManagementDomain;
      ROLE CARDINALITY CONSTRAINT (1..N);
REGISTERED AS ??;
```

11. Acknowledgments

We acknowledge the contributions of the authors of the previous versions of this document. These authors are listed below:

- Authors of 1993 version: Lars Richter (Telia Research, Sweden), Masaki Wakano (NTT, Japan) and Hiroshi Oshigiri (NEC, Japan).
- Authors of 1994/1995 version: Motoharu Kawanishi (Oki, Japan), Magnus Lengdell (Telia Research, Sweden), Lars Richter (Telia Research, Sweden), Hiroshi Oshigiri NEC, Japan), and Juan Pavon (Alcatel SESA, Spain).

We thank all Core-team members (96-97) for their review comments and all members of the Resource Stream for discussions and ideas that have helped us to develop this revised Network Resource Information Model. Special thanks to Frank Steegmans and Jarno Rajahalme for several suggestions that improved the model and the type definitions.

We also thank Naiem Dathi (HP) who validated the Network Resource Information Model and provided valuable feedback to the model.

Finally, we thank the following external reviewers for their comments and suggestions that have led to further improvements in the model: Daniel Ranc (Alcatel), Christine Pageot-Millet (CNET/France Telecom), Tero Koskivirta (Nokia), Sakae Chikara (NTT), Anne-Grete Karasen (Telenor), Terje Henriksen (Telenor), and Leif Bystrom (Telia).

N. Natarajan
Bellcore
U.S.A.

Hannu Flinck
Nokia
Finland

R. M. Rosli
Telecom Malaysia
Malaysia

Appendix A: Changes from Last Version

The major changes made to the previous version of the NRIM document, i.e., NRIM 95 document [4], are listed below:

- The following new concepts have been introduced:
  - Connectivity Layer Network
  - Layer Network Domain
  - Local Layer Network Domain
  - Foreign Layer Network Domain
  - Link and its relationship with Topological Link
  - CPE
  - Stream Flow Connection
  - Network Flow Connection
  - Terminal Flow Connection
  - Stream Flow End Point
  - Stream Flow End Point Pool
  - Network Flow End Point
  - Network Flow End Point Pool
- The concepts of Logical Connection Graph, Physical Connection Graph, and Nodal Connection Graph have been revised. The new model is based on a uniform set of concepts, such as source end point, sink end point, and branch. The NRIM 95 concepts of Port, Vertex, and Line have been removed.
- The following changes have been made to align NRIM with ITU-T SG4/Q18 and ATM Forum M4 Network View specifications
  - The NRIM 95 concept of Connection has been renamed as Link Connection
  - The NRIM 95 concept of Edge has been revised to align with the concept of Subnetwork Termination Point defined in SG4/Q18 and M4 Network View specifications
- The Network Fragment and Connectivity Fragment have been revised to position them as views from the perspective of a connectivity provider
- A new relationship called PertainsTo has been introduced in the Domain and Management Support Fragment. It relates a LogRecord object to the object that represents the resource to which the information in the log record pertains.
- In the Resource Configuration Fragment, the following NRIM 95 objects have been removed to position NRIM as a pure resource information model that is not specific to any management functional architecture:
  - RCData
  - ConfigData

- The Accounting Management Fragment is new
- A clear separation has been made between the objects in the resource fragments (Network, Connectivity, and Termination Point) and objects in the management function fragments (Resource Configuration, Fault Management, and Accounting Management). Objects on in the resource fragment (e.g, Subnetwork) are not derived from objects in the management function fragments (e.g., Configurable). Such derivations are now outside the scope of NRIM and are viewed as decisions pertaining to the design of specific management functions (or services).
- The Adapter Fragment of NRIM 95 has been removed. Different forms of adaptation functions are represented using relationships in Network Fragment, Connectivity Fragment, and Termination Point Fragment.
- The Quasi-GDMO+GRM definitions have been simplified. Role Bindings are described in-line in relationship definitions following GRM conventions. This makes the specification less verbose.
- Chapter 3 of NRIM 95 has been revised as follows. Comparisons with relevant standards have been moved to Appendix C. The chapter is expanded to provide a comprehensive introduction to the basic concepts used in NRIM.
- The following appendices of NRIM 95 have been removed:
  - Appendix A: Usage of Rumbaugh Graphical Notation (The description was too brief to be of any use)
  - Appendix C: Usage of objects (the text was incorrect; the usage is better illustrated through examples in the new version)
  - Appendix D: Origin of Managed Objects (this description is now contained in Appendix C of the new version)
  - Appendix E: Reuse Fragment within NRIM (the necessary definitions have been included in the appropriate fragment in the body of the document to increase readability)

Appendix B: Type Definitions

