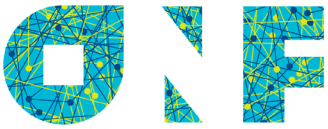


OPEN NETWORKING
FOUNDATION

Core Information Model (CoreModel)

Version 1.1
November 30, 2015

ONF TR-512

ONF Document Type: Technical Recommendation
ONF Document Name: Core Information Model version 1.1

Disclaimer

Open Networking Foundation
2275 E. Bayshore Road, Suite 103, Palo Alto, CA 94303
www.opennetworking.org

List of Figures

Document History

Version	Date	Description of Change
1.0	March 30, 2015	Initial version of the base document of the "Core Information Model" fragment of the ONF Common Information Model (ONF-CIM).
1.1	November 24, 2015	Version 1.1

1 Introduction

An information model describes the things in a domain in terms of objects, their properties (represented as attributes), and their relationships. This ONF Technical Recommendation (TR) focuses on the CoreModel of the ONF Common Information Model (ONF-CIM). The ONF-CIM provides a representation of data plane resources[1] for the purpose of management-control. In accordance with the SDN architecture **[ONF SDN Arch],** this management-control is expected to be achieved by an SDN controller. The controller expresses a view of the network, in terms of ONF-CIM artefacts, to client SDN controllers/applications to meet the needs of that client.

The ONF CIM is divided into a number of pieces and is centered on a core fragment that is independent of specific data plane technology. The model includes pieces that provide data plane technology (forwarding technology) specific structures and properties (such as OTN, Ethernet, and MPLS-TP). These can be used to augment the CoreModel to provide a data plane technology specific representation. The model also includes application specific information models (e.g. applicable to representation of networking in a storage context). Most importantly, the ONF-CIM is modeled independently of the protocols used in the control interfaces.

There are a number of other documents associated with this document:

[1] It is focused on representation of the functions/resources that have the primary purpose of supporting information forwarding (transfer and transform functions), that form a network that realizes virtual adjacency, for the purpose of control of those functions/resources. Those resources are referred to as data plane resources. The information model is not intended to cover resources that have a primary purpose of supporting storage or compute solutions.

- The **[ONF CIM Guidelines]** specifies the principles and guidelines for the development and use of the ONF-CIM, including guidelines for deriving purpose-specific information model views (through pruning and re-factoring selected subsets of artifacts from the ONF-CIM), and mapping to data schemas for protocol-specific control interfaces.
- The ONF-CIM is expressed in a formal language called UML (Unified Modeling Language). UML has a number of basic model elements, called UML artifacts. In order to assure consistent modeling, only a subset of the UML artifacts is used in the development of the ONF-CIM. The selected subset of UML artifacts is documented in **[ONF UML Guidelines]**.
- The **[ONF Papyrus Guidelines]** specifies the guidelines for using the Papyrus tool used in the development of the ONF-CIM. This guidelines document also describes how the Common IM modeling teams can cooperate in the GitHub environment for separate and coordinated development of the ONF-CIM fragments.

In the development of the Core IM, information model work from other SDOs has been used as input, including [TR215], [TMF TR225], [TMF SID 5LR], [ITU-T G.7711], [ITU-T G.874.1], [ITU-T G.8052], and [ITU-T G.8152]. The Core IM is being shared with other bodies via various mechanisms including publication of a view of the model as an IETF draft [draft-lam].

1.1 Summary of main changes between version 1.0 and 1.1

Changes to the model and/or related documentation:

- Class name improvements (no change in semantics):
  - The class called LinkEnd in 1.0 is now LinkPort
  - The class called Endpoint in 1.0 is now FcPort
- Addition of directionality model
- Enhancements to the state model
- Adding the stereotype «LikelyToChange» to the NE class to indicate that work is underway to refine the model in this area
- Addition of a data dictionary section to this document to provide easily accessible model detail
  - Note that the data dictionary is generated by tooling from the model and is fully aligned with the model.
- Various improvements to text and figures

2 References

[ONF SDN Arch]	ONF SDN Architecture 1.0, June 2014 (https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf)
[ONF CIM Overview]	TR-513 (V1.1): ONF Common Information Model Overview (https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/Common_Information_Model_V1.0.pdf)

[ONF UML Guidelines] TR514 (V1.1): ONF UML Model Guidelines (https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/UML_Modeling_Guidelines_V1.0.pdf)

[ONF Papyrus Guidelines] TR515 (V1.1): ONF Papyrus Guidelines (https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/Papyrus_Guidelines_V1.0.pdf)

[TMF TR225] TM Forum TR225 (R15.0.0), Logical Resource: Network Function Model (liaised to ONF)

[TMF TR215] TMF TR215 (V0.5.3) Logical Resource Network Model Advancements and Insights (liaised to ONF)

[TMF SID 5LR] TM Forum GB922 (R15.0.0) Information Framework (SID) Addendum 5LR (liaised to ONF)

[ITU-T G.7711] Recommendation ITU-T G.7711/Y.1702 (ex. G.gim) (08/2015), *Generic Protocol-Neutral Information Model for Transport Resources*

[ITU-T G.874] Recommendation ITU-T G.874 (08/2013), *Management aspects of optical transport network elements*

[ITU-T G.874.1] Recommendation ITU-T G.874.1 (10/2012), *Optical transport network: Protocol-neutral management information model for the network element view,* plus Amendment 1 (08/2013)

[ITU-T G.8052] Recommendation ITU-T G.8052/Y.1346 (08/2013), *Protocol-neutral management information model for the Ethernet Transport capable network element*

[ITU-T G.8152] Recommendation ITU-T G.8152/Y.1375 (draft), *Protocol-neutral management information model for the MPLS-TP network element*

[ITU-T G.852.2] Recommendation ITU-T G.852.1 (11/1996), *Enterprise viewpoint description of transport network resource model*

[TMF 612] TM Forum MTOSI (4.0), Multi-Technology OS Interface

[ISO/IEC 19505] ISO/IEC 19505:2012 Information technology -- Object Management Group Unified Modeling Language (OMG UML)

[draft-lam] IETF draft-lam-teas-usage-info-model-net-topology-02, Usage of IM for network topology to support TE Topology YANG Module Development

[ITU-T G.805] Recommendation ITU-T G.805 (03/2000), *Generic functional architecture of transport networks*

[ITU-T G.800] Recommendation ITU-T G.800 (02/2012), *Unified functional architecture of transport networks*

[ITU-T M.3100] Recommendation ITU-T M.3100 (04/2015) , *Generic network information model*

[ITU-T G.808.1] Recommendation ITU-T G.808.1 (05/2014), *Generic protection switching – Linear trail and subnetwork protection*

[ITU-T G.8081]	Recommendation ITU-T G.8081 (02/2012), *Terms and definitions for automatically switched optical networks*
[ITU-T G.8001]	Recommendation ITU-T G.8001/Y.1354 (09/13), *Terms and definitions for Ethernet frames over transport*
[ITU-T X.731]	Recommendation ITU-T X.731 (01/1992), *Information technology - Open Systems Interconnection - Systems management: State management function*

3 Definitions

3.1 Terms defined elsewhere

This document uses terms defined elsewhere. These terms are highlighted in section 4 Abbreviations and acronyms below by referring to the definition source document.

3.2 Terms defined in this TR

The primary purpose of this document is to define terms and hence terms are defined throughout the document. Key terms are highlighted in section 4 Abbreviations and acronyms below by referring to the section in this document where the term is defined.

4 Abbreviations and acronyms

This TR uses the following abbreviations and acronyms (Note that some cross references are included here rather than in the References section where the cross reference is only relevant for abbreviation/acronym interpretation purposes):

AP	Access Point [ITU-T G.805]
CNM	Customer Network Management
CP	Connection Point [ITU-T G.805]
CTP	Connection Termination Point. Note that definitions differ between TM Forum [TMF 612] and [ITU-T M.3100]. Both usages apply here when referring to legacy cases and the abbreviation is qualified in all cases of use.
ECC	Embedded Communications Channel [ITU-T G.874]
EMS	Element Management System [definition reference ITU-T M.3400 - TMN][2]
ETH	Ethernet MAC Layer [definition reference ITU-T G.8001]
ETY	Ethernet Physical Layer [definition ITU-T G.8001]

[2] This term is not intended for use other than in reference to legacy systems.

FC	ForwardingConstruct (defined in the ONF-CIM - see 6.1 Overview of the CoreNetworkModule of the CoreModel below). • Note that at this point the definition is subtly different to that in [TMF TR225]. The aim is to align the terms usage
FDFr	FlowDomainFragment [TMF 612]
FRE	ForwardingRelationshipEncapsulation [TMF TR215]
FTP	FloatingTerminationPoint [TMF 612]
GitHub	See www.github.com
GUID	Globally Unique IDentifier (see www.wikipedia .org/Globally_unique_identifier)
IM	Information Model (see 1 Introduction above)
IMP	Inverse MultiPlexing [ITU-T G.805]
ISO	International Organization for Standardization (see www.iso.org)
ITU-T	International Telecommunications Union (see www.itu.int)
LP	LayerProtocol (defined in the ONF-CIM - see 6.1 Overview of the CoreNetworkModule of the CoreModel below). Note that there are two related terms: • layer-protocol: used to refer to the information transfer protocol (or Characteristic Information of the signal) • layerProtcolName: used to refer to the attribute in the LP that carries the value that identifies the characteristic layer-protocol of the LP • LayerProtocolName: used to refer to the data type that holds the formal name of the layer-protocol
LTP	LogicalTerminationPoint (defined in the ONF-CIM - see 6.1 Overview of the CoreNetworkModule of the CoreModel below)
MAC	Media Access Control
MEP	Maintenance End Point
MFDFr	MatrixFlowDomainFragment
MLSN	MultiLayerSubNetwork [TMF 612]
MP2MP	Multi-Point to Multi-Point
MPLS-TP	Multi-Protocol Label Switching Transport Profile [definition reference RFC6378]
NCD	NetworkControlDomain
NE	NetworkElement
OAM	Operations Administration and Maintenance
OCh	Optical Channel
ODU	Optical Data Unit
OMS	Optical Multiplex Section
ONF-CIM	ONF Common Information Model
OPS	Optical Protection Switch

OS	Operations System (essentially OSS - Operation Support System)
OTN	Optical Transport Network
OTS	Optical Transmission Section
OTU	Optical channel Transport Unit
P2MP	Point to Multi-Point
P2P	Point to Point
PON	Passive Optical Network
PTP	Physical Termination Point [TMF 612]
RMP	Rooted Multi-Point
SDN	Software Defined Networking [ONF]
SDO	Standards Development Organization
SNC	SubNetworkConnection [TMF 612]
SNP	SubNetworkPoint [ITU-T G.8081]
TBD	To Be Defined
TCP	Termination Connection Point [ITU-T G.805]
TDM	Time Division Multiplex
TMF	TeleManagement Forum (see www.tmforum.org)
TP	Termination Point [ITU-T M.3100]
TPE	TerminationPointEncapsulation [TMF TR215]
TR	Technical Recommendation [ONF] Technical Report [TM Forum]
TRI	Transport Resource Identifier [ITU-T G.8081]
TTP	Trail Termination Point [ITU-T M.3100]
UML	Unified Modelling Language (see www.omg.org)
UUID	Universally Unique IDentifier (see https://en.wikipedia.org/wiki/Universally_unique_identifier)
VCAT	Virtual Concatenation
VNE	Virtual Network Element
XC	CrossConnection

5 Conventions

5.1 UML modeling conventions

The information model defined in this TR is expressed in Unified Modeling Language (UML). UML defines a number of basic model elements, called UML artifacts. In order to assure consistent modeling, only a subset of these artifacts is used in the development of the ONF-CIM. The selected subset of UML artifacts is documented in [ONF UML Guidelines].

5.2 Lifecycle Stereotypes

Lifecycle stereotypes are applied to entities in the model to indicate their degree of maturity[3]. These are made visible in many of the figures in this document.

The following stereotypes appear in this document:

- «Experimental»: Indicates that the entity is at a very early stage of development and will almost certainly change. The entity is NOT mature enough to be used in implementation[4].
- «Preliminary»: Indicates that the entity is at a relatively early stage of development and is likely to change but is mature enough to be used in implementation.

If no stereotype is shown the entity is mature. There are other lifecycle stereotypes that are not relevant in this document.

5.3 Diagram Keys

This document includes a number of UML diagrams. The UML symbol set is suitably explained in [ONF UML Guidelines]. Many of the UML diagrams in this document have small font (due to density of information conveyed). It will be necessary for the reader to zoom in and pan across the figure to see the detail[5].

This document also contains a number of non-UML diagrams, which use the symbols highlighted below in pictorial representations of network examples.

[3] The whole model including all degrees of work in progress has been published to allow the user maximum opportunity to set a most consistent direction with the work at hand. It is considered important to expose work in progress especially where this may have an impact on a choice of implementation. There may be some experimental structure that contains some very stable parts, without that structure those parts might be quite uninterpretable. A user who decides to take a low risk approach can ignore preliminary and experimental parts. A user who is more inclined to take a risk or who is looking for inspiration for their work can take the experimental and preliminary parts, understanding the risk involved.

[4] The implementer can clearly choose to use the item at risk (expecting change and accounting for this in deployments etc)

[5] The aim is to improve the figure readability in future releases.

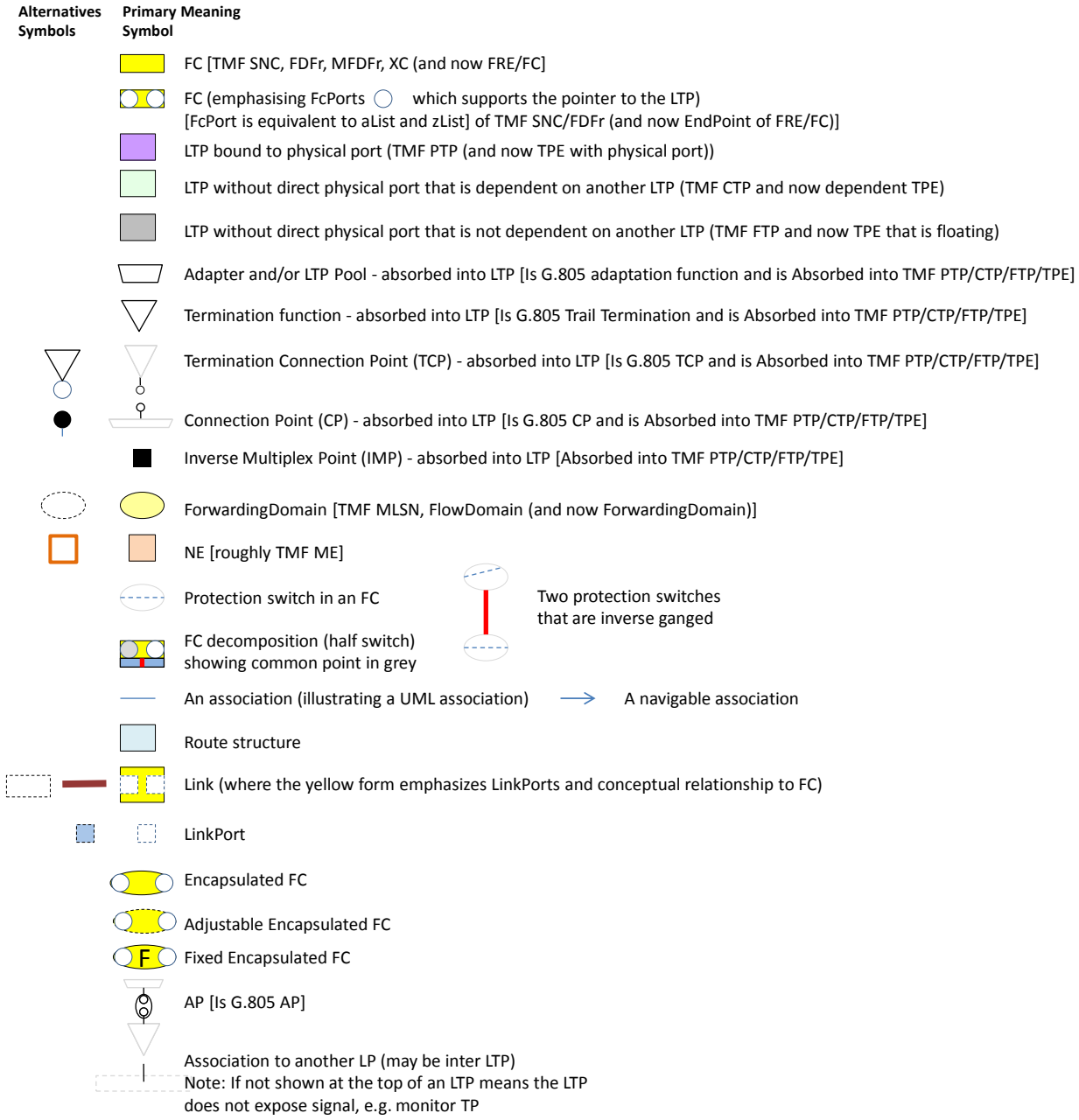


Figure 5-1 Network Diagram Symbol Key[6]

The relationship between some of the entities in the ONF-CIM and other familiar models are shown in the next figure. The figure also provides a key to some additional symbols. Further mappings are provided in section 11 Addendum Terminology Translation table on page 161

[6] It should be noted that in this version and future versions the terms ForwardingDomain (FD) and ForwardingConstruct (FC) are used in place of SubNetworkConnection (SNC) and SubNetwork (SN) (used in the earlier versions of the ONF-CIM).

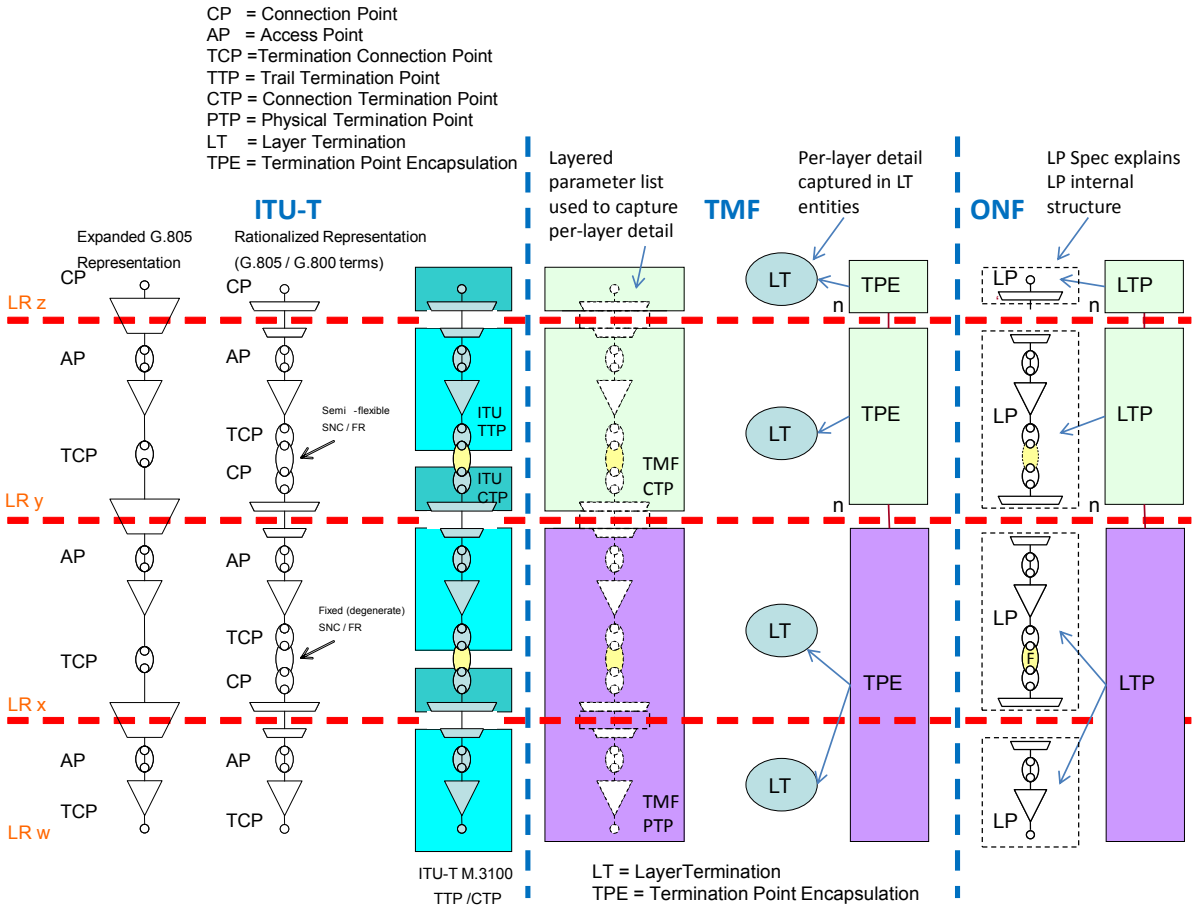


Figure 5-2 Mapping from ITU-T and TM Forum Termination models to the ONF Core[7]

6 Overview of the CoreModel Fragment

The focus of this document is the CoreModel Fragment of the ONF-CIM. The CoreModel Fragment is divided into several UML packages.[8] The following UML packages are discussed in detail in this document:

- CoreNetworkModule: Covers the essentials for modeling of the Network providing an overview of the key classes.
- CoreFoundationModule: Covers aspects common to many classes such as identifiers and naming.

Some of the contents of the CoreModelEnhancements UML package are also discussed.

[7] It should be noted that in this version and future versions the terms ForwardingDomain (FD) and ForwardingConstruct (FC) are used in place of SubNetworkConnection (SNC) and SubNetwork (SN) (respectively used in the earlier versions of the ONF Information Model).

[8] Some UML packages are used to hold model fragments, some to hold modules and other subsets of a module or fragment. The UML packages are labelled accordingly.

The CoreNetworkModule encompasses Topology, Termination and Forwarding aspects. These aspects are described in this document as follows:

- Termination Subset of the CoreNetworkModule: Covers the modeling of the processing of transport characteristic information, such as termination, adaptation, OAM, etc.
    - Note that technology specific details are covered in the ForwardingTechnologyModelFragments package of the ONF-CIM (this aspect is not in the scope of this document)
- Forwarding Subset of the CoreNetworkModule: Covers the details of forwarding entities, including:
    - The Basic Forwarding
    - The Forwarding Construct Specification
- Topology Subset of the CoreNetworkModule: Covers the modeling of network topology information in detail[9] and describes the attributes relevant when working with network topology.

Figures showing fragments of the model using standard UML symbols and also figures illustrating application of the model using the symbols in Figure 5-1 Network Diagram Symbol Key are provided throughout this document. Many of the application view figures also provide corresponding UML class diagrams of fragments. All UML diagrams depict a subset of the relationships among the object classes, such as inheritance (i.e. specialization), association relationships (such as aggregation and composition), and conditional features or capabilities. Some UML diagrams also show further details of the individual classes, such as their attributes and the data types used by the attributes.

In some of the figures the LP is depicted with a view of the internal details. The following figure shows the cases illustrated in figures. In a realization the LP detail structure would be expressed by some form of specification. The intention is to expand the model to include support for machine interpretable specification (see section 10.2 LTP Specification on page 157).

[9] The information described in this subset can be used for example for path computation and to provide views of network capacity/capability with information maintained in a topology database.

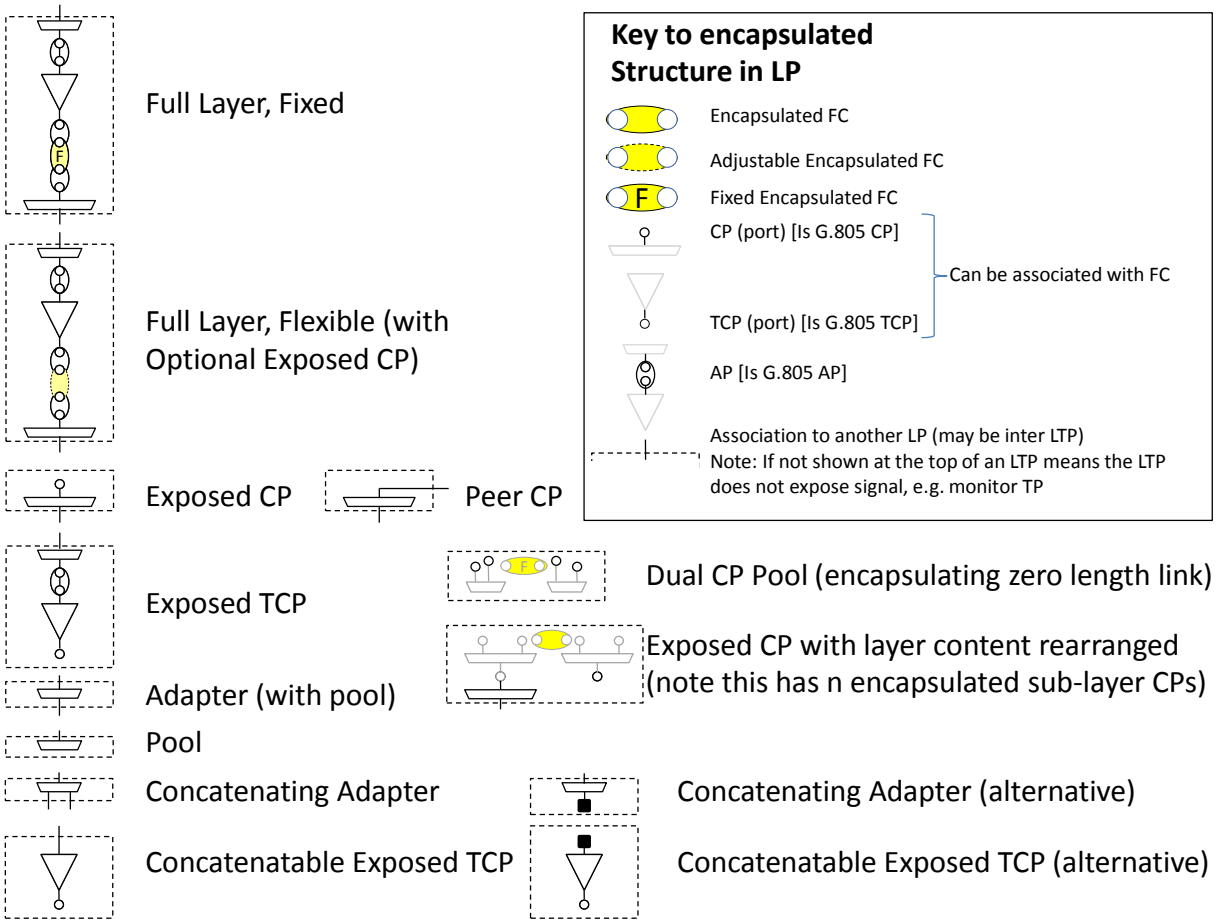


Figure 6-1 LP Cases

A data dictionary that sets out the details of all classes, data types and attributes is also provided. (see 9 Data Dictionary on page 67).

6.1 Overview of the CoreNetworkModule of the CoreModel

This section provides a high-level overview of the generic information model. Figure 6-1 below is a skeleton class diagram illustrating the interrelationships between key object classes defined in the CoreNetworkModule of the CoreModel. The classes are colored to help recognize key groupings in the model. The colors are chosen to match the key entity colors in Figure 5-1 Network Diagram Symbol Key (with the Link in the alternative color for clarity). This color scheme for class diagrams is used in some of the later figures.

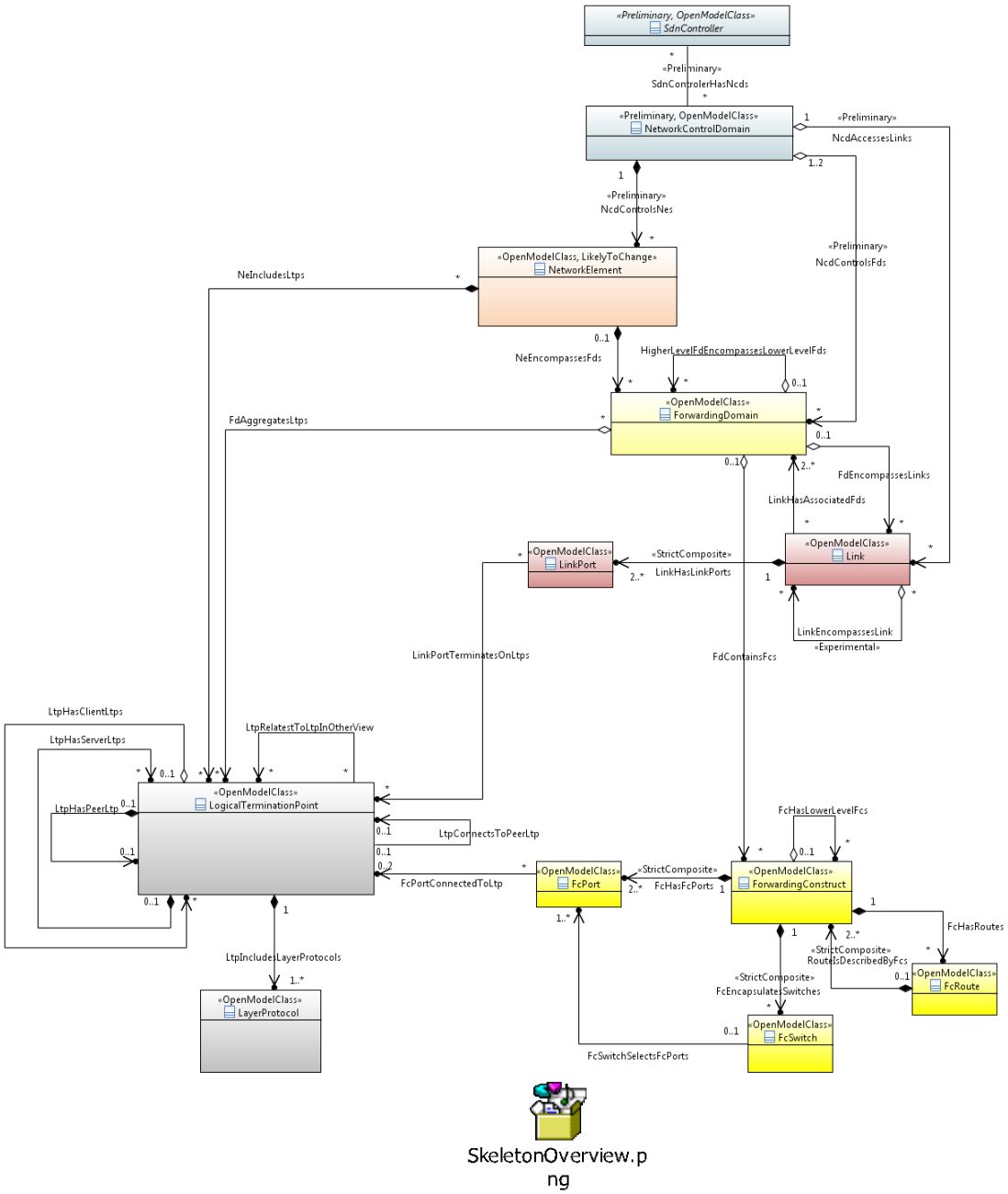


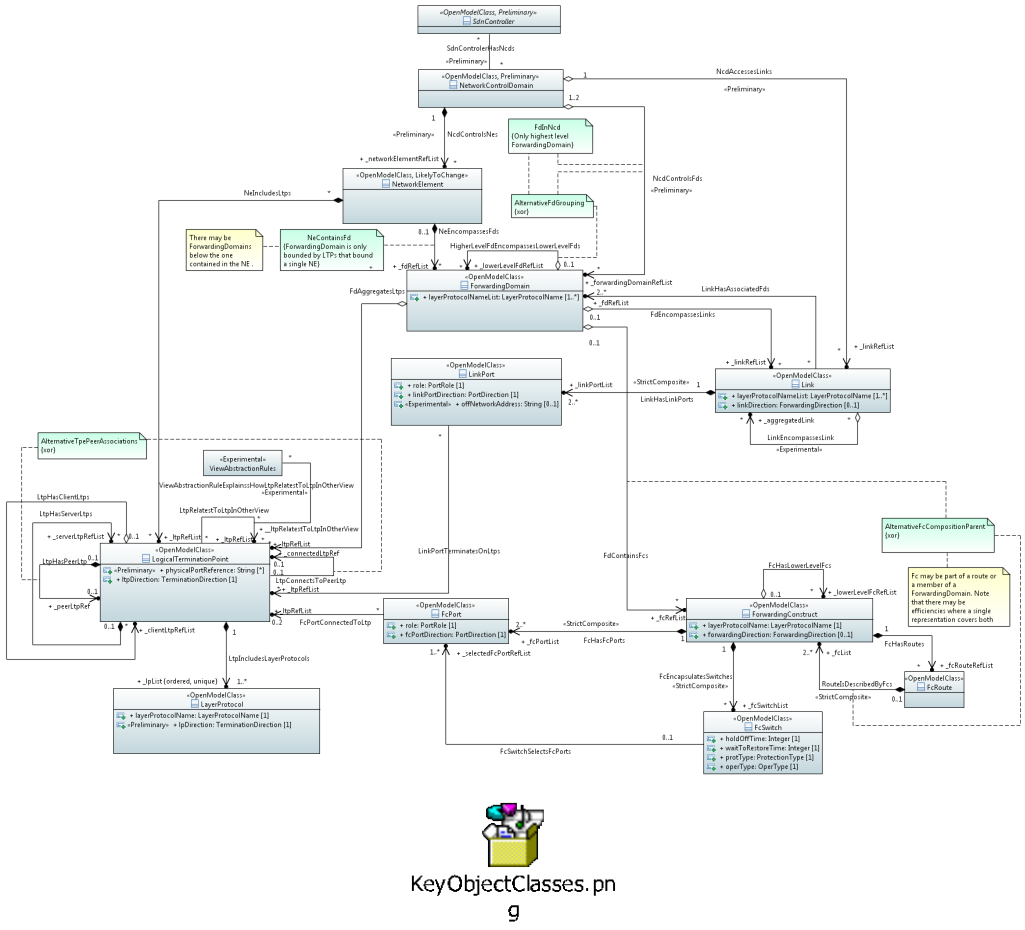
Figure 6-2 Skeleton Class Diagram of key object classes

CoreModel diagram: HighLevelColouredSkeletonOverview

Note: A more convenient way to view the details of the diagram is by using the companion PNG file, which can be independently zoomed in and out without impacting the viewing of the main document. Available in the document attachments when downloaded.

