OPEN NETWORKING FOUNDATION

Core Information Model (CoreModel)

TR-512.2

Forwarding and Termination

Version 1.3.1
January 2018

OpenFlow

ONF Document Type: Technical Recommendation
ONF Document Name: Core Information Model version 1.3.1

Disclaimer

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

Any marks and brands contained herein are the property of their respective owners.

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

©2018 Open Networking Foundation. All rights reserved.

Open Networking Foundation, the ONF symbol, and OpenFlow are registered trademarks of the Open Networking Foundation, in the United States and/or in other countries. All other brands, products, or service names are or may be trademarks or service marks of, and are used to identify, products or services of their respective owners.

Important note

This Technical Recommendations has been approved by the Project TST, but has not been approved by the ONF board. This Technical Recommendation is an update to a previously released TR specification, but it has been approved under the ONF publishing guidelines for 'Informational' publications that allow Project technical steering teams (TSTs) to authorize publication of Informational documents. The designation of '-info' at the end of the document ID also reflects that the project team (not the ONF board) approved this TR.

Table of Contents

 © 2018 Open Networking Foundation

List of Figures

 © 2018 Open Networking Foundation

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
1.2	September 20, 2016	Version 1.2 [Note Version 1.1 was a single document whereas 1.2 is broken into a number of separate parts]
1.3	September 2017	Version 1.3 [Published via wiki only]
1.3.1	January 2018	Addition of text related to approval status.

 © 2018 Open Networking Foundation

1 Introduction

This document is an addendum to the TR-512 ONF Core Information Model and forms part of the description of the ONF-CIM. For general overview material and references to the other parts refer to TR-512.1.

1.1 References

For a full list of references see TR-512.1.

1.2 Definitions

For a full list of definition see TR-512.1.

1.3 Conventions

See TR-512.1 for an explanation of:

- UML conventions
- Lifecycle Stereotypes
- Diagram symbol set

1.4 Viewing UML diagrams

Some of the UML diagrams are very dense. To view them either zoom (sometimes to 400%) or open the associated image file (and zoom appropriately) or open the corresponding UML diagram via Papyrus (for each figure with a UML diagram the UML model diagram name is provided under the figure or within the figure).

1.5 Understanding the figures

Figures showing fragments of the model using standard UML symbols and also figures illustrating application of the model are provided throughout this document. Many of the application-oriented figures also provide UML class diagrams for the corresponding model fragments (see TR-512.1 for diagram symbol sets). All UML diagrams depict a subset of the relationships between the 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.

2 Introduction to the Core Network Model

The focus of this document is the key parts of Core Network Model of the ONF-CIM. The Core Network Model covers the essentials for modeling of the Network providing all of the key classes.

The CoreNetworkModel encompasses all aspects of Termination and Forwarding. The focus of this document is:

 © 2018 Open Networking Foundation

- Termination aspects of the CoreNetworkModel covering the modeling of the processing of transport characteristic information, such as termination, adaptation, OAM, etc.
- Forwarding aspects of the CoreNetworkModel covering the details of forwarding entities

The Core Network Model also encompasses a number of other areas which are covered in detail in related documents:

- Topology (see TR-512.4) covering the modeling of network topology information in detail[1] and describes the attributes relevant when working with multi-layered network topology.
- Resilience (see TR-512.5) covering the modeling of switches and configuration/switch control

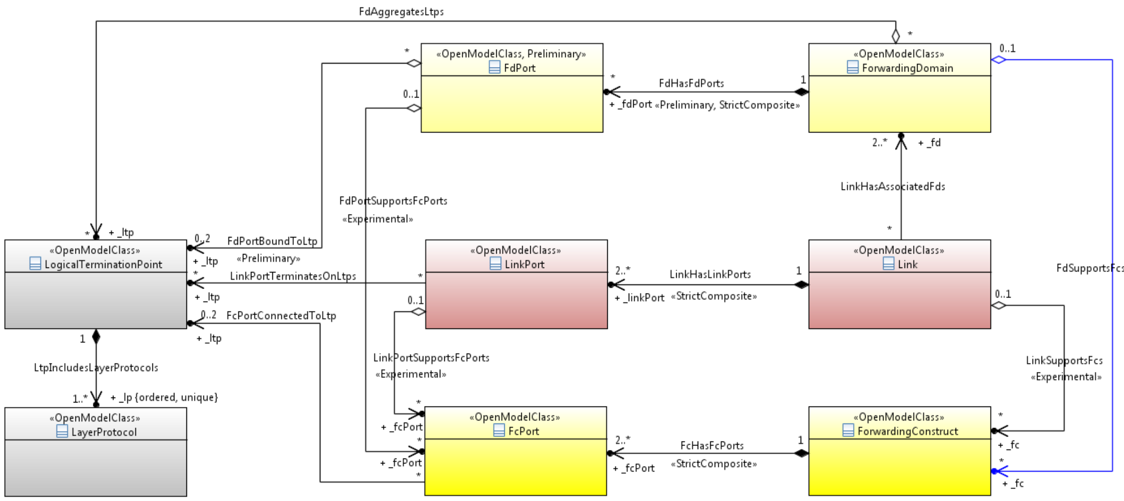
A data dictionary that sets out the details of all classes, data types and attributes is also provided (TR-512.DD).

3 Forwarding and Termination model detail

The Forwarding and Termination model is at the heart of the CoreModel. The figure below provides a view of the structure of the model. Further structure related to other aspects of the model is provided in other sections (especially relevant areTR-512.4 and TR-512.5). The diagram below highlights key interrelationships between key 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 the diagram symbol set referenced in section 1.3 Conventions on page 6 (with the Link in the alternative color for clarity). This color scheme for class diagrams is used in some of the later figures.

[1] 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.

© 2018 Open Networking Foundation



CoreModel diagram: Forwarding-AlignedSkeletonOverview

Figure 3-1 Skeleton Class Diagram of key object classes

The model in the figure above provides the essential entities for representation of Forwarding and Termination. Several patterns can be seen in the model:

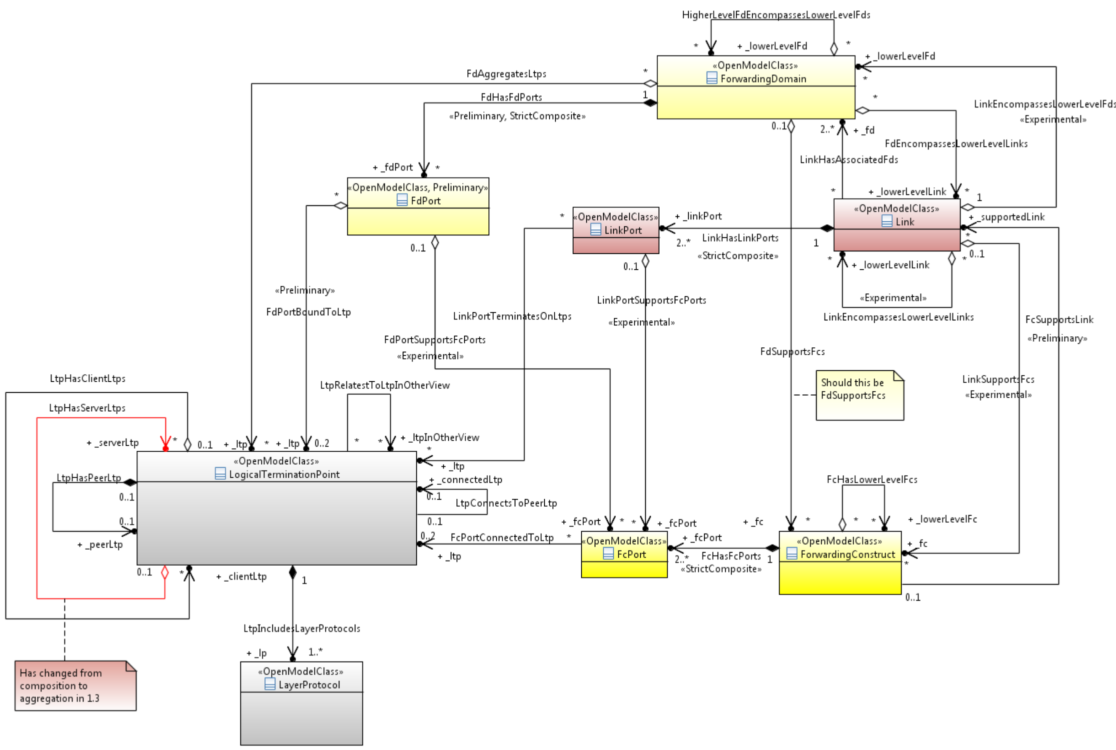
- Component-Port: An aspect of the Component-System pattern discussed in TR-512.A.2. The Link, ForwardingConstrict (FC) and ForwardingDomain (FD) have ports
- Symmetric function: An FD can be seen as a symmetric function and can be associated directly with an LTP (bypassing the FdPort). A Link can also be seen as a symmetric function in the context of an FD and be directly associated to an FD bypassing the FdPort, LinkPort and intervening LogicalTerminationPoint (LTP)
- Port support: If the ports on one class (e.g. FD) support the ports of another class (e.g.FC such that FdPortSupportsFcPort) then there is also a support relationship between the classes (e.g. FdSupportsFcs[2])
- Enablement: The FD and Link represent the potential to enable constrained forwarding. Both FD and Link can support enabled constrained forwarding represented by the FC

When applying the information model to a specific 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 class or part of a class. 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. The Pruning and Refactoring method is described in [ONF TR-513].

The figure below provides more detail highlighting peer and interlayer associations between LTPs. The figures in section 4.2 Termination on page 29 explain the uses of the associations using simple pattern examples.

[2] Note that the name of this association has been changed in this release to emphasise the pattern.

 © 2018 Open Networking Foundation



CoreModel diagram: Forwarding-LtpInterLayerSkeletonOverview

Figure 3-2 Skeleton Class Diagram of key classes showing layering

The figure also shows inter-layer and intra-layer associations between FD, FC and Link. Details of FC to link layering and other FC, Link and FD considerations are provided in TR-512.4.