```
-- --------------------------------------------------

-- Alphabetical list of defined types within TINA NRIM

-- --------------------------------------------------


AccountingDataList::= SEQUENCE OF AccountingReportData
    -- one element in the sequence per accounting interval


AccountingInterval::= REAL          -- in seconds


AccountingReportData ::= SEQUENCE {
    startTime       GeneralizedTime,
    endTime         GeneralizedTime,
    trafficVolume   REAL        -- in bits
    }

AccountingState ::= ENUMERATED {
    disabled (0),
    enabled (1)
    }

AdministrativeState ::= ENUMERATED {            -- adopted from X.731
    locked (0),             -- prohibited from providing services
    shuttingDown (1),       -- prohibited from serving new users
    unlocked (2)            -- permitted to perform services
    }

AlarmSeverityAssignment ::= SEQUENCE {
    problem                                 [0] ProbableCause,
    severityAssignedServiceAffecting        [1] Severity OPTIONAL,
    severityAssignedNonServiceAffecting     [2] Severity OPTIONAL,
    severityAssignedNonServiceIndependent   [3] Severity OPTIONAL
    }
```

```
AlarmSeverityAssignmentList ::= SET OF AlarmSeverityAssignment

AlarmStatus ::= ENUMERATED {
        underRepair (0),
        critical (1),
        minor (2),
        major (3),
        alarmOutstanding (4)
        }

AlarmStatusList ::= SET OF AlarmStatus

ConnectionTopology ::= ENUMERATED {
    pt-pt-unidirectional (0),
    pt-pt-bidirectional (1) ,
    pt-multipt-unidirectional (2)
    }

CharacteristicInfo ::= OBJECT IDENTIFIER        -- adopted from M.3100

DiscriminatorConstruct ::= GraphicString

Destination ::= GraphicString

EventType::= ENUMERATED {
    objectCreation (0),
    objectDeletion (1),
    attrValChange (2),
    stateChange (3),
    alarm (4),
    accounting (5)
    }

FlowType ::= OBJECT IDENTIFIER
```

```
LinkDirectionality ::= ENUMERATED {
    unidirectional (0),
    bidirectional (1)
    }

LogFullAction ::= ENUMERATED {
    ignore (0),        -- do not write to the log
    overwrite (1)      -- overwrite from the beginning
    }

OperationalState ::= ENUMERATED {               -- adopted from X.731
    disabled (0),
    enabled (1)
    }

ProbableCause ::= OBJECT IDENTIFIER

ProbableCauseList ::= SET OF ProbableCause

QoSAttribute ::= SEQUENCE {
    id       OBJECT IDENTIFIER,
    value    ANY
    }

QoSAttributeList ::= SET OF QoSAttribute

Severity ::= ENUMERATED {
        indeterminate (0),
        warning (1)
        minor (2),
        major (3),
        critical (4),
        }
```