When applying the information model to a specific-purpose interface, only a subset of the overall information model may be needed. Depending on the scope of the interface, pruning of the information model may be necessary, such as excluding a whole object class or part of an object class. For example, in the interface from an SDN controller directly to a Network Element in the infrastructure layer, the NetworkControlDomain object class is beyond the scope of the Network Element[10] interface and thus not needed. In addition, re-factoring of the selected model artifacts may be necessary to meet the specific-purpose needs. However, re-factoring of the model artifacts should not add semantics beyond those defined in the information model.

Figure 6-2 below is a more comprehensive class diagram than the previous one showing some attributes and constraints of the associations in the model.



KeyObjectClasses.png

Figure 6-3 Class Diagram of all key object classes showing attributes and constraints

CoreModel diagram: HighLevelDetail

[10] The Network Element scope of the direct interface from a SDN controller to a Network Element in the infrastructure layer is similar to the EMS-to-NE management interface defined in the information models [ITU-T G.874.1] (OTN), [ITU-T G.8052] (Ethernet), and draft [ITU-T G.8152] (MPLS-TP). On the other hand, the network scope of the interface between two SDN controllers is similar to the OS-to-OS management interface defined in the TMF 612 information model.

The above diagram shows owned attributes of the key object classes in the model. Not all classes are shown and the classes in the diagram have additional attributes related to associations to those classes as well as some inherited attributes and some experimental attributes. The diagram should be reviewed in conjunction with the material in section 9.1 Core Network Module data dictionary on page 67 which provides a detailed description of the classes, attributes, data types etc. There are also subsections in the data dictionary on other Core Model fragments. Each subsection lists classes etc in alphabetical order. All class diagrams in this document should be read in conjunction with the data dictionary. The table of contents provides links to each class etc. for ease of navigation. The following sections provide some additional clarifications to assist in reading the diagram and data dictionary content.

6.1.1 LogicalTerminationPoint (LTP) and LayerProtocol (LP)

- Rules for forming and interrelating LTP instances are provided in section 6.3 Termination on page 28.
- Transport layer-protocol[11] specific properties (such as technology specific termination and adaptation properties) are modeled as attributes of conditional packages (called "_Pacs" in the UML notation of the ONF-CIM) associated with the LP object class.[12,13]
    - Where a technology specific termination has a complex structuring of internal parts, these parts will be modeled as local classes with instances identified in the context of the LTP global class (see section 6.2.1 Naming and identifiers on page 23).

6.1.2 ForwardingDomain (FD)

- See "subnetwork" topological component in G.852.2 and TMF 612
- Can support more than one layer-protocol
- See Figure 6-14 in clause 6.5.1 for further details on partitioning of FDs
    - An FD also aggregates Links: see Figure 6-14 in clause 6.5.1.
    - A Network Element can encompass multiple switch matrixes (represented by FDs): see Figure 6-15 in clause 6.5.1.

6.1.3 ForwardingConstruct (FC), FcPort, FcRoute and FcSwitch

- FC partitioning reflects FD partitioning. The Cross-Connection in an NE is not necessarily the lowest level of FC partitioning.
- Note that the LinkConnections [ITU-T G.800] associated with the Links that are exposed as part of the internal structure of the FD are not modeled at this point.
- The route is an alternative view of the internal structure of the FC. There are cases where a route is the most appropriate representation and cases where the aggregation is the most appropriate form.

[11] The specific transport technology Characteristic Information (see defiinition of LP in section 3.2 Terms defined in this TR on page 8)
[12] It is likely that the technology specific parameter will be associated indirectly through the specification classes
[13] Note that some implementation languages may allow the addition and removal of _Pacs and attributes from instances of a running system and others may be restricted such that _Pacs/attributes cannot be added/removed once an instance has been created. The model supports both modes of operation.

- FcSwitch usage is explained further in 6.4.2 Forwarding Construct Specification and other details of Forwarding on page 33
- See 6.4.2 Forwarding Construct Specification and other details of Forwarding on page 33
- The FcSwitch approach supports all forms of protection described in [ITU-T 808.1].

6.1.4 Link and LinkPort

- At this point the model supports point to point links fully.
    - The model allows multi-point but anything above 2 (i.e., 3..*) is preliminary
- The model supports an experimental attribute, offNetworkAddress[14], in the LinkPort to cover cases where the FD that the Link ends on is outside the visibility (and hence off network).
    - A Link with an Off-network end cannot be encompassed by an FD.
- A Link may offer parameters such as capacity and delay (see section 6.5.5 Off-network reference and the clients view on age 55).
    - These parameters depend on the type of technology that supports the link.
- The FdEncompassesLinks association can be inferred from the higherLevelFdEncompassesLowerLevelFd association together with the linkHasAssociatedFds association.
    - Note that Link decomposition can also be represented using the LinkEncompassesLink association (similar to the FdEncompassesFd usage for the ForwardingDomain (this association is experimental).

6.1.5 NetworkElement, NetworkControlDomain and SdnController

- These three classes offer a rudimentary controller model that does require some advancement.
- The Network Element concept is well known in the industry and it is normal practice to represent it as in this model. However there would appear to be a number of potential issues with this traditional representation. These potential issues will be explored in a future release and there may be changes made to this entity. Because of the familiarity it has NOT been marked preliminary.

6.2 CoreFoundationModule

This module includes all aspect of the core that are relevant to all other parts of the ONF CIM. The class, data types etc. are defined in detail in section 9.2 Core Foundation Module data dictionary on page 121

6.2.1 Naming and identifiers

To communicate about a thing it is necessary to have some way of referring to that thing, i.e., to have some reference. Terms such as name and identifier are often used when describing the reference. Unfortunately these terms in general usage have ambiguity in their definition that

[14] This is a referece shared between the parties either side of the network boundary. The assumption is that the provider knows the mapping between network port and offNetworkAddress and the client knows the mapping between the client port and the offNetworkAddress and that the offNetworkAddress references some common point or pool of points. It may represent some physical location where the hand-off takes place.

leads to erroneous system behavior. To ensure that the controller system behavior is not erroneous, the model will adopt the following principal definitions:

- Entity: Has identity, defined boundary, properties, functionality and lifecycle in a global context.
    - Examples: A circuit pack, an LTP[15]
- Entity Feature: An inseparable, externally distinguishable part of an entity.
    - Examples: A pin on an integrated circuit, the ports of a FC[16], a face of a cube, the handle of a cup.
    - Note that this is important from a modeling perspective as the representation appears similar to that of an Entity
    - Represented using a UML class
- Role: A specific structure of responsibilities, knowledge, skills, and attitudes in the context of some activity or greater structure. The role has an identity and identifier.
- Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself globally unique, and immutable. An identifier carries no semantics with respect to the purpose or state of the entity.
- Universally Unique Identifier[17] (UUID): An identifier that is universally unique.
- Local ID: An identifier that is unique in the context of some scope that is less than the global scope.
- Name: A property of an entity with a value that is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.
- Label: A property of an entity with a value that is not expected to be unique and is allowed to change. A label carries no semantics with respect to the purpose of the entity and has no effect on the entity behavior or state.
    - A label can be used to carry a freeform text string for any operator purpose. The contents of a label in one view may happen to be the value of a name or identifier in another view. From the perspective of the view with the label there is no expectation other than the value is a string.
- Address: A structure of named values[18] in some address space that defines a location (a volume in that address space) where the structure is a nested hierarchy.
    - A named value may be a name or identifier, the name of the value may be a name or identifier
- Route: the way (via specified intermediate locations and paths) to get to one location from another.
- Property: A quality associated with a thing, structure or location.
- Semantics: Meaning.
- Reference: Data in a communication between two applications that allows a shared understanding of the individual things.

[15] An Entity is represented using a UML class
[16] A Feature of an Entity is represented using a UML class
[17] The term GUID was used in the previous version of the model. The change is in recognition of the more generally applicability of UUID
[18] A named value is simply a tuple with two terms, one being a value and the other being the name of that value. For example in a street address a value may be “London” and the name of that value would be “City”.

- This could be an identifier (including a UUID), a name, an address, or a route, depending upon the needs

Note:

- An entity may be known to be at a place in some functional or physical structure.
- A role may be known to be at a place in some process or behavioral structure.

Figure 6-3 below illustrates the naming/identifier-related attributes defined in the ONF-CIM. They are Universally Unique ID (UUID), Local ID, Name and Label.

The model includes two abstract classes that provide names and identifiers, the GlobalClass and the LocalClass.[19] A GlobalClass represents a type of thing that has instances which can exist in their own right (independently of any others). A LocalClass represents a type of thing that is inseparable from a GlobalClass, but that is a distinct feature of that GlobalClass such that the instances of LocalClass are able to have associations with other instances. The mandatory LocalId of the LocalClass instance is unique in the context of the GlobalClass instance, from which it is inseparable.

The model also includes Extension which is not related to naming/identification. Extension provides an opportunity to define properties not declared in the class that extend the class enabling a realization with simple ad-hoc extension of standard classes to be conformant.

Note that a UUID is applicable only to global type object classes (i.e., subclass of GlobalClass) that their instances can exist on their own right, i.e., NCD, NetworkElement, LTP, FD, Link, FC, and SdnController. The other naming/identifier-related attributes are applicable to both global type object classes and local type object classes (i.e., subclass of LocalClass).[20]

[19] The model also provides ConditionalPackage to supply names and identifiers to _Pac classes but this is currently experimental.

[20] The intention is that only classes from the Core Model are shown in the figure. The classes shown are essentially illustrative. There is another figure in the model that captures Core Model inheritance in detail. All classes from all fragments should inherit from GlobalClass, LocalClass or ConditionalPackage. There is no issue with model dependency as the inheritance association is maintained with the class that is inheriting properties. Although not mandatory, it would seem advisable to maintain a figure per fragment that shows all classes from that fragment and their inheritance.

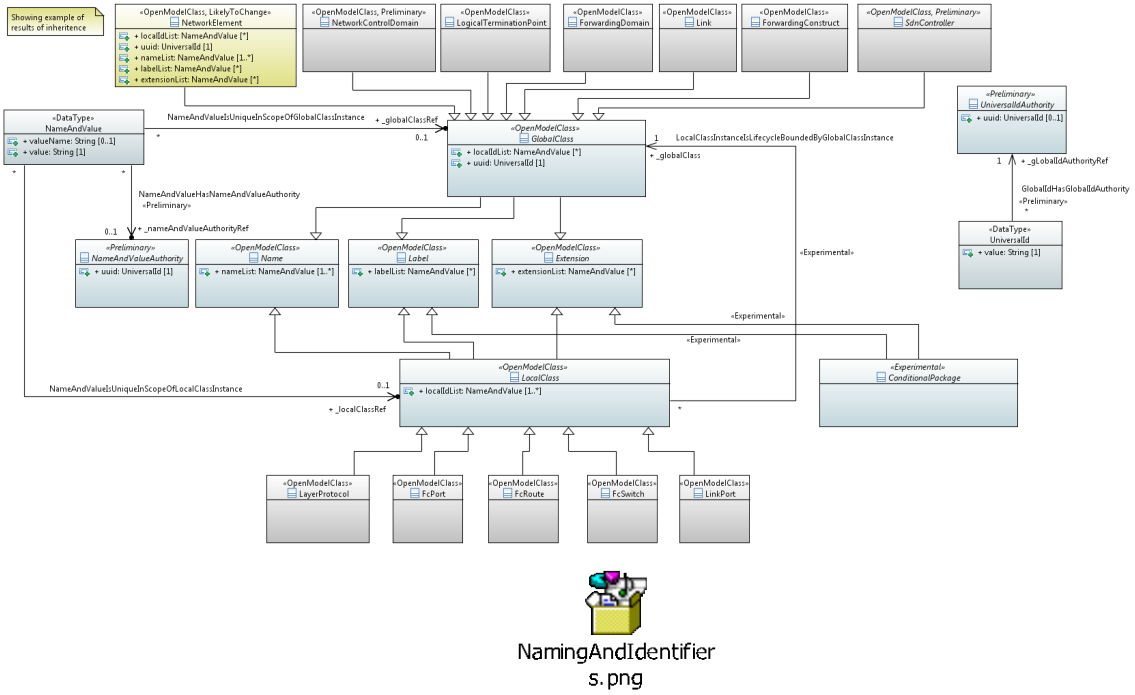


Figure 6-4 Class Diagram for Naming and Identifier of Objects

CoreModel diagram: CommonPackages-NoNotes

6.2.2 States

The Core Foundation module also defines a State_Pac artifact, which provides state attributes. The work on states is preliminary at this stage (it is derived from [ITU-T X.731]). The State_Pac is inherited by GlobalClass and LocalClass object classes. The use of these states provides a consistent way represent the overall operability, usability and current usage of the resource.

It should be noted that the states are «Preliminary»/«Experimental».

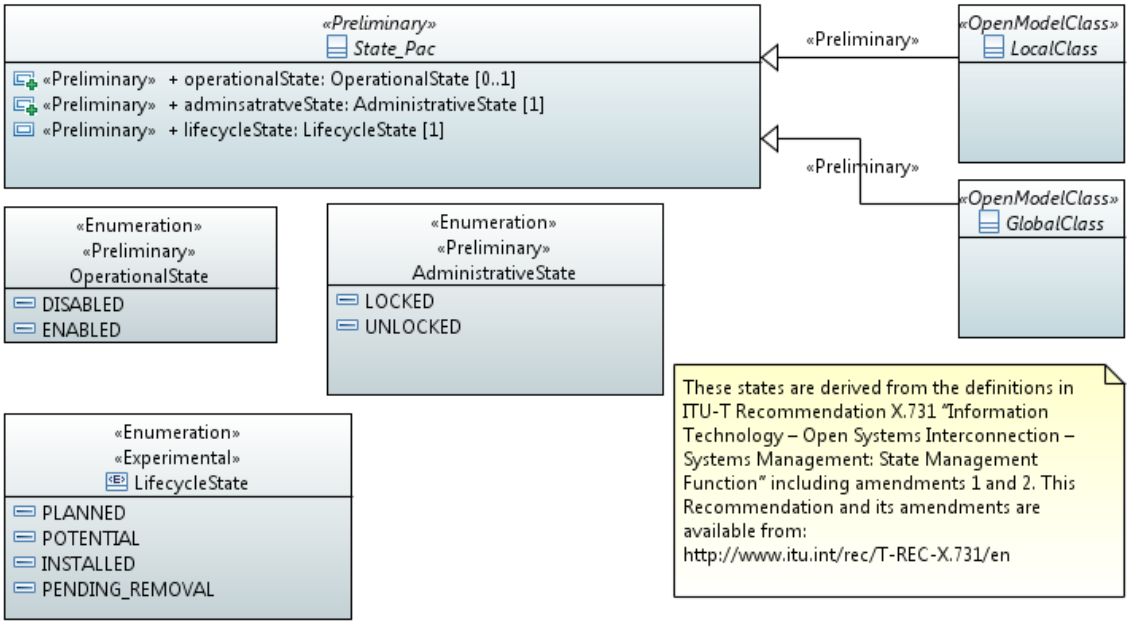


Figure 6-5 States for all Objects

CoreModel diagram: GeneralizedStates

6.2.2.1 Relationship between states in Provider context

If the lifecycleState is PLANNED then the operationalState must be DISABLED and the administrativeState should be LOCKED.

In all other circumstances the states are independent.

6.2.2.2 Relationship between states in the client and provider context

The table below lists the states in the provider context that influence the states in the client context.

Provider State	Value	Client State	Value
operationalState	DISABLED	operationalState	DISABLED
administrativeState	LOCKED	operationalState	DISABLED
lifecycleState	PLANNED	operationalState	DISABLED

Table 1: Influence of Provider state on Client state

No other states in the client context have a dependency on the state in the provider context.

None of the states in the client contextinfluence the states in the provider context.

The administrativeState in the provider context is not visible in the client context. The client context may maintain an independent administrativeState.

The provider controls the lifecycleState that is visible to the client context.

6.3 Termination Fragment

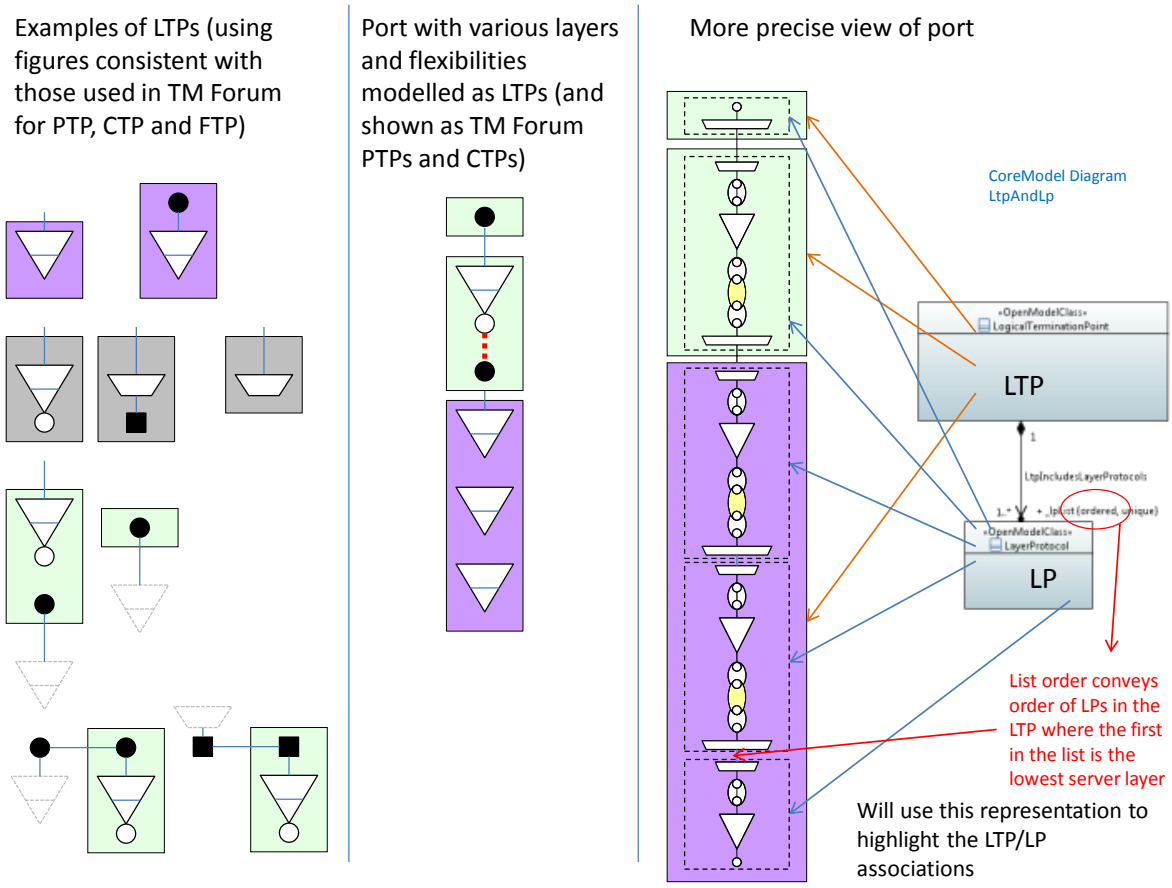


Figure 6-6 Representations of LTPs

In the figure above, the pictorial form shows a number of representations of LTPs (purple, grey and green) representing the layering associated with physical ports (purple), their connectable clients (green) and floating LTPs (grey). The right most pictorial form shows the relationship between the LTP and the LP in terms of a detailed symbol derived from work by TM Forum and ITU-T.[21] An LP instance represents all aspects of termination of a single layer-protocol. An LTP is composed of 1 or more LPs, where the LPs represent the stack of terminations relevant to the LTP as depicted in the pictorial view. A termination stack may spread across several LTPs. The

[21] The work has been liaised by TM Forum and related to Recommendation ITU-T G.805.

reason for this split includes multiplicity, connection flexibility and flow orientation transitions (see also 5.3 Diagram Key).

In the model:

- The flow of signal through the aspects of the LP shown in the figure is not currently formally represented,
    - The LTP specification work (see section 10.2 LTP Specification on page 157) which is currently experimental provides the basis for formal representation in a following release.
- The flow between LPs within an LTP is represented via list order (see the note on the figure above)
- The flow between LPs in different LTPs in a hierarchy is represented by the specific LTP relationship (see Figure 6-8 LTP relationships illustrated in a simple Network Element context on page 31) and the corresponding LP list order in the LTP
    - In the figure above the Sink[22] signal flowing from the top of the upper LP of the purple LTP (i.e. the last entry in the LP list of that LTP) passes to the bottom of the LP in the associated green LTP

There are a number of different cases of LTP which are depicted in figure below.

[22] See section 6.6 Directionality on page 56.

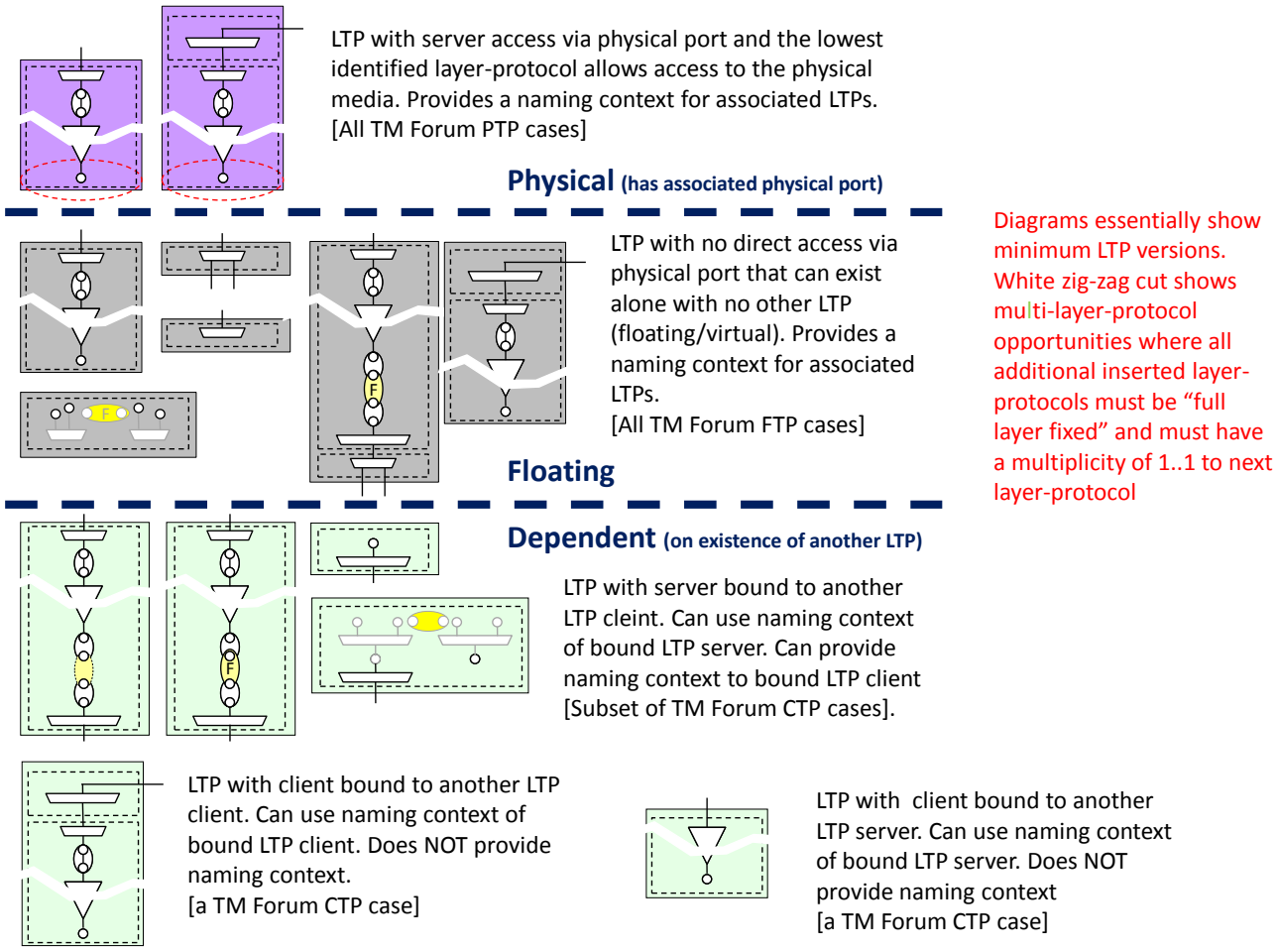


Figure 6-7 LTP Cases

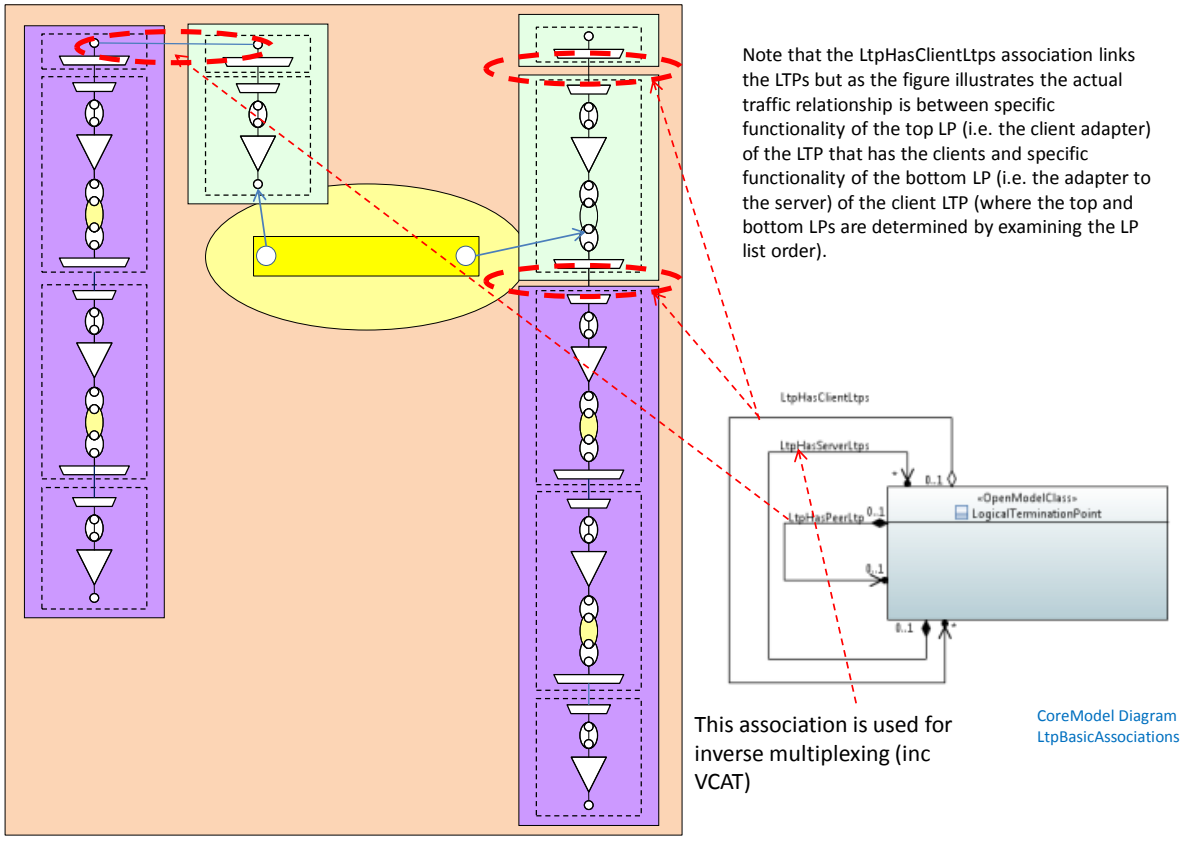


Figure 6-8 LTP relationships illustrated in a simple Network Element context

In Figure 6-6 above, the pictorial form shows a number of LTPs (purple and green) representing the layering associated with physical ports (purple) and their connectable clients (green) as described in the previous section. This figure shows in more detail the partitioning of the layer stack between LTPs. Several different relationships are available for use at the split. The choice depends upon the orientation of traffic flow.

Consider the left most LTP pair in the pictorial form and a signal entering the bottom of the purple LTP (at a physical port)' The signal would be de-multiplexed up to the top of the purple LTP and then re-multiplexed as it travels down the green LTP. The association between the two is essentially a degenerate point-to-point FC. The LTPs are split because of the change in flow orientation (multiplexing orientation). The association supporting this relationship is shown in the UML diagram in the figure above.

Considering the right most LTPs in the pictorial form and a signal entering the bottom of the purple LTP (at a physical port), the signal would be de-multiplexed up to the top of the purple LTP and then further de-multiplexed in the client LTPs. The LTPs are split because of a change in multiplicity or the opportunity to connect with an FC. The association supporting this relationship is shown in the UML diagram in the figure above.

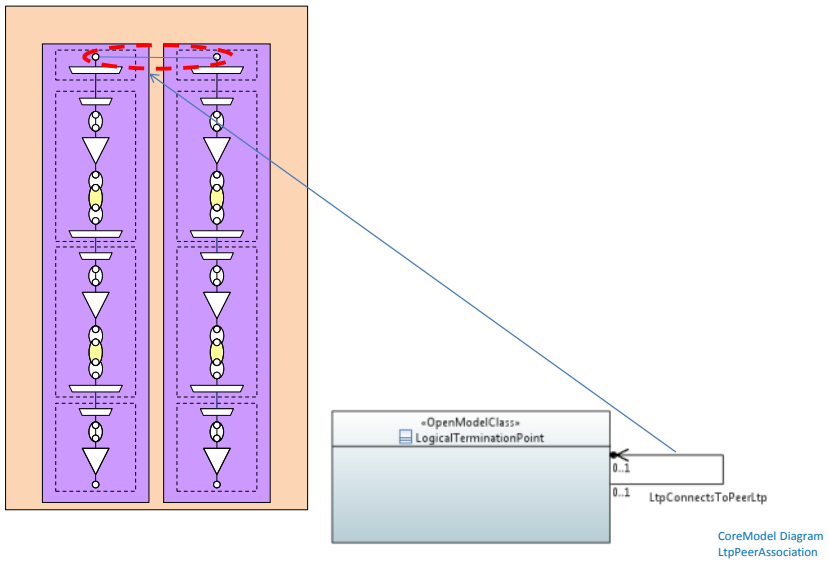


Figure 6-9 LtpConnectsToPeerLtp illustrated in an Amplifier/Regenerator context

In Figure 6-7 above, the final LTP to LTP association is highlighted. This allows two LTPs that are associated with physical ports without the need for an FC. This is only allowed in a case when the relationship between the LTPs is such that the whole signal from one LTP must flow to the other with no flexibility. The association effectively represents a degenerate FC.