Note that not all attributes are shown for the classes below (see TR-512.DD for a list of all attributes). Only those attributes that are relevant for this document are shown.

3.1 Termination model

3.1.1 LogicalTerminationPoint (LTP)

Qualified Name: CoreModel::CoreNetworkModel::ObjectClasses::LogicalTerminationPoint

The LogicalTerminationPoint (LTP) class encapsulates the termination and adaptation functions of one or more transport layers represented by instances of LayerProtocol.
The encapsulated transport layers have a simple fixed 1:1 client-server relationship defined by association end ordering.
The structure of LTP supports all transport protocols including circuit and packet forms.

Inherits properties from:

- GlobalClass

 © 2018 Open Networking Foundation

Table 1: Attributes for LogicalTerminationPoint

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
physicalPortReference	Preliminary	One or more text labels for the unmodeled physical port associated with the LTP. In many cases there is no associated physical port.
ltpDirection		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
_serverLtp		References contained LTPs representing servers of this LTP in an inverse multiplexing configuration (e.g. VCAT).
_clientLtp		References contained LTPs representing client traffic of this LTP for normal cases of multiplexing.
_lp		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).
_connectedLtp		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.
_peerLtp		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 this LTP.
_ltpInOtherView	Preliminary	References one or more LTPs in other views that represent this LTP. The referencing LTP is the provider of capability.
_accessPort	Experimental	Provides a reference to the place where the signal is accessed. It may represent a physical place (some part of one or more connectors) or a virtual equivalent where there is no further protocol layering (visible).
_transferCapacity_Pac	Experimental	The LTP has as an inherent capacity derived from underlying capability. The capacity of a particular LTP may be dependent upon other uses of resource in the device and hence it may vary over time. The capacity of a Link is dependent upon the capacity of the LTPs at its ends. An LTP 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.
_ltpSpecReference:ClassRef	SpecReference Experimental	Provides a reference to a specification which is in the form of a class definition. An instance of LTP will reference a class (by a universally unique id) that provides definition that extends the LTP including attributes and structure that are present in the LTP instance but that are not defined in the native LTP class.

 © 2018 Open Networking Foundation

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_fdRuleGroup	Experimental	An LTP can reference FD rules that the FD that aggregates it also references so that the rules can then apply to the LTP.
_embeddedClock		See referenced class

An explanation of the structure and usage of the specification referenced by "_ltpSpec" is provided in TR-512.7. Rules for forming and interrelating LTP instances are provided in section 4.2 Termination on page 29.

3.1.2 LayerProtocol (LP)

Qualified Name: CoreModel::CoreNetworkModel::ObjectClasses::LayerProtocol

The projection of an LTP into each transport layer is represented by a LayerProtocol (LP) instance. A LayerProtocol instance 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 there is a n:1 relationship between client and server, the layers must be split over two separate instances of LTP.

Inherits properties from:

- LocalClass

Table 2: Attributes for LayerProtocol

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
layerProtocolName		Indicate the specific layer-protocol described by the LayerProtocol entity.
lpDirection		The overall directionality of the LP. - A BIDIRECTIONAL LP will have some SINK and/or SOURCE flows. - 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		Indicates whether the layer is terminated and if so how.
_lpSpecReference:ClassRef	SpecReference Experimental	See referenced class

© 2018 Open Networking Foundation

Transport layer-protocol[3] specific properties (such as technology specific termination and adaptation properties) are not modeled directly in LayerTermination. These attributes are defined in specifications (see TR-512.7) that are used to augment the model. Where a technology specific termination has a complex structuring of internal parts, these parts will be modeled in the specification

3.2 Forwarding

3.2.1 ForwardingDomain (FD)

Qualified Name: CoreModel::CoreNetworkModel::ObjectClasses::ForwardingDomain

The ForwardingDomain (FD) class models the topological component that represents a forwarding capability that provides the opportunity to enable forwarding (of specific transport characteristic information at one or more protocol layers) between points.
The FD object provides the context for and constrains the formation, adjustment and removal of FCs and hence offers the potential to enable forwarding.
The FCs may be formed between LTPs at the boundary of the FD or between physical ports at the boundary of the FD (for media layers).
A number of FDs (related by Links) may be grouped and abstracted to form an FD where that FD represents the effect of the underlying FDs but where the detailed structure is not apparent.
This grouping and abstraction is potentially recursive.
An FD represents an abstraction of some combination of software behavior, electronic behavior and physical structure that provides a forwarding capability.
At a lower level of recursion an FD could represent a forwarding capability within a device.
A device may encompass two or more disjoint forwarding capabilities and may support more than one layer protocol, hence more than one FD.
A routing fabric may be logically partitioned to represent connectivity constraints, hence the FD representing the routing fabric may be partitioned into a number of FDs representing the constraints.
The FD represents a subnetwork [ITU-T G.800], FlowDomain [TMF 612] and a MultiLayerSubNetwork (MLSN) [TMF 612].
As in the TMF concept of MLSN the FD can support more than one layer-protocol.
Note that the ITU-T G.800 subnetwork is a single layer entity.

Inherits properties from:

- GlobalClass
- ForwardingEntity

Table 3: Attributes for ForwardingDomain

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
layerProtocolName		One or more protocol layers at which the FD represents the opportunity to enable forwarding between LTP that bound it.

[3] The specific transport technology Characteristic Information (see [ITU-T G.805])

 © 2018 Open Networking Foundation

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_lowerLevelFd		The FD 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 an 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).
_fc		An FD aggregates one or more FCs. An aggregated FC connects LTPs that bound the FD.
_ltp		An instance of FD is associated with zero or more LTP objects. The LTPs that bound the FD provide capacity for forwarding. For asymmetric FDs, the association to the LTP is via the FdPort.
_lowerLevelLink		The FD encompasses Links that interconnect lower level FDs and collect Links that are wholly within the bounds of the FD. See also _lowerLevelFd.
_fdSpec	Experimental	See referenced class

3.2.2 FdPort

Qualified Name: CoreModel::CoreNetworkModel::ObjectClasses::FdPort

The association of the FD to LTPs may be direct for symmetric FDs and may be via FdPort for asymmetric FDs.
The FdPort class models the role of the access to the FD function.
The capability to set up FCs between the associated FdPorts of the FD depends upon the type of FD. It is asymmetry in this capability that brings the need for FdPort.
The FD can be considered as a component and the FdPort as a Port on that component.

Inherits properties from:

- LocalClass

This class is Preliminary.

Table 4: Attributes for FdPort

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_ltp		An instance of FD is associated with zero or more LTP objects. The LTPs that bound the FD provide capacity for forwarding. For asymmetric FDs, the association to the LTP is via the FdPort.

 © 2018 Open Networking Foundation

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
role		Each FdPort of the FD has a role (e.g., symmetric, hub, spoke, leaf, root) in the context of the FD with respect to the FD capability.
fdPortDirection		The orientation of the defined flow at the FdPort.
_fcPort	Experimental	Where an FD is asymmetric and hence has FdPorts and where that FD and supports FCs, appropriate FdPorts of that FD support the corresponding FcPorts.

3.2.3 ForwardingConstruct (FC)

Qualified Name: CoreModel::CoreNetworkModel::ObjectClasses::ForwardingConstruct

The ForwardingConstruct (FC) class models enabled constrained potential for forwarding between two or more FcPorts at a particular specific layerProtocol .
Like the LTP, the FC supports any transport protocol including all analogue, circuit and packet forms.
For digital layer networks it is used to effect forwarding of transport characteristic (layer protocol) information.
An FC can be in only one ForwardingDomain (FD).
The FC 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 structures for linear, ring or mesh protection schemes.
When applied to media, the FC represents the ability for a flow/wave (potentially containing information), to be propagated between FcPorts.
The existence of a FC instance is independent of the presence (or absence) of a flow/wave (and any information encoded within it).
A flow/wave cannot propagate in the absence of a FC instance.

Inherits properties from:

- GlobalClass
- ForwardingEntity

Table 5: Attributes for ForwardingConstruct

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
layerProtocolName		The layerProtocol at which the FC enables the potential for forwarding.

 © 2018 Open Networking Foundation

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_lowerLevelFc		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. FC aggregation reflects FD aggregation. The FC represents what would have traditionally been considered as a "Cross-Connection" in an "NE". The "Cross-Connection" in an "NE" is not necessarily the lowest level of FC partitioning.
_fcPort		The FcPorts define the boundary of the FC. The FC is accessed via the FcPorts. Flow within the FC is defined in terms of its FcPorts.
forwardingDirection		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. In the case of media the FcPorts and FC may also be omni-directional.
_fcSpecReference:ClassRef	SpecReference Experimental	Reference to the Spec (Class).

3.2.4 FcPort

Qualified Name: CoreModel::CoreNetworkModel::ObjectClasses::FcPort

The association of the FC to LTPs is always made via FcPorts.
In the case of media the association between FCs is made via their FcPorts and the association of an FC to the physical Pin is made via the FcPort.
The FcPort 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 ForwardingConstruct can be considered as a component and the FcPort as a Port on that component.

Inherits properties from:

- LocalClass

Table 6: Attributes for FcPort

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description

 © 2018 Open Networking Foundation

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_ltp		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-directional LTPs Zero LTP - BreakBeforeMake transition - Planned LTP not yet in place - Off-network LTP referenced through other mechanism.
role		Each FcPort of the FC has an assigned role (e.g., working, protection, protected, symmetric, hub, spoke, leaf, root) in the context of the FC with respect to the FC function. The role is fixed by the referenced FcSpec.
fcPortDirection		The orientation of the defined flow at the FcPort.

3.2.5 Link

Qualified Name: CoreModel::CoreNetworkModel::ObjectClasses::Link

The Link class models effective adjacency between two or more ForwardingDomains (FD).
For digital layer networks, 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).