```
SeverityList ::= SET OF Severity

TPDirectionality ::= ENUMERATED {
    source (0),
    sink (1),
    bidirectional (2)
    }
```

Appendix C: Standards and other Sources that influenced the Specification

Many of the modelling concepts used in NRIM are based on similar concepts defined or used in other standards in network modelling. Table C-1 lists, for each key concept used in the NRIM, the original standard source that defined the concept, and other standards that have adopted a similar concept. The list is organized in alphabetical order.

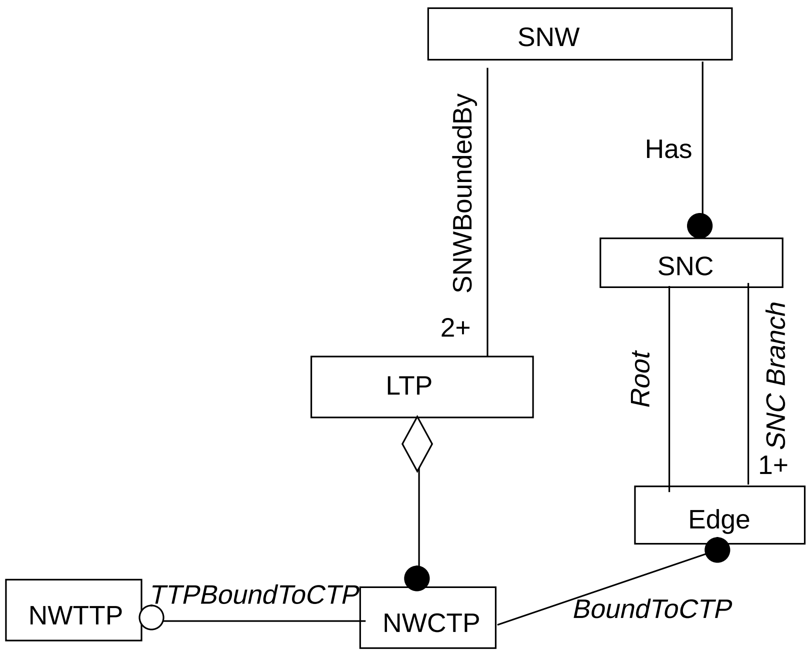
Table C-1. Origin of NRIM Concepts and Related Standards

NRIM Concept	Origin	Related Standards
Connectivity Layer Network	TINA-C	-
Edge	INA	ITU-T G.853, ATMF M4 Network View (called Subnetwork Termination Point)
Layer Network	ITU-T G.803, G.805	ITU-T G.853, ATMF M4 Network View, ETSI GOM
Layer Network Domain	-	ITU-T G.853, ATMF M4 Network View, ETSI GOM
Link	ITU-T G.803, G.805	ITU-T G.853, ATMF M4 Network View, ETSI GOM
Link Connection	ITU-T G.803, G.805	ITU-T G.853, ATMF M4 Network View, ETSI GOM
Link Termination Point	TINA-C	ATMF M4 Network View
Logical Connection Graph	TINA-C	-
Network Connection Termination Point	ITU-T M.3100 (NE View)	ITU-T G.853, ATMF M4 Network View, ETSI GOM
Network Flow Connection	TINA-C	-

NRIM Concept	Origin	Related Standards
Network Flow End Point	TINA-C	-
Network Trail Termination Point	ITU-T M.3100 (NE View)	ITU-T G.853, ATMF M4 Network View, ETSI GOM
Nodal Connection Graph	TINA-C	-
Physical Connection Graph	TINA-C	-
Stream Flow Connection	RM-ODP, TINA-C	-
Stream Flow End Point	TINA-C	-
Subnetwork	ITU-T G.803, G.805	ITU-T G.853, ATMF M4 Network View, ETSI GOM
Subnetwork Connection	ITU-T G.803, G.805	ITU-T G.853, ATMF M4 Network View, ETSI GOM
Tandem Connection	ITU-T G.803, G.805	ITU-T G.853
Terminal Flow Connection	TINA-C	-
Topological Link	TINA-C	-
Topological Link Termination Point	TINA-C	-
Trail	ITU-T G.803, G.805	ITU-T G.853, ATMF M4 Network View, ETSI GOM

Appendix D: Relationship to Network Element Level Aspects

This section shows the relationship between the objects in the network resource information model specification (this document) and objects describing network element level aspects. The objects of interest are depicted in Figure D-1.



SNW Subnetwork
SNC Subnetwork Connection
LTP Link Termination Point
NWCTP Network Connection Termination Point
NWTTP Network Trail Termination Point

Figure D-1. Network Resource Objects

The smallest subnetwork, i.e., a subnetwork that is not decomposed further into smaller subnetworks, corresponds to a single network element. This section describes how objects related to the smallest subnetwork correspond to objects defined in the following technology specific models:

- G.774 SDH model
- ATM Forum M4 NE View model (or GR-1114 ATM NE model).

These models are switching and technology specific models describing network element level aspects, i.e., they describe the resource structure within a network element. G.774 and M4 NE View are both based upon M.3100 Generic Network Information Model.

D.1 Relationship to G.774

G.774 is based on M.3100 Generic Information Model. Basically, G.774 introduces a large number of SDH specific termination point object classes. These are all specializations of the M.3100 class Termination Point, and they describe the standardized signal types and bit rates. G.774 does not introduce new objects for describing connectivity between the termination points, instead the M.3100 objects such as cross connection, multipoint XC (crossconnect) and fabric are reused. For completeness, both G.774 and the M.3100 object classes are shown in Figure D-2.

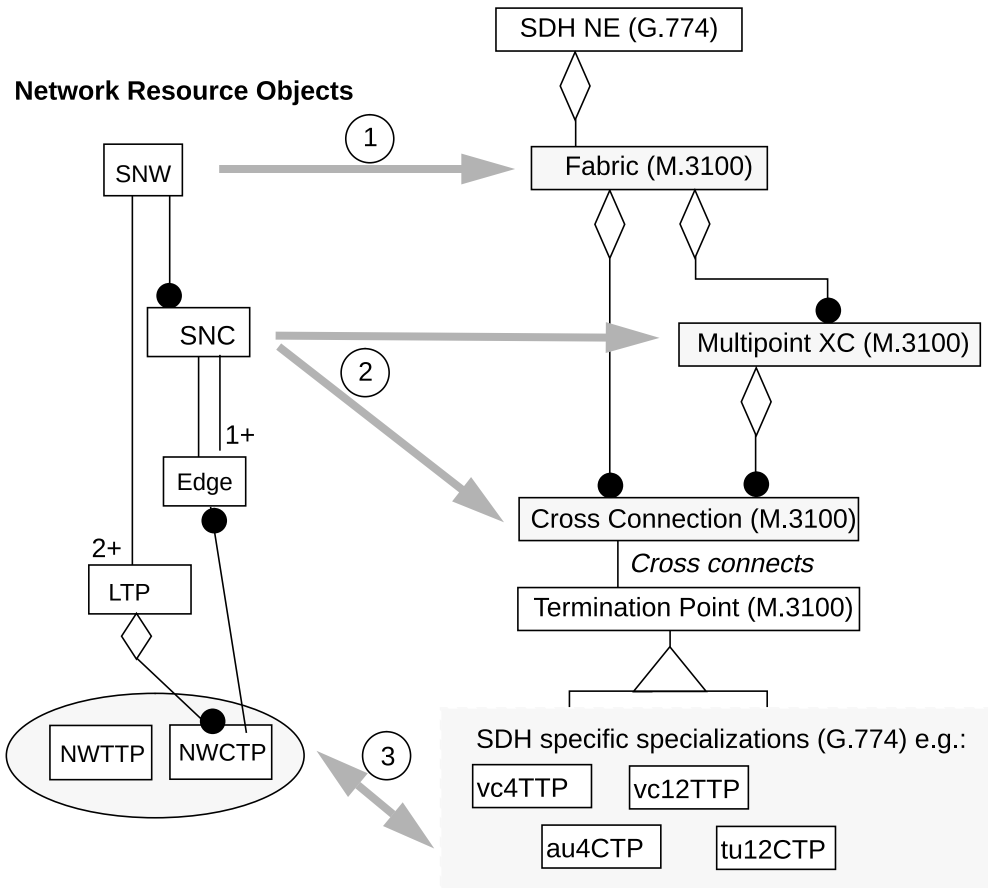