6.4 Forwarding Fragment

6.4.1 Basic Forwarding

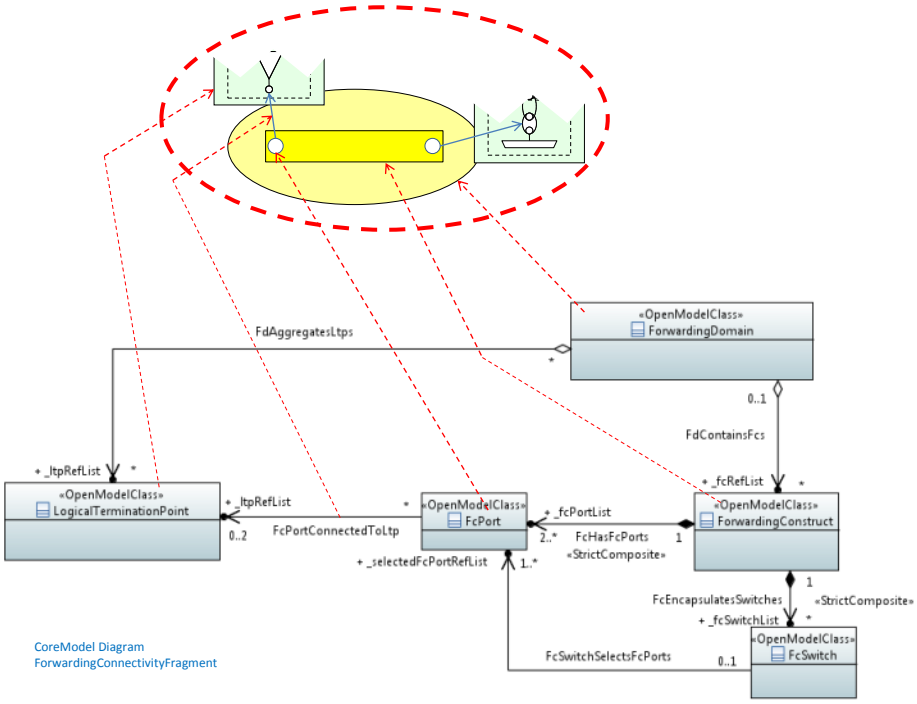


Figure 6-10 Forwarding fragment

The pictorial form in Figure 6-8 above shows the ForwardingConstruct (FC) in the context of two LTPs (a part of an earlier figure). The FcPort of the FC is shown within the FC, emphasizing the strict whole-part relationship and lifecycle dependency of the FcPort on the FC. The FcPorts are effectively FC component ports. The FC shown has two FcPorts but the model allows for two or more FcPorts [2..*] where in some cases the FcPort could be selected as a source or destination for switching. The protection switching capability is explained elsewhere in this document.

The [0..2] multiplicity of _ltpRefList (at the end of the association "FcPortConnectedToLtp" allows for a bidirectional FcPort to associate with two unidirectional LTPs.

6.4.2 Forwarding Construct Specification and other details of Forwarding

Prior to embarking on a brief description of the FC specification and associated classes it is important to explain the concept of specification classes in general. In this model the specification classes provide a mechanism to express the restrictions of a particular case of application of a specific class or set of classes. For example an FC may in general have [2..*] FcPorts while a specific case of FC may have exactly 4. This case may also be such that it has 2 switches and such that these switches affect specific flows in the FC. The FcSpec is designed to allow the expression of cases of this sort.

At this point only limited work has been done on specification in general with a focus on the FcSpec and associated classes. It is anticipated that in general specification classes would be developed for all entities in the model. The classes, data types etc. related to the specification are defined in section 9.3 Core Enhancements data dictionary on page 132.

In the diagram below, the FcSpec and supporting FcPortSetSpec describe the capabilities of the FC in terms of MultiSwitchedUniFlows, each of which has [1..*] IngressFcPortSets and [1..*] EgressFcPortSets. Each MultiSwitchedUniFlow may have [0..1] ingress switches and [0..1] egress switches where the ingress switch may select only one set member from one set and the egress switch may select [1..*] set members from the egress set. The ingress and egress switch selections are controlled by the ConfigurationAndSwitchControl element that may be:

- embedded in the switch when there is no coordination of switches required
- embedded in the FC when the coordination of switches is only in the scope of the FC
- independent of the FC and described by the ConfigurationGroupSpec where there is multi-FC coordination required

The behavior of the ConfigurationAndSwitchControl element is described by ControlRules.

The model has been exercised for a number of different cases (not detailed here). Figure 6-9 below provides the class diagram of the FC specification fragment.

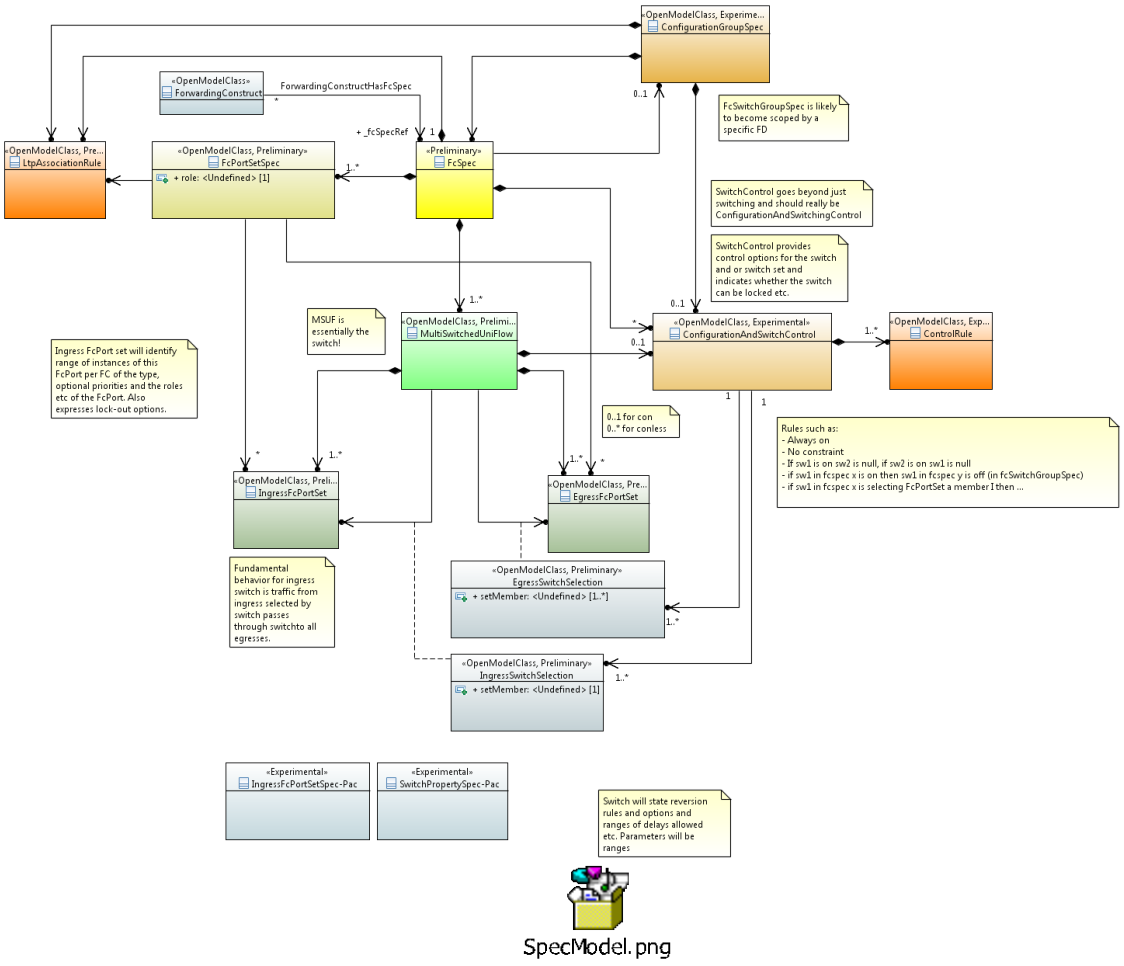


Figure 6-11 Class Diagram of the Spec Model of Connection Control

CoreModel diagram: FcCapabilitySpec

The diagrams below show a pictorial view of some of the classes above (the colors used in the figure are consistent with those used in the model above).

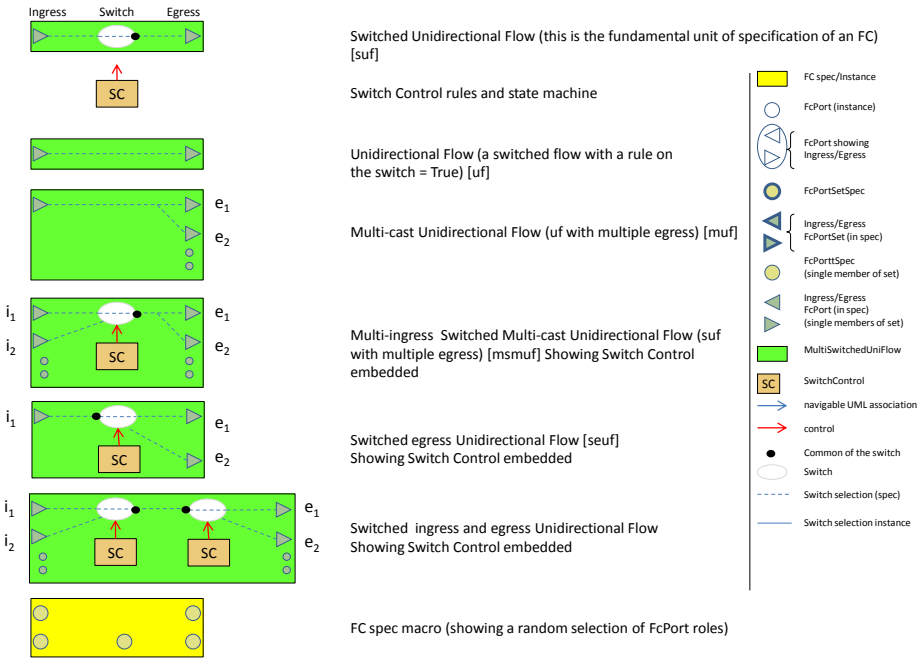


Figure 6-12 Pictorial view of the Spec Model of Configuration Control

The diagrams below show a pictorial view of a case of FcSpec. The lower element of the diagram shows specification class instances and the upper element shows an instance of FC abiding by the spec.

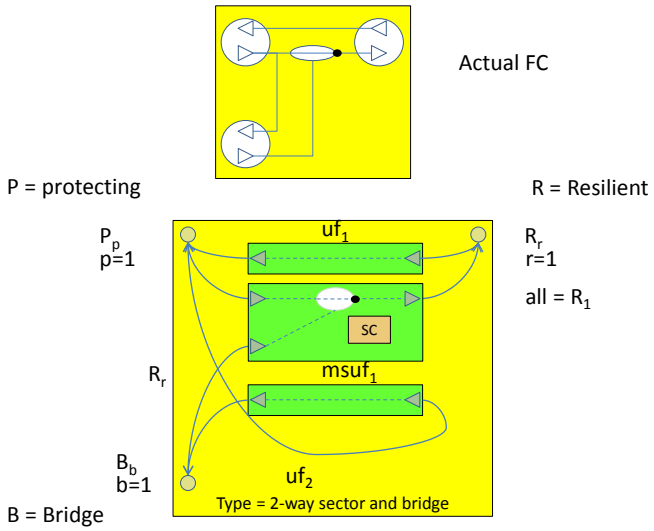


Figure 6-13 Pictorial view of spec model and resulting FC instance

The figure below provides the class diagram of further detailed FC and protection switching related object classes. The figure shows development of the controller of the FcSwitch. This area of model is experimental work in progress as highlighted by the «Experimental» stereotypes and on that basis is very likely to change.

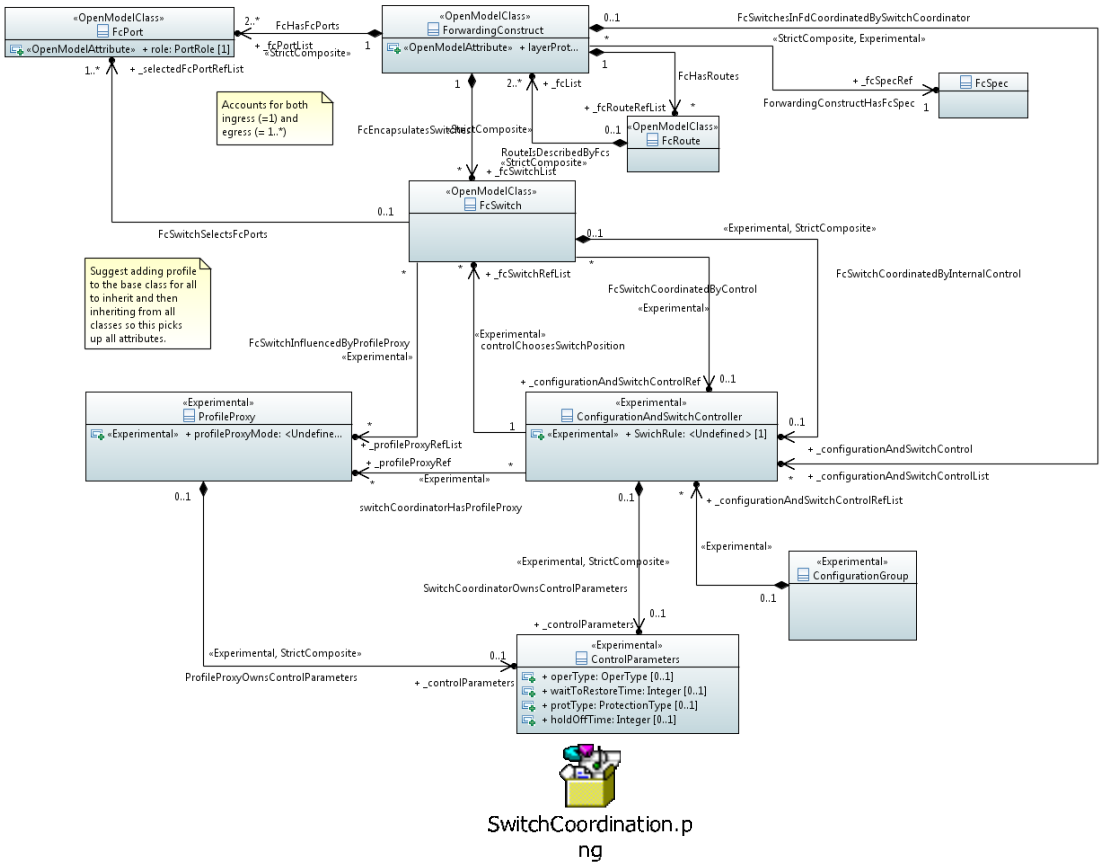


Figure 6-14 Switch coordination and control

CoreModel diagram: SketchOfSwitchCoordinatorAndProfile

The controller of switching that is often embedded in the FC can, as discussed, be pulled out to coordinate switching of several FCs (as per traditional protection group) and further can cause the creation/deletion of FC and hence is part of the continuum of management-control. This experimental model fragment offers flexibility in the way the FcSwitch gains its ControlParameters and provides an instantiable ConfigurationAndSwitchController that can be positioned with an appropriate scope of control for any particular case. The ConfigurationAndSwitchController can provide the control parameter to the FcSwitch (directly or via a ProfileProxy) or the FcSwitch can reference a profile (also available to the ConfigurationAndSwitchController). The ConfigurationAndSwitchController can be contained in the FcSwitch, can be contained in the FC and referenced by the FcSwitch or can be contained in some ConfigurationGroup entity with a scope greater than the FC and the FcSwitch.

Not covered at this point:

- This controllers need to be controlled/managed
- The controllers need a capability spec

6.5 Topology Fragment

The topology subset is summarized in the following figure.

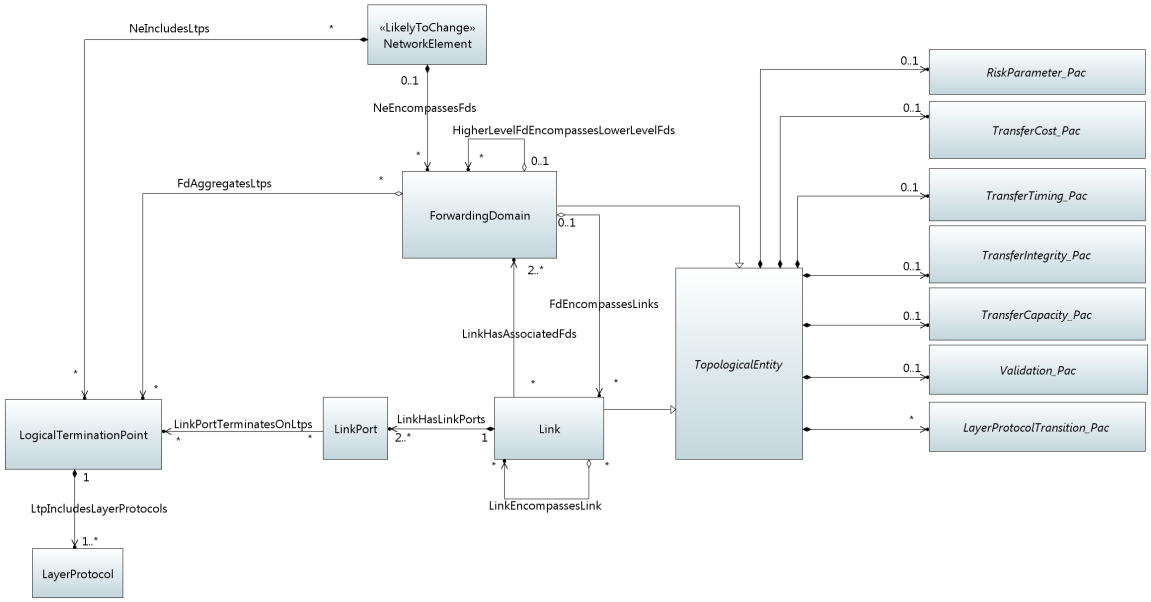


Figure 6-15 Classes of the Topology Subset

CoreModel diagram: Topology-HighLevelOverviewOfStructureAndPacs-LargeText

The figure above shows a lightweight view of the model omitting the attributes (described later in this section). The figure focuses on interrelationships and shows that:

- An FD may be a subordinate part of a NetworkElement, may coincide with a NetworkElement or may be larger than, and independent of, any NetworkElement (See for example FDs A.1 and A.3 in Figure 6-15).

- An FD may encompass lower level FDs. This may be such that:

  - An FD directly contained in a NetworkElement is divided into smaller parts

  - An FD not encompassed by a NetworkElement is divided into smaller parts some of which may be encompassed by NetworkElements (see Figure 6-17 ForwardingDomain recursion with link and NetworkElement on page 42)

  - The FD represents the whole network

  Note that an FD at the lowest level of abstraction (a fabric or some piece of a fabric) does not encompass FDs while an FD at the highest level of abstraction (i.e., the FD representing the whole network) is not encompassed by any higher level FDs.

- An FD encompasses Links that interconnect any FDs encompassed by the FD

  Note that Offnet Links are not encompassed by any FD. All other Links are always encompassed by one FD which may be the FD representing the whole network. As a consequence, the FD representing the whole network shall always be instantiated.

- A Link may aggregate Links in several ways
  - In parallel where several links are considered as one
  - In series where Links chain to form a Link of a greater span
    - Note that this case requires further development in the model
- A Link has associated FDs that it interconnects
  - A Link may interconnect 2 or more FDs[23]
    - Note that it is usual for a Link to interconnect 2 FDs but there are cases where many may be interconnected by a Link
- A Link has LinkPorts that represent the ports of the Link itself
  - LinkPorts are especially relevant for multi-ended asymmetric Link
- An FD aggregates LogicalTerminationPoints (LTPs) that bound it. An LTP represent a stack of layer-protocol terminations, where the details of each is held in the LayerProtocol (LP). An LTP may be:
  - Part of a NetworkElement
  - Conceptually independent from any NetworkElement[24]
- A Link terminates on LTPs via its contained LinkPorts.

Both the Link and FD are TopologicalEntities (an abstract class, i.e., a class that will never instantiate) and hence they can acquire contents from the conditional packages (_Pacs). The conditional packages provide all key topology properties.

6.5.1 Basic Topology

In this section a basic stylized network example is used to illustrate some of the associations in the Topology model fragment. The first two figures focus on the ForwardingDomain class and

[23] An off-network link with two LinkPorts does not interconnect any FDs in the view.
[24] The assumption is that the LTP can be floating (representing a pool) in the context of a network as a whole (represented by an FD). The LTP has a UUID to allow it to be identified. Under these circumstances it does not need to be contained in anything (although it is pooled by the FD representing the network). It clearly does need to be accessible via a controller.

the recursive aggregation relationship as well as the relationship between the ForwardingDomain, Link and the NetworkElement.

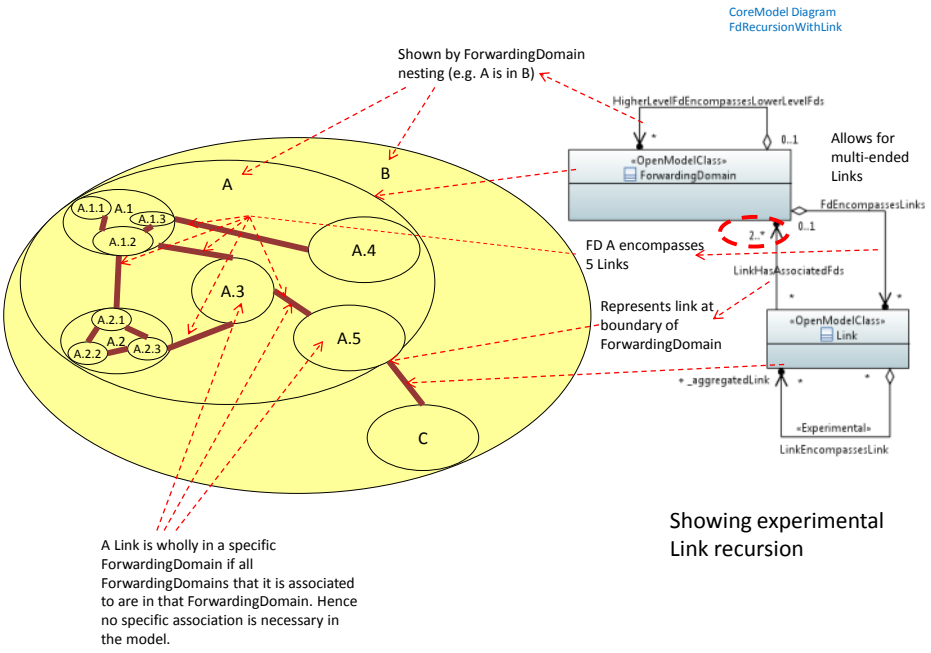


Figure 6-16 ForwardingDomain recursion with Link[25]

The figure above shows a UML fragment including the Link and ForwardingDomain (FD). For simplicity it is assumed here that the Links and FDs are for a single layer-protocol although an FD can support a list of layer-protocols.

The pictorial form shows a number of instances of FD interconnected by Links and shows nesting of FDs. The recursive aggregation HigherLevelFdEncompassesLowerLevelFds relationship (aggregation is represented by an open diamond) supports the ForwardingDomain nesting, but it should be noted that this is intentionally showing no lifecycle dependency between the lower ForwardingDomains and the higher ones that nest them (to do this composition, a black diamond would have been used instead of an open diamond). This is to allow for rearrangements of the ForwardingDomain hierarchy (e.g., when regions of a network are split or merged) and to emphasize that the nesting is an abstraction rather than decomposition. The underlying network still operates regardless of how it is perceived in terms of aggregating ForwardingDomains. The model allows for only one hierarchy.

In the example in the figure above, there are fourteen FD instances with the following instances of the "HigherLevelFdEncompassesLowerLevelFds" relationships:

- B encompasses two FDs: A and C

[25] The numbering on the figure implies strict and fixed hierarchy. It should be noted that the association is aggregation and hence the hierarchy can change and an FD may move from being encompassed by one FD to being encompassed by another. Consider the numbering as simply a view of the current structure.

- A encompasses five FDs: A.1, A.2, A.3, A.4 and A.5
- A.1 encompasses three FDs: A.1.1, A.1.2 and A.1.3
- A.2 encompasses three FDs: A.2.1, A.2.2 and A.2.3

When one FD is removed, the "HigherLevelFdEncompassesLowerLevelFds" relationships are modified. For example, if FD A.1 in Figure 4-2 is removed, the instances of the "HigherLevelFdEncompassesLowerLevelFds" relationships will be modified as follows:

- B encompasses two FDs: A and C
- A encompasses seven FDs: A.1.1, A.1.2, A.1.3, A.2, A.3, A.4 and A.5[26]
- A.2 encompasses three FDs: A.2.1, A.2.2 and A.2.3

An FD can also be added. Initially it will have no associated lower level FDs. Existing FDs can be moved as appropriate to form the new hierarchy.

The association between Link and FD allows a Link to be terminated on two or more FDs (see Figure 6-18 ForwardingDomain, Link and LTP associations on page 43). Through this the model supports point to point Links as well as cases where the server ForwardingConstruct is multi-point terminated giving rise to a multi-pointed Link. Multi-pointed links occur in PON and Layer 2 MAC in MAC.[27]

It should be noted that the model includes LinkPort which further details the relationship between FD and Link. This is explained below.

[26] Clearly the FD naming in the figure is for ease of reading the diagram and does not represent hierarchy.
[27] Work supporting this was liaised from TM Forum.

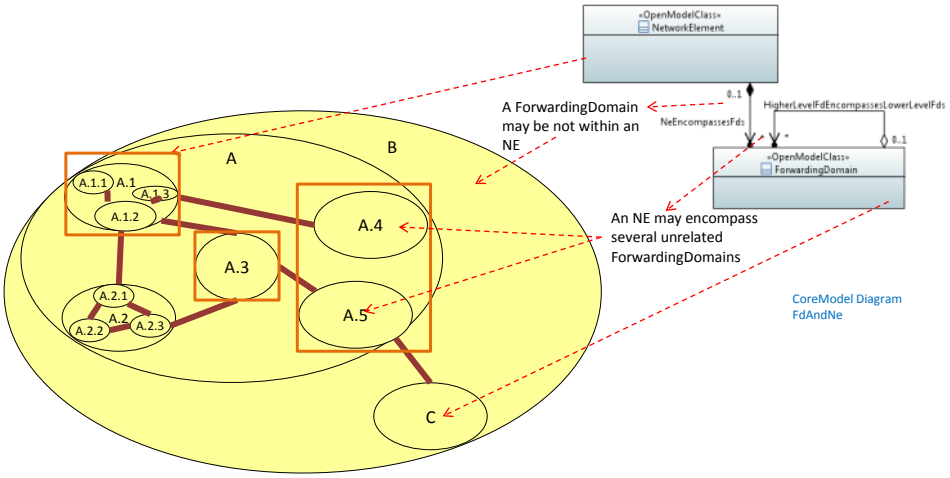


Figure 6-17 ForwardingDomain recursion with link and NetworkElement

The figure above the pictorial form shows an overlay of NetworkElement on the ForwardingDomains and a corresponding fragment of UML showing only the ForwardingDomain and NetworkElement classes.

The figure emphasizes that at and below one particular level of abstraction of ForwardingDomain, the ForwardingDomains are all bounded by a specific NetworkElement (brown square). This is represented in the UML fragment by the composition association (black diamond) that explains that there is a lifecycle dependency in that the ForwardingDomain at this level cannot exist without the NetworkElement. The figure also shows that a ForwardingDomain need not be bounded by a NetworkElement (as explained in the UML fragment by the 0..1 composition), and that a ForwardingDomain may have a smaller scope than the whole NetworkElement (even when considering only a single layer-protocol as noted earlier). In one case depicted (e.g., the right hand side NetworkElement encompassing two FDs), the two ForwardingDomains in the NetworkElement are completely independent. In the other cases depicted (e.g., the left hand side NetworkElement encompassing three FDs), the subordinate ForwardingDomains are themselves joined by Links emphasizing that the NetworkElement does not necessarily represent the lowest level of relevant network decomposition.

The figure also emphasizes that just because one ForwardingDomain at a particular level of decomposition of the network happens to be the one bounded by a NetworkElement does not mean that all ForwardingDomains at that level are also bounded by NetworkElements.[28]

[28] It should be noted that a NetworkElement is never within the bounds of an FD. The NetworkElement is associated with levels in the FD hierarchy.

The following figure zooms in on a fragment of the network used in previous figures. The figure shows detail of the LinkPort and LTP (intentionally omitted from the earlier figures). The key points are highlighted in the figure.

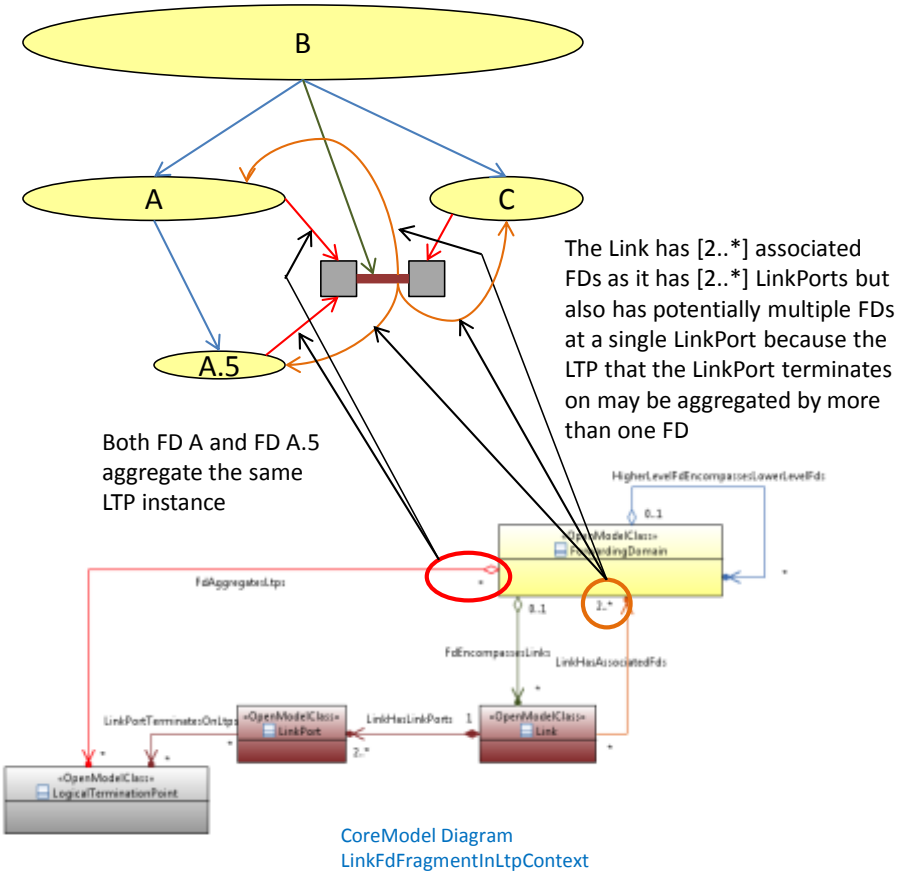


Figure 6-18 ForwardingDomain, Link and LTP associations

An alternative way of depicting the topology of the example is shown in the next figure. The link shown in the figure above is shown twice in the next figure as highlighted in the figure.

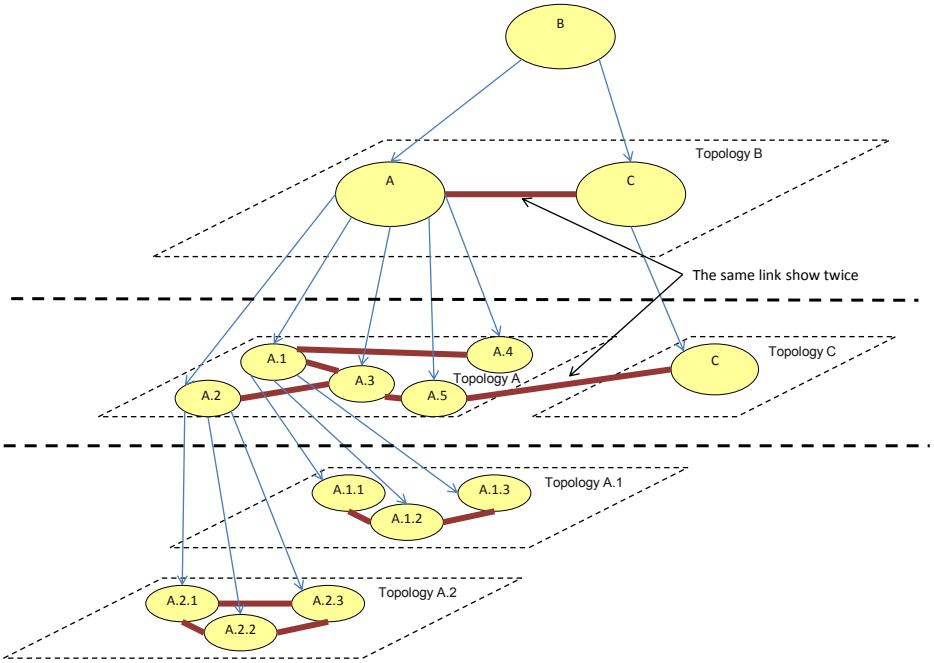


Figure 6-19 FDs and Topology

The following figure considers the topology further in combination with the FCs. The figure shows LTP aggregated in two (or more) FDs and highlights that the FD may be of different layer-protocols than the LTP. The LTP will be aggregated by an FD if a LayerProtocol of the LTP:

- Is for the same layer-protocol of the FD (and the LTP is on the FD boundary)
- Adapts to the layer-protocol of the FD (and the LTP is on the FD boundary)

Note that the figure shows an NE context to assist in understanding that the principle also applies in a fully "virtualized" case (simply remove the NE boxes).