Inherits properties from:

- GlobalClass
- ForwardingEntity

Table 7: Attributes for Link

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
layerProtocolName		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.
_fd		The Link associates with two or more FDs. This association provides a direct summarization of the association via LinkPort and LTP.

 © 2018 Open Networking Foundation

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_linkPort		The association of the Link to LTPs is made via LinkPort (essentially the ports of the Link).
_lowerLevelLink	Experimental	A Link may be formed from subordinate links (similar FD formations from subordinate FDs). This association is intended to cover concepts such as serial compound links.
linkDirection		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.
_fdRuleSet		The rules related to a Link.
_linkSpec	Experimental	See referenced class
_linkSpecReference:ClassRef	SpecReference Experimental	See referenced class

- 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
- A Link may offer parameters such as capacity and delay (see TR-512.4).
  - These parameters depend on the type of technology that supports the link.

3.2.6 LinkPort

Qualified Name: CoreModel::CoreNetworkModel::ObjectClasses::LinkPort

The association of the Link to LTPs is made via LinkPort.
The LinkPort 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.

Inherits properties from:

- LocalClass

Table 8: Attributes for LinkPort

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description

 © 2018 Open Networking Foundation

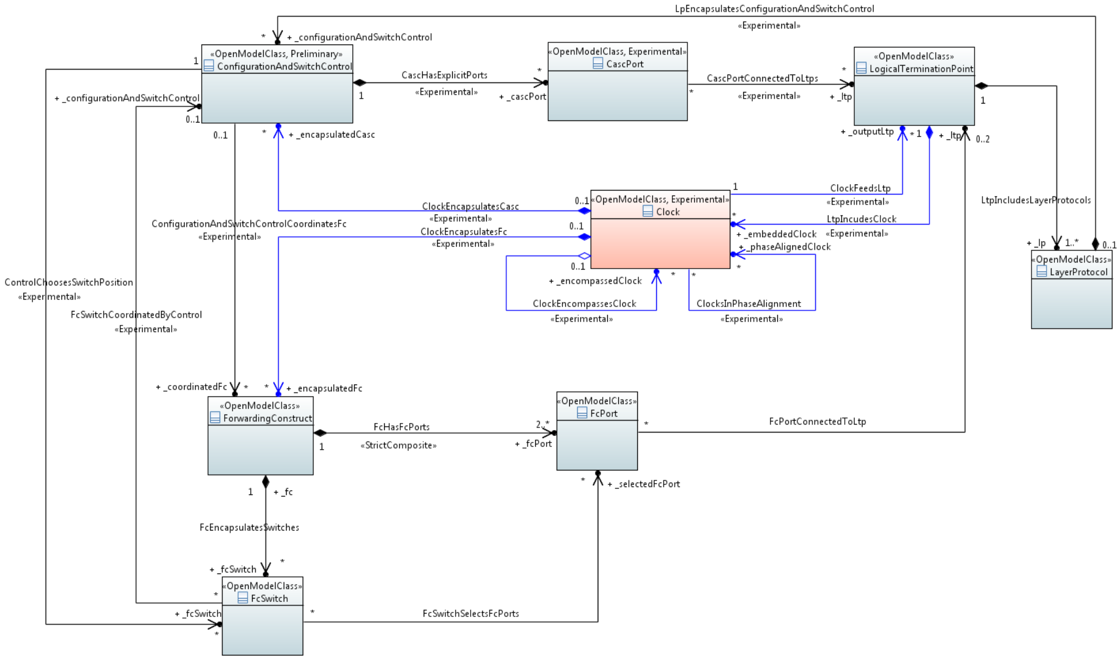
Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_ltp		The LinkPort may be associated with more than one LTP when the LinkPort is bidirectional and the LTPs are unidirectional. Multiple LTP - Bidirectional LinkPort to two Uni-directional LTPs Zero LTP - BreakBeforeMake transition - Planned LTP not yet in place - Off-network LTP referenced through other mechanism.
role		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 capability.
offNetworkAddress	Experimental	A freeform opportunity to express a reference for a Port of the Link that is not visible and hence is outside the scope of the control domain (off-network). 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 is a reference 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. This attribute is used when an LTP cannot be referenced. A Link with an Off-network end cannot be encompassed by an FD.
linkPortDirection		The orientation of the defined flow at the LinkPort.
_fcPort	Experimental	Where a Link supports FCs each LinkPort of that Link supports the corresponding FcPorts.

3.3 Clock, Timing and Synchronization

Propagation of timing information (frequency, time or both) is a fundamental aspect of networking. The timing model includes a representation of the clock used to provide timing for the functionality of a device. For many applications the clock in a device needs to be synchronized to clocks in other devices (i.e. build a synchronized network). The model represents the control of capability that allows for the clock be synchronized with other devices that provide a timing signal and of capability that allows the clock to provide a timing signal to other devices.

The timing signals are supported by various network protocols and synchronization is achieved by various techniques. The model described here is a generalized model that can be applied to any network protocol (see TR-512.A.8 for more details). For any particular case the specific properties for the corresponding protocol/technique via the specification approach (see TR-512.7).

 © 2018 Open Networking Foundation



CoreModel diagram: Synchronization-ClockInContext

Figure 3-3 Clock related to LTP, C&SC and FC

The figure above shows the clock (highlighted in red) and associations (highlighted in blue) that form the clock model.

Qualified Name: CoreModel::CoreNetworkModel::ObjectClasses::Timing::Clock

Clock function processes the input sync information (frequency and ssm or time stamp and PTP announce messages) and provides the modified sync information to the sync distribution function. If none of the inputs meet the quality defined by the controller the clock may enter a hold-over or free run mode.
The status of the clock will be reported to the controller.

Inherits properties from:

- LocalClass

This class is Experimental.

Table 9: Attributes for Clock

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
runMode	Experimental	The run-mode of the frequency system clock, such as free-run, locked, and holdover.
_encompassedClock	Experimental	A clock may be emergent from and may effectively encompass several clocks in a resilient solution.

 © 2018 Open Networking Foundation

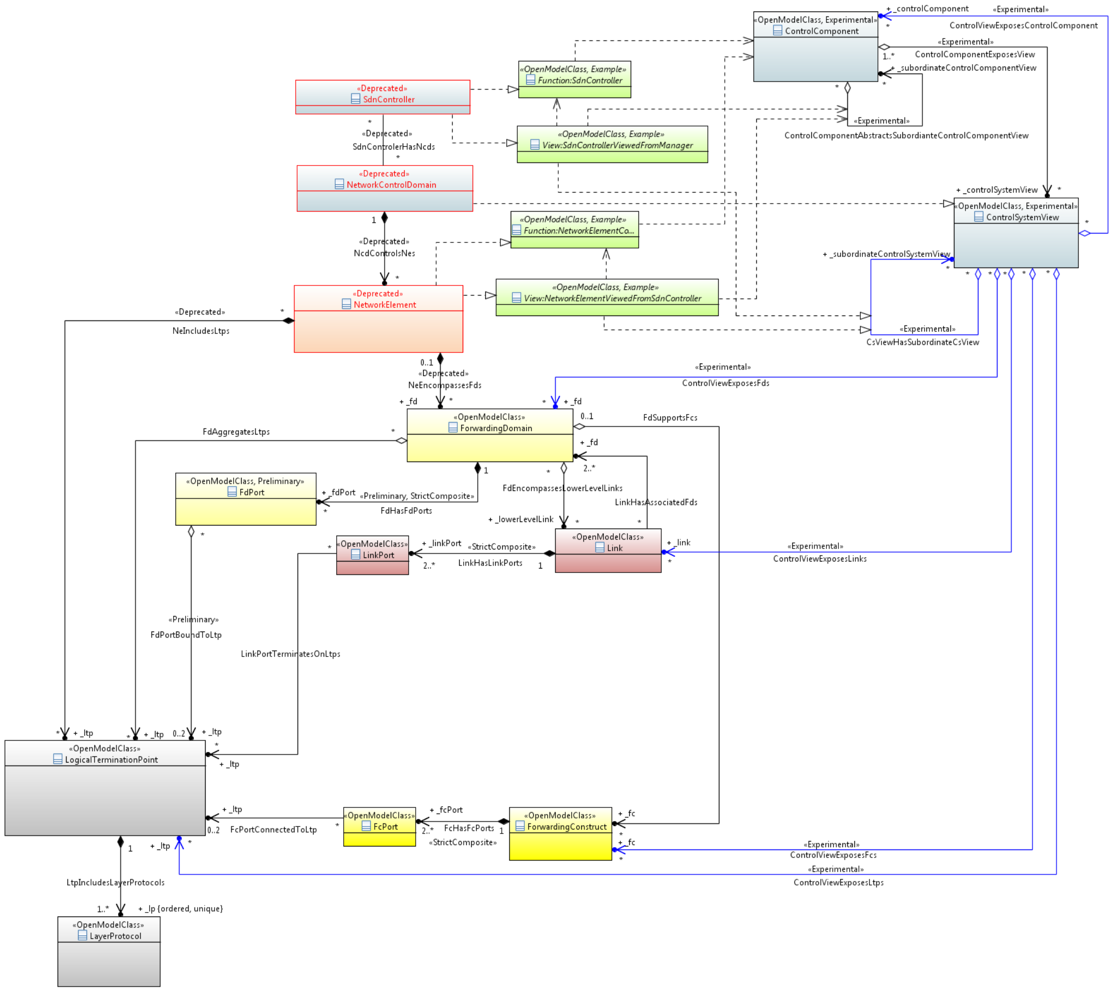
Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_encapsulatedFc	Experimental	A Clock may encapsulate an FC related to resilience where the clock provides an output that is essentially that of one of several other clocks in the resilience scheme.
_outputLtp	Experimental	A clock may feed one or more LTPs with timing information to propagate across the network (it may feed no LTPs).
_encapsulatedCasc	Experimental	The clock may encapsulate a complex FC where there is a resilience mechanism active and that FC will need to be controlled. The Casc to control the FC can be encapsulated in the Clock.
_phaseAlignedClock	Experimental	One or more clocks can be actively phase aligned (this is especially relevant in a hitless resilience scheme).