Figure D-2. G.774 and M.3100 object class mapping

The figure also shows the related object classes in the network resource information model. The mapping is shown as shaded arrows (they should be interpreted as follows):

1. Each instance of the lowest level of subnetwork (SNW) maps to one fabric instance in a network element (SDH NE)

2. The mapping of a SNC to a NE level cross-connection depends on the topology type:

   2.1 If it is a point-to-point connection, the subnetwork connection (SNC) object maps to a cross connect object.

   2.2 If the connection topology is point-to-multipoint, the subnetwork connection (SNC) object maps to many i cross connect objects. It also maps to a multipoint crossconnect object (multipoint XC)

3. Each instance of a network trail termination point (NWTTP) object or network connection termination point (NWCTP) object, maps to one instance of the SDH specific specializations.

D.2 Relationship to ATMF M4 NE View (or GR-1114)

The ATM Forum M4 NE View specifies an information model for the management of ATM switching systems. It is also based on M.3100. Figure D-3 illustrates some of the important object classes in M4 NE View.

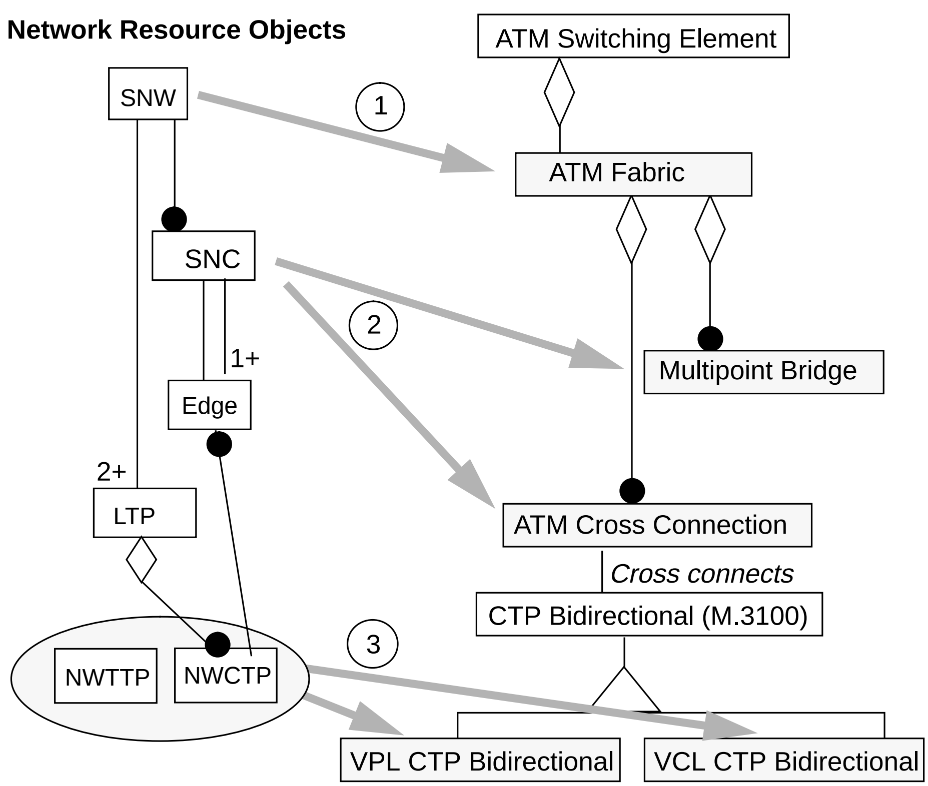


Figure D-3. M4 NE View object class mapping

ATM switching element object class represents an ATM switching system.

ATM fabric object is responsible for the establishment and release of ATM cross connections, i.e., to cross connect two or many termination points.

The ATM cross connection object represents the assignment relationship between the termination points. There are two types of connections in ATM. These are virtual channel links (VCL) and virtual path links (VPL).

The multipoint bridge represent the ATM bridging function required by ATM switching systems for support of multipoint connections.

The related network resource information model classes are also shown in Figure D-3. The mappings are as follows:

1. Each instance of the lowest level of subnetwork (SNW) maps to one ATM fabric instance in an ATM switching element
2. The mapping of a SNC to NE level cross-connections depends on the topology type:
   - 2.1 If it is a point-to-point connection, the subnetwork connection (SNC) object maps to an ATM cross connection object.
   - 2.2 If the connection topology is point-to-multipoint, the subnetwork connection (SNC) object maps to many ATM cross connection objects. It also maps to a multipoint bridge object
3. Each instance of a network connection termination point (NWCTP) object maps to either a VPL CTP bidirectional or VCL CTP bidirectional object class.

References

TINA-C Deliverables

[1] *Network Resource Architecture*, Version 3.0, Document No. NRA_v3.0_97_02_10, TINA-C, February 1997.

[2] *Information Modelling Concepts*, Document No. TB_EAC.001_3.0_94, TINA-C, December 1994.

[3] *Computational Modelling Concepts*, Document No. TB_HC.012_3.0_95, TINA-C, August 1995.

[4] *Network Resource Information Model*, Document No. TB_LR.010_2.1_95, TINA-C, August 1995.

[5] *The ConS Reference Point*, Version 1.0, February 1997.

[6] *Management Architecture*, Document No. TB_GN.010_2.0_93, TINA-C, December 1994.

Standards

[7] ITU-T Recommendation M.3010, *Principles for a Telecommunications Management Network,* 1992.

[8] ITU-T Recommendation M.3100, *Generic Network Information Model*, 1992.

[9] ITU-T Recommendation G.803, *Architectures of Transport Networks Based on the Synchronous Digital Hierarchy (SDH),* June 1992.

[10] ITU-T Recommendation G.805, *Architectures of Transport Networks,* June 1995.

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