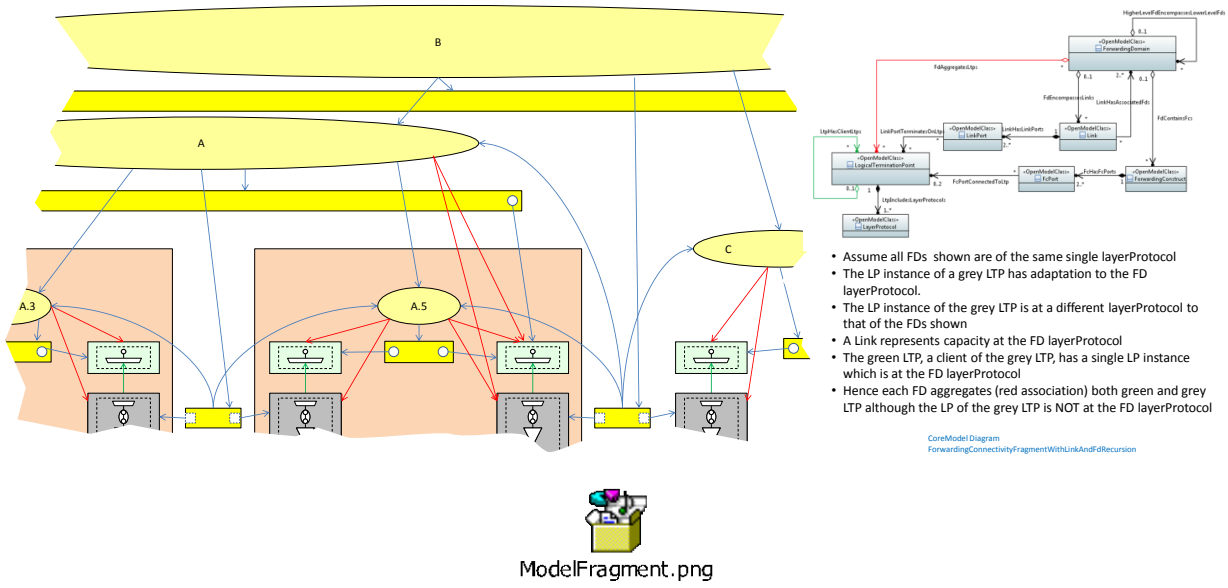


Figure 6-20 LTPs Encompassed by FDs (at one layer-protocol)

Clearly an LTP with multiple LPs may be aggregated by FDs at multiple layer-portocols. An LTP may be aggregated by FDs of different layer-protocols even where the LTP only has one LP. The figure below shows a case where there is a floating LTP, A, containing a single LP where the LTP is aggregated in FD B at layer-protocol X and FD C and FD D at layer-protocol Y.

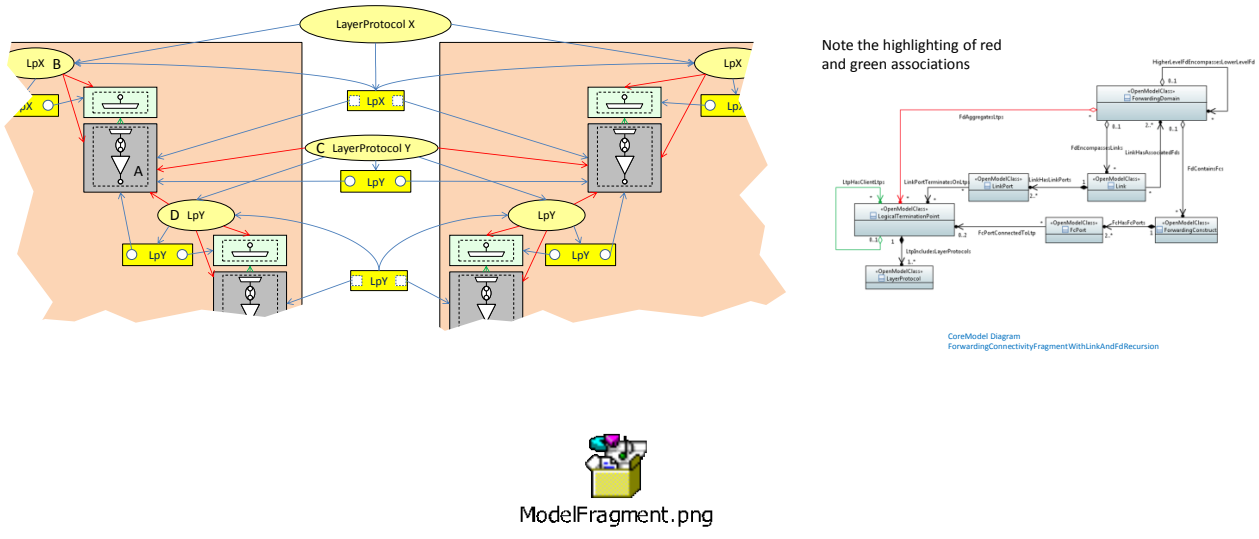


Figure 6-21 LTPs Encompassed by FDs (at several layer-protocols)

6.5.2 Topology and views

- For cases where there is no physical LTP a “floating” LTP is used.
- Where the situation is fully virtualized a “floating” LTP with only the pooling function is used.
- An inter-view relationship to link contents of a “floating” LTP with the contents of a physically bound LTP is shown (preliminary). This is essentially internally to the controller

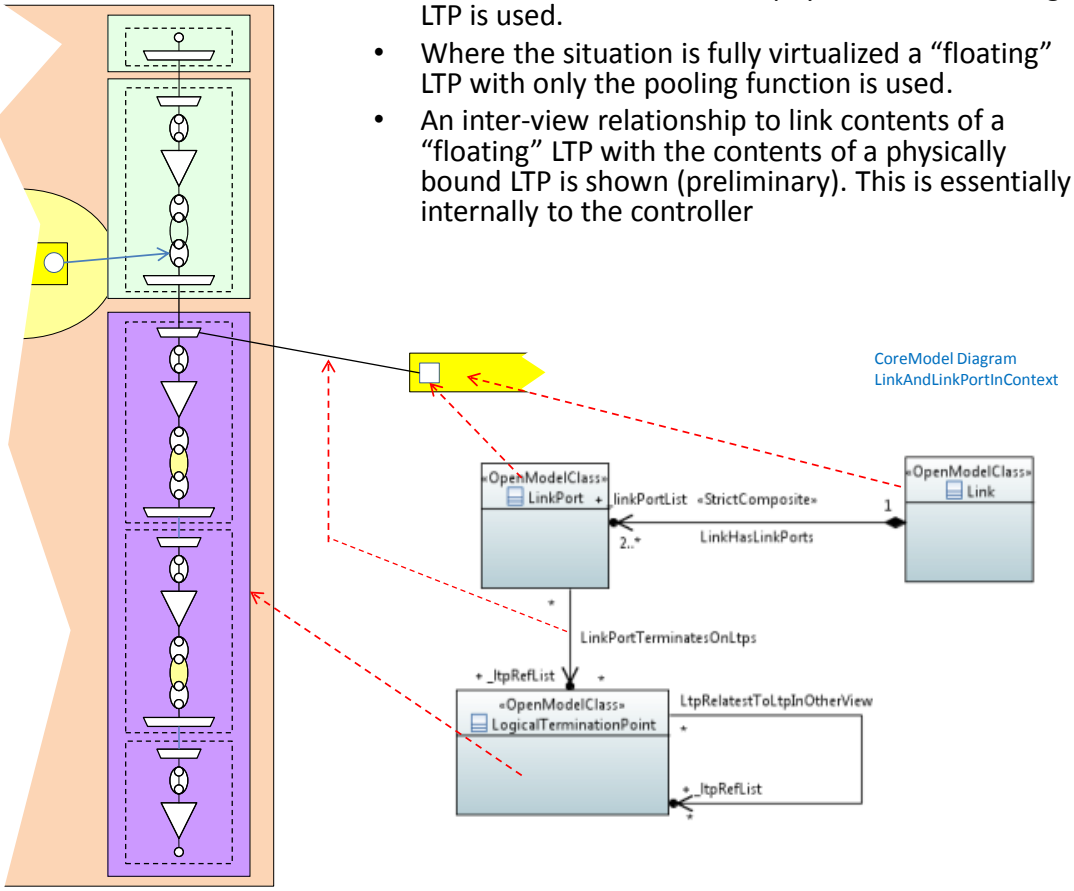


Figure 6-22 LTP "pooling" client LTPs

Figure 6-16 above shows how the Link terminates on the LTP via the LinkPort.

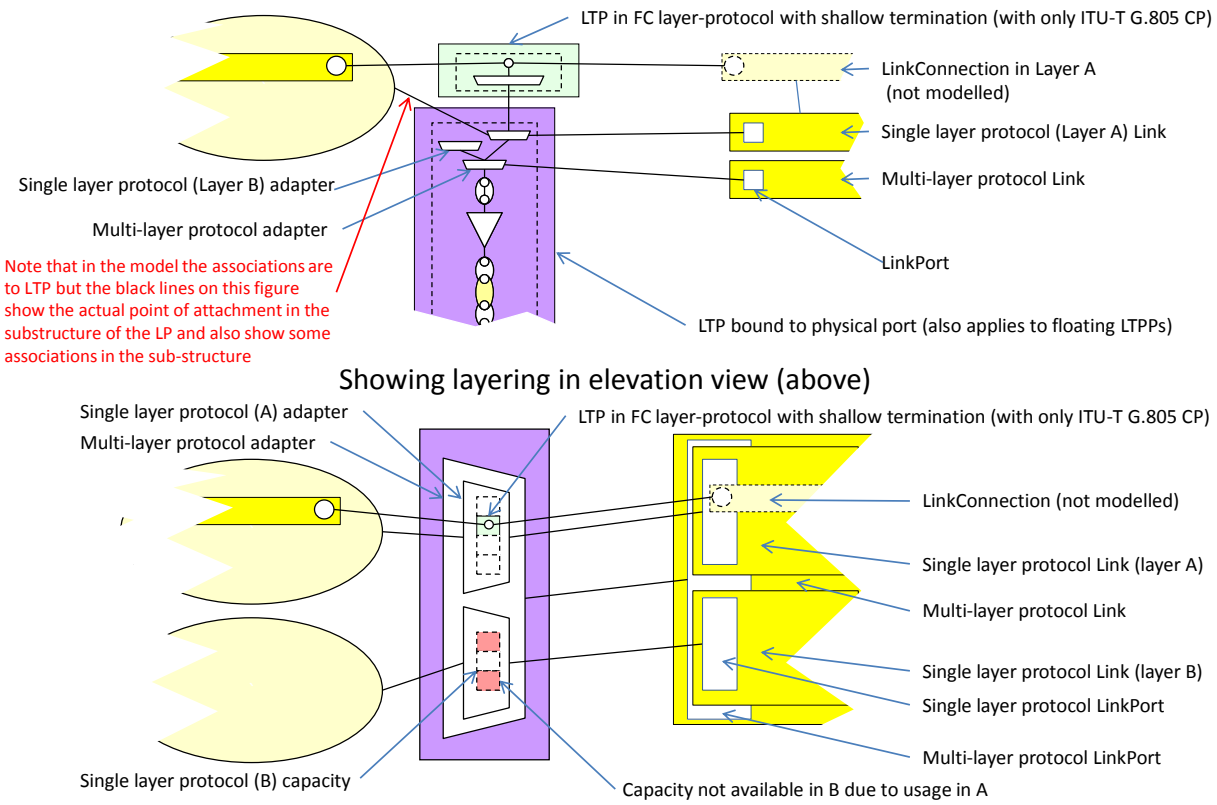


Figure 6-23 Views of Link, LinkPort and LTP showing LTP pooling

The LTP may have the capability[29] to map to multiple client layer-protocols where there is an interaction between the client mappings (e.g., if capacity/channel x of client layer-protocol A is used then capacity/channel set y of client layer-protocol B is no longer available). The capacity of the Link is determined by evaluating the "intersection" of capabilities of the LTPs at the ends (which is complex in a multi-ended case).

The used capacity is determined by considering which client LTPs exist as a result of their being FCs.

A Link may be multi-layered and hence may represent the whole client capacity of an LTP or it may be single layered.

[29] This capability of the LTP is not currently modeled but work is under way to construct an LTP specification model

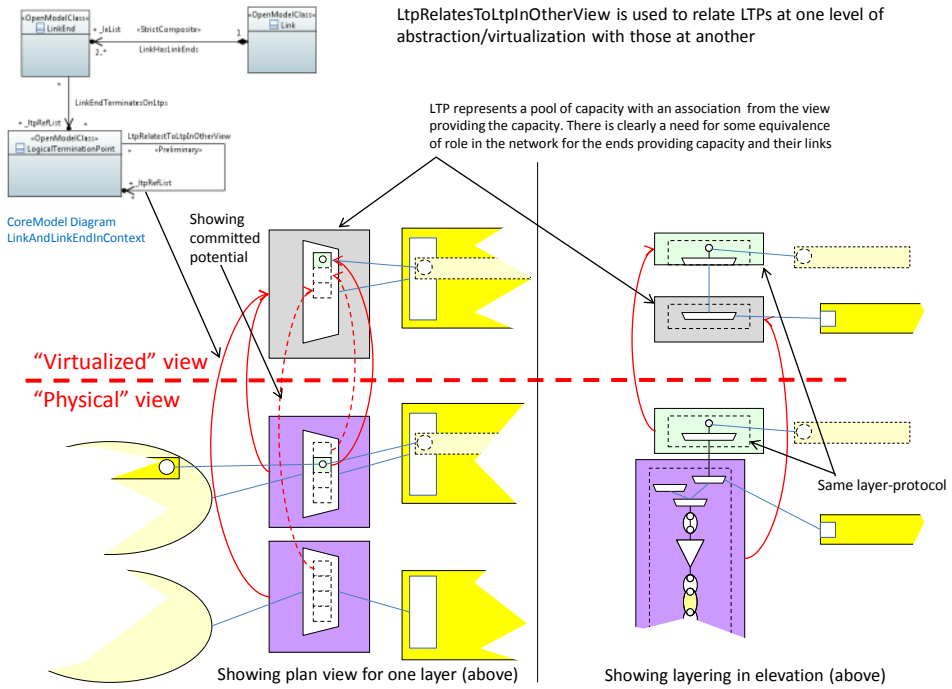


Figure 6-24 Views of "virtualization" [30] of LTPs with server side LTP representing a pool

Some capacity may be taken from each of a number of Links supporting a particular layer-protocol and offered in a "virtualized" view perhaps for use in a particular application etc. The "virtualized" view will normally be referenced in a different name space. The rules for grouping capacity into Links in the "virtualized" view have not yet been documented. The same model is used for Links and LTPs in the "virtualized" view as is used in the "physical" view.

It is important to emphasize that the Virtual/Physical split is a gross simplification. In reality a server provides an abstracted/virtualized view of an underlying system to its client where that underlying system is provided by further servers hence "Physical" view obscures this complexity (but is sufficient for this description).

- Both views are virtualized where the lower view is "providing" to the upper view
- Using the term "physical" at this point is tolerable as it enables easier case oriented interpretation of the figures and concepts.
- Something like "provider's resource context" and "provider's client view context" may be better terms in the long run

The figures below provide a view of sequence of realization of a virtualized view.

The first figure provides a starting position. The figure depicts a virtualized view and a completely disassociated physical view. At this point although the network does exist it has no capacity allocated to the client or used in any way.

[30] The terms "physical", "virtualization" and "virtualized" are used loosely here. The "physical" aspects are shown in the context of LTPs bound to physical but in general this is really the "provider view" and the "virtualized" aspects are really "provider's client view context" (which is essentially what the provider exposes to the client".

However, the client has been offered some pre-planned resources and has chosen usage of some resources. These resources are clearly not operable. This is analogous to pre-provisioning an equipment slot in an NE.

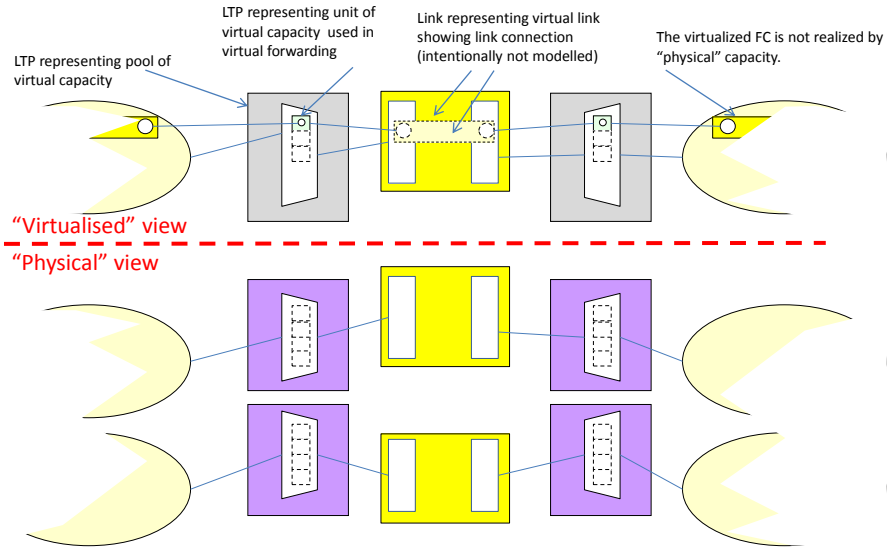


Figure 6-25 Starting condition

The operator then chooses to allocate capacity is allocated by the provider from the "Physical" view to the "Virtualized" view. Note that the association "LtpRelatesToLtpInOtherView" is from the "Physical" view to the "Virtualized" view and is from both levels of LTP (LayerProtocol Client and LayerProtocol Server). This orientation emphasizes that real resources are provided and that the actual client will not see anything other than the virtualized view. Clearly in some places in an actual solution realization both directions of association may be beneficial.

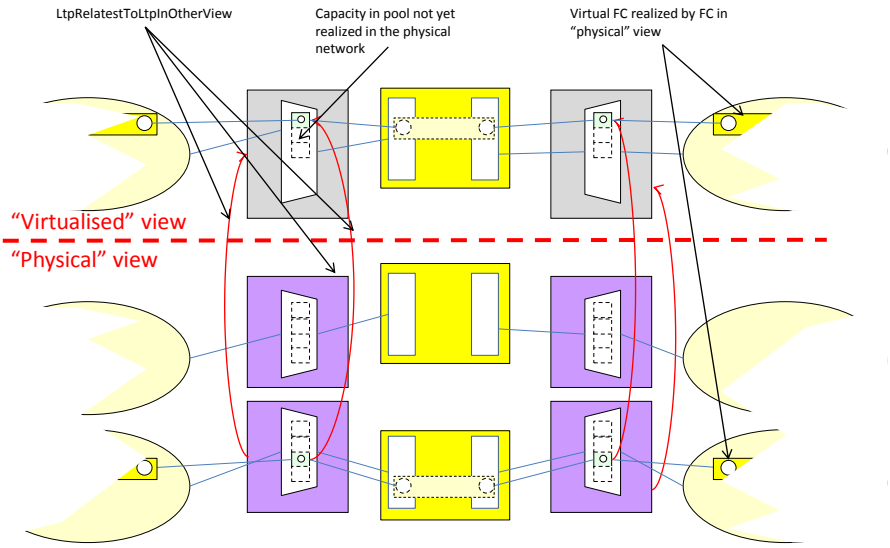


Figure 6-26 Resource allocation

The figure illustrates that there is no necessary ordering/numbering consistency between the "Physical" view and the "Virtualized" view.

At some future point the provider my decide to reallocate resources in the network such that the "Virtualized" view LTP is now supported by a different "Physical" view LTP (as shown in the figure below). Clearly there are various sequencing considerations to minimize impact. The essential thing to note is that the naming/addressing in the "Virtualized" view is unchanged through the process.

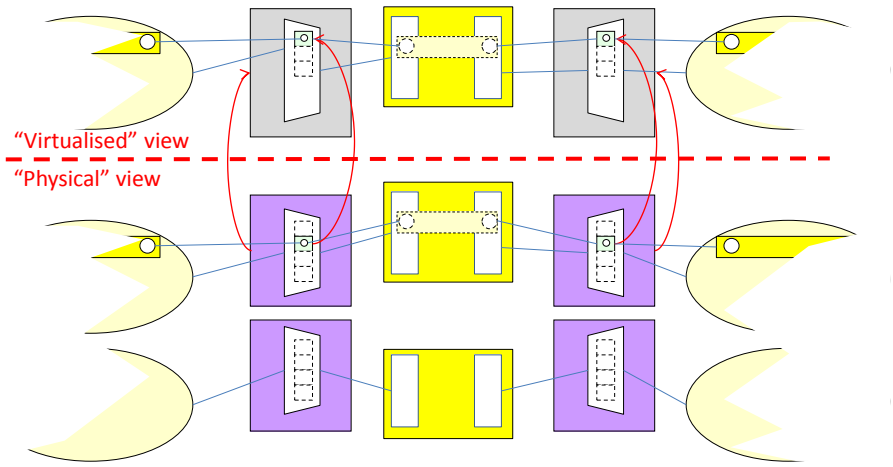


Figure 6-27 Move of allocation with no change to "Virtualized" view

After some further time the operator may choose to add capacity to the "Virtualized" view as illustrated in the next figure.

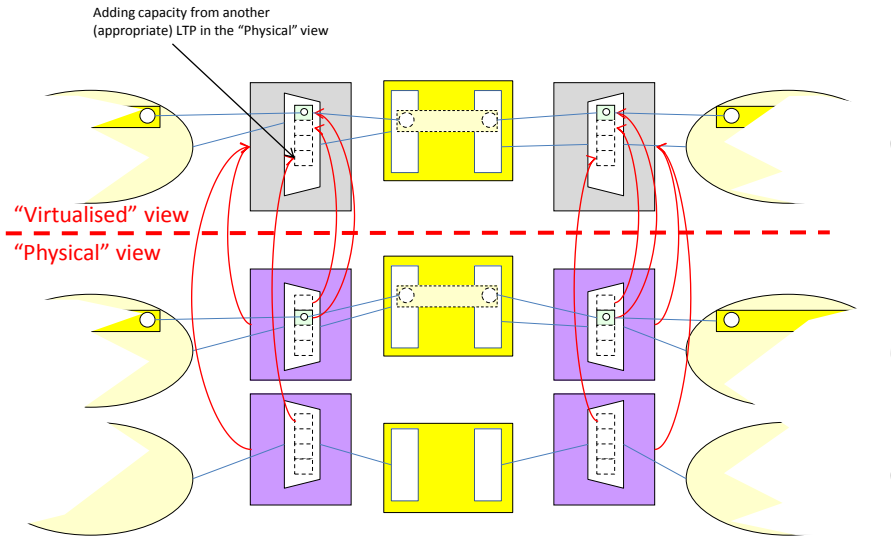


Figure 6-28 Capacity from server LayerProtocol Server LTPs

In this case there are two "Physical" view LayerProtocol Server LTPs associated with a single "Virtualized" view Pool LTP. Note that the Server LTP plays a Pool role.

The above illustration sequence leads to the following observations:

- There is no fixed association between the resources represented in the "Virtualized" view and the resources represented in the "Physical" view.
    - The identifiers in the two spaces must be different. This will be discussed in a following section.
- The "LtpRelatesToLtpInOtherView" association can provide all necessary view interrelationships

6.5.3 View boundaries and intermediates

In the previous section the "Virtualized" view had no physical ports. However, clearly a client to a network may need to connect at a physical port. The following figure shows several network cases as simple sketches where the outer ellipse boundary represents the actual commercial network boundary. A normal interworking case the operator exposes nothing of the interior of the network so the network is opaque and only the physical edge detail is provided (as show in the upper left diagram in the figure below). In some cases the operator may choose to expose apparent interior structure to perhaps explain capacity limitations. The network is essentially semi-transparent. It is possible that the network edge is essentially in the cloud so that even the interconnects are virtualized. A fully virtualized case where there is some exposure of internal constraints is shown in the lower right diagram in the figure below.

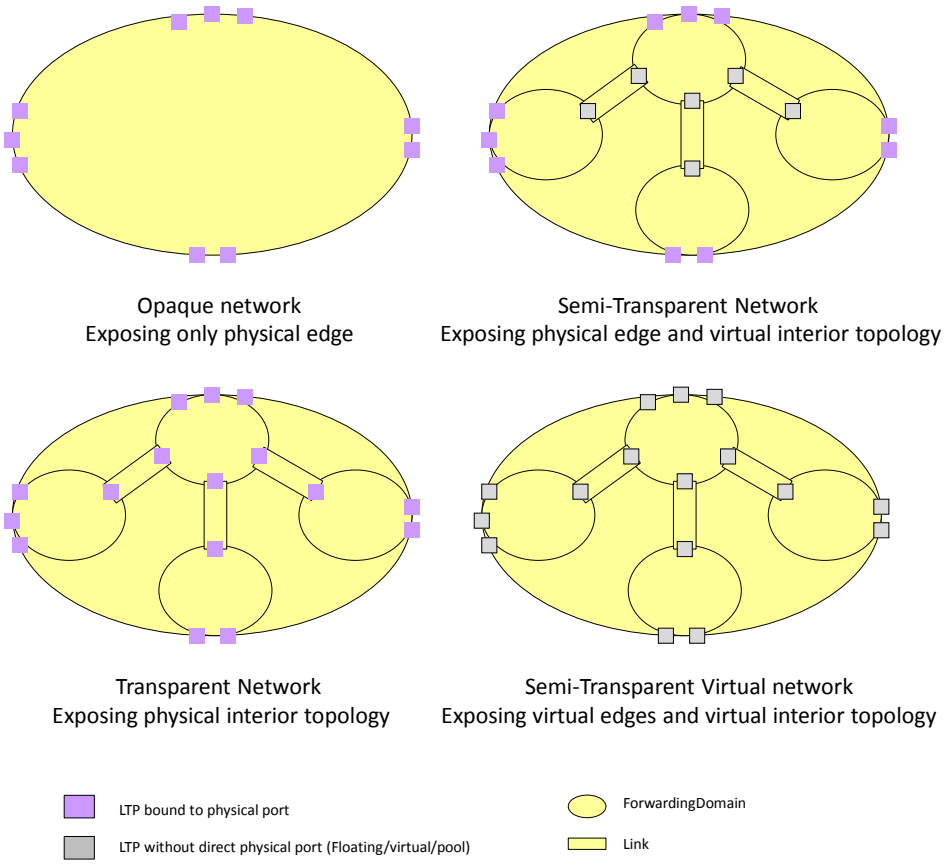


Figure 6-29 Various view boundaries

6.5.4 More on views and names/identifiers

Each view may have its own name spaces and/or identifier spaces. An entity, regardless of which view it is in, will expose the appropriate name and identifiers using the attributes highlighted in section 6.2.1 Naming and identifiers on page 23. An entity may have several names and several identifires. An entity may be referred to using an address (a sequence of names/identifiers) where the names/identifiers have a local scope smaller than the context in which the entity needs to be uniquely determined.

In the following figure a number of views are exposed where each has its own namespace and where the LTPs relate via the "LtpRelatesToLtpInOtherView" association as discussed in the earlier section. There could be more or less views in the recursion and the discussion here is not on the absolute number of levels but instead on how they relate and on how the things in the views are referenced.

The most abstracted view (Abstract Intent[31]) shows the FC bounded represents a "Service"[32] or Call [ITU-T G.8081][33]. The FC is, as usual by LTPs, at the actual physical edge of the

[31] The term “Intent” is being used loosely here
[32] The usage of the term “Service” is intentionally vague here.
[33] The choice of term depends upon the terminology context and the usages are not always directly analogous in detail (but at this level of description are sufficient).

administration of the network. There is a two level hierarchy of LTPs shown where the lower (grey) represents the pool of physical network access ports and the upper LTP represents the per-"Service"/Call forwarding termination[34]. The layering of the upper LTP is that of the "Service"/Call.

These LTPs have abstract references. A common acronym for references at this level of abstraction is TRI. The TRI will carry a reference that is known by both either side of the administrative demarcation.

Depending upon the approach to the TRI generation, the TRI may be structured with a number of fields as an address or may be a single opaque field. Depending upon the quality of the TRI scheme, the TRI could be considered as either a name or an identifier (or address of names or identifiers). Regardless, the name "TRI" would be conveyed in the valueName field of the NameAndValue type (used for the appropriate localId or for the appropriate name).

At the next level of abstraction shown (Detailed Intent) the FC represents a "Service" decomposition or a Connection etc. The same approach is used for the SNP reference relevant at the next level of abstraction. The layering here is more precise, representing the effect of the network as viewed through the physical port. In this particular case, each LTP bounding the call is realized by a pair of LTPs in the connection[35].

In the final two levels of abstraction ("Realization" and "Physical Network") the FCs and LTPs take their more familiar roles.

[34] The layering of the lower LTPs depends upon the variety of accesses and will not be discussed further here.
[35] This level may be decomposed into “connection” segments and there may be many recursions of abstraction decomposition.

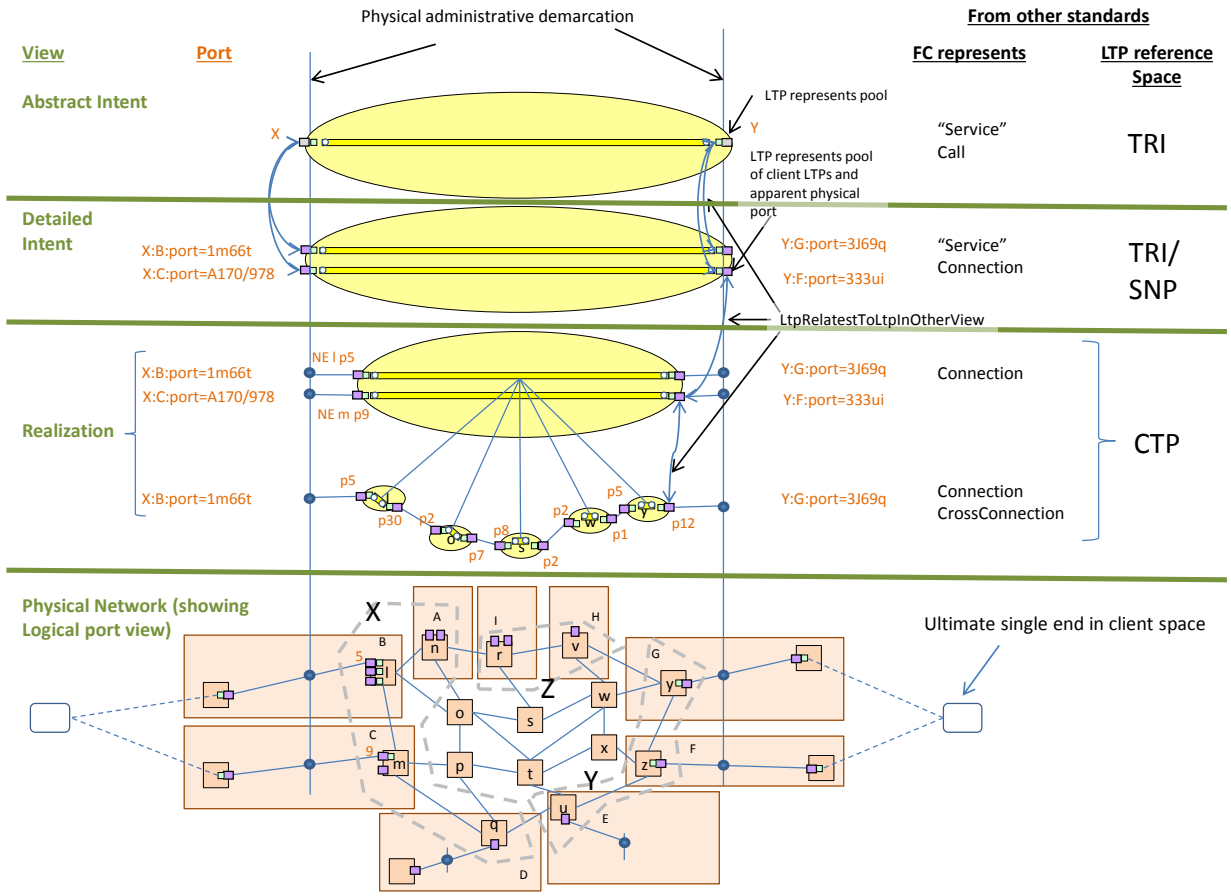


Figure 6-30 Various interrelated network views in a multi-party context

A final consideration at the edge of the network is the layering perceived by the client in a case where there is a device at the edge of the network that is not operating at the layer of the service. The figure below shows such a case. The key observation is that the layering of ports deep in the network is projected through the ports at the edge to form a hybrid apparent layering structure that is then exposed to the client. The exposure is exactly what would be seen if the client were to "look into" the edge port.