3.4 NetworkElement, NetworkControlDomain and SdnController

As explained in TR-512 V1.2 the classes SdnController, NetworkControlDomain and NetworkElement[4] have been reassessed and new classes have been developed in this release. The figure below shows the relationship between the V1.2 classes (that have been deprecated – highlighted with red text and borders) and the new V1.3 classes via some expanded example classes (highlighted in green) that show two aspects of a control entity, the controller itself and the view of the controller.

[4] 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).

 © 2018 Open Networking Foundation



CoreModel diagram: Forwarding-SkeletonWithControllers

Figure 3-4 Skeleton Class Diagram of key object classes showing Controllers

The new model is explained in full detail in TR-512.8. The key consideration is that the ControlComponent (SDN Controller) exposes a ControlView which includes via aggregation (highlighted in blue) all controlled entities (where a controlled entity is allowed to be in many ControlViews). The SDN Controller exposes itself as a ControlComponentView and also exposes the NEs as SubordinateExposedViews which provides a further ControlView that includes the NE ControlComponentView (i.e. the control aspects of the NE) and the aggregation of the subset of the entities from the SDN Controller ControlView that the NE controls. These are presented in the terminology and naming of the SDN Controller.

4 Explanatory Figures

This section provides figures that illustrate the application of the model to various network scenarios. The section covers both forwarding and termination. The forwarding views are relatively lightweight. More sophisticated forwarding views are provided in TR-512.4 and TR-512.5.

 © 2018 Open Networking Foundation

For an explanation of the symbol set being used in the figures see section 1.3 Conventions on page 6.

4.1 Forwarding

4.1.1 Basic Forwarding

The basic forwarding model, described in previous sections, offers the capability to enable constrained forwarding between LTPs. The figure below provides a basic nodal view.

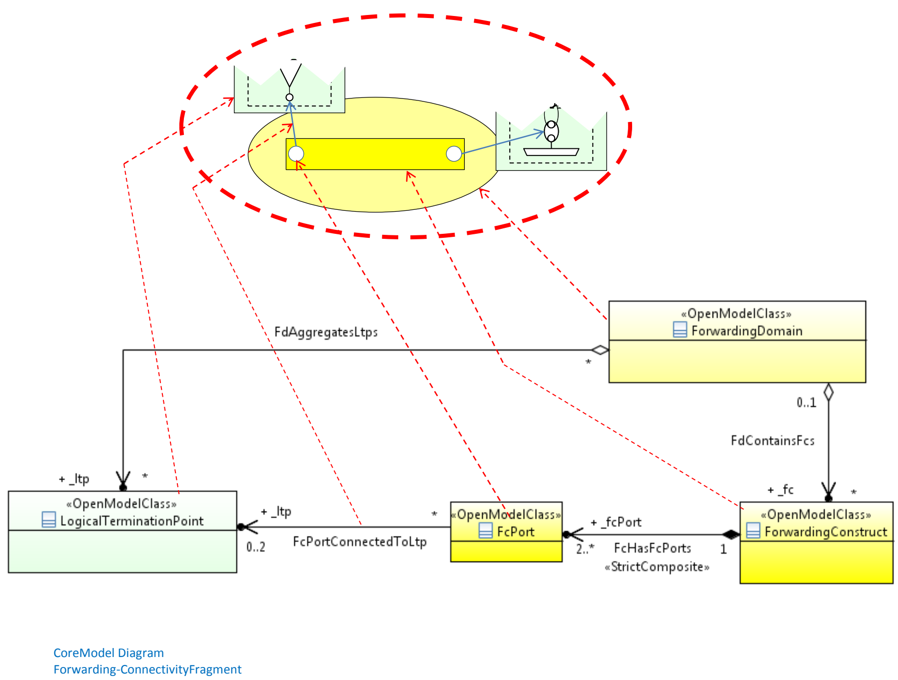


Figure 4-1 Forwarding fragment in a nodal view

The pictorial form in the figure above shows the ForwardingConstruct (FC) in the context of two LTPs. The FC defines the enabled constrained forwarding between the LTPs (in the figure it is point to point). 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.

4.1.2 Forwarding topology

The FC defining the enable constrained forwarding between a set of LTPs can be considered in the context of a network topology offering capacity.

 © 2018 Open Networking Foundation

The figure below shows a network for a single layer protocol in terms of the basic topology of FDs, Links and LTPs (grey) that provide capability and capacity for the layer protocol and the signal forwarding using FCs (X, Y and Z) and LTPs (green) enabling information flow for the layer protocol.

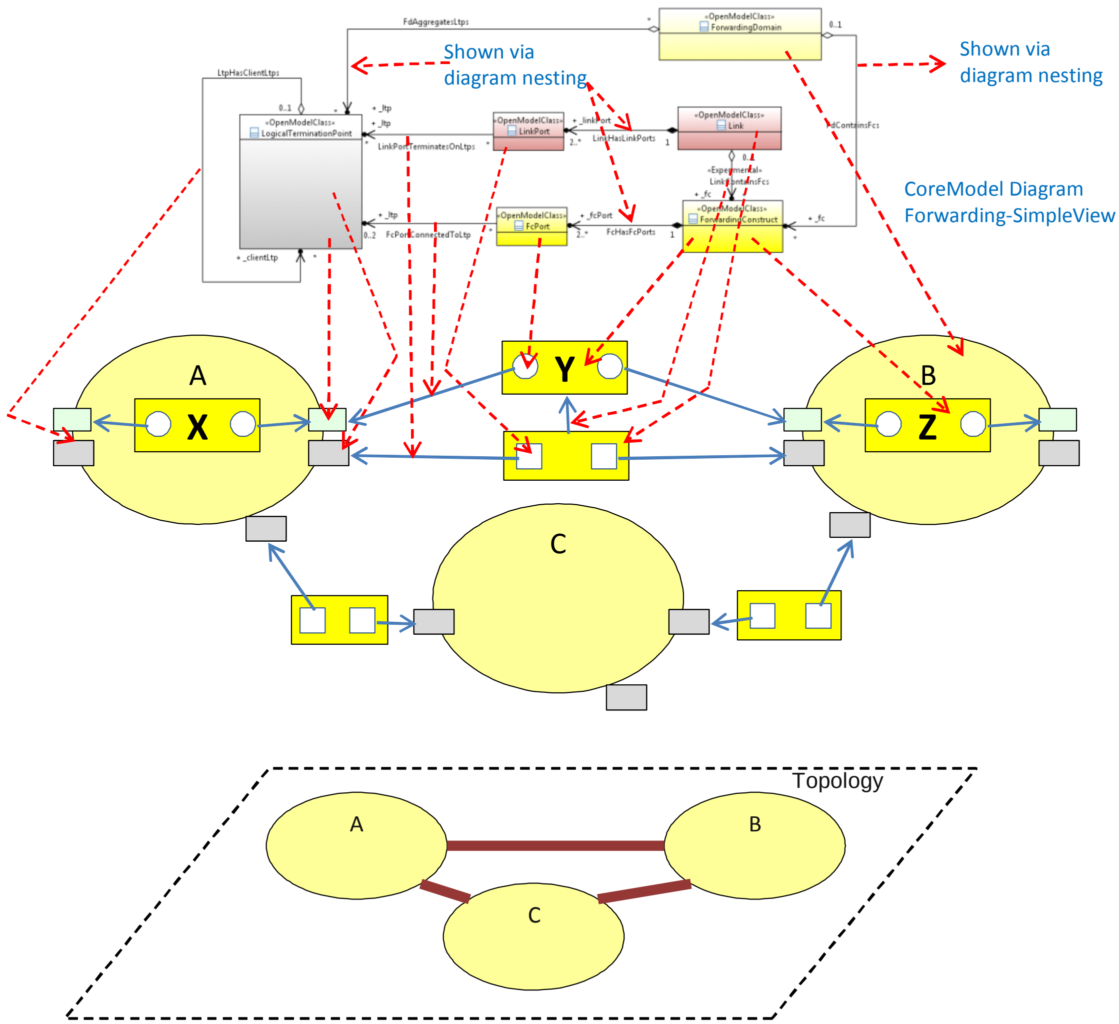


Figure 4-2 Forwarding in a single layer

The following section deals with LTP layering considered in the context of singe FDs. More sophisticated multi-layer multi-FD and multi-view considerations are covered in detail in TR-512.4.

4.1.3 Forwarding in the media layer

Consider media (glass, copper, air etc) through which information is to be transferred. An appropriate medium is relatively transparent to some particular wave/particle flow (light, microwaves, electrons etc), i.e. when the medium is present it is inherently enabled to forward some specific flows "uninhibited".

 © 2018 Open Networking Foundation

Also consider a wave/particle flow that is modulated by a source of information such that the information can be recovered by observing the flow (and hence termination some or all of the flow) at some point remote from the source. When the modulated flow is applied to an appropriate medium it is possible to transfer information from a source to a distant receiver at the boundary of the medium.

On this basis and considering the definition of ForwardingConstruct (FC) it is clear that the effect of a medium can be represented by an FC and that the FC is always essentially present. The existence of an FC instance is independent of the presence (or absence) of a flow/wave and information encoded within it (if any). A wave/flow cannot propagate without an FC. In some cases the FC may support the propagation of the wave/flow, but the characteristics of the FC may prevent the transfer of information.

In general, a medium imposes degradation on the flow where the specific characteristics of the medium interact with the characteristics of the flow for example:

- Absorption causing loss of power
- Interference between flows causing loss of integrity
- Dispersion causing loss of integrity
- Adding noise causing loss of integrity
- Restricting bandwidth (frequency slot) causing loss of information integrity

As flow takes place in a length of a medium it can be represented by an FC with certain characteristics.

On some occasions the characteristic can be used to advantage. For example the interaction that takes place in an Erbium doped fiber causes amplification of the power in one flow.