[12] ISO/IEC IS 10165-1 / ITU-T Recommendation X.720, *Information Technology - Open Systems Interconnection - Structure of Management Information - Part 1: Management Information Model,* International Organization for Standardization and International Electrotechnical Committee, August 1991.

[13] ISO/IEC 10165-2 / ITU-T Recommendation X.721, *Information Technology - Open Systems Interconnection - Structure of Management Information (SMI) - Part 2: Definition of Management Information,* International Organization for Standardization and International Electrotechnical Committee, September 1991.

[14] ISO/IEC DIS 10165-4 / ITU-T Recommendation X.722, *Information Technology - Open Systems Interconnection - Structure of Management Information - Part 4: Guidelines for the Definition of Managed Objects (GDMO),* International Organization for Standardization and International Electrotechnical Committee, September 1991.

[15] ISO/IEC 10164-2 / ITU-T Recommendation X.731: *Information Technology - Open*

Systems Interconnection - Systems Management - Part 1: State Management Function, International Organization for Standardization and International Electrotechnical Committee, August 1991.

[16] ISO/IEC 10164-4 / ITU-T Recommendation X.733: *Information Technology - Open Systems Interconnection - Systems Management - Part 1: Alarm Report Function,* International Organization for Standardization and International Electrotechnical Committee, August 1991.

[17] ISO/IEC 10164-5 / ITU-T Recommendation X.734: *Information Technology - Open Systems Interconnection - Systems Management - Part 1: Event Report Function,* International Organization for Standardization and International Electrotechnical Committee, August 1991.

[18] ISO/IEC 10164-6 / ITU-T Recommendation X.735: *Information Technology - Open Systems Interconnection - Systems Management - Part 1: Log Control Function,* International Organization for Standardization and International Electrotechnical Committee, June 1991.

[19] ITU-T Draft Recommendation G.774, *Synchronous Digital Hierarchy (SDH) Management Information Model,* November 1991.

[20] CCITT Recommendation Q.821, *Stage 2 and Stage 3 Description for the Q3 Interface - Alarm Surveillance*.

Other Consortia and Organizations

[21] ATM Forum, *CMIP Specification for the M4 Interface (Network Element View)*, Version 1.0, af-nm-0027-001, September 1995.

[22] ATM Forum, *M4 Network View CMIP MIB Specification*, Version 1.0, af-nm-0073-000, January 1997.

[23] Bellcore, GR-NWT-001114, *Issue 1, Generic Requirements for Operations Interfaces Using OSI Tools: Broadband ATM Network Operations, Issue 2,* October 1995.

[24] Bellcore, TR-NWT-001042, *SONET Transport Information Model.* March, 1992.

[25] Bellcore, TR-NWT-000836, *Transport Configuration and Surveillance for Network Elements, Issue 1*, March 1992.

[26] Bellcore SR-NWT-002287, *INA Cycle 1 Management Information Model,* Issue 2, April 1993.

[27] EURESCOM Project EU-P103 *Evolution of the Intelligent Network,* Technical Report TR 7, *Network Resource Model*, December 1994.

[28] ETSI DE/TM-2201 *SDH, Information Model,* September 1992.

[29] ETSI, GOM Network Level Class Library, 1995.

Acronyms

- **ASN.1:** Abstract Syntax Notation One
- **ATM:** Asynchronous Transfer Mode
- **B-ISDN:** Broadband Integrated Services Digital Network
- **CLNW:** Connectivity Layer Network
- **CPE:** Customer Premises Equipment
- **CTP:** Connection Termination Point
- **DN:** Distinguished Name
- **DPE:** Distributed Processing Environment
- **EFD:** Event Forwarding Discriminator
- **EML:** Element Management Layer
- **ETSI:** European Telecommunication Standards Institute
- **FCAPS:** Fault, Configuration, Accounting, Performance, and Security
- **FLND:** Foreign Layer Network Domain
- **FM:** Fault Management