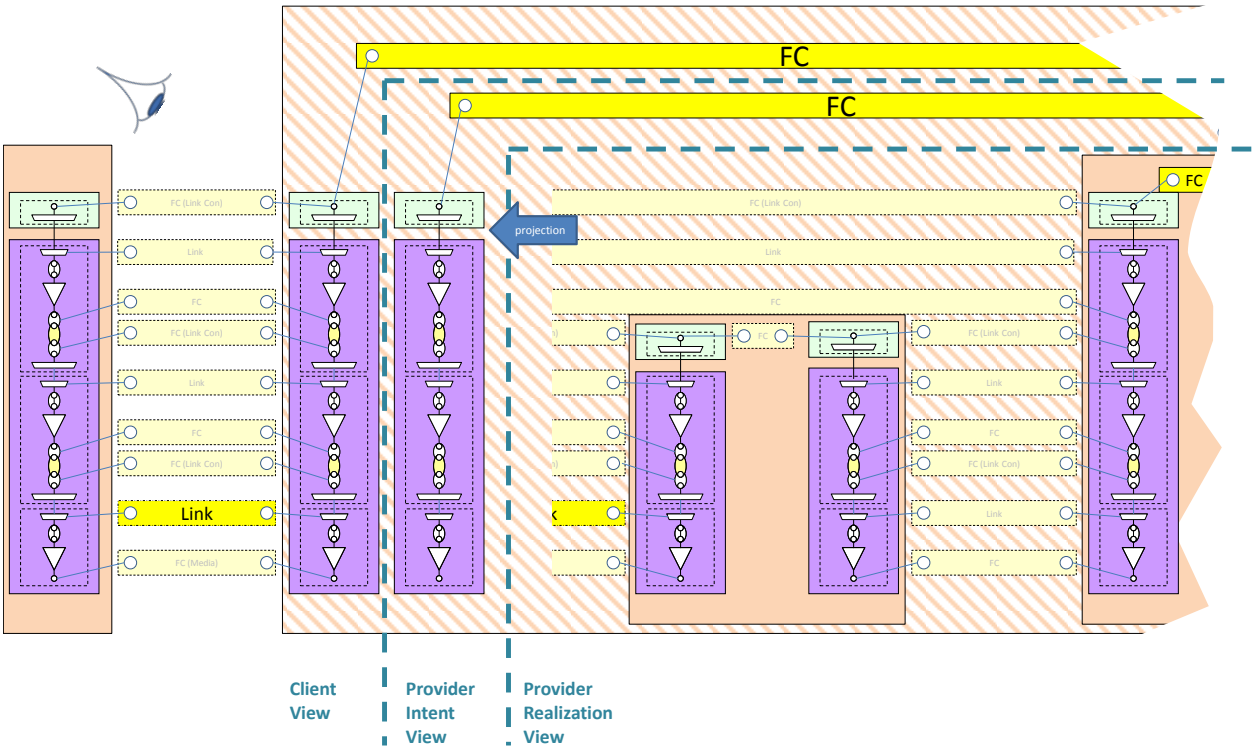


Figure 6-31 Complex network edge

6.5.5 Off-network reference and the clients view

The following figure shows the positioning of a link with an LinkPort that will use the "offNetworkAddress" attribute rather than a fully resolved LTP. Each blue dot in the figure represents an off-network address.

Unlike the case of the Client in the previous section, the Provider does not need to have any knowledge of the client port, the client does not need to present any view of their network to the Provider. The provider could create a dummy LTP to represent the client port or could simply end the Link with an off-network reference (offNetworkAddress)[36] in the LinkPort.

[36] Note that the attribute in the model is «Experimental»

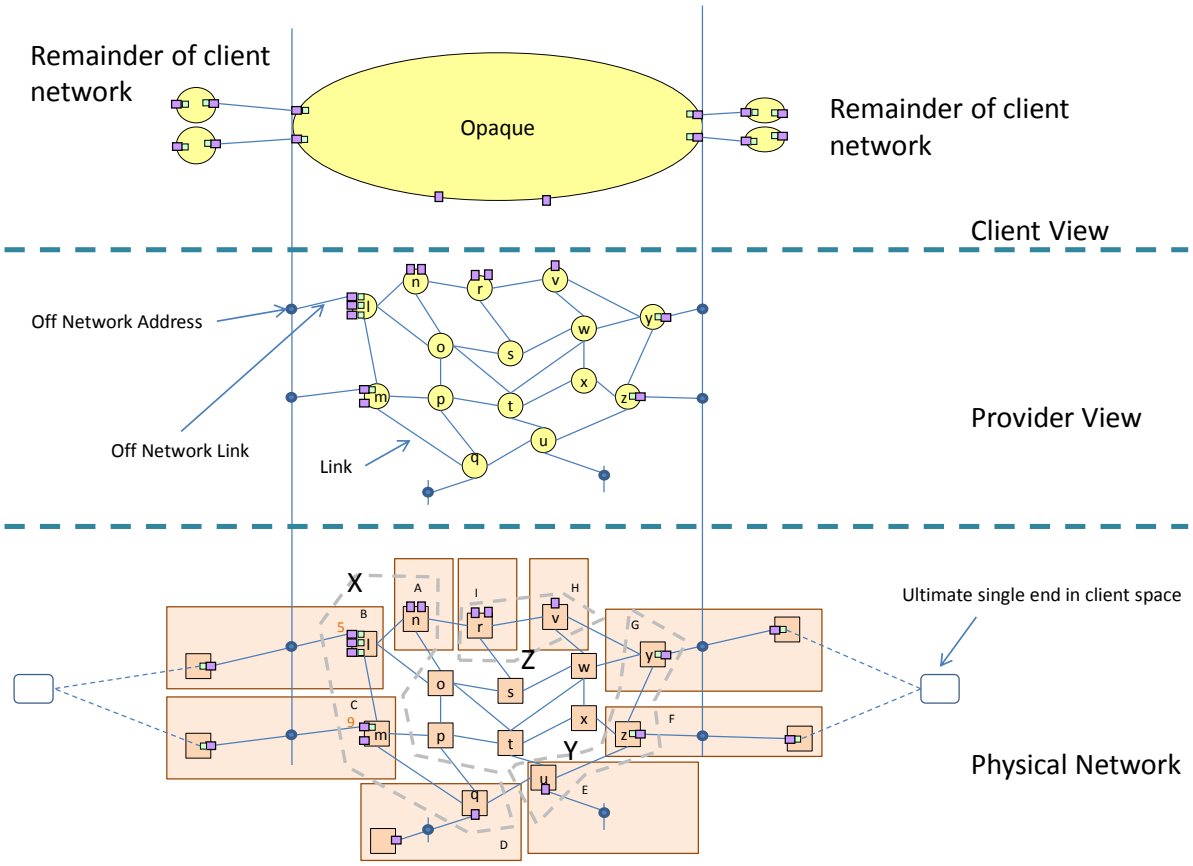


Figure 6-32 Complex network edge

6.5.6 Physical port reference

The port is not modeled as a distinct entity but the "name" of the port is provided in the LTP in a port structure. The connector is not currently modeled as a distinct entity but the connector details for the port ccould be provided in the port structure as relevant. At this point a basic physical port reference is supported in the model from the LTP. The attribute allows for levels of structuring of the reference where relevant. The connector and pin details of the port could be left to the specification rather than expressed in the operational model.

There are many potential arrangements of association of LTP to physical port/connector. The following sequence of figures provides a view of some of the variety.

Port is a "virtual" concept related to some coherent traffic flow. As can be seen from the sequence of figures there is no fixed relationship from port to connector/pin.

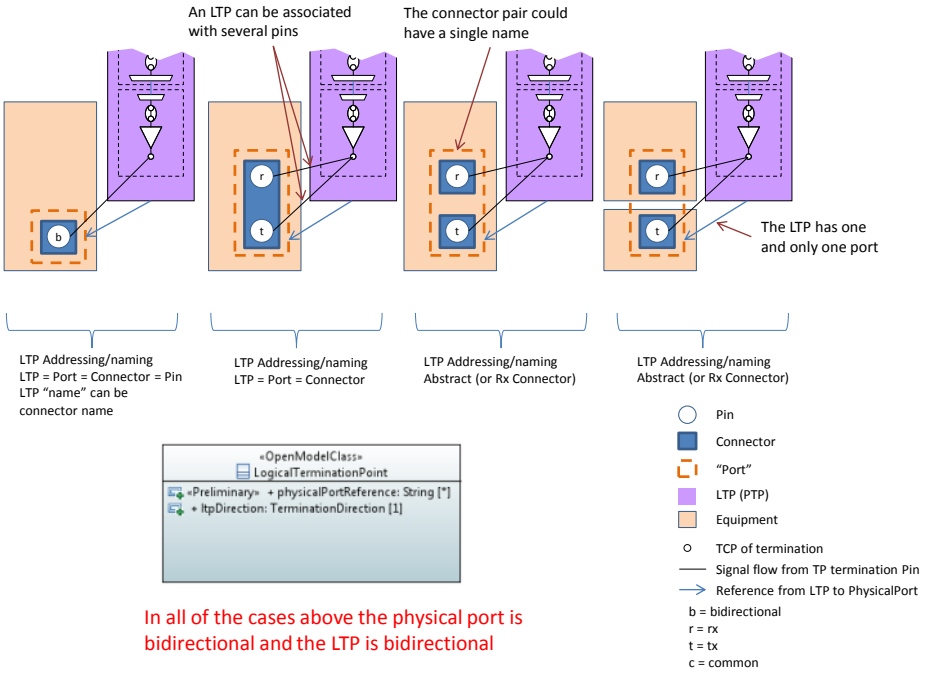


Figure 6-33 Basic cases of Physical Port Reference

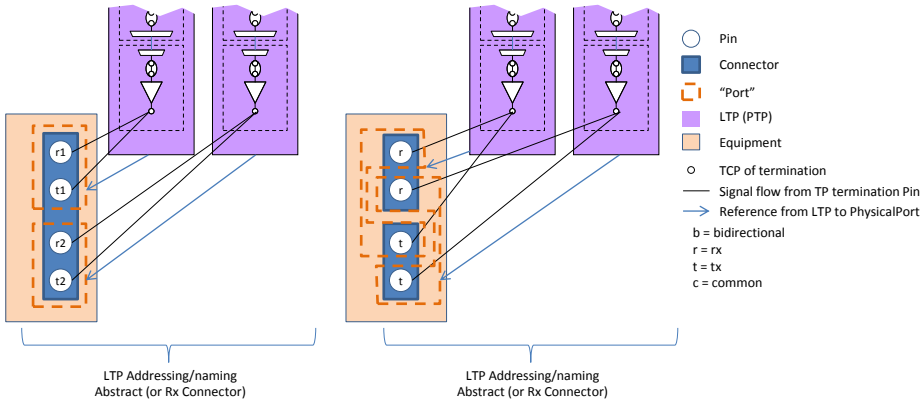


Figure 6-34 More Complex cases of intertwined connectors

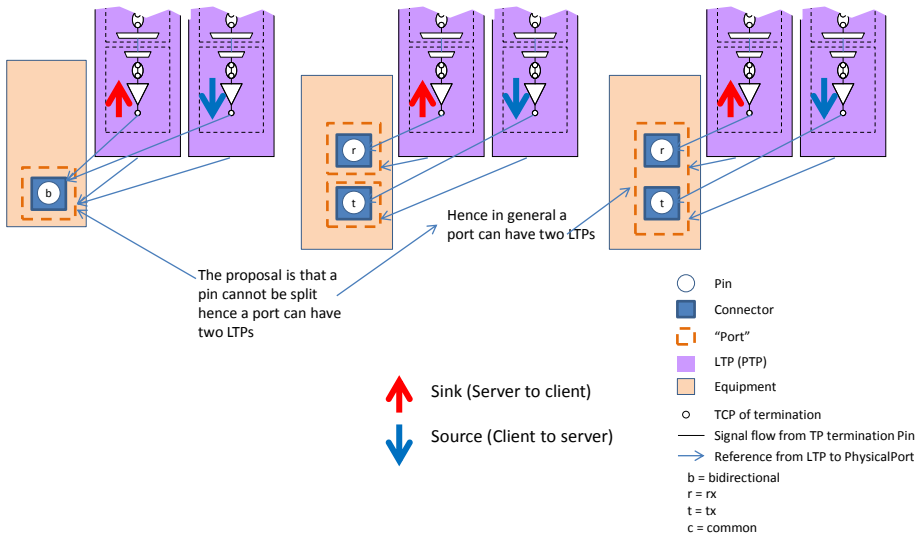


Figure 6-35 Unidirectional Cases

6.5.7 Detailed properties of Topology

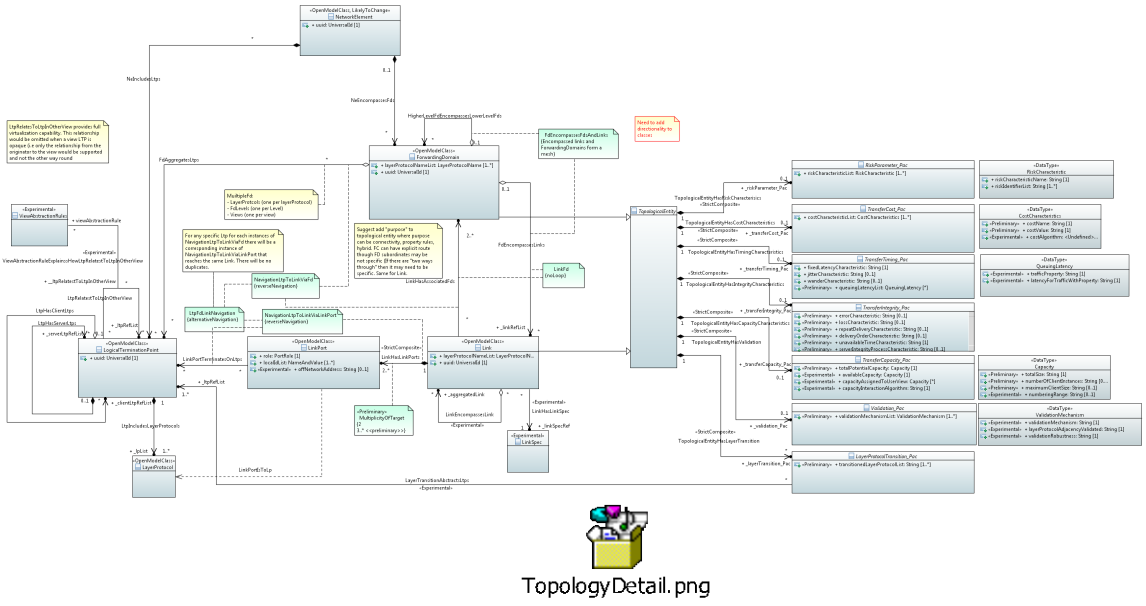


Figure 6-36 Topology detail

CoreModel diagram: TopologyDetail

The figure above shows finalized, preliminary and experimental extensions of the Topology model. The model recognizes that both ForwardingDomain and Link share topological properties (the TopologicalEntity, which is abstract and hence not intended to be instantiated, provides the

linkage[37]). The classes related to TopologicalEntity, the _Pacs, are «StrictComposite » and hence are essentially part of the ForwardingDomain and of the Link. The _Pacs are optional as in some cases of Link/ForwardingDomain they are not relevant.

The figure below shows the _Pacs in more detail.

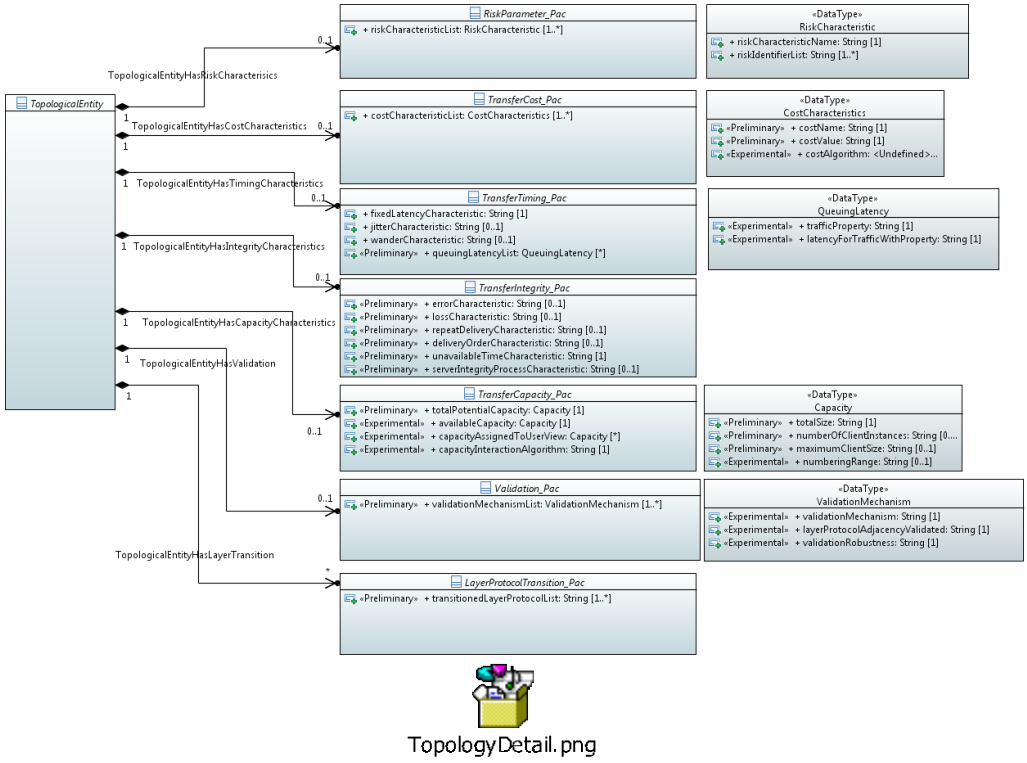


Figure 6-37 Topology _Pac detail

CoreModel diagram: TopologicalEntity-DetailedProperties

As shown in the figure, an object class "TopologicalEntity" has been defined to collect topology-related properties (characteristics, etc.) that are common for FD and Link.

Both the TopologicalEntity and the _Pacs are detailed in section 9.1 Core Network Module data dictionaryon page 67.

Note that a number of areas are still under development (highlighted using Experimental or Preliminary).

[37] TopologicalEntity has no direct attributes and only relationships that the ForwardingDomain and Link inherit.

6.6 Directionality

The model supports bidirectional, unidirectional and mixed directionality constructs. The following figure shows the directionality attributes and data types.

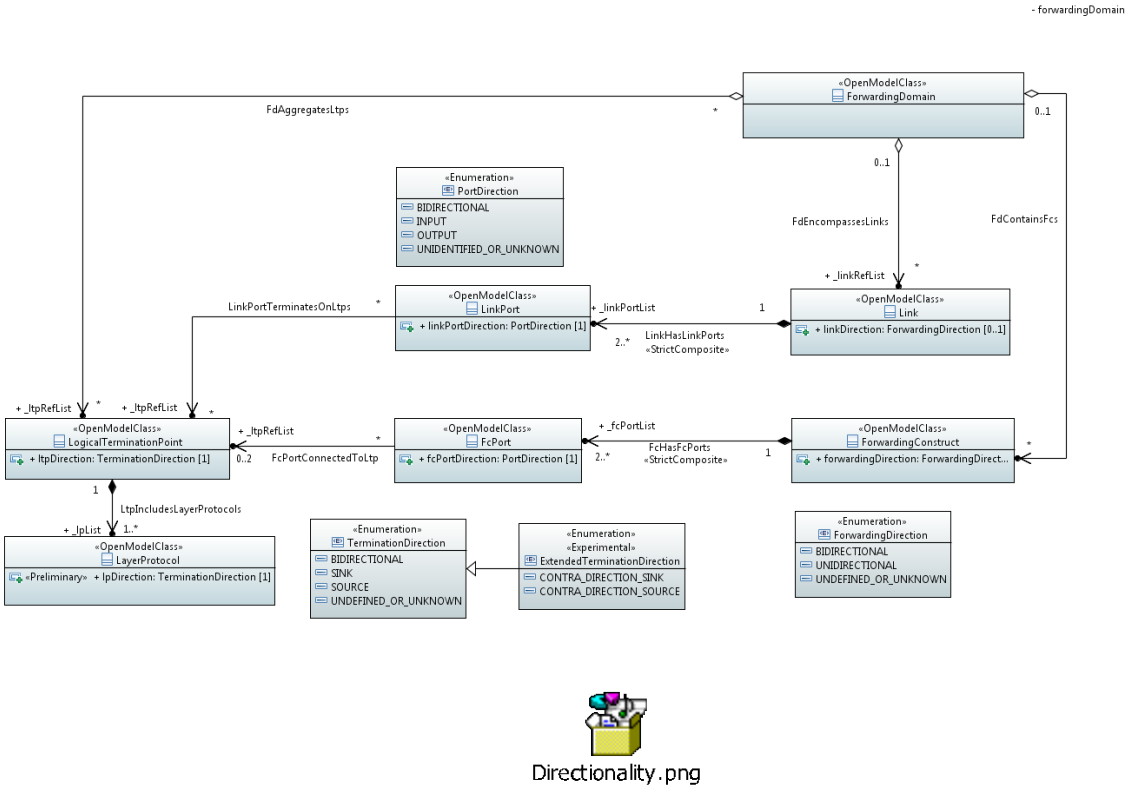


Figure 6-38 Model highlighting directionality

CoreModel diagram: ForwardingConnectivityFragmentWithLtAndDirection

The following figure shows in pictorial form the meaning of the key direction attributes in the model.

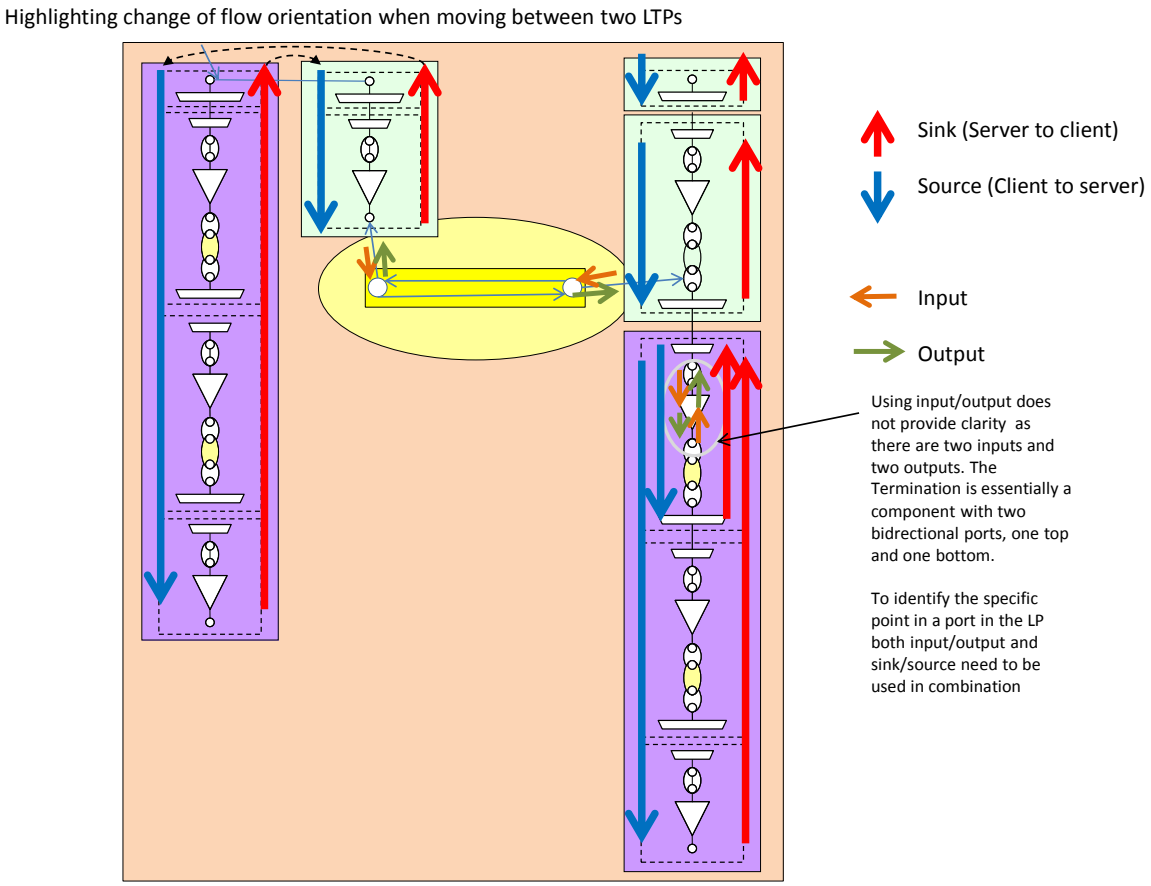


Figure 6-39 Interpreting the direction attributes

The figure above shows bidirectional LTPs and an FC in an NE context. It should be noted that the terms Sink and Source are consistent with Input and Output at the base of the LTP/LP (but counterintuitive at the top of the LTP/LP (where a Sink outputs signal). The specific terminology is aligned to that used in ITU-T. Sink/Source are defined in terms of "flow orientation" in the layer stack (i.e. client to server or server to client).

There are a number of legal combinations of bidirectional and unidirectional LTPs and FCs. The following sequence of figures provides an overview.

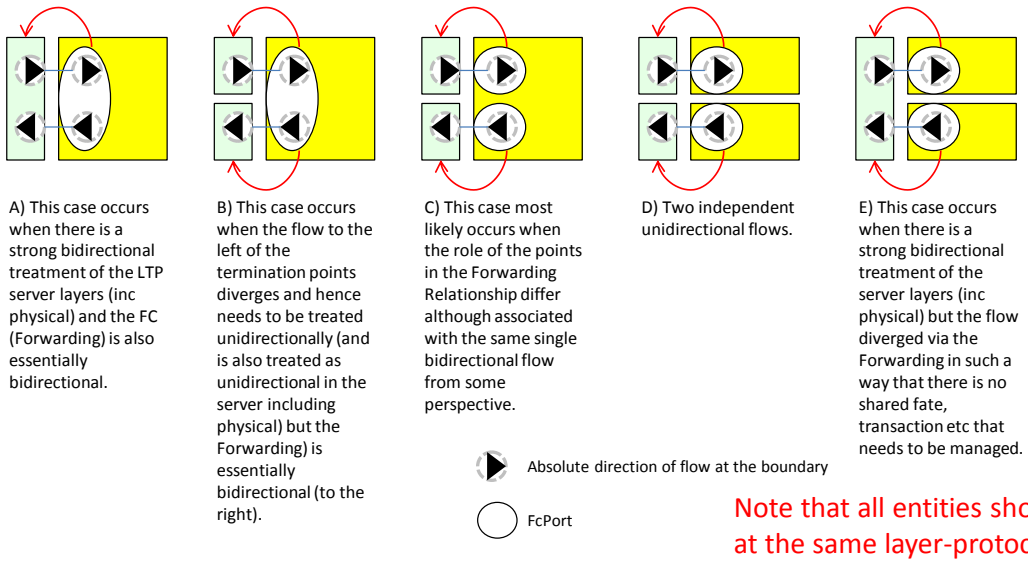


Figure 6-40 Various mixed directionality forms

The following figure shows how to relate two unidirectional LTPs to a single FC where the two LTPs are intended to carry the same traffic. The pattern also applies to bidirectional LTPs and FCs.

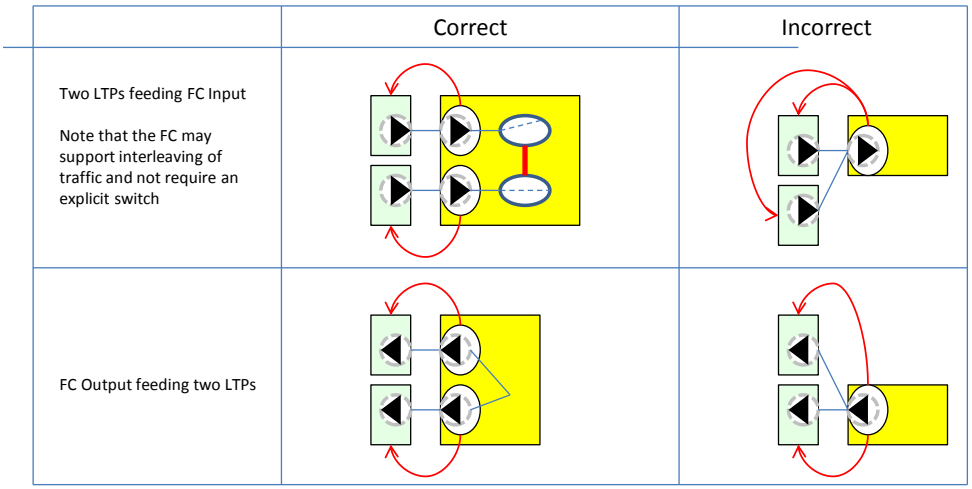


Figure 6-41 Interrelationship between a pair of unidirectional LTPs and a unidirectional FC

The following figure shows how to relate two unidirectional FCs to a single LTP where the two FCs are intended to carry the same traffic. The pattern also applies to bidirectional LTPs and FCs.

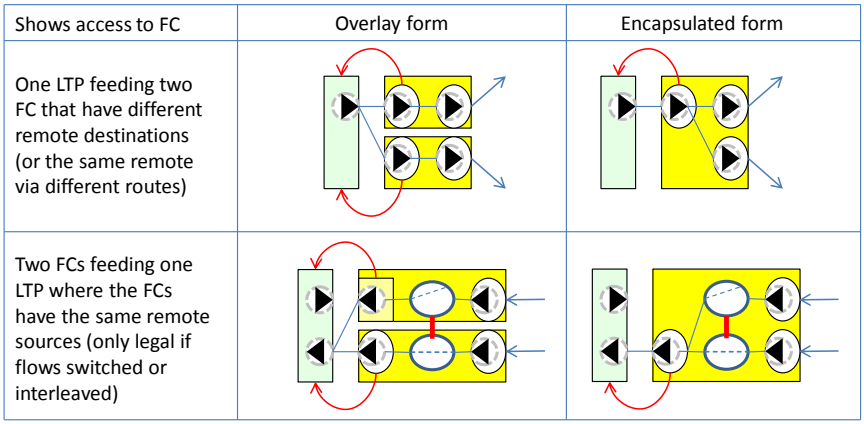


Figure 6-42 Interrelationship between a pair of unidirectional FCs and a single LTP

In some network cases the LP encapsulates several terminations functions with the same essential orientation of flow. The figure below shows a case with non-intrusive monitoring in an LTP (green). In that LTP, the two cases of Sink flow are distinguished by recognizing that one is in the normal orientation (red flow) with respect to standard traffic flow, i.e. the signal passed from the Server LTP is further terminated, whereas the other is in a non-normal orientation, i.e. the signal that would be expected to be encoded by (multiplexed etc.) by the server LTP is actually terminated (blue going to brown flow). The non-normal orientation is called ContraSink.