We could represent fibre adjacency with a media Link/FD and hence the bridge between the (atomic/static) physical consideration and the photonic functional considerations. A medium will have some non-linear characteristics. Because the medium can modify the flow in a complex way, and considering that there is media both within a device and between devices the choice of whether to represent an element supporting the FC as Link or FD is driven purely by its position.

A physical medium is passive in nature, but when stimulated with the appropriate flows, both linear and non-linear characteristics of the medium cause complex activity that yields relevant emergent functionality. In some media the characteristics are such that:

- Power may be transferred from a flow of one characteristic to a flow of another characteristic in the medium
  - Linear
    - Electronic to photonic (e.g. a laser)
    - Photonic to electronic (e.g. a photodiode)
  - Non-linear
    - Photonic to Photonic (an optical amplifier)
    - Photonic to Photonic in a fibre (causing interference)
- Modulation may be transferred from a flow of one characteristic to a flow of another characteristic[5] in the medium

[5] The transfer may only account for one form of modulation (e.g. phase information may be lost)

 © 2018 Open Networking Foundation

- Electronic to photonic (e.g. a laser[6])
- Photonic to electronic (e.g, a photodiode)

On this basis a FC may be fed with inputs of different characteristics and as a result the power and/or modulation may transfer from one domain to another (e.g. electronic to photonic). There may also be a media change with no change of domain of flow (e.g. an air gap in an optical network where the light passes temporarily into a gaseous medium, e.g. at the junction between the medium of the laser and the fiber).

Clearly some media are constructed to minimize the undesirable effect of non-linear characteristics (e.g. the fiber between sites) and other media are constructed to take advantage of non-linear characteristics (e.g. Erbium Doped Fiber Amplifier (EDFA)).

A physical medium necessarily occupies three dimensional space. Within that space a flow can be in any direction and can spread. There is no simple relevant quantization, it is not slotted or grid based, it is essentially continuous. In some cases overlaying a conceptual grid structure is beneficial but this is a view and not an inherent aspect of the medium. The model does not assume any grid structure. If there is a relevant grid/band formulation this will be detailed in a specification model (as described in TR-512.7).

The "any direction" characteristic is termed "omni-directionality". To cater for this aspect the directionality enumerations have been extended appropriately. In some cases a medium will restrict the flow (due to diminished physical dimensions) to be only meaningful in one dimension (e.g. an optical fiber), but the case is still considered as omni-directional. Further restrictions using appropriate active elements will prevent flow in one direction in an essentially one dimensional case. Under these circumstances the more usual unidirectional representations will apply. As is the case for other layers, bidirectional is applied to an FC that is an abstraction of an assembly of two oppositely directed unidirectional FCs.

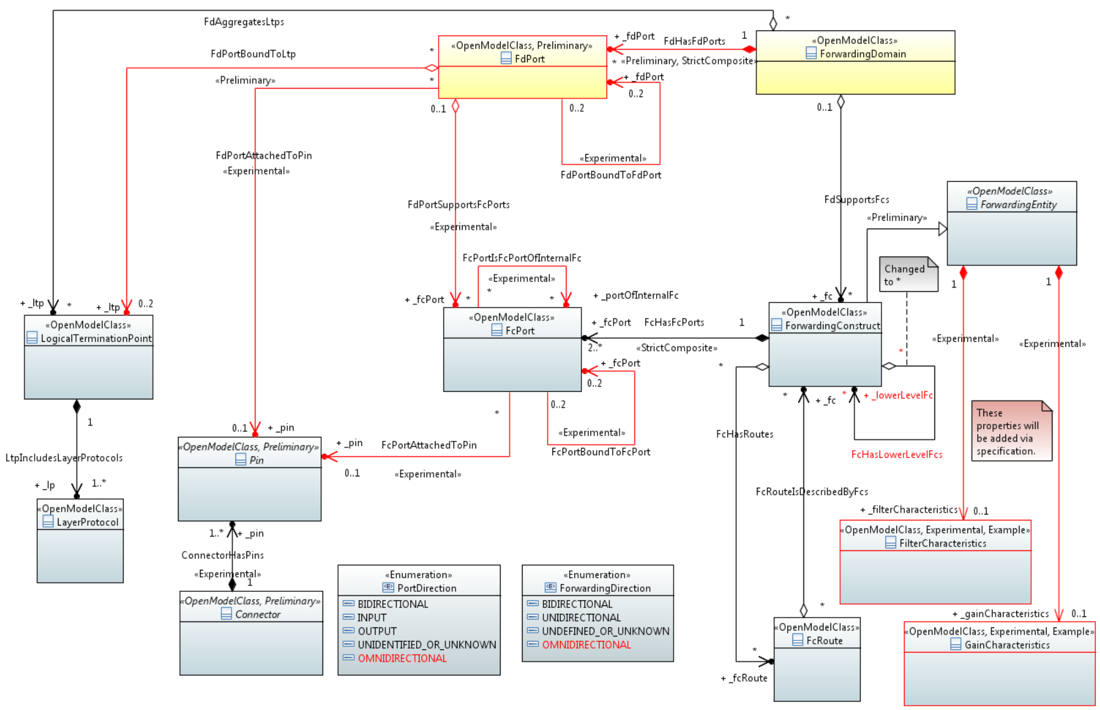
In the media layer, bidirectional abstractions are normally only used as higher-level abstraction for the purpose of managing network connectivity[7]. For other purposes, bidirectional representations are rarely useful and in most cases omni-directional or unidirectional representations are used.

The following figure highlights the key enhancements made to the model to support media.

[6] The medium in a laser has relatively high impedance to photons compared to glass and to electrons compared to copper but in this case the key is transfer of power/modulation from the electronic to photonic domain. The change of domain happens within the medium.

[7] For example, in a higher level control/management system it may be convenient to associate a pair of flexible grid filters that are omni directional as a bidirectional entity where one filter is used for each direction of wave/flow propagation. This would allow a single command to cause the configuration of two FCs.

 © 2018 Open Networking Foundation



CoreModel diagram: Forwarding-EnhancementsRelatedToMedia

Figure 4-3 Enhancements related to media (highlighted in red)

The figure above highlights in red the additions and adjustments made to the model to accommodate media. There are two additional ForwardingEntity Pacs both of which provide information on analogue characteristics[8].

Often the media forwarding devices have a fixed number of FcPorts where the forwarding characteristics per port are variable. The following figure shows an FC that represents a multi-port tunable filter.

The figure shows internal FCs and uses the FcPortIsFcPOrtOfInternalFc association. The internal FCs could be represented as elements of a FcRoutes (rather than directly via the FcHasLowerLevelFcs), if they need to be explicitly exposed, or simply as bundles of attributes explained by the FcSpec where, amongst other things, the FcSpec would define the frequency and transfer characteristics (loss/gain, gain tilt (for an amplifier), chromatic dispersion, "distortion" etc)[9]. When using the spec the SubordinateForwarding specification mechanism is used (see TR-512.7).

[8] Depending on the application it may not be necessary to use some or any of the analogue characteristics. It is expected that the _Pac approach will be replaced by a Specification approach in the next release and this will provide the necessary flexibility.
[9] An indication if the parameter is inherently omni directional (i.e. independent of the direction of signal propagation) or "uni" directional will also be provided in the specification.

 © 2018 Open Networking Foundation

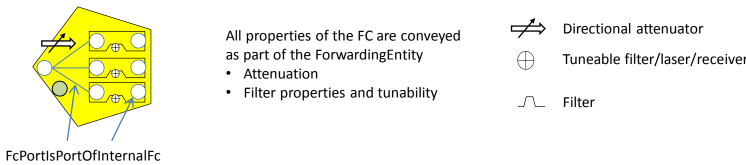


Figure 4-4 Broadband coupler/splitter with tuneable filter (FC contains FC)

Where the FcSpec is used, the FcSpec define internal routes and parameters of the route. Where the properties are fixed, they could be defined only in the spec as usual. So fixed coupler/splitters just need the FC with spec but variable devices also need route, reflecting the spec against which to hold the controls. Like FcSwitch this can be a sparse model so the Route FC is present unconnected or just the measure is provided against the main FC with NO route where the measure is numbered as per the spec and probably with respect to the port numbering.

As for all media, a fiber has a particular transfer characteristic that essentially filters incoming flow allowing only photons with characteristics in a relatively small range to pass almost unimpeded[10]. Photons well outside this range will not pass at all. A flow of photons with characteristics near the edge of the range may be attenuated and the phase characteristics of the modulated information may be impaired. The transfer characteristics of a fiber are complex. Photons within a range of characteristics that may readily pass are considered to be in a band.

A media network is constructed from an arrangement of units of media. Considering the omni-directional nature of the media, some of the units of media will broadcast a flow (splitting the power) and others will merge flows, as shown in the figure above. All units of media will apply some degree of filtering. A band of characteristics for which there is minimal attenuation can be considered as a channel through which a flow can pass relatively unimpeded. A particular medium may have several channels (as it may have several bands).

To determine if a flow of particular characteristic will pass through a chain of units of media in a network the characteristic of those units must be accumulated appropriately. For the filtering characteristics, the method of accumulation can be considered as intersection. The intersection of channel definitions for each unit of media defines the channel through the chain. [ITU-T G.872] defines the term these channels to be "media channel".

As discussed earlier, an FC is used to represent the enabled flow. An FC is therefore used to represent a channel through a medium[11] and also through any chain of units of media. As a chain can be considered as starting anywhere and ending anywhere, this leads to the FC being a grouping of any series of FC and hence grouping FCs are NOT in a simple hierarchy.

[10] There is considerable complexity in the characteristics of a medium that has been ignored here especially around the edges filter pass band. The notion of a frequency slot is not fully covered.
[11] The FC is also used to represent all parallel channels through the medium.