- **GDMO:** Guidelines for the Definition of Managed Objects
- **GRM:** General Relationship Model
- **ITU-T:** International Telecommunication Union - Telecommunication Standardization Sector
- **LC:** Link Connection
- **LCG:** Logical Connection Graph
- **LLND:** Local Layer Network Domain
- **LND:** Layer Network Domain
- **LNW:** Layer Network
- **LTP:** Link Termination Point
- **NCG:** Nodal Connection Graph
- **NE:** Network Element
- **NFEP:** Network Flow End Point
- **NFC:** Network Flow Connection
- **NMF:** Network Management Forum
- **NML:** Network Management Layer
- **NRIM:** Network Resource Information Model
- **NWCTP:** Network Connection Termination Point
- **NWTP:** Network Termination Point

- **NWTTP:** Network Trail Termination Point
- **ODP:** Open Distributed Processing
- **OMT:** Object Modelling Technique
- **PCG:** Physical Connection Graph
- **PCS:** Personal Communication Services
- **PDH:** Plesiochronous Digital Hierarchy
- **POTS:** Plain Old Telephone Service
- **QoS:** Quality of Service
- **RCM:** Resource Configuration Management
- **SDH:** Synchronous Digital Hierarchy
- **SFC:** Stream Flow Connection
- **SFEP:** Stream Flow End Point
- **SNC:** Subnetwork Connection
- **SNW:** Subnetwork
- **TC:** Tandem Connection
- **TFC:** Terminal Flow Connection
- **TINA-C:** Telecommunications Information Networking Architecture Consortium
- **TL:** Topological Link
- **TLTP:** Topological Link Termination Point
- **TMN:** Telecommunications Management Network
- **TP:** Termination Point
- **TTP:** Trail Termination Point
- **VC:** Virtual Channel
- **VP:** Virtual Path

Glossary

Only the terms defined in this document are included in this glossary. See TINA Glossary for a complete list of all glossary terms used in TINA-C specifications.

- **Business Management Layer:** The business management layer is defined in TMN standards. It contains functions that span the total enterprise, and is the layer at which agreements between operators are made. This layer is not a TINA-C concern.
- **Characteristic Information:** A signal of characteristic rate, coding, and format which is transferred within and between subnetworks in a layer network and presented by a “client” network to an adaptation function for transport by a “server” network.
- **Circuit Layer Network:** A layer network that provides end-users with telecommunications services such as ATM cell relay, packet switching, and leased lines. Multiple circuit layer networks can be defined according to the services provided.
- **Client Layer Network:** A layer network which requests a trail from other layer networks. (Cf. “Server Layer Network”).
- **Connectivity Layer Network**: A transport network that is made up of one or more layer networks. The characteristic information accepted by a connectivity layer network can be different from the characteristic information delivered by the connectivity layer network. Within a connectivity layer network, a layer network may be directly connected with one or more other layer networks. Since different layer networks transport different characteristic information, the layer networks interconnection involves adaptation of characteristic information; i.e., information transported by one layer network is adapted and converted into information transported by another layer network.
- **Connectivity Session**: A set of Network Flow Connections that have been grouped together for some purpose, e.g., connections that are part of the same communication session, connections that have related lifetimes, and so on.
- **Edge**: An end point of a subnetwork connection.
- **Element Layer:** See “Network Element Layer”.