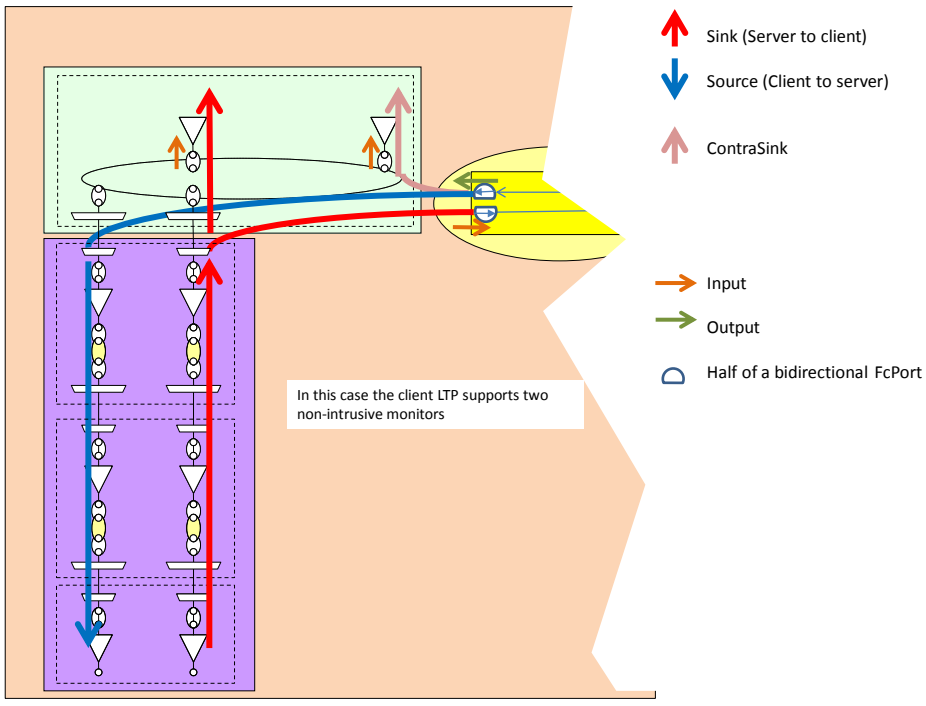


Figure 6-43 Contra-directionality showing monitors

The same logic applies to the Source terminations as depicted in the following figure where the LTP has both non-intrusive monitoring (as in the previous figure) and the potential for active test signal injection in an LTP (green)

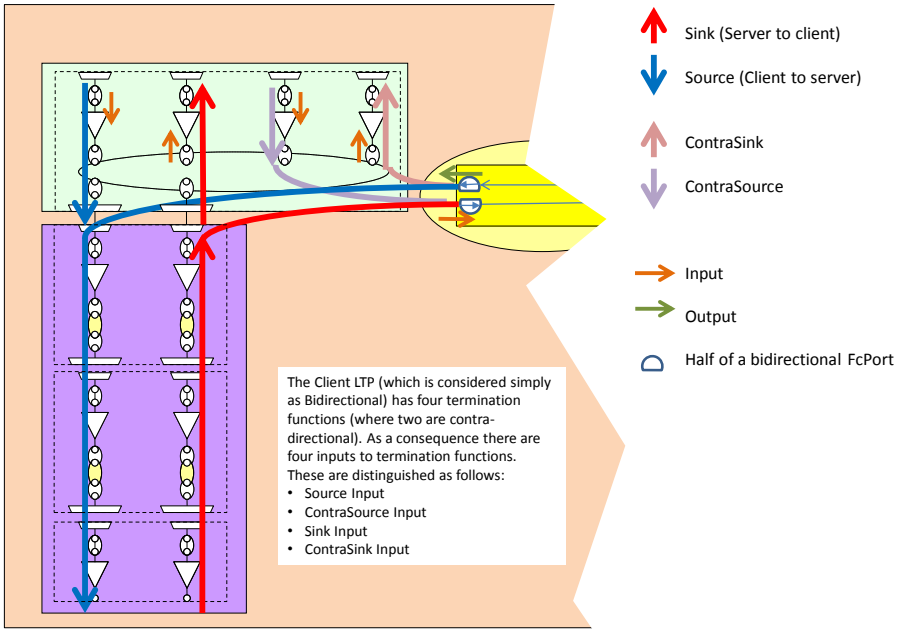


Figure 6-44 Contra-directionality showing monitors and signal sources

In the above two cases the LTPs/LPs are bidirectional. In the following example where there is a deep inspection capability, although the LTPs are bidirectional, the upper LP of the green LTP is unidirectional Sink. This illustrates one case where an LTP directionality is different from the directionality of an included LP.

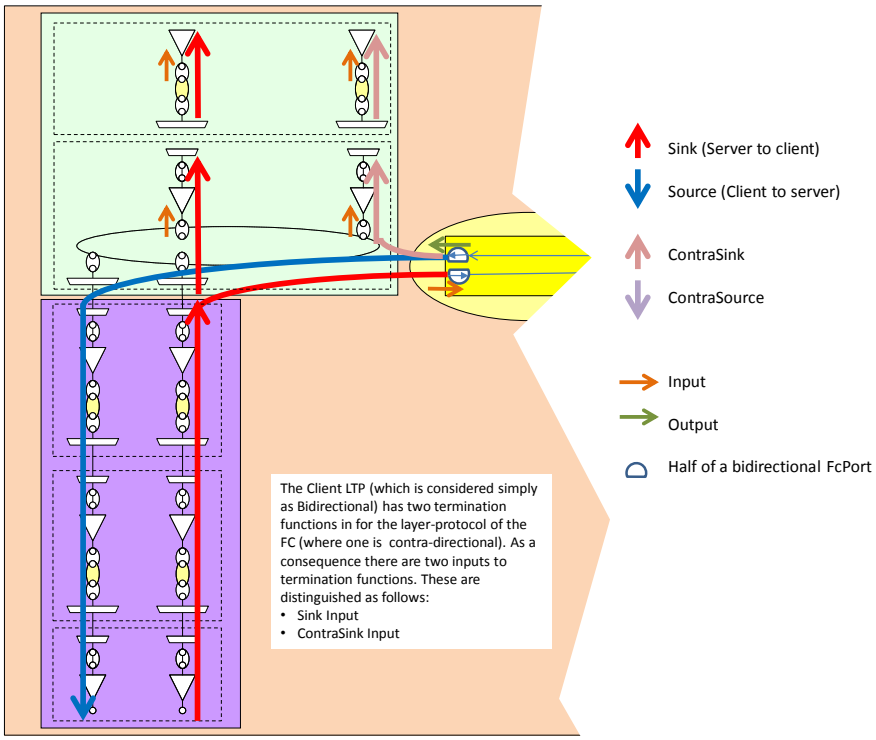


Figure 6-45 Contra-directionality showing deep inspection

7 Future CoreModel areas

Potential future areas of work in the CoreModel include

- Profiles, Templates and Specifications Module
  - Including LTP spec
- Policy Module
- Management-Control Component Module
  - Controller model (including NE, NCD, Controller component, Network core control, etc.)
- ECC Subset
- Interface patterns including
  - Notification model
- Generalized OAM functions e.g., generalized MEPs
- Assurance Module
- Dependency graph
- Consider ITU-T work on Timing and Synchronization model (frequency and time/phase) Subset
- Physical Equipment Module
- General processing construct model (and the component system-pattern)

- Understand the pattern underlying Link/FC/FD and minimally represent that pattern and show derivation of Link/FC/FD from that pattern.
- Link and Topology enhancements
- Further clarification of off-network "things"(could be link topology)
- Enhancements to view abstraction examples and cases, including FD view, FC view, Call view, Service view, Connection view
- Enhanced mapping to OpenFlow
- Support for the Intent model, NBI model work in general, TAPI and OTWG
- Tooling enhancement including more YANG generator work, JSON generator and support for pruning and refactoring process
- Improvements to documentation
- Rename the LayerProtocol class.

8 UML model files

8.1 Papyrus File

This section provides the link to the information model file and the companion Open Model Profile file specified using the "Papyrus" modeling tool and also some model sketches to assist the understanding of the model.

Link of the Core Model files: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/CoreInformationModel.V1.0.zip.

The file is structured as follows:

- SupportingDocuments: Consists of two informational documents that may assist when using the model and that provide a view of work in progress
  - These documents may have internal inconsistencies
- OnfModel-CoreModel: consists of four files:
  - .project,
  - CoreModel.di,
  - CoreModel.notation
  - CoreModel.uml
- OpenModelProfile: Consists of four files:
-
  - .project
  - OpenModel_Profile.profile.di
  - OpenModel_Profile.profile.notation
  - OpenModel_Profile.profile.uml

In order to view and further extend or modify the information model, one will need to install the open source Eclipse software and the Papyrus tool. The installation guide for Eclipse and Papyrus can be found at https://www.eclipse.org/papyrus/updates/index.php.

9 Data Dictionary

The data dictionary is divided up into four sections based upon the division of the CoreModel and maturity of work. The first three sections provide key information about the CoreNetworkModule (which includes Topology), the CoreFoundationModule and the enhancements that have been developed to a reasonable degree (from the CoreModelEnhancements package). The final section provides sketch fragment enhancements that have only been partially developed (from the CoreModelEnhancements package).

9.1 Core Network Module data dictionary

9.1.1 Classes

This section provides a view of the Classes of the Core Network Module. It shows all attributes for each class including those that are inherited. Inherited attributes are highlighted and are duplicates of attributes identified in other parts of the data dictionary.

9.1.1.1 FcPort

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::FcPort

The association of the FC to LTPs is made via FcPorts. The FcPort object class models the access to the FC function. The traffic forwarding between the associated FcPorts of the FC depends upon the type of FC and may be associated with FcSwitch object instances. In cases where there is resilience the FcPort may convey the resilience role of the access to the FC. It can represent a protected (resilient/reliable) point or a protecting (unreliable working or protection) point. The FcPort replaces the Protection Unit of a traditional protection model. The ForwadingConstruct can be considered as a component and the FcPort as a Port on that component

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 2: Attributes for FcPort

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description

_ltpRefList	LogicalTerminationPoint	0..2	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The FcPort may be associated with more than one LTP when the FcPort is bidirectional and the LTPs are unidirectional. Multiple Ltp - Bidirectional FcPort to two Uni Ltps Zero Ltp - BreakBeforeMake transition - Planned Ltp not yet in place - Off-network LTP referenced through other mechanism
role	PortRole	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Each FcPort of the FC has a role (e.g., working, protection, protected, symmetric, hub, spoke, leaf, root) in the context of the FC with respect to the FC function.
fcPortDirection	PortDirection	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The orientation of defined flow at the FcPort.
localIdList Inherited	NameAndValue	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An identifier that is unique in the context of some scope that is less than the global scope. (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself unique, and immutable. The identifier therefore represents the identity of the entity/role. An identifier carries no semantics with respect to the purpose of the entity.)
nameList Inherited	NameAndValue	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of names.
labelList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of labels.
extensionList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of simple name-value extentions
operationalState Inherited	OperationalState	0..1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The operational state is used to indicate whether or not the resource is installed and working

administrativeControl Inherited	AdministrativeControl	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The administrativeControl state provides control of the availability of specific resources without modification to the provisioning of those resources. The value is the current control target. The actual administrativeState may or may not be at target.
adminsatratveState Inherited	AdministrativeState	1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Shows whether or not the client has permission to use or has a prohibition against using the resource. The administrative state expresses usage permissions for specific resources without modification to the provisioning of those resources.
lifecycleState Inherited	LifecycleState	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Used to track the planned deployment, allocation to clients and withdrawal of resources.

9.1.1.2 FcRoute

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::FcRoute

The FC Route (FcRoute) object class models the individual routes of an FC. The route is an alternative view of the internal structure of the FC. The route of an FC object is represented by a list of FCs at a lower level with the implicit understanding that unmodelled link connections are interleaved between the lower level FCs. Note that depending on the service supported by an FC, an the FC can have multiple routes. Also applicable where NE level ForwardingDomain may be decomposed into subordinate ForwardingDomains. Applies to both virtual and real NE cases.

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 3: Attributes for FcRoute

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description

_fcList	ForwardingConstruct	2..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The list of FCs describing the route of an FC.
localIdList Inherited	NameAndValue	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An identifier that is unique in the context of some scope that is less than the global scope. (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself unique, and immutable. The identifier therefore represents the identity of the entity/role. An identifier carries no semantics with respect to the purpose of the entity.)
nameList Inherited	NameAndValue	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of names.
labelList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of labels.
extensionList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of simple name-value extentions
operationalState Inherited	OperationalState	0..1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The operational state is used to indicate whether or not the resource is installed and working
administrativeControl Inherited	AdministrativeControl	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The administrativeControl state provides control of the availability of specific resources without modification to the provisioning of those resources. The value is the current control target. The actual administrativeState may or may not be at target.

adminsatratveState Inherited	AdministrativeState	1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Shows whether or not the client has permission to use or has a prohibition against using the resource. The administrative state expresses usage permissions for specific resources without modification to the provisioning of those resources.
lifecycleState Inherited	LifecycleState	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Used to track the planned deployment, allocation to clients and withdrawal of resources.

9.1.1.3 FcSwitch

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::FcSwitch

The FcSwitch object class models the switched forwarding of traffic (traffic flow) between FcPorts and is present where there is protection functionality in the FC. If an FC exposes protection (having two FcPorts that provide alternative identical inputs/outputs), the FC will have one or more associated FcSwitch objects to represent the alternative flow choices visible at the edge of the FC. The FC switch represents and defines a protection switch structure encapsulated in the FC. Essentially performs the functoion of the Protection Group in a traditional model. Associates to 2 or more FcPorts each playing the role of a Protection Unit. One or more protection FcPorts (standby/backup) provide protection for one or more working (i.e. regular/main/preferred) FcPorts where eith protection or working can feed one or more protected FcPort. May be used in revertive or non-revertive (symmetric) mode. When in revertive mode may define waitToRestore time. May be used in one of several modes including source switch, destination switched, source and destination switched etc (covering cases such as 1+1 ane 1:1).. May be lockout (prevented from switching), force switched or manual switched. Will indicate switch state and change of state.

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 4: Attributes for FcSwitch

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description

holdOffTime	Integer	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	This attribute indicates the time, in seconds, between declaration of unacceptable quality of signal on the currently selected FcPort, and the initialization of the protection switching algorithm.
waitToRestoreTime	Integer	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	If the protection system is revertive, this attribute specifies the amount of time, in seconds, to wait after the preferred FcPort returns to an acceptable state of operaion (e.g a fault has cleared) before restoring traffic to that preferred FcPort.
protType	ProtectionType	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Indicates the protection scheme that is used for the ProtectionGroup.
operType	OperType	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	This attribute whether or not the protection scheme is revertive or non-revertive.
_selectedFcPortRefList	FcPort	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Indicates which points are selected by the switch.
_profileProxyRefList	ProfileProxy	0..*	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Provides a set of predefined values for switch control in place of the direct values avaiable via the FcSwitch or via _configurationAndSwitchControl
_configurationAndSwitchControlRef	ConfigurationAndSwitchController	0..1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	A multi-switch controller external to the FcSwitch. The multi-switch controller coordinates multiple switches in the same FC or across multple FCs
_configurationAndSwitchControl	ConfigurationAndSwitchController	0..1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	A switch controller encapsulated in the FcSwitch.

localIdList Inherited	NameAndValue	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An identifier that is unique in the context of some scope that is less than the global scope. (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself unique, and immutable. The identifier therefore represents the identity of the entity/role. An identifier carries no semantics with respect to the purpose of the entity.)
nameList Inherited	NameAndValue	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of names.
labelList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of labels.
extensionList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of simple name-value extentions
operationalState Inherited	OperationalState	0..1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The operational state is used to indicate whether or not the resource is installed and working
administrativeControl Inherited	AdministrativeControl	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The administrativeControl state provides control of the availability of specific resources without modification to the provisioning of those resources. The value is the current control target. The actual administrativeState may or may not be at target.
adminsatratveState Inherited	AdministrativeState	1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Shows whether or not the client has permission to use or has a prohibition against using the resource. The administrative state expresses usage permissions for specific resources without modification to the provisioning of those resources.

lifecycleState Inherited	LifecycleState	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Used to track the planned deployment, allocation to clients and withdrawal of resources.

9.1.1.4 ForwardingConstruct

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::ForwardingConstruct

The ForwardingConstruct (FC) object class models enabled potential for forwarding between two or more LTPs at a particular specific layerProtocol. Like the LTP the FC supports any transport protocol including all circuit and packet forms. It is used to effect forwarding of transport characteristic (layer protocol) information. An FC can be in only one FD. The ForwardingConstruct is a Forwarding entity. At a low level of the recursion, a FC represents a cross-connection within an NE. It may also represent a fragment of a cross-connection under certain circumstances. The FC object can be used to represent many different structures including point-to-point (P2P), point-to-multipoint (P2MP), rooted-multipoint (RMP) and multipoint-to-multipoint (MP2MP) bridge and selector structure for linear, ring or mesh protection schemes.

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 5: Attributes for ForwardingConstruct

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
layerProtocolName	LayerProtocolName	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The layerProtocol at which the FC enables potential for forwarding.
_lowerLevelFcRefList	ForwardingConstruct	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An FC object supports a recursive aggregation relationship such that the internal construction of an FC can be exposed as multiple lower level FC objects (partitioning). Aggregation is used as for the FD to allow changes in hierarchy.

_fcRouteRefList	FcRoute	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An FC object can have zero or more routes, each of which is defined as a list of lower level FC objects describing the flow across the network.
_fcPortList	FcPort	2..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The association of the FC to LTPs is made via FcPorts (essentially the ports of the FC).
_fcSwitchList	FcSwitch	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	If an FC exposes protection (having two FcPorts that provide alternative identical inputs/outputs), the FC will have one or more associated FcSwitch objects. The arrangement of switches for a particular instance is described by a referenced FcSpec
_configurationAndSwitchControlList	ConfigurationAndSwitchController	0..*	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	A multi-switch controller encapsulated in the FC. The multi-switch controller coordinates multiple switches in the same FC.
_fcSpecRef	FcSpec	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	References the specification that describes the capability and internal structure of of the FC (e.g. The arrangement of switches for a particular instance is described by a referenced FcSpec). The specification allows interpretation of FcPort role and switch configurations etc.
forwardingDirection	ForwardingDirection	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The directionality of the ForwardingConstruct. Is applicable to simple ForwardingConstructs where all FcPorts are BIDIRECTIONAL (the ForwardingConstruct will be BIDIRECTIONAL) or UNIDIRECTIONAL (the ForwardingConstruct will be UNIDIRECTIONAL). Is not present in more complex cases.

localIdList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An identifier that is unique in the context of some scope that is less than the global scope. (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself unique, and immutable. The identifier therefore represents the identity of the entity/role. An identifier carries no semantics with respect to the purpose of the entity.)
uuid Inherited	UniversalId	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	UUID: An identifier that is universally unique (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself globally unique, and immutable. An identifier carries no semantics with respect to the purpose or state of the entity)
nameList Inherited	NameAndValue	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of names.
labelList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of labels.
extensionList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of simple name-value extentions
operationalState Inherited	OperationalState	0..1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The operational state is used to indicate whether or not the resource is installed and working
administrativeControl Inherited	AdministrativeControl	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The administrativeControl state provides control of the availability of specific resources without modification to the provisioning of those resources. The value is the current control target. The actual administrativeState may or may not be at target.

adminsatratveState Inherited	AdministrativeState	1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Shows whether or not the client has permission to use or has a prohibition against using the resource. The administrative state expresses usage permissions for specific resources without modification to the provisioning of those resources.
lifecycleState Inherited	LifecycleState	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Used to track the planned deployment, allocation to clients and withdrawal of resources.

9.1.1.5 ForwardingDomain

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::ForwardingDomain

The ForwardingDomain (FD) object class models the topological component which represents the opportunity to enable forwarding (of specific transport characteristic information at one or more protocol layers) between points represented by the LTP in the model. The FD object provides the context for instructing the formation, adjustment and removal of FCs and hence offers the potential to enable forwarding. The LTPs available are those defined at the boundary of the FD. At a lowere level of recursion, an FD (within a network element (NE)) represents a switch matrix (i.e., a fabric). Note that an NE can encompass multiple switch matrices (FDs) and the FD representing the switch matrix can be further partitioned.

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 6: Attributes for ForwardingDomain

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
layerProtocolNameList	LayerProtocolName	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	One or more protocol layers at which the FD represents the opportunity to enable forwarding between LTP that bound it.

_lowerLevelFdRefList	ForwardingDomain	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The FD object class supports a recursive aggregation relationship (HigherLevelFdEncompassesLowerLevelFds) such that the internal construction of an FD can be exposed as multiple lower level FDs and associated Links (partitioning). The aggregated FDs and Links form an interconnected topology that provides and describes the capability of the aggregating FD. Note that the model actually represents aggregation of lower level FDs into higher level FDs as views rather than FD partition, and supports multiple views. Aggregation allow reallocation of capacity from lower level FDs to different higher level FDs as if the network is reorganized (as the association is aggregation not composition).
_fcRefList	ForwardingConstruct	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An FD contains one or more FCs. A contained FC connects LTPs that bound the FD.
_ltpRefList	LogicalTerminationPoint	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An instance of FD is associated with zero or more LTP objects. The LTPs that bound the FD provide capacity for forwarding.
_linkRefList	Link	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The FD encompasses Links that interconnect lower level FDs and collect links that are wholly within the bounds of the FD. See also _lowerLevelFdRefList.
localIdList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An identifier that is unique in the context of some scope that is less than the global scope. (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself unique, and immutable. The identifier therefore represents the identity of the entity/role. An identifier carries no semantics with respect to the purpose of the entity.)

uuid Inherited	UniversalId	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	UUID: An identifier that is universally unique (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself globally unique, and immutable. An identifier carries no semantics with respect to the purpose or state of the entity)
nameList Inherited	NameAndValue	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of names.
labelList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of labels.
extensionList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of simple name-value extentions
operationalState Inherited	OperationalState	0..1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The operational state is used to indicate whether or not the resource is installed and working
administrativeControl Inherited	AdministrativeControl	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The administrativeControl state provides control of the availability of specific resources without modification to the provisioning of those resources. The value is the current control target. The actual administrativeState may or may not be at target.
adminsatratveState Inherited	AdministrativeState	1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Shows whether or not the client has permission to use or has a prohibition against using the resource. The administrative state expresses usage permissions for specific resources without modification to the provisioning of those resources.

lifecycleState Inherited	LifecycleState	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Used to track the planned deployment, allocation to clients and withdrawal of resources.
_riskParameter_Pac Inherited	RiskParameter_Pac	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if risk information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if risk is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made.	
_transferCost_Pac Inherited	TransferCost_Pac	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if cost information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if cost is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made.	
_transferTiming_Pac Inherited	TransferTiming_Pac	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if transfer timing information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if transfer timing is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made.	

_transferCapacity_Pac Inherited	TransferCapacity_Pac	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if transfer capacity information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if transfer capacity is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made.	
_transferIntegrity_Pac Inherited	TransferIntegrity_Pac	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: OPTIONAL • condition:Present if transfer integrity information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if transfer integrity is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made.	
_validation_Pac Inherited	Validation_Pac	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: OPTIONAL • condition:Present if validation information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if validation is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made. Note that validation may not be possible for the specific layer protocol or in the particular case.	

_layerTransition_Pac Inherited	LayerProtocolTransition_Pac	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if layer transition information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if layer transiotio is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made. Note that layer transition occurs in a limited number of cases.	

9.1.1.6 LayerProtocol

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::LayerProtocol

Each transport layer is represented by a LayerProtocol (LP) instance. The LayerProtocol instances it can be used for controlling termination and monitoring functionality. It can also be used for controlling the adaptation (i.e. encapsulation and/or multiplexing of client signal), tandem connection monitoring, traffic conditioning and/or shaping functionality at an intermediate point along a connection. Where the client – server relationship is fixed 1:1 and immutable, the layers can be encapsulated in a single LTP instance. Where the is a n:1 relationship between client and server, the layers must be split over two separate instances of LTP.

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 7: Attributes for LayerProtocol

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
layerProtocolName	LayerProtocolName	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Indicate the specific layer-protocol described by the LayerProtocol entity.

_lpSpec	LpSpec	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The LpSpec identifies the interna structure of the LP explaining internal flexibilities, degree of termination and degree of adaptation on both client and server side.
configuredClientCapacity	invalid	0..1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Provides a summarized view of the client capacity that is configurable for use. Note the cleint LTP association should provide all necessary detail hence this attribute is questionable.
lpDirection	TerminationDirection	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The overall directionality of the LP. - A BIDIRECTIONAL LP will have some SINK and/or SOURCE flowss. - A SINK LP can only contain elements with SINK flows or CONTRA_DIRECTION_SOURCE flows - A SOURCE LP can only contain SOURCE flows or CONTRA_DIRECTION_SINK flows
terminationState	invalid	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Indicates whether the layer is terminated and if so how.
localIdList Inherited	NameAndValue	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An identifier that is unique in the context of some scope that is less than the global scope. (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself unique, and immutable. The identifier therefore represents the identity of the entity/role. An identifier carries no semantics with respect to the purpose of the entity.)
nameList Inherited	NameAndValue	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of names.
labelList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of labels.

extensionList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of simple name-value extentions
operationalState Inherited	OperationalState	0..1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The operational state is used to indicate whether or not the resource is installed and working
administrativeControl Inherited	AdministrativeControl	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The administrativeControl state provides control of the availability of specific resources without modification to the provisioning of those resources. The value is the current control target. The actual administrativeState may or may not be at target.
adminsatratveState Inherited	AdministrativeState	1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Shows whether or not the client has permission to use or has a prohibition against using the resource. The administrative state expresses usage permissions for specific resources without modification to the provisioning of those resources.
lifecycleState Inherited	LifecycleState	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Used to track the planned deployment, allocation to clients and withdrawal of resources.

9.1.1.7 LayerProtocolTransition_Pac

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::TopologyPacs::LayerProtocolTransition_Pac

Relevant for a Link that is formed by abstracting one or more LTPs (in a stack) to focus on the flow and deemphasize the protocol transformation. This abstraction is relevant when considering multi-layer routing. The layer protocols of the LTP and the order of their application to the signal is still relevant and need to be accounted for. This is derived from the LTP spec details. This Pac provides the relevant abstractions of the LTPs and provides the necessary association to the LTPs involved. Links that included details in this Pac are often referred to as Transitional Links.

Applied stereotypes:

- OpenModelClass

- objectCreationNotification: NA
- objectDeletionNotification: NA
- support: MANDATORY

Table 8: Attributes for LayerProtocolTransition_Pac

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
transitionedLayerProtocolList	String	1..*	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Provides the ordered structure of layer protocol transitions encapsulated in the TopologicalEntity. The ordering relates to the LinkPort role.
_ltpRefList	LogicalTerminationPoint	1..*	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Lists the LTPs that define the layer protocol transition of the transitional link.

9.1.1.8 Link

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::Link

The Link object class models effective adjacency between two or more ForwardingDomains (FD). In its basic form (i.e., point-to-point Link) it associates a set of LTP clients on one FD with an equivalent set of LTP clients on another FD. Like the FC, the Link has ports (LinkPort) which take roles relevant to the constraints on flows offered by the Link (e.g., Root role or leaf role for a Link that has a constrained Tree configuration). The Link is a Forwarding entity.

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 9: Attributes for Link

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description

layerProtocolNameList	LayerProtocolName	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The Link can support multiple transport layer protocols via the associated LTP object. For implementation optimization, where appropriate, multiple layer-specific links can be merged and represented as a single Link instance as the Link can represent a list of layer protocols. A link may support different layer protocols at each Port when it is a transitional link.
_fdRefList	ForwardingDomain	2..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The Link associates with two or more FDs. This association provides a direct summarization of the association via LinkPort and LTP.
_linkPortList	LinkPort	2..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The association of the Link to LTPs is made via LinkPort (essentially the ports of the Link).
_linkSpecRef	LinkSpec	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	References the specification that describes the capability and internal structure of of the Link (e.g. asymmetric flows between points). The specification allows interpretation of LinkPort role and switch configurations etc. See also ForwardingConstruct.
_aggregatedLink	Link	0..*	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	A link may formed from subordinate links (similar FD formations from subordiate FDs). This association is intended to cover concepts such as serial compound links.
linkDirection	ForwardingDirection	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The directionality of the Link. Is applicable to simple Links where all LinkPorts are BIDIRECTIONAL (the Link will be BIDIRECTIONAL) or UNIDIRECTIONAL (the Link will be UNIDIRECTIONAL). Is not present in more complex cases.

localIdList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An identifier that is unique in the context of some scope that is less than the global scope. (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself unique, and immutable. The identifier therefore represents the identity of the entity/role. An identifier carries no semantics with respect to the purpose of the entity.)
uuid Inherited	UniversalId	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	UUID: An identifier that is universally unique (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself globally unique, and immutable. An identifier carries no semantics with respect to the purpose or state of the entity)
nameList Inherited	NameAndValue	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of names.
labelList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of labels.
extensionList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of simple name-value extentions
operationalState Inherited	OperationalState	0..1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The operational state is used to indicate whether or not the resource is installed and working
administrativeControl Inherited	AdministrativeControl	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The administrativeControl state provides control of the availability of specific resources without modification to the provisioning of those resources. The value is the current control target. The actual administrativeState may or may not be at target.

adminsatratveState Inherited	AdministrativeState	1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Shows whether or not the client has permission to use or has a prohibition against using the resource. The administrative state expresses usage permissions for specific resources without modification to the provisioning of those resources.
lifecycleState Inherited	LifecycleState	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Used to track the planned deployment, allocation to clients and withdrawal of resources.
_riskParameter_Pac Inherited	RiskParameter_Pac	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if risk information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if risk is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made.	
_transferCost_Pac Inherited	TransferCost_Pac	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if cost information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if cost is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made.	

_transferTiming_Pac Inherited	TransferTiming_Pac	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if transfer timing information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if transfer timing is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made.	
_transferCapacity_Pac Inherited	TransferCapacity_Pac	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if transfer capacity information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if transfer capacity is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made.	
_transferIntegrity_Pac Inherited	TransferIntegrity_Pac	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: OPTIONAL • condition:Present if transfer integrity information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if transfer integrity is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made.	

_validation_Pac Inherited	Validation_Pac	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: OPTIONAL • condition:Present if validation information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if validation is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made. Note that validation may not be possible for the specific layer protocol or in the particular case.	
_layerTransition_Pac Inherited	LayerProtocolTransition_Pac	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if layer transition information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if layer transiotio is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made. Note that layer transition occurs in a limited number of cases.	

9.1.1.9 LinkPort

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::LinkPort

The association of the Link to LTPs is made via LinkPort. The LinkPort object class models the access to the Link function. The traffic forwarding between the associated LinkPorts of the Link depends upon the type of Link. In cases where there is resilience the LinkPort may convey the resilience role of the access to the Link. The Link can be considered as a component and the LinkPort as a Port on that component

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA

- support: MANDATORY

Table 10: Attributes for LinkPort

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
_ltpp	invalid	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY Obsolete	OBSOLETE. Was reference to LtpPool. The pool has now been subsumed into the LTP. This will be deleted in the next release.
_ltpRefList	LogicalTerminationPoint	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The LinkPort may be associated with more than one LTP when the LinkPort is bidirectional and the LTPs are unidirectional. Multiple Ltp - Bidirectional LinkPort to two Uni Ltps Zero Ltp - BreakBeforeMake transition - Planned Ltp not yet in place - Off-network LTP referenced through other mechanism
role	PortRole	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Each LinkPort of the Link has a role (e.g., symmetric, hub, spoke, leaf, root) in the context of the Link with respect to the Link function.
offNetworkAddress	String	0..1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	A freeform opportunity to express a reference for an Port of the Link that is not outside the scope of the control domain. This attribute is expected to convey a foreign identifier/name/address or a shared reference for some mid-span point at the boundary between two administrative domains. This attribute is used when an LTP cannot be referenced.
linkPortDirection	PortDirection	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The orientation of defined flow at the LinkPort.

localIdList Inherited	NameAndValue	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An identifier that is unique in the context of some scope that is less than the global scope. (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself unique, and immutable. The identifier therefore represents the identity of the entity/role. An identifier carries no semantics with respect to the purpose of the entity.)
nameList Inherited	NameAndValue	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of names.
labelList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of labels.
extensionList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of simple name-value extentions
operationalState Inherited	OperationalState	0..1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The operational state is used to indicate whether or not the resource is installed and working
administrativeControl Inherited	AdministrativeControl	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The administrativeControl state provides control of the availability of specific resources without modification to the provisioning of those resources. The value is the current control target. The actual administrativeState may or may not be at target.
adminsatratveState Inherited	AdministrativeState	1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Shows whether or not the client has permission to use or has a prohibition against using the resource. The administrative state expresses usage permissions for specific resources without modification to the provisioning of those resources.

lifecycleState Inherited	LifecycleState	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Used to track the planned deployment, allocation to clients and withdrawal of resources.

9.1.1.10 LogicalTerminationPoint

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::LogicalTerminationPoint

The LogicalTerminationPoint (LTP) object class encapsulates the termination and adaptation functions of one or more transport layers. The structure of LTP supports all transport protocols including circuit and packet forms.

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 11: Attributes for LogicalTerminationPoint

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
_serverLtpRefList	LogicalTerminationPoint	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	References contained LTPs representing servers of this LTP in an inverse multiplexing configuration (e.g. VCAT).
_clientLtpRefList	LogicalTerminationPoint	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	References contained LTPs representing client traffic of this LTP for normal cases of multiplexing.
_lpList	LayerProtocol	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Ordered list of LayerProtocols that this LTP is comprised of where the first entry in the list is the lowest server layer (e.g. physical)
_connectedLtpRef	LogicalTerminationPoint	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Applicable in a simple context where two LTPs are associated via a non-adjustable enabled forwarding. Reduces clutter removing the need for two additional LTPs and an FC with a pair of FcPorts.