 © 2018 Open Networking Foundation

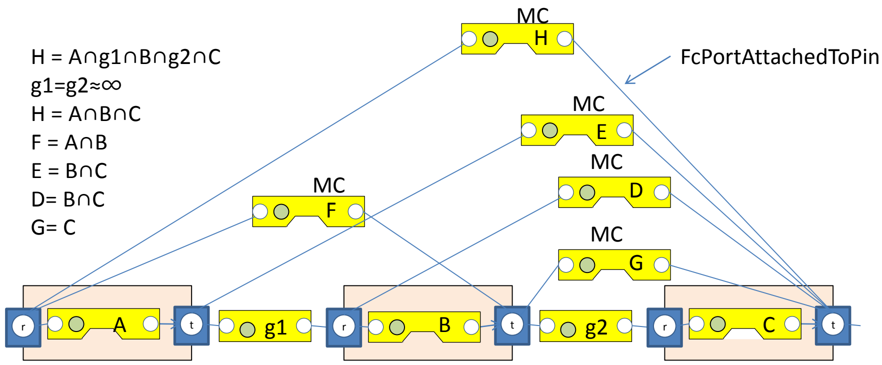


Figure 4-5 Chain of filters and fibers

The figure above shows a simple arrangement of units of media represented as FCs and emphasizes the non-hierarchical nature of the treatment. The filter characteristics of the units of media are considered and such that the characteristics of an FC is the intersection of the characteristics of the FCs it is composed of. The FcPort to Pin of the Connector is explained in section 4.5 Relationships to the physical port on page 50.

The figure below shows a view of the same chain of filters and fibers with a purely FC based abstraction. The figure illustrates the use of two different associations between FcPorts, one allowing chaining of FCs and the other allowing nesting of FCs. This abstraction could also be applied to layers other than media to provide a flow only view where the specific termination considerations are not relevant and where the connected FCs are alternately in a Link and in an FD (potentially in a Route context).

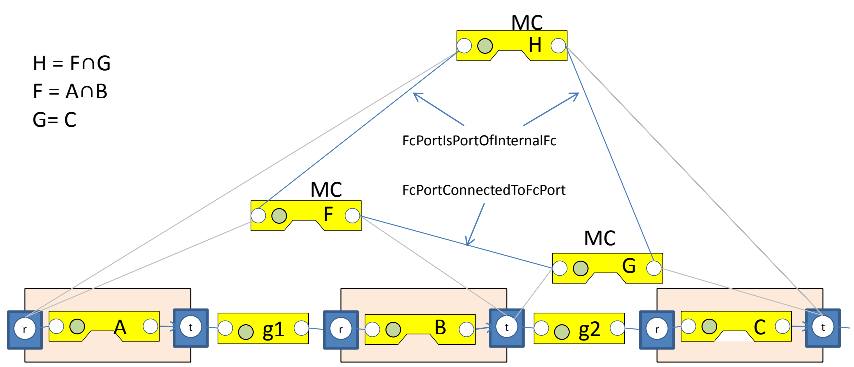


Figure 4-6 Abstraction of chain of filters and fibers

The figure below illustrates a more complex arrangement with a splitter/coupler and shows three FCs that have a point in common to the left of the diagram and show the effects of the individual ports of the coupler splitter when viewed at the points to the right of the diagram.

 © 2018 Open Networking Foundation

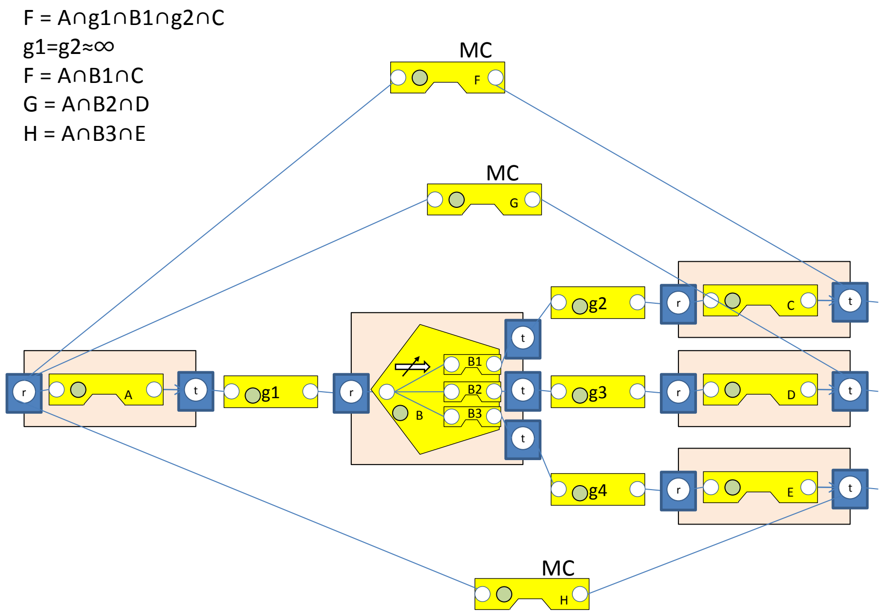


Figure 4-7 Complex assembly of chains of filters and fibres

4.2 Termination

4.2.1 Cases of LTP and LP

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 a specification as described in TR-512.7.

 © 2018 Open Networking Foundation

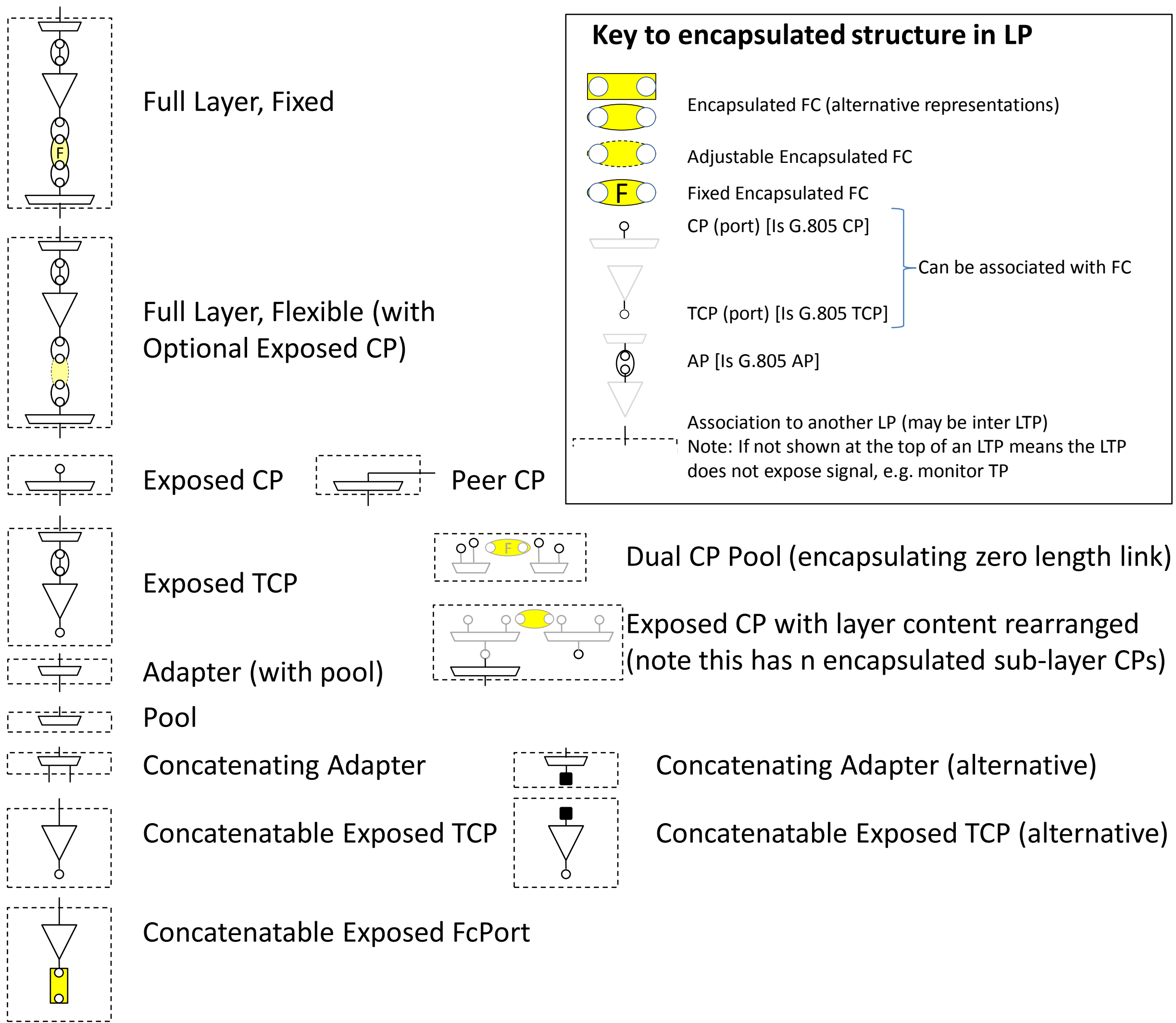


Figure 4-8 LP Cases

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 TR-512.TM.