- **Element Management Layer:** A sublayer of resource management functions defined in TMN standards that consists of functions that manage individual network elements or subsets of network elements (which may be viewed by network management layer functions as subnetworks).

- **Functional Area:** A task-specific grouping of required network management functions. The OSI defines five management functional areas. The TINA-C architecture defines six TINA functional areas by dividing the OSI Configuration Management functional area into Resource Configuration and Connection Management. The six TINA functional areas are: Accounting Management, Connection Management, Fault Management, Performance Management, Resource Configuration Management, and Security Management.
- **Layer Network**: A transport network made up of components of a specific transmission and/or switching technology that transports information of a specific format, coding and rate. The information type transported by a layer network is called its characteristic information. The termination points at which a layer network accepts or delivers its characteristic information are called trail termination points.
- **Layer Network Domain**: A part of a layer network that is under the control of one connectivity provider.
- **Link:** A network resource that represents a topological relationship and the potential for connectivity between either two subnetworks or a subnetwork and a CPE. It is configured using one or more topological links.
- **Link Connection:** A network resource that transports information across a link between two Network Connection termination Points.
- **Link Termination Point:** An end point of a link.
- **Logical Connection Graph:** An information object that represents a set of stream flow connections that have been grouped for some purpose.
- **Network Connection Termination Point**: An end point of a link connection.
- **Network Element Layer:** The category of functions defined in TMN standards that are linked to the technology or architecture of the network resources that provide the basic telecommunications services. These functions may be accessed by the element management layer functions using standard or open information specifications that may hide vendor-specific functions within network resources.
- **Network Element Level Aspect:** A category of information. The network element level aspect is concerned with the information that is required to manage specific equipment resources that provide network element layer functions.This refers to the information required to manage the physical, telecommunications and support functions within one network element.
- **Network Element Management Layer:** See “Element Management Layer”.
- **Network Flow Connection**: A network resource that transports information across a connectivity layer network between two or more Network Flow Endpoints. The characteristic information associated with the different flow endpoints of a flow connection may be different. A Network Flow Connection is made up of one or more trails.
- **Network Flow Endpoint**: An abstraction that represents in a technology independent manner a termination of a Network Flow Connection. Has a one-to-one association with a trail termination point.

- **Network Flow Endpoint Pool**: A topological component of a Connectivity Layer Network. Represents the potential for terminations of Network Flow Connections in a CPE. A network flow endpoint pool is either a collection of Link Termination Points that are collocated on the same CPE or a portion of a Link Termination Point.