_peerLtpRef	LogicalTerminationPoint	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	References contained LTPs representing the reversal of orientation of flow where two LTPs are associated via a non-adjustable enabled forwarding and where the referenced LTP is fully dependent on the this LTP.
_ltpSpec	LtpSpec	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The specification of the LTP defines internal structure of the LTP. The specification allows interpretation of organisatoon of LPs making up the LTP and also identifies which inter-LTP associations are valid.
physicalPortReference	String	0..*	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	One or more text labels for the unmodelled physical port associated with the LTP. In many cases there is no associated physical port
_ltpRefList	LogicalTerminationPoint	0..*	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	References one or more LTPs in other views that represent this LTP. The referencing LTP is the rovider of capability.
ltpDirection	TerminationDirection	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The overall directionality of the LTP. - A BIDIRECTIONAL LTP must have at least some LPs that are BIDIRECTIONAL but may also have some SINK and/or SOURCE LPs. - A SINK LTP can only contain SINK LPs - A SOURCE LTP can only contain SOURCE LPs
localIdList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An identifier that is unique in the context of some scope that is less than the global scope. (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself unique, and immutable. The identifier therefore represents the identity of the entity/role. An identifier carries no semantics with respect to the purpose of the entity.)

uuid Inherited	UniversalId	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	UUID: An identifier that is universally unique (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself globally unique, and immutable. An identifier carries no semantics with respect to the purpose or state of the entity)
nameList Inherited	NameAndValue	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of names.
labelList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of labels.
extensionList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of simple name-value extentions
operationalState Inherited	OperationalState	0..1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The operational state is used to indicate whether or not the resource is installed and working
administrativeControl Inherited	AdministrativeControl	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The administrativeControl state provides control of the availability of specific resources without modification to the provisioning of those resources. The value is the current control target. The actual administrativeState may or may not be at target.
adminsatratveState Inherited	AdministrativeState	1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Shows whether or not the client has permission to use or has a prohibition against using the resource. The administrative state expresses usage permissions for specific resources without modification to the provisioning of those resources.

lifecycleState Inherited	LifecycleState	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Used to track the planned deployment, allocation to clients and withdrawal of resources.

9.1.1.11 NetworkControlDomain

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::NetworkControlDomain

The Network Control Domain (NCD) object class represents the scope of control that a particular SDN controller has with respect to a particular network, (i.e., encompassing a designated set of interconnected network elements). In the interfaces between SDN controllers where virtualization is necessary, e.g., in client/server SDN controller relationship, the NCD object defines the scope of control of the client SDN controller on the virtual network that has been provided by the server SDN controller (i.e., the scope of control relates to the partitioned provider resources allocated to that particular client). The NCD provides the scope of naming space for identifying objects representing the virtual resources within the virtual network.

Applied stereotypes:

- Preliminary
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 12: Attributes for NetworkControlDomain

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
_forwardingDomainRefList	ForwardingDomain	0..*	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The FDs accessible via the NCD.
_linkRefList	Link	0..*	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The links accessible in the scope of the NCD. The domain is bounded by off-network links.
_networkElementRefList	NetworkElement	0..*	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The network elements within the scope of the NCD where each NE is within one and only one domain.

localIdList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An identifier that is unique in the context of some scope that is less than the global scope. (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself unique, and immutable. The identifier therefore represents the identity of the entity/role. An identifier carries no semantics with respect to the purpose of the entity.)
uuid Inherited	UniversalId	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	UUID: An identifier that is universally unique (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself globally unique, and immutable. An identifier carries no semantics with respect to the purpose or state of the entity)
nameList Inherited	NameAndValue	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of names.
labelList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of labels.
extensionList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of simple name-value extentions
operationalState Inherited	OperationalState	0..1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The operational state is used to indicate whether or not the resource is installed and working
administrativeControl Inherited	AdministrativeControl	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The administrativeControl state provides control of the availability of specific resources without modification to the provisioning of those resources. The value is the current control target. The actual administrativeState may or may not be at target.

adminsatratveState Inherited	AdministrativeState	1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Shows whether or not the client has permission to use or has a prohibition against using the resource. The administrative state expresses usage permissions for specific resources without modification to the provisioning of those resources.
lifecycleState Inherited	LifecycleState	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Used to track the planned deployment, allocation to clients and withdrawal of resources.

9.1.1.12 NetworkElement

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::NetworkElement

The Network Element (NE) object class represents a network element (traditional NE) in the data plane. A data plane network element is essentially a consolidation of capabilities that can be viewed and controlled through a "single" management-control port. In the direct interface from an SDN controller to a network element in the data plane, the NetworkElement object defines the scope of control for the resources within the network element For example internal transfer of user information between the external terminations (ports of the NE), encapsulation, multiplexing/demultiplexing, and OAM functions, etc. The NetworkElement provides the scope of the naming space for identifying objects representing the resources within the data plane network element. NE is really a product bundling or some view of management scope, management access, session. The NE is not directly part of topology but brings meaning to the FD context and the LTP context (and hence the links).

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY
- LikelyToChange

Table 13: Attributes for NetworkElement

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description

_fdRefList	ForwardingDomain	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Represents the FD that is completely within the boundary of the NE. At a low level of recursion, an FD (within a network element (NE)) represents a switch matrix (i.e., a fabric). Note that an NE can encompass multiple switch matrices (FDs) and the FD representing the switch matrix can be further partitioned. Where an FD is referenced by the NeEncompassesFd association, any FDs that it encompasses (i.e., that are associated with it by HigherLevelFdEncompassesLowerLevelFds), must also be encompassed by the NE and hence must have the NeEncompassesFd association.
_ltppList	invalid	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY Obsolete	OBSOLETE. Was reference to LtpPool. The pool has now been subsumed into the LTP. This will be deleted in the next release.
_ltpRefList	LogicalTerminationPoint	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An NE has associated LTPs that are at its boundary. The NeEncompassesFd association occurs for FDs that are within the bounds of the NetworkElement definition such that the FD is bounded by LTPs, all of which are on the boundary of the NetworkElement or are within the NetworkElement. An LTP can be independent of an NE.
localIdList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An identifier that is unique in the context of some scope that is less than the global scope. (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself unique, and immutable. The identifier therefore represents the identity of the entity/role. An identifier carries no semantics with respect to the purpose of the entity.)

uuid Inherited	UniversalId	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	UUID: An identifier that is universally unique (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself globally unique, and immutable. An identifier carries no semantics with respect to the purpose or state of the entity)
nameList Inherited	NameAndValue	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of names.
labelList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of labels.
extensionList Inherited	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of simple name-value extentions
operationalState Inherited	OperationalState	0..1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The operational state is used to indicate whether or not the resource is installed and working
administrativeControl Inherited	AdministrativeControl	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The administrativeControl state provides control of the availability of specific resources without modification to the provisioning of those resources. The value is the current control target. The actual administrativeState may or may not be at target.
adminsatratveState Inherited	AdministrativeState	1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Shows whether or not the client has permission to use or has a prohibition against using the resource. The administrative state expresses usage permissions for specific resources without modification to the provisioning of those resources.

lifecycleState Inherited	LifecycleState	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Used to track the planned deployment, allocation to clients and withdrawal of resources.

9.1.1.13 RiskParameter_Pac

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::TopologyPacs::RiskParameter_Pac

The risk characteristics of a TopologicalEntity come directly from the underlying physical realization. The risk characteristics propagate from the physical realization to the client and from the server layer to the client layer, this propagation may be modified by protection. A TopologicalEntity may suffer degradation or failure as a result of a problem in a part of the underlying realization. The realization can be partitioned into segments which have some relevant common failure modes. There is a risk of failure/degradation of each segment of the underlying realization. Each segment is a part of a larger physical/geographical unit that behaves as one with respect to failure (i.e. a failure will have a high probability of impacting the whole unit (e.g. all cables in the same duct). Disruptions to that larger physical/geographical unit will impact (cause failure/errors to) all TopologicalEntities that use any part of that larger physical/geographical entity. Any TopologicalEntity that uses any part of that larger physical/geographical unit will suffer impact and hence each TopologicalEntity shares risk. The identifier of each physical/geographical unit that is involved in the realization of each segment of a Topological entity can be listed in the RiskParameter_Pac of that TopologicalEntity. A segment has one or more risk characteristic. Shared risk between two TopologicalEntities compromises the integrity of any solution that use one of those TopologicalEntity as a backup for the other. Where two TopologicalEntities have a common risk characteristic they have an elevated probability of failing simultaneously compared to two TopologicalEntities that do not share risk characteristics.

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 14: Attributes for RiskParameter_Pac

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
riskCharacteristicList	RiskCharacteristic	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	A list of risk characteristics for consideration in an analysis of shared risk. Each element of the list represents a specific risk consideration.

9.1.1.14 SdnController

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::SdnController

Represents the SDN controller.

Applied stereotypes:

- Preliminary
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

9.1.1.15 TopologicalEntity

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::TopologyPacs::TopologicalEntity

A TopologicalEntity is an abstract representation of the emergent effect of the combined functioning of an arrangement of components (running hardware, software running on hardware etc). The effect can be considered as the realization of the potential for apparent communication adjacency for entities that are bound to the terminations at the boundary of the TopologicalEntity. The TopologicalEntity enables the creation of constrained forwarding to achieve the apparent adjacency. The apparent adjacency has intended performance degraded from perfect adjacency and a statement of that degradation is conveyed via the attributes of the packages associated with this class. In the model both ForwardingDomain and Link are TopologicalEntities. This abstract class is used as a modeling approach to apply packages of attributes to both Link and ForwardingDomain. Link and ForwardingDomain are the key TopologicalEntities.

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 15: Attributes for TopologicalEntity

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description

_riskParameter_Pac	RiskParameter_Pac	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if risk information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if risk is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made.	
_transferCost_Pac	TransferCost_Pac	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if cost information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if cost is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made.	
_transferTiming_Pac	TransferTiming_Pac	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if transfer timing information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if transfer timing is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made.	

_transferCapacity_Pac	TransferCapacity_Pac	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if transfer capacity information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if transfer capacity is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made.	
_transferIntegrity_Pac	TransferIntegrity_Pac	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: OPTIONAL • condition:Present if transfer integrity information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if transfer integrity is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made.	
_validation_Pac	Validation_Pac	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: OPTIONAL • condition:Present if validation information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if validation is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made. Note that validation may not be possible for the specific layer protocol or in the particular case.	

_layerTransition_Pac	LayerProtocolTransition_Pac	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if layer transition information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if layer transiotio is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made. Note that layer transition occurs in a limited number of cases.	

9.1.1.16 TransferCapacity_Pac

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::TopologyPacs::TransferCapacity_Pac

The TopologicalEntity derives capacity from the underlying realization. A TopologicalEntity may be an abstraction and virtualization of a subset of the underlying capability offered in a view or may be directly reflecting the underlying realization. A TopologicalEntity may be directly used in the view or may be assigned to another view for use. The clients supported by a multi-layer TopologicalEntity may interact such that the resources used by one client may impact those available to another. This is derived from the LTP spec details. Represents the capacity available to user (client) along with client interaction and usage. A TopologicalEntity may reflect one or more client protocols and one or more members for each profile.

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 16: Attributes for TransferCapacity_Pac

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
totalPotentialCapacity	Capacity	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An optimistic view of the capacity of the TopologicalEntity assuming that any shared capacity is available to be taken.

availableCapacity	Capacity	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Capacity available to be assigned.
capacityAssignedToUserView	Capacity	0..*	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Capacity already assigned
capacityInteractionAlgorithm	String	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	A reference to an algorithm that describes how various chunks of allocated capacity interact (e.g. when shared)

9.1.1.17 TransferCost_Pac

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::TopologyPacs::TransferCost_Pac

The cost characteristics of a TopologicalEntity not necessarily correlated to the cost of the underlying physical realization. They may be quite specific to the individual TopologicalEntity e.g. opportunity cost. Relates to layer capacity There may be many perspectives from which cost may be considered for a particular TopologicalEntity and hence many specific costs and potentially cost algorithms. Using an entity will incur a cost.

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 17: Attributes for TransferCost_Pac

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
costCharacteristicList	CostCharacteristics	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY • condition:	The list of costs where each cost relates to some aspect of the TopologicalEntity.

9.1.1.18 TransferIntegrity_Pac

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::TopologyPacs::TransferIntegrity_Pac

Transfer intergrity characteristic covers expected/specified/acceptable characteristic of degradation of the transfered signal. It includes all aspects of possible degradation of signal content as well as any damage of any form to the total TopologicalEntity and to the carried signals. Note that the statement is of total impact to the TopologicalEntity so any partial usage of the TopologicalEntity (e.g. a signal that does not use full capacity) will only suffer its portion of the impact.

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 18: Attributes for TransferIntegrity_Pac

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
errorCharacteristic	String	0..1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if errorCharacteristics information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if errorCharacteristics is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made. Note that this only applies to TDM.	Describes the degree to which the signal propagated can be errored. Applies to TDM systems as the errored signal will be propagated and not packet as errored packets will be discarded.

lossCharacteristic	String	0..1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if lossCharacteristics information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if lossCharacteristics is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made. Note that this only applies to packet systems.	Describes the acceptable characteristic of lost packets where loss may result from discard due to errors or overflow. Applies to packet systems and not TDM (as for TDM errored signals are propagated unless grossly errored and overflow/underflow turns into timing slips).
repeatDeliveryCharacteristic	String	0..1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if repeatCharacteristics information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if repeatCharacteristics is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made. Note that this primarily applies to packet systemswhere a packet may be delivered more than once (in fault recovery for example). Note that it can also apply to TDM where several frames may be received twice due to switching in a system with a large differential propagation delay.	Primarily applies to packet systems where a packet may be delivered more than once (in fault recovery for example). It can also apply to TDM where several frames may be received twice due to switching in a system with a large differential propagation delay.

deliveryOrderCharacteristic	String	0..1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if deliveryOrderCharacteristics information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if deliveryOrderCharacteristics is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made. Note that this only applies to packet systems.	Describes the degree to which packets will be delivered out of sequence. Does not apply to TDM as the TDM protocols maintain strict order.
unavailableTimeCharacteristic	String	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Describes the duration for which there may be no valid signal propagated.
serverIntegrityProcessCharacteristic	String	0..1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if serverIntegrityProcessCharacteristics information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if serverIntegrityProcessCharacteristics is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made. Note that this only applies where the server has some error recovery mechanism alters the characteristics of the link from a normal distribution.	Describes the effect of any server integrity enhancement process on the characteristics of the TopologicalEntity.

9.1.1.19 TransferTiming_Pac

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::TopologyPacs::TransferTiming_Pac

A TopologicalEntity will suffer effects from the underlying physical realization related to the timing of the information passed by the TopologicalEntity.

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 19: Attributes for TransferTiming_Pac

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
fixedLatencyCharacteristic	String	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	A TopologicalEntity suffers delay caused by the realization of the servers (e.g. distance related; FEC encoding etc.) along with some client specific processing. This is the total average latency effect of the TopologicalEntity
jitterCharacteristic	String	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if jitterCharacteristics information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if jitterCharacteristics is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made. Note that this only applies to TDM.	High frequency deviation from true periodicity of a signal and therefore a small high rate of change of transfer latency. Applies to TDM systems (and not packet).

wanderCharacteristic	String	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if wanderCharacteristics information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. Note that if wanderCharacteristics is relevant but consistent ststement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made. Note that this only applies to TDM.	Low frequency deviation from true periodicity of a signal and therefore a small low rate of change of transfer latency. Applies to TDM systems (and not packet).
queuingLatencyList	QueuingLatency	0..*	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: CONDITIONAL • condition:Present if queuingLatencyCharacteristics information is relevant to usage and statement can be made that applies equally to all flows that can be supported by the TopologicalEntity. There may be more than one instance if the queuing behavior depends upon traffic properties. Note that if queuingLatencyCharacteristics is relevant but consistent statement cannot be made then the TopologicalEntity should be described in terms of subordinate parts against which coherent statements can be made. Note that this only applies to packet system.	The effect on the latency of a queuing process. This only has significant effect for packet based systems and has a complex characteristic.

9.1.1.20 Validation_Pac

Qualified Name: CoreModel::CoreNetworkModule::ObjectClasses::TopologyPacs::Validation_Pac

Validation covers the various adjacenct discovery and reachability verification protocols. Also may cover Information source and degree of integrity.

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA

- support: MANDATORY

Table 20: Attributes for Validation_Pac

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
validationMechanismList	ValidationMechanism	1..*	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Provides details of the specific validation mechanism(s) used to confirm the presence of an intended topologicalEntity.

9.1.2 Data Types

9.1.2.1 Capacity

Qualified Name: CoreModel::CoreNetworkModule::TypeDefinitions::TopologyPacs::Capacity

Informtion on capacity of a particular TopologicalEntity.
Applied stereotypes:

Table 21: Attributes for Capacity

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
totalSize	String	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Total capacity of the TopologicalEntity in MB/s
numberOfClientInstances	String	0..1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Where there is some limit to the number of client (e.g. in a channelized case).
maximumClientSize	String	0..1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Where a client is of variable capacity but due to some underlying realization the maximum size of the client is smaller than the totalSize.
numberingRange	String	0..1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Method for identifying units of capacity via some numbering scheme.

9.1.2.2 CostCharacteristics

Qualified Name: CoreModel::CoreNetworkModule::TypeDefinitions::TopologyPacs::CostCharacteristics

The information for a particular cost characteristic.
Applied stereotypes:

Table 22: Attributes for CostCharacteristics

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
costName	String	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The cost characteristic will related to some aspect of the TopologicalEntity (e.g. $ cost, routing weight). This aspect will be conveyed by the costName.
costValue	String	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The specific cost.
costAlgorithm	invalid	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The cost may vary based upon some properties of the TopologicalEntity. The rules for the variation are conveyed by the costAlgorithm.

9.1.2.3 LayerProtocolName

Qualified Name: CoreModel::CoreNetworkModule::TypeDefinitions::LayerProtocolName

Provides a controlled list of layer protocol names and indicates the naming authority. Note that it is expected that attributes will be added to this structure to convey the naming authority name, the name of the layer protocol using a human readable string and any particular standard reference. Layer protocol names include: - Layer 1 (L1): OTU, ODU - Layer 2 (L2): Carrier Grade Ethernet (ETY, ETH), MPLS-TP (MT)
Applied stereotypes:

- Preliminary

9.1.2.4 PortRole

Qualified Name: CoreModel::CoreNetworkModule::TypeDefinitions::PortRole

The role of a port in the context of the function of the forwarding entity that it bounds
Applied stereotypes:

- Preliminary

9.1.2.5 ProtectionType

Qualified Name: CoreModel::CoreNetworkModule::TypeDefinitions::ProtectionType

Identifies the type of rotection of an FcSwitch.
Applied stereotypes:

- Experimental

9.1.2.6 QueuingLatency

Qualified Name: CoreModel::CoreNetworkModule::TypeDefinitions::TopologyPacs::QueuingLatency

Provides information on latency characteristic for a particular stated trafficProperty.
Applied stereotypes:

Table 23: Attributes for QueuingLatency

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
trafficProperty	String	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The identifier of the specific traffic property to which the queuing latency applies.
latencyForTrafficWithProperty	String	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The specific queuing latency for the traffic property.

9.1.2.7 RiskCharacteristic

Qualified Name: CoreModel::CoreNetworkModule::TypeDefinitions::TopologyPacs::RiskCharacteristic

The information for a particular risk characteristic where there is a list of risk identifiers related to that characteristic.
Applied stereotypes:

Table 24: Attributes for RiskCharacteristic

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description

riskCharacteristicName	String	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The name of the risk characteristic. The characteristic may be related to a specific degree of closeness. For example a particular characteristic may apply to failures that are localized (e.g. to one side of a road) where as another characteristic may relate to failures that have a broader impact (e.g. both sides of a road that crosses a bridge). Depending upon the importance of the traffic being routed different risk characteristics will be evaluated.
riskIdentifierList	String	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	A list of the identifiers of each physical/geographic unit (with the specific risk characteristic) that is related to a segment of the TopologicalEntity.

9.1.2.8 ValidationMechanism

Qualified Name: CoreModel::CoreNetworkModule::TypeDefinitions::TopologyPacs::ValidationMechanism

Identifies the validation mechanism and describes the characteristics of that mechanism
Applied stereotypes:

Table 25: Attributes for ValidationMechanism

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
validationMechanism	String	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Name of mechanism used to validate adjacency
layerProtocolAdjacencyValidated	String	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	State of validatiion
validationRobustness	String	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Quality of validation (i.e. how likely is the stated validation to be invalid)

9.1.3 Enumeration Types

9.1.3.1 Directionality

Qualified Name: CoreModel::CoreNetworkModule::TypeDefinitions::Directionality

OBSOLETE: The enumeration with the options for directionality of the termination point.

Applied stereotypes:
- Obsolete

Contains Enumeration Literals:
- SINK:
  - Applied stereotypes:
- SOURCE:
  - Applied stereotypes:
- BIDIRECTIONAL:
  - Applied stereotypes:

9.1.3.2 ExtendedTerminationDirection

Qualified Name: CoreModel::CoreNetworkModule::TypeDefinitions::ExtendedTerminationDirection

Extended to include contra-direction considerations. Only applies to LP and elements of LP not to LTP??

Applied stereotypes:
- Experimental

Contains Enumeration Literals:
- CONTRA_DIRECTION_SINK:
  - The essential flow of the Termination entity is SINK (i.e. up the layer stack) but the INPUT flow of the Termination entity was provided by a SOURCE OUTPUT or taken from a SOURCE INPUT (duplicating the input signal) hence reversing the flow orientation from down the layer stack to up the layer stack.
  - Applied stereotypes:
- CONTRA_DIRECTION_SOURCE:

- The essential flow of the Termination entity is SOURCE (i.e. down the layer stack) but the OUTPUT flow of the Termination entity was fed to (and replaces) a SINK OUTPUT or was fed to a SINK INPUT (replacing the normal flow) hence reversing the flow orientation from down the layer stack to up the layer stack.
- Applied stereotypes:

9.1.3.3 ForwardingDirection

Qualified Name: CoreModel::CoreNetworkModule::TypeDefinitions::ForwardingDirection

The directionality of a Forwarding entity.

Applied stereotypes:

Contains Enumeration Literals:

- BIDIRECTIONAL:
  - The Fowarding entity supports both BIDIRECTIONAL flows at all Ports (i.e. all Ports have both an INPUT flow and an OUTPUT flow defined)
  - Applied stereotypes:
- UNIDIRECTIONAL:
  - The Forwarding entity has Ports that are either INPUT or OUTPUT. It has no BIDIRECTIONAL Ports.
  - Applied stereotypes:
- UNDEFINED_OR_UNKNOWN:
  - Not a normal state. The system is unable to determine the correct value.
  - Applied stereotypes:

9.1.3.4 OperType

Qualified Name: CoreModel::CoreNetworkModule::TypeDefinitions::OperType

The operation type associated with the protection mechanism (either non-revertive or revertive).

Applied stereotypes:

Contains Enumeration Literals:

- REVERTIVE:
  - Traffic will return to a preferred route on recovery of that route (potentiall after some hold-off time)
  - Applied stereotypes:

- NON-REVERTIVE:
  - Traffic will be rerouted on failure and not on recovery of any particular route.
  - Applied stereotypes:

9.1.3.5 OperationalState

Qualified Name: CoreModel::CoreNetworkModule::TypeDefinitions::OperationalState

OBSOLETE: The list of valid operational states for the connection.

Applied stereotypes:
- Obsolete

Contains Enumeration Literals:
- ENABLED:
  - Applied stereotypes:
- DISABLED:
  - Applied stereotypes:

9.1.3.6 PortDirection

Qualified Name: CoreModel::CoreNetworkModule::TypeDefinitions::PortDirection

The orientation of flow at the Port of a Forwarding entity

Applied stereotypes:

Contains Enumeration Literals:
- BIDIRECTIONAL:
  - The Port has both an INPUT flow and an OUTPUT flow defined.
  - Applied stereotypes:
- INPUT:
  - The Port only has definition for a flow into the Forwarding entity (i.e. an ingress flow).
  - Applied stereotypes:
- OUTPUT:
  - The Port only has definition for a flow out of the Forwarding entity (i.e. an egress flow).

    - Applied stereotypes:
- UNIDENTIFIED_OR_UNKNOWN:
    - Not a normal state. The system is unable to determine the correct value.
    - Applied stereotypes:

9.1.3.7 TerminationDirection

Qualified Name: CoreModel::CoreNetworkModule::TypeDefinitions::TerminationDirection

The directionality of a termination entity

Applied stereotypes:

Contains Enumeration Literals:

- BIDIRECTIONAL:
    - A Termination with both SINK and SOURCE flows.
    - Applied stereotypes:
- SINK:
    - The flow is up the layer stack from the server side to the client side. Considering an example of a Termination function within the termination entity, a SINK flow: - will arrive at at the base of the termination function (the server side) where it is essentially at an INPUT to the termination component - then will be decoded and deconstructed - then relevant parts of the flow will be sent out of the termination function (the client side) where it is essentially at an OUTPUT from the termination component A SINK termination is one that only supports a SINK flow. A SINK termiation can be bound to an OUTPUT Port of a Forwarding entity
    - Applied stereotypes:
- SOURCE:
    - The flow is down the layer stack from the server side to the client side. Considering an example of a Termination function within the termination entity, a SOURCE flow: - will arrive at at the top of the termination function (the client side) where it is essentially at an INPUT to the termination component - then will be assembled with various overheads etc and will be coded - then coded form of the assembly of flow will be sent out of the termination function (the server side) where it is essentially at an OUTPUT from the termination component A SOURCE termination is one that only supports a SOURCE flow. A SOURCE termiation can be bound to an INPUT Port of a Forwarding entity
    - Applied stereotypes:
- UNDEFINED_OR_UNKNOWN:
    - Not a normal state. The system is unable to determine the correct value.

- Applied stereotypes:

9.1.3.8 TerminationState

Qualified Name: CoreModel::CoreNetworkModule::TypeDefinitions::TerminationState

Provides support for the range of behaviours and specific states that an LP can take with respect to termination of the signal. Indicates to what degree the LayerTermination is terminated.

Applied stereotypes:

- Experimental

Contains Enumeration Literals:

- LP_CAN_NEVER_TERMINATE:
  - A non-flexible case that can never be terminated.
  - Applied stereotypes:
    - Experimental
- LT_NOT_TERMINATED:
  - A flexible termination that can terminate but is currently not terminated.
  - Applied stereotypes:
    - Experimental
- TERMINATED_SERVER_TO_CLIENT_FLOW:
  - A flexible termination that is currently terminated for server to client flow only.
  - Applied stereotypes:
    - Experimental
- TERMINATED_CLIENT_TO_SERVER_FLOW:
  - A flexible termination that is currently terminated for client to server flow only.
  - Applied stereotypes:
    - Experimental
- TERMINATED_BIDIRECTIONAL:
  - A flexible termination that is currently terminated in both directions of flow.
  - Applied stereotypes:
    - Experimental
- LT_PERMENANTLY_TERMINATED:

    - A non-flexible termination that is always terminated (in both directions of flow for a bidirectional case and in the one direction of flow for both unidirectional cases).
    - Applied stereotypes:
        - Experimental
- TERMINATION_STATE_UNKNOWN:
    - There TerminationState cannot be determined.
    - Applied stereotypes:
        - Experimental

9.1.4 Primitive Types

9.2 Core Foundation Module data dictionary

9.2.1 Classes

This section provides a view of the Classes of the Core Foundation Module. It shows owned attributes for each class.

9.2.1.1 ConditionalPackage

Qualified Name: CoreModel::CoreFoundationModule::SuperClassesAndCommonPackages::ObjectClasses::ConditionalPackage

The base class for conditional packages.

Applied stereotypes:

- Experimental
- OpenModelClass
    - objectCreationNotification: NA
    - objectDeletionNotification: NA
    - support: MANDATORY

9.2.1.2 Extension

Qualified Name: CoreModel::CoreFoundationModule::SuperClassesAndCommonPackages::ObjectClasses::Extension

Extension provides an opportunity to define properties not declared in the class that extend the class enabling a realization with simple ad-hoc extension of standard classes to be conformant.

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 26: Attributes for Extension

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
extensionList	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of simple name-value extentions

9.2.1.3 GlobalClass

Qualified Name: CoreModel::CoreFoundationModule::SuperClassesAndCommonPackages::ObjectClasses::GlobalClass

Represents a type of thing (an Entity) that has instances which can exist in their own right (independently of any others). Entity: Has identity, defined boundary, properties, functionality and lifecycle in a global context. (consider in the context of an Object Class: (usage) The representation of a thing that may be an entity or an inseparable Entity Feature)

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 27: Attributes for GlobalClass

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description

localIdList	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An identifier that is unique in the context of some scope that is less than the global scope. (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself unique, and immutable. The identifier therefore represents the identity of the entity/role. An identifier carries no semantics with respect to the purpose of the entity.)
uuid	UniversalId	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	UUID: An identifier that is universally unique (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself globally unique, and immutable. An identifier carries no semantics with respect to the purpose or state of the entity)

9.2.1.4 Label

Qualified Name: CoreModel::CoreFoundationModule::SuperClassesAndCommonPackages::ObjectClasses::Label

A property of an entity with a value that is not expected to be unique and is allowed to change. A label carries no semantics with respect to the purpose of the entity and has no effect on the entity behavior or state.

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 28: Attributes for Label

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
labelList	NameAndValue	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of labels.

9.2.1.5 LocalClass

Qualified Name: CoreModel::CoreFoundationModule::SuperClassesAndCommonPackages::ObjectClasses::LocalClass

A LocalClass represents a Feature of an Entity. It is inseparable from a GlobalClass but is a distinct feature of that GlobalClass such that the instances of LocalClass are able to have associations to other instances.. Feature of an Entity: An inseparable, externally distinguishable part of an entity. The mandatory LocalId of the LocalClass instance is unique in the context of the GlobalClass from which it is inseparable. (consider in the context of an Object Class: (usage) The representation of a thing that may be an entity or an inseparable feature of an entity)

Applied stereotypes:

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 29: Attributes for LocalClass

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
localIdList	NameAndValue	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	An identifier that is unique in the context of some scope that is less than the global scope. (consider in the context of Identifier: A property of an entity/role with a value that is unique within an identifier space, where the identifier space is itself unique, and immutable. The identifier therefore represents the identity of the entity/role. An identifier carries no semantics with respect to the purpose of the entity.)

9.2.1.6 Name

Qualified Name: CoreModel::CoreFoundationModule::SuperClassesAndCommonPackages::ObjectClasses::Name

Name: A property of an entity with a value that is unique in some namespace but may change during the life of the entity. A name carries no semantics with respect to the purpose of the entity.

Applied stereotypes:

- OpenModelClass

- objectCreationNotification: NA
- objectDeletionNotification: NA
- support: MANDATORY

Table 30: Attributes for Name

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
nameList	NameAndValue	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	List of names.

9.2.1.7 NameAndValueAuthority

Qualified Name: CoreModel::CoreFoundationModule::SuperClassesAndCommonPackages::ObjectClasses::NameAndValueAuthority

Represents the authority that controls the legal valuse for the names and values of a name/value attribute.

Applied stereotypes:

- Preliminary
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 31: Attributes for NameAndValueAuthority

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
uuid	UniversalId	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The UUID for the NameValueAuthority.

9.2.1.8 State_Pac

Qualified Name: CoreModel::CoreFoundationModule::StateModel::ObjectClasses::State_Pac

Provides general state attributes.

Applied stereotypes:

- Preliminary

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 32: Attributes for State_Pac

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
operationalState	OperationalState	0..1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The operational state is used to indicate whether or not the resource is installed and working
administrativeControl	AdministrativeControl	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The administrativeControl state provides control of the availability of specific resources without modification to the provisioning of those resources. The value is the current control target. The actual administrativeState may or may not be at target.
adminsatratveState	AdministrativeState	1	R	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Shows whether or not the client has permission to use or has a prohibition against using the resource. The administrative state expresses usage permissions for specific resources without modification to the provisioning of those resources.
lifecycleState	LifecycleState	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Used to track the planned deployment, allocation to clients and withdrawal of resources.

9.2.1.9 UniversalIdAuthority

Qualified Name: CoreModel::CoreFoundationModule::SuperClassesAndCommonPackages::ObjectClasses::UniversalIdAuthority

Represents the authority that controls the allocation of UUIDs.

Applied stereotypes:

- Preliminary

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 33: Attributes for UniversalIdAuthority

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
uuid	UniversalId	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The UUID for the UUID authority.

9.2.2 Data Types

9.2.2.1 DateAndTime

Qualified Name: CoreModel::CoreFoundationModule::TypeDefinitions::DateAndTime

This primitive type defines the date and time according to the following structure: "yyyyMMddhhmmss.s[Z|{+|-}HHMm]" where: yyyy "0000".."9999" year MM "01".."12" month dd "01".."31" day hh "00".."23" hour mm "00".."59" minute ss "00".."59" second s ".0".."9" tenth of second (set to ".0" if EMS or NE cannot support this granularity) Z "Z" indicates UTC (rather than local time) {+|-} "+" or "-" delta from UTC HH "00".."23" time zone difference in hours Mm "00".."59" time zone difference in minutes.
Applied stereotypes:

9.2.2.2 NameAndValue

Qualified Name: CoreModel::CoreFoundationModule::SuperClassesAndCommonPackages::TypeDefinitions::NameAndValue

A scoped name-value pair
Applied stereotypes:

Table 34: Attributes for NameAndValue

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
valueName	String	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The name of the value. The value need not have a name.

value	String	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The value
_nameAndValueAuthorityRef	NameAndValueAuthority	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The authority that defines the named value.
_globalClassRef	GlobalClass	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The scope of the name uniqueness
_localClassRef	LocalClass	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The scope of the name uniqueness

9.2.2.3 UniversalId

Qualified Name: CoreModel::CoreFoundationModule::SuperClassesAndCommonPackages::TypeDefinitions::UniversalId

The univeral ID value where the mechanism for generation is defned by some authority not directly referenced in the structure.
Applied stereotypes:

Table 35: Attributes for UniversalId

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
value	String	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The specific value of the universal id

9.2.3 Enumeration Types

9.2.3.1 AdministrativeControl

Qualified Name: CoreModel::CoreFoundationModule::StateModel::TypeDefinitions::AdministrativeControl

The possible values of the current target administrative state.
Reflects the current control action when the entity is not in the desired state.