 © 2018 Open Networking Foundation

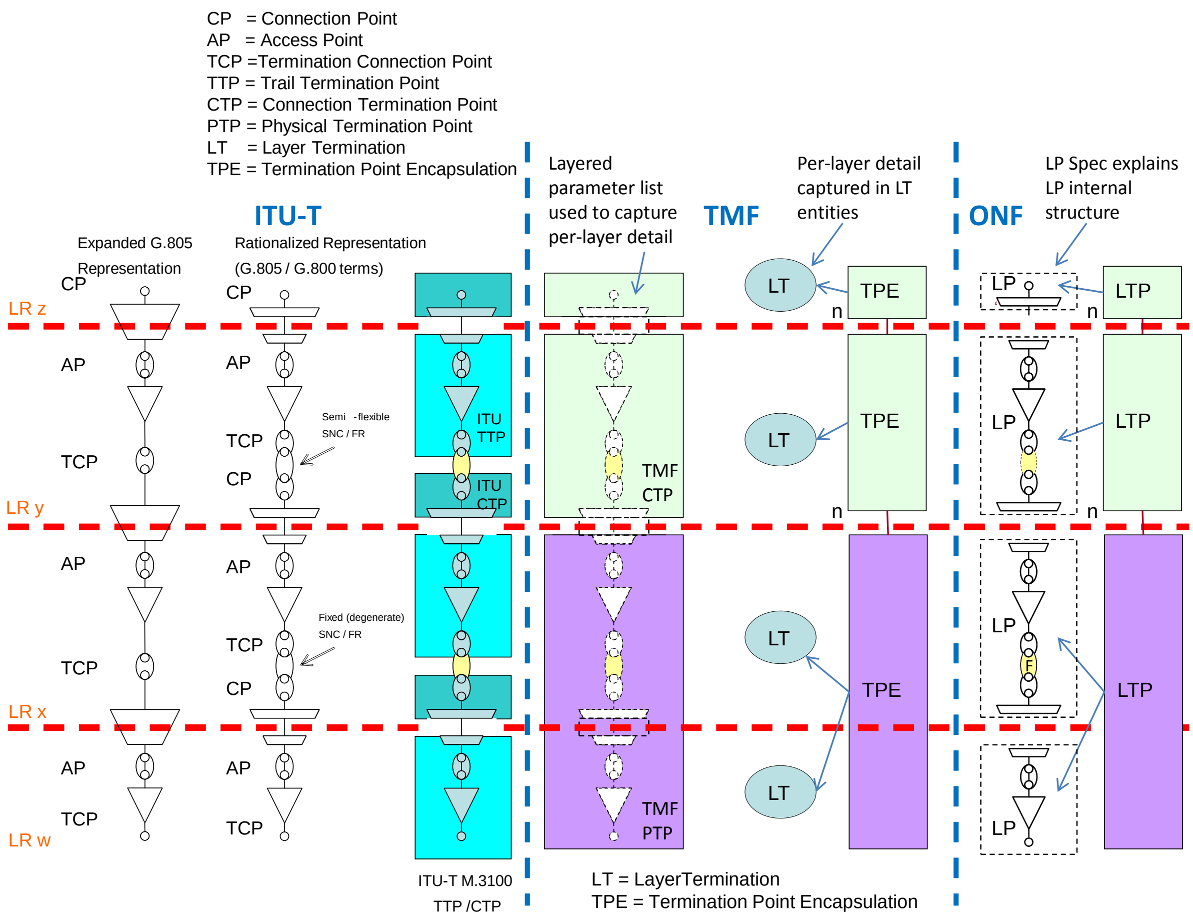


Figure 4-9 Mapping from ITU-T and TM Forum Termination models to the ONF Core[12]

[12] 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).

© 2018 Open Networking Foundation

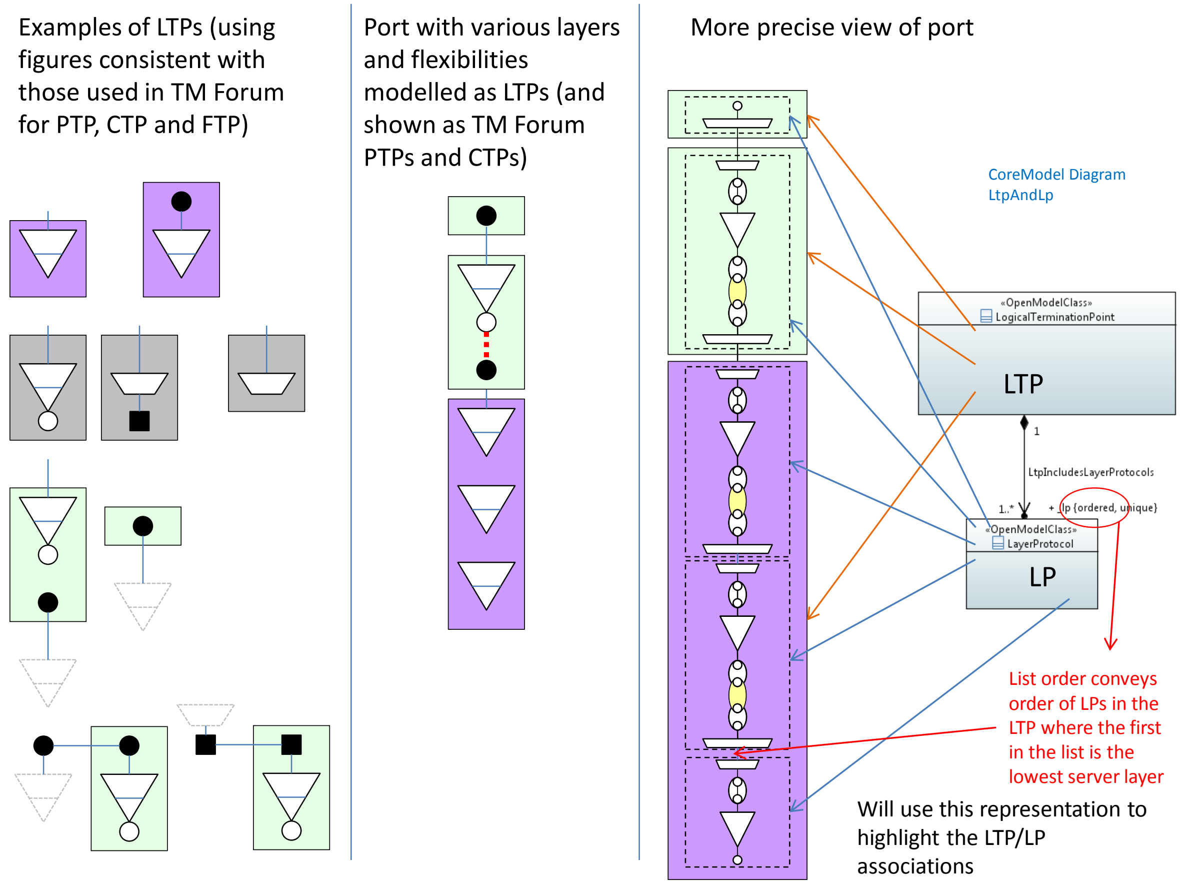


Figure 4-10 Representations of LTPs

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.[13] 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 reason for this split includes multiplicity, connection flexibility and flow orientation transitions (see also 1.3 Conventions on page 6 for reference to the diagram keys etc).

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 TR-512.7) which is currently experimental provides the basis for formal representation in a following release.

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

 © 2018 Open Networking Foundation

- 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 4-12 LTP relationships illustrated in a simple Network Element context on page 34) and the corresponding LP list order in the LTP
    - In the figure above, the Sink[14] 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 the figure below.

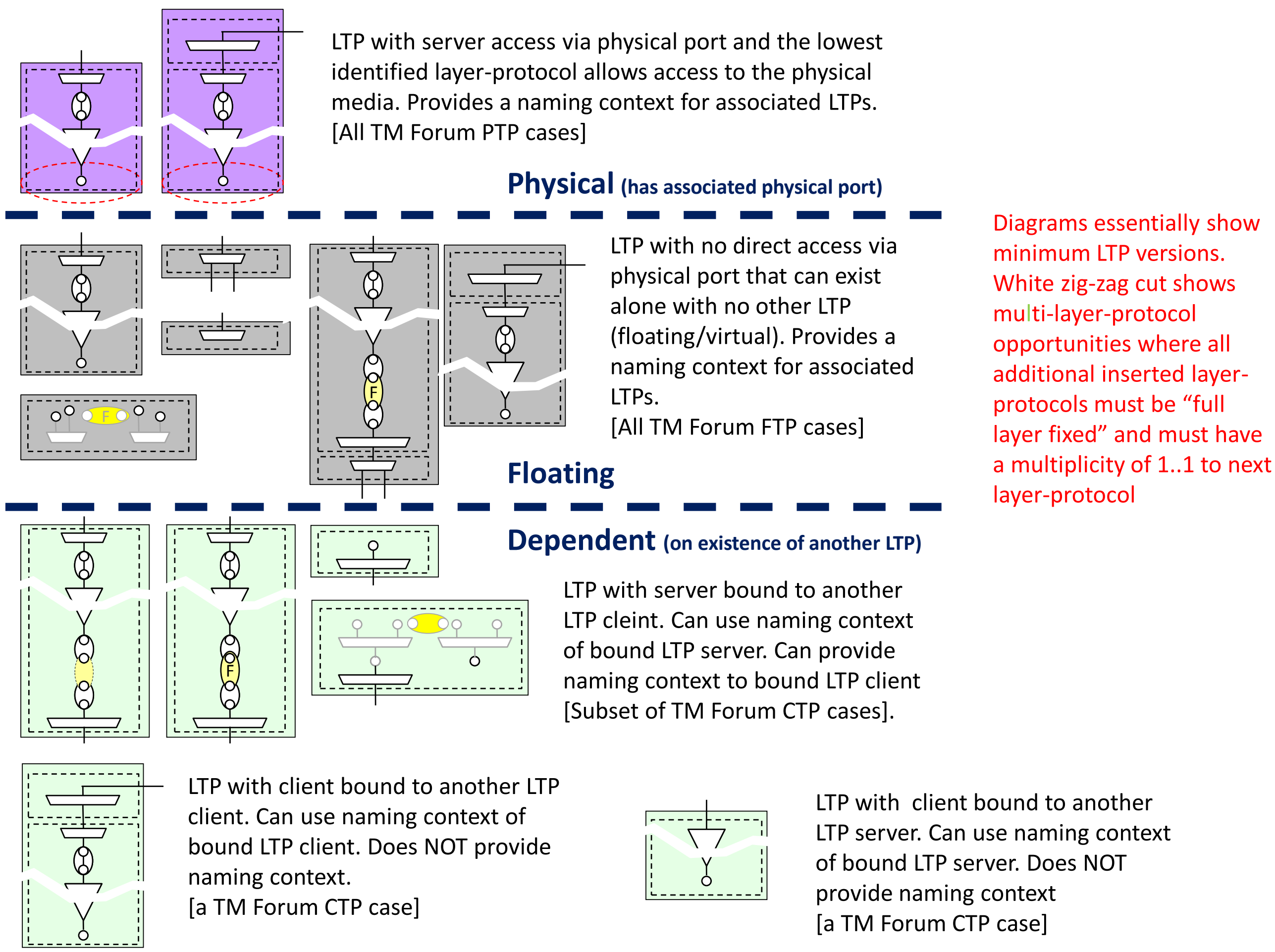


Figure 4-11 LTP Cases

[14] See section 4.4 Directionality on page 56.