- **Network Management Layer:** A sublayer of network resource management functions defined in TMN standards that have the responsibility for the management of all the network elements, as presented by the element management layer. It is not concerned with how a particular network element provides service internally. Complete visibility of the whole network is typical, and a vendor independent view will need to be maintained. The functions in this layer interact with the service management layer on end-to-end connections, performance, faults, etc. across the network.
- **Nodal Connection Graph:** An information object that represents a set of terminal flow connections in a CPE that form a terminal part for a group of stream flow connections.
- **Physical Connection Graph:** An information object that represents a set of network flow connections that form the network part for a group of stream flow connections.
- **Resource Configuration Management:** One of the six TINA network management functional areas. Functions in this category establish and maintain the configuration of managed objects which have corporate data associated with them; they manipulate point-to-point network connections, and manage the network resources needed for manipulating network connections.
- **Resource Management:** The activities within a network which provide management services to end-user service applications, and which are responsible for management of network element level functions either individually or in aggregation. Includes Network Management Layer and Element Management Layer functions.
- **Service Management Layer:** The category of functions defined in the TMN standards that provide end-user service specific functions including service logic and service management.
- **Stream Flow Connection**: A network resource that transports information in a unidirectional manner between applications in a TINA network. The information is transported from a Source Stream Flow End Point to one more Sink Stream Flow End Points.
- **Stream Flow End Point**: An end point of a Stream Flow Connection.
- **Subnetwork:** A subset of the network resources such that the resources, having common operations properties (e.g., manufacturer, common function, or common geographical location) cooperate to support some aspect or portion of one or more telecommunications services. It may contain resources of different suppliers, and may consist of several nodes that are operated as a cohesive entity. In the context of Connection Management the subnetwork is used as a topological component to effect routing and management. It can be partitioned into subnetworks that are interconnected by links.

- **Subnetwork Connection:** A network resource that transports information across a subnetwork between two or more termination points (edges).
- **Tandem Connection**: A portion of a trail that exists in a layer network domain. The end points of a tandem connection are either Network Trail Termination Points or Network Connection Termination Points.
- **Terminal Flow Connection**: A network resource that transports information within a CPE either from a Stream Flow End Point to a Network Flow End Point, or vice versa.
- **Topological Link:** A network resource that represents either the potential for connectivity or a bundle of connections between either two subnetworks or a subnetwork and a CPE. It is configured using a trail in a server layer network.
- **Topological Link Termination Point:** An end point of a topological link.
- **Trail**: A network resource that transports information across a Layer Network between two or more Trail Termination Points and ensures the integrity of the information transfer.
- **Trail Termination Point**: A termination of a trail where the characteristic information associated with the trail is accepted and/or delivered in a layer network.