Applied stereotypes:

- Experimental

Contains Enumeration Literals:

- UNLOCK:
  - The intention is for the entity to become unlocked. The entity may already be UNLOCKED.
  - Applied stereotypes:
- LOCK_PASSIVE:
  - The intention is for the entity to become locked but no effort is expected to move to the Locked state (the state will be achieved once all users stop using the resource). The entity may be LOCKED
  - Applied stereotypes:
- LOCK_ACTIVE:
  - The intention is for the entity to become locked and it is expected that effort will be made to move to the Locked state (users will be actively removed). The entity may already be LOCKED.
  - Applied stereotypes:
- LOCK_IMMEDIATE:
  - The intention is for the entity to become locked and it is expected to move to the Locked state immediately (users will be force removed). The entity may already be LOCKED.
  - Applied stereotypes:

9.2.3.2 AdministrativeState

Qualified Name: CoreModel::CoreFoundationModule::StateModel::TypeDefinitions::AdministrativeState

The possible values of the administrativeState.

Applied stereotypes:

- Preliminary

Contains Enumeration Literals:

- LOCKED:
  - Users are administratively prohibited from making use of the resource.
  - Applied stereotypes:
    - Preliminary
- UNLOCKED:
  - Users are allowed to use the resource
  - Applied stereotypes:
    - Preliminary

9.2.3.3 ExtendedAdminState

Qualified Name: CoreModel::CoreFoundationModule::StateModel::TypeDefinitions::ExtendedAdminState

Possible extentions to AdministrativeState

Applied stereotypes:

- Experimental

Contains Enumeration Literals:

- SHUTTING_DOWN_ACTIVE:
  - The entity is administratively restricted to existing instances of use only. There are specific actions to remove existing uses. There may be no new instances of use enabled. This corresponds to a control of LOCK_ACTIVE.
  - Applied stereotypes:
    - Experimental
- SHUTTING_DOWN_PASSIVE:
  - The entity is administratively restricted to existing instances of use only. There may be no new instances of use enabled. This corresponds to a control of LOCK_PASSIVE.
  - Applied stereotypes:
    - Experimental

9.2.3.4 LifecycleState

Qualified Name: CoreModel::CoreFoundationModule::StateModel::TypeDefinitions::LifecycleState

The possible values of the lifecycleState.

Applied stereotypes:

- Experimental

Contains Enumeration Literals:

- PLANNED:
  - The resource is planned but is not present in the network.
  - Applied stereotypes:
    - Experimental
- POTENTIAL:

    - The supporting resources are present in the network but are shared with other clients; or require further configuration before they can be used; or both. o When a potential resource is configured and allocated to a client it is moved to the "installed" state for that client. o If the potential resource has been consumed (e.g. allocated to another client) it is moved to the "planned" state for all other clients.
    - Applied stereotypes:
        - Experimental
- INSTALLED:
    - The resource is present in the network and is capable of providing the service expected.
    - Applied stereotypes:
        - Experimental
- PENDING_REMOVAL:
    - The resource has been marked for removal
    - Applied stereotypes:
        - Experimental

9.2.3.5 OperationalState

Qualified Name: CoreModel::CoreFoundationModule::StateModel::TypeDefinitions::OperationalState

The possible values of the operationalState.

Applied stereotypes:

- Preliminary

Contains Enumeration Literals:

- DISABLED:
    - The resource is unable to meet the SLA of the user of the resource. If no (explicit) SLA is defined the resource is disabled if it is totally inoperable and unable to provide service to the user.
    - Applied stereotypes:
        - Preliminary
- ENABLED:
    - The resource is partially or fully operable and available for use
    - Applied stereotypes:
        - Preliminary

9.2.4 Primitive Types

9.2.4.1 BitString

Qualified Name: CoreModel::CoreFoundationModule::TypeDefinitions::BitString

This primitive type defines a bit oriented string. The size of the BitString will be defined in the valueRange property of the attribute; according to ASN.1 (X.680). The semantic of each bit position will be defined in the Documentation field of the attribute.

Applied stereotypes:

9.2.4.2 PrintableString

Qualified Name: CoreModel::CoreFoundationModule::TypeDefinitions::PrintableString

A string that only includes printable characters

Applied stereotypes:

9.2.4.3 Real

Qualified Name: CoreModel::CoreFoundationModule::TypeDefinitions::Real

This primitive type maps to the "realnumber" defined in Recommendation X.680.

Applied stereotypes:

9.3 Core Enhancements data dictionary

This section provides a view of the Classes of the Core Enhancements. It shows owned attributes for each class.

[Editor's note: Some enhancements to descriptions are required in this section. They will be provided in a future release.]

9.3.1 Classes

9.3.1.1 AdapterPropertySpec

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::LtpCapability-Developed::ObjectClasses::AdapterPropertySpec

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 36: Attributes for AdapterPropertySpec

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
_poolPropertySpecList	PoolPropertySpec	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_mappingInteractionRuleRefList	MappingInteractionRule	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_mappingInteractionRuleList	MappingInteractionRule	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.3.1.2 ClientSpec

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::LtpCapability-Developed::ObjectClasses::ClientSpec

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 37: Attributes for ClientSpec

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description

_mappingInteractionRuleRefList	MappingInteractionRule	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.3.1.3 ConfigurationAndSwitchControl

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::FcCapability-Developed::ObjectClasses::ConfigurationAndSwitchControl

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 38: Attributes for ConfigurationAndSwitchControl

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
switchControlRule	ControlRule	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
switch	IngressSwitchSelection	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
egressSelection	EgressSwitchSelection	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.3.1.4 ConfigurationAndSwitchController

Qualified Name: CoreModel::CoreModelEnhancements::FcSwitchEnhancements-Developed::ConfigurationAndSwitchController

Sketch representation of a cntroller with basic capability to control switched and add/delete/modify FCs.

Applied stereotypes:

- Experimental

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 39: Attributes for ConfigurationAndSwitchController

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
SwichRule	invalid	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	A sketch of the presence of complex rules governing the switch behavior.
_fcSwitchRefList	FcSwitch	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The switch being controlled.
_controlParameters	ControlParameters	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The control parameters to be aplied if local parameters are used rather than profiles
_profileProxyRef	ProfileProxy	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Applied profiles.

9.3.1.5 ConfigurationGroup

Qualified Name: CoreModel::CoreModelEnhancements::FcSwitchEnhancements-Developed::ConfigurationGroup

Represents a scope of control for one or more Controllers

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 40: Attributes for ConfigurationGroup

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description

_configurationAndSwitchControlRefList	ConfigurationAndSwitchController	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	A controller operating in the scope defined.

9.3.1.6 ConfigurationGroupSpec

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::FcCapability-Developed::ObjectClasses::ConfigurationGroupSpec

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 41: Attributes for ConfigurationGroupSpec

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
switchControl	ConfigurationAndSwitchControl	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
fcSpec	FcSpec	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
ltpAssociationRule	LtpAssociationRule	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.3.1.7 ConnectionPointAndAdapterSpec

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::LtpCapability-Developed::ObjectClasses::ConnectionPointAndAdapterSpec

Applied stereotypes:

- Experimental

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 42: Attributes for ConnectionPointAndAdapterSpec

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
_connectionSpec	ConnectionSpec	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.3.1.8 ConnectionSpec

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::LtpCapability-Developed::ObjectClasses::ConnectionSpec

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

9.3.1.9 ControlParameters

Qualified Name: CoreModel::CoreModelEnhancements::FcSwitchEnhancements-Developed::ControlParameters

A list of control parameters to apply to a switch

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 43: Attributes for ControlParameters

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
operType	OperType	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	This attribute whether or not the protection scheme is revertive or non-revertive.
waitToRestoreTime	Integer	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	If the protection system is revertive, this attribute specifies the amount of time, in seconds, to wait after a fault clears before restoring traffic to the protected protectionUnit that initiated the switching. Valid values for this attribute are integers.
protType	ProtectionType	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	Indicates the protection scheme that is used for the ProtectionGroup.
holdOffTime	Integer	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	This attribute indicates the time, in seconds, between declaration of signal degrade or signal fail, and the initialization of the protection switching algorithm. Valid values are integers in units of seconds.

9.3.1.10 ControlRule

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::FcCapability-Developed::ObjectClasses::ControlRule

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 44: Attributes for ControlRule

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description

switchControl_switch_1	invalid	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.3.1.11 EgressFcPortSet

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::FcCapability-Developed::ObjectClasses::EgressFcPortSet

Applied stereotypes:

- Preliminary
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

9.3.1.12 EgressSwitchSelection

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::FcCapability-Developed::ObjectClasses::EgressSwitchSelection

Applied stereotypes:

- Preliminary
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 45: Attributes for EgressSwitchSelection

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
setMember	invalid	1..*	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.3.1.13 FcPortSetSpec

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::FcCapability-Developed::ObjectClasses::FcPortSetSpec

Applied stereotypes:

- Preliminary
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 46: Attributes for FcPortSetSpec

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
ingressFcPortSet	IngressFcPortSet	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
egressFcPortSet	EgressFcPortSet	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
ltpAssociationRule	LtpAssociationRule	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
role	invalid	1	RW	Preliminary OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.3.1.14 FcSpec

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::FcCapability-Developed::ObjectClasses::FcSpec

Applied stereotypes:

- Preliminary

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 47: Attributes for FcSpec

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
multiSwitchedUniFlow	MultiSwitchedUniFlow	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
switchControl	ConfigurationAndSwitchControl	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
fcPortSpec	FcPortSetSpec	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
fcSwitchGroupSpec	ConfigurationGroupSpec	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
ltpAssociationRule	LtpAssociationRule	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.3.1.15 IngressFcPortSet

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::FcCapability-Developed::ObjectClasses::IngressFcPortSet

Applied stereotypes:

- Preliminary
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

9.3.1.16 IngressFcPortSetSpec-Pac

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::FcCapability-Developed::ObjectClasses::IngressFcPortSetSpec-Pac

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

9.3.1.17 IngressSwitchSelection

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::FcCapability-Developed::ObjectClasses::IngressSwitchSelection

Applied stereotypes:

- Preliminary
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 48: Attributes for IngressSwitchSelection

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
setMember	invalid	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.3.1.18 LpSpec

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::LtpCapability-Developed::ObjectClasses::LpSpec

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 49: Attributes for LpSpec

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
_adapterSpec	ConnectionPointAndAdapterSpec	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_terminationSpec	TerminationSpec	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_adapterPropertySpecList	AdapterPropertySpec	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_providerViewSpec	ProviderViewSpec	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_serverSpecList	ServerSpec	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.3.1.19 LtpAssociationRule

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::FcCapability-Developed::ObjectClasses::LtpAssociationRule

Applied stereotypes:

- Preliminary
- OpenModelClass
  - objectCreationNotification: NA

- objectDeletionNotification: NA
- support: MANDATORY

9.3.1.20 LtpSpec

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::LtpCapability-Developed::ObjectClasses::LtpSpec

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 50: Attributes for LtpSpec

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
_lpSpecList	LpSpec	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.3.1.21 MappingInteractionRule

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::LtpCapability-Developed::ObjectClasses::MappingInteractionRule

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

9.3.1.22 MultiSwitchedUniFlow

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::FcCapability-Developed::ObjectClasses::MultiSwitchedUniFlow

Applied stereotypes:

- Preliminary
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 51: Attributes for MultiSwitchedUniFlow

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
_ingressFcPort	IngressFcPortSet	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_egressFcPort	EgressFcPortSet	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
switchControl	ConfigurationAndSwitchControl	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
ingressFcPortSet	IngressFcPortSet	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
egressFcPortSet	EgressFcPortSet	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.3.1.23 PoolPropertySpec

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::LtpCapability-Developed::ObjectClasses::PoolPropertySpec

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 52: Attributes for PoolPropertySpec

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
_clientSpec	ClientSpec	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
clientCapacity	invalid	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_adapterPropertySpecRefList	AdapterPropertySpec	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.3.1.24 ProfileProxy

Qualified Name: CoreModel::CoreModelEnhancements::FcSwitchEnhancements-Developed::ProfileProxy

A lightweight sketch placeholder for the profile model.

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 53: Attributes for ProfileProxy

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description

profileProxyMode	invalid	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	A parameter profile may be used in a number of different ways: - Forces the values on the target with no opportunity to see or override the values in the target - Sets the values on the target that can be seen on the target - Sets the values on the target and supports override on the target so the target can be set away from the value in the profile - etc
_controlParameters	ControlParameters	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	The control parameters that can be set int the profile and applied to the target. Not all parameters need be selected and applied.

9.3.1.25 ProviderViewSpec

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::LtpCapability-Developed::ObjectClasses::ProviderViewSpec

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 54: Attributes for ProviderViewSpec

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
_poolPropertySpecList	PoolPropertySpec	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.3.1.26 ServerSpec

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::LtpCapability-Developed::ObjectClasses::ServerSpec

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

9.3.1.27 SwitchPropertySpec-Pac

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::FcCapability-Developed::ObjectClasses::SwitchPropertySpec-Pac

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

9.3.1.28 TerminationSpec

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::LtpCapability-Developed::ObjectClasses::TerminationSpec

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 55: Attributes for TerminationSpec

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description

_connectionSpec	ConnectionSpec	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.3.2 Data Types

9.3.3 Enumeration Types

9.3.4 Primitive Types

9.4 Other Core Enhancement Fragments data dictionary

9.4.1 Classes

This section provides a view of the Classes of other Core Enhancement fragments. It shows owned attributes for each class.

9.4.1.1 AnyEntityClass

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::SpecAndProfile-Sketch::AnyEntityClass

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

9.4.1.2 AnyEntityInstance

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::SpecAndProfile-Sketch::AnyEntityInstance

Applied stereotypes:

- Experimental

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 56: Attributes for AnyEntityInstance

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
_profileRefList	ProfileInstance	0..*	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_specRef	SpecInstance	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_class	AnyEntityClass	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.4.1.3 AttributePackage

Qualified Name: CoreModel::CoreModelEnhancements::InformationArchitectureAndPatterns::EssentialStructure-Sketch::AttributePackage

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

9.4.1.4 Component

Qualified Name: CoreModel::CoreModelEnhancements::InformationArchitectureAndPatterns::EssentialStructure-Sketch::Component

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 57: Attributes for Component

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
_port	Port	1..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_attributePackageList	AttributePackage	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_boundComponent	Component	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_encapsulatedSystem	System	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.4.1.5 LinkSpec

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::LinkCapability-Sketch::LinkSpec

Describes the capabilities of the Link focussing on link asymmetries and constraints.

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

9.4.1.6 Port

Qualified Name: CoreModel::CoreModelEnhancements::InformationArchitectureAndPatterns::EssentialStructure-Sketch::Port

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 58: Attributes for Port

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
Role	invalid	1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_boundPort	Port	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.4.1.7 ProfileClass

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::SpecAndProfile-Sketch::ProfileClass

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

9.4.1.8 ProfileInstance

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::SpecAndProfile-Sketch::ProfileInstance

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 59: Attributes for ProfileInstance

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
_specInstanceRef	SpecInstance	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_profileClass	ProfileClass	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_class	AnyEntityClass	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.4.1.9 Sketch::FC

Qualified Name: CoreModel::CoreModelEnhancements::InterViewRelationships-Sketch::Sketch::FC

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 60: Attributes for Sketch::FC

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
_supportingFcInOtherViewRef	Sketch::FC	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_suportedFcInOtherView	Sketch::FC	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.4.1.10 Sketch::LTP

Qualified Name: CoreModel::CoreModelEnhancements::InterViewRelationships-Sketch::Sketch::LTP

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 61: Attributes for Sketch::LTP

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
_supportingLtpInOtherViewRef	Sketch::LTP	0..1	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_supportedLtpInOtherViewRef	Sketch::LTP	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.4.1.11 SpecClass

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::SpecAndProfile-Sketch::SpecClass

Applied stereotypes:

- Experimental

- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

9.4.1.12 SpecInstance

Qualified Name: CoreModel::CoreModelEnhancements::ProfilesTemplatesAndSpecificationsModule::SpecAndProfile-Sketch::SpecInstance

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 62: Attributes for SpecInstance

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
specClass	SpecClass	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	
_classRef	AnyEntityClass	1	RW	Experimental OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.4.1.13 System

Qualified Name: CoreModel::CoreModelEnhancements::InformationArchitectureAndPatterns::EssentialStructure-Sketch::System

Applied stereotypes:

- Experimental
- OpenModelClass

- objectCreationNotification: NA
- objectDeletionNotification: NA
- support: MANDATORY

9.4.1.14 ViewAbstractionRules

Qualified Name: CoreModel::CoreModelEnhancements::ViewAbstractionRule-Sketch::ViewAbstractionRules

Provides rules and access to policies that govern and explain the view abstraction. At this point it appears that this is not necessary however it has been left in the model as there is still some view abstraction work to be done.

Applied stereotypes:

- Experimental
- OpenModelClass
  - objectCreationNotification: NA
  - objectDeletionNotification: NA
  - support: MANDATORY

Table 63: Attributes for ViewAbstractionRules

Attribute Name	Type	Multiplicity	Access	Stereotypes	Description
__ltpRelatestToLtpInOtherView	LtpRelatestToLtpInOtherView	0..*	RW	OpenModelAttribute • AVC: NA • valueRange: no range constraint • support: MANDATORY	

9.4.2 Data Types

9.4.3 Enumeration Types

9.4.4 Primitive Types

10 Additional figures related to potential extensions

10.1 State extensions

There are a number of experimental items in the state model as shown below. These are included in the main data dictionary.

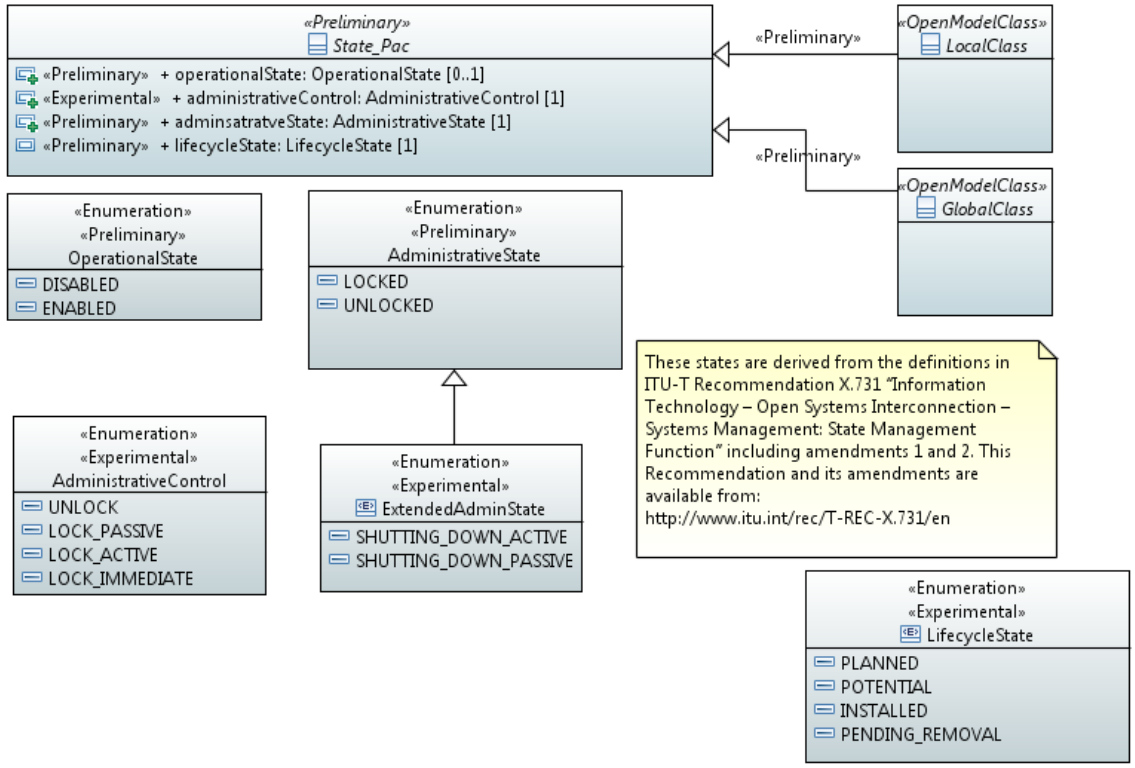


Figure 10-1 State model with enhancements

CoreModel diagram: EnhancedStates

10.2 LTP Specification

The LTP specification is planned for the next phase of work. Some experimental work has already been carried out as depicted below.

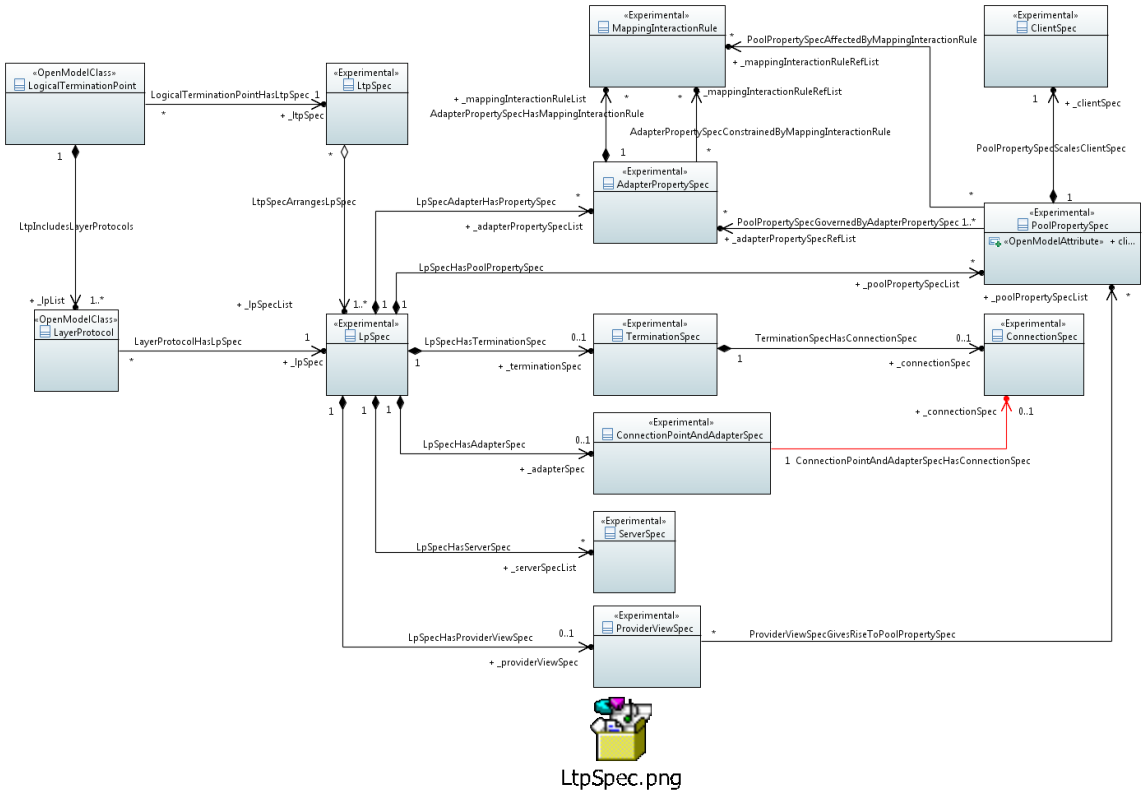


Figure 10-2 LTP/LP Spec Model

CoreModel diagram: LtpCapabilitySpec

10.3 Model structure rules

Some of the associations in the model are interrelated by some pattern. The following figures explore ways of expressing the patterns and interrelationships.

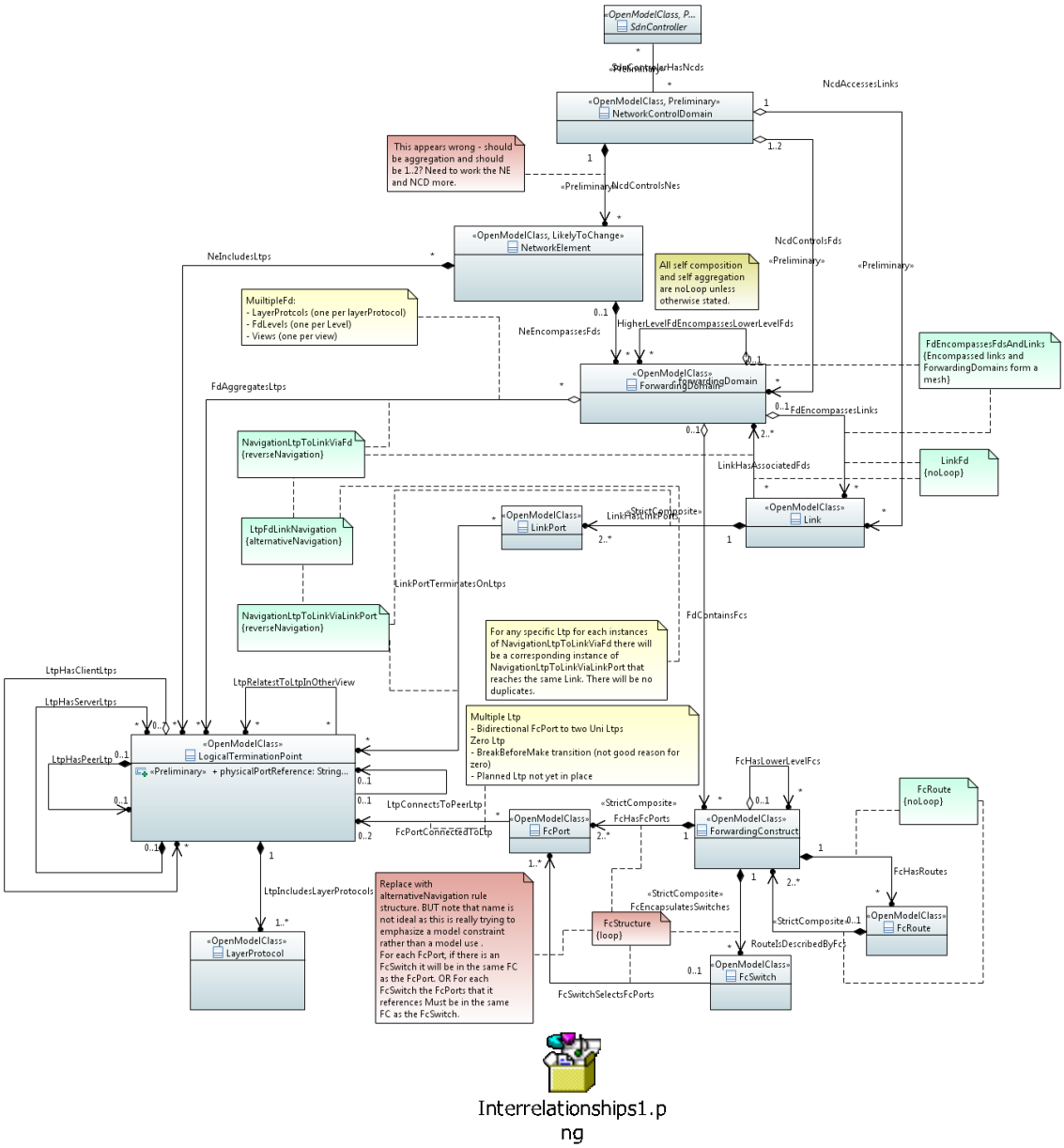


Figure 10-3 Association interrelationship rules alternative 1

CoreModel diagram: HighLevelSkeltonOverviewWithLoopsHighlighted-Altenative1

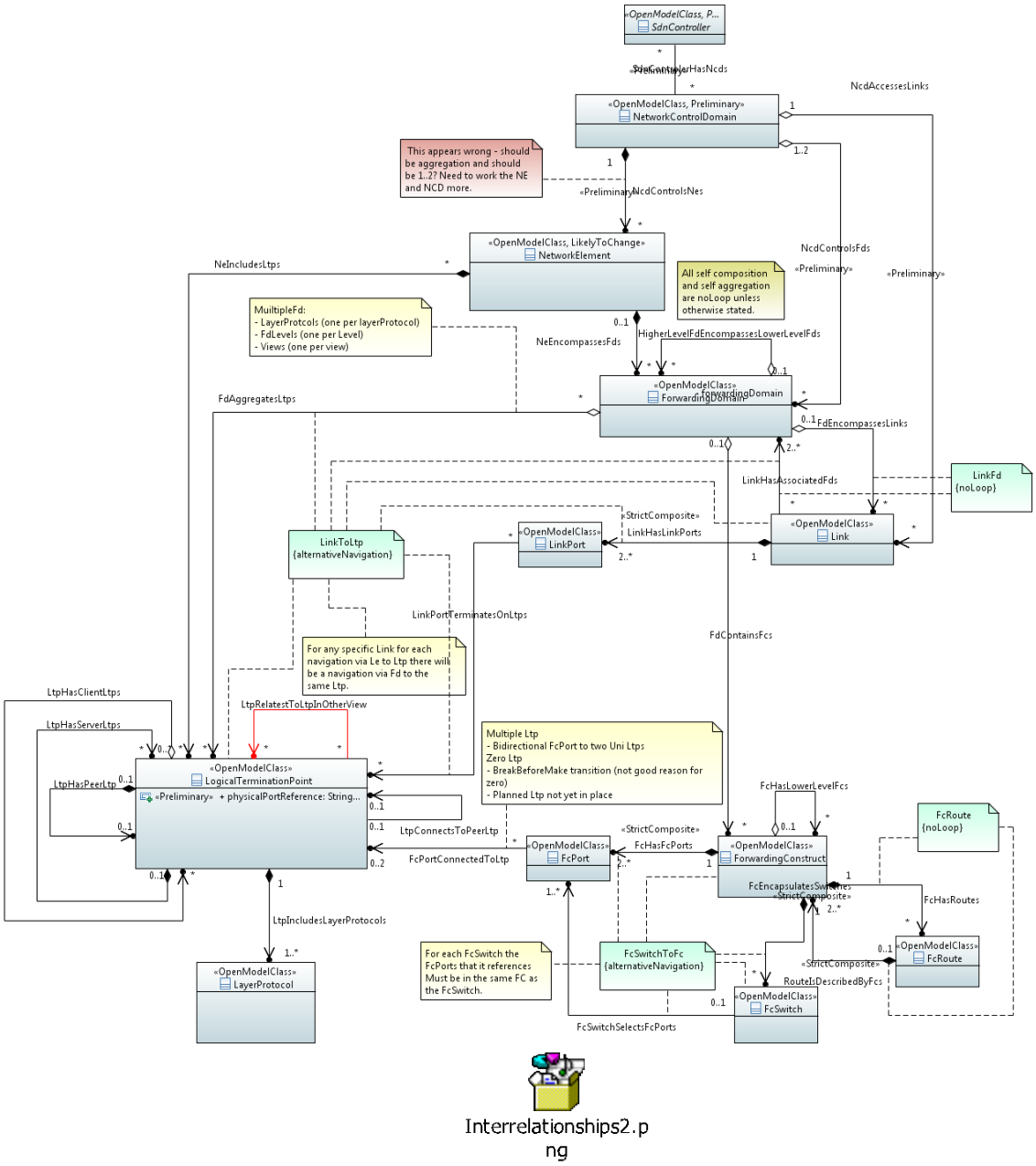


Figure 10-4 Association interrelationship rules alternative 2

CoreModel diagram: HighLevelSkeltonOverviewWithLoopsHighlighted-Alternative2

11 Addendum Terminology Translation table

The translations provided in this release are early draft. There may be errors in the table and the table is not complete. It should be used for guidance only.

Table 64: Concept mappings
Please download the document and see attachments

MappingTable.xlsx

12 Back matter

12.1 Editors

Kam LAM, Alcatel-Lucent

Nigel DAVIS, Ciena

12.2 Contributors

Dieter BELLER	Alcatel-Lucent
Sergio BELOTTI	Alcatel-Lucent
Kam LAM	Alcatel-Lucent
Eve VARMA	Alcatel-Lucent
Nigel DAVIS	Ciena
Christopher Hartley	Cisco
Jonathan SADLER	Coriant
Bernd ZEUNER	Deutsch Telekom
Dave HOOD	Ericsson
Meral SHIRAZIPOUR	Ericsson
Scott MANSFIELD	Ericsson
Xiang YUN	FiberHome
Yuji TOCHIO	Fujitsu
Italo BUSI	Huawei
Maarten VISSERS	Huawei
Raymond CHEH	Juniper
Karthik SETHURAMAN	NEC
Malcolm BETTS	ZTE