 © 2018 Open Networking Foundation

4.2.2 LTP in an NE context

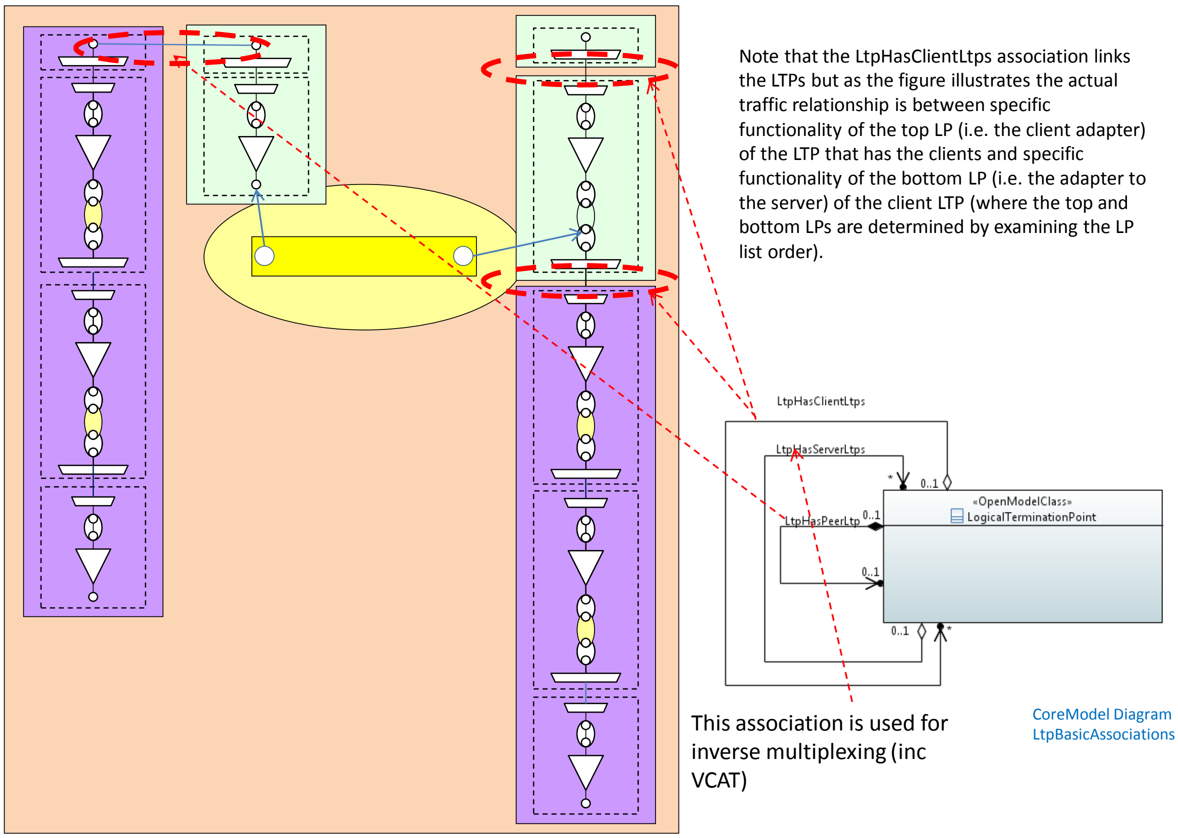


Figure 4-12 LTP relationships illustrated in a simple Network Element context

In the figure 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.

 © 2018 Open Networking Foundation

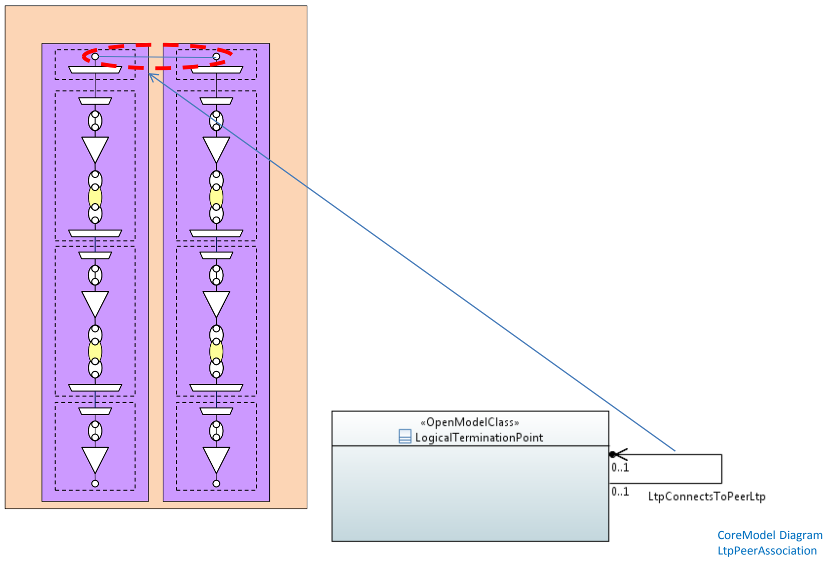


Figure 4-13 LtpConnectsToPeerLtp illustrated in an Amplifier/Regenerator context

In the figure 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.

The following figure shows a standard case of an FC between two LTPs (green) which are clients of LTPs (purple) where those LTPs support multiple clients.

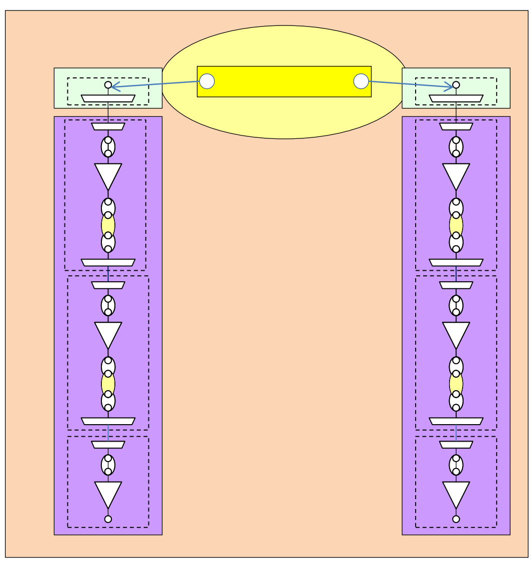


Figure 4-14 FC between LTPs

 © 2018 Open Networking Foundation

The following figure shows a standard case of an FC between two LTPs (purple) where there is forwarding flexibility but the LTP supports only one signal flow.

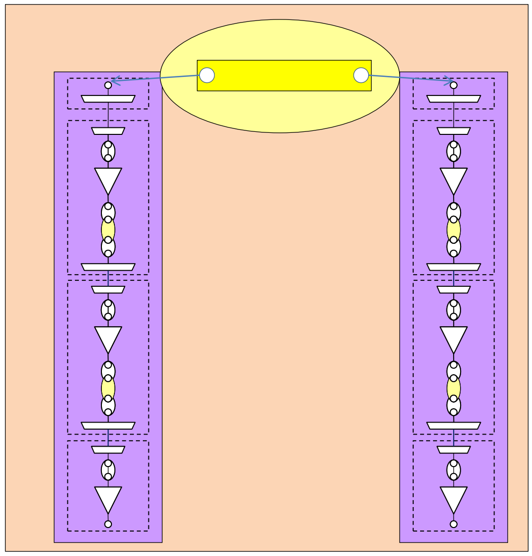


Figure 4-15 FC between LTPs supporting only one flow

4.2.3 Inverse multiplexing

It is sometimes necessary to carry a single information flow of that has a particular characteristic rate over a network where the bearers are too small to carry that rate of information transfer. Under these circumstances it is necessary to use a mechanism that divides the information flow into parts to be conveyed over several of the bearers in parallel such that it can be reassembled at the far end of the bearer into a flow that is indistinguishable from the original.

The dividing of an information flow into parts is called Inverse Multiplexing. There are a number of different schemes for inverse multiplexing (Link Aggregation Group (LAG), Virtual Concatenation (VCAT) etc). Some schemes take advantage of other characteristics of the information flow such as the packet nature. The scheme provides distinct properties and also distinct measures. Regardless of the specific scheme the essential model is the same.

In the case of the LAG it is possible to use some of the bearers to protect others by simply overprovisioning. Again, this does not change the essential model but may change the encapsulation and certainly affects the parameters and measures.

In the figure below:

- The "Expanded Representation" diagram shows a view of the essential model of Inverse Multiplexing as an arrangement of basic generalized functions.
    - The FC is shown with a selector that operates at signal rate selecting fragment by fragment from different inputs (where the fragments may be packets, frames, frame fragments) and feeds this as a stream towards the client.

  This form is overly complex and there is opportunity for simplification
- The "Encapsulated FC and CSC" diagram shows the chosen simplified form where the C&SC and the FC have been encapsulated in the LTP

 © 2018 Open Networking Foundation

   - This encapsulation could be exposed within the spec of an LP of the LTP or could be summarized as attributes of the LP of the LTP

  This is the model for Inverse Multiplexing
- There are two specific cases shown dealing with different multiplicities
   - 'n clients and n "channels" on the server' shows the use of the full "Encapsulated.." model
   - '1 client and 1 "channel" on the server' shows the most reduced form

  The most likely case is 'n clients and 1 "channel" on the server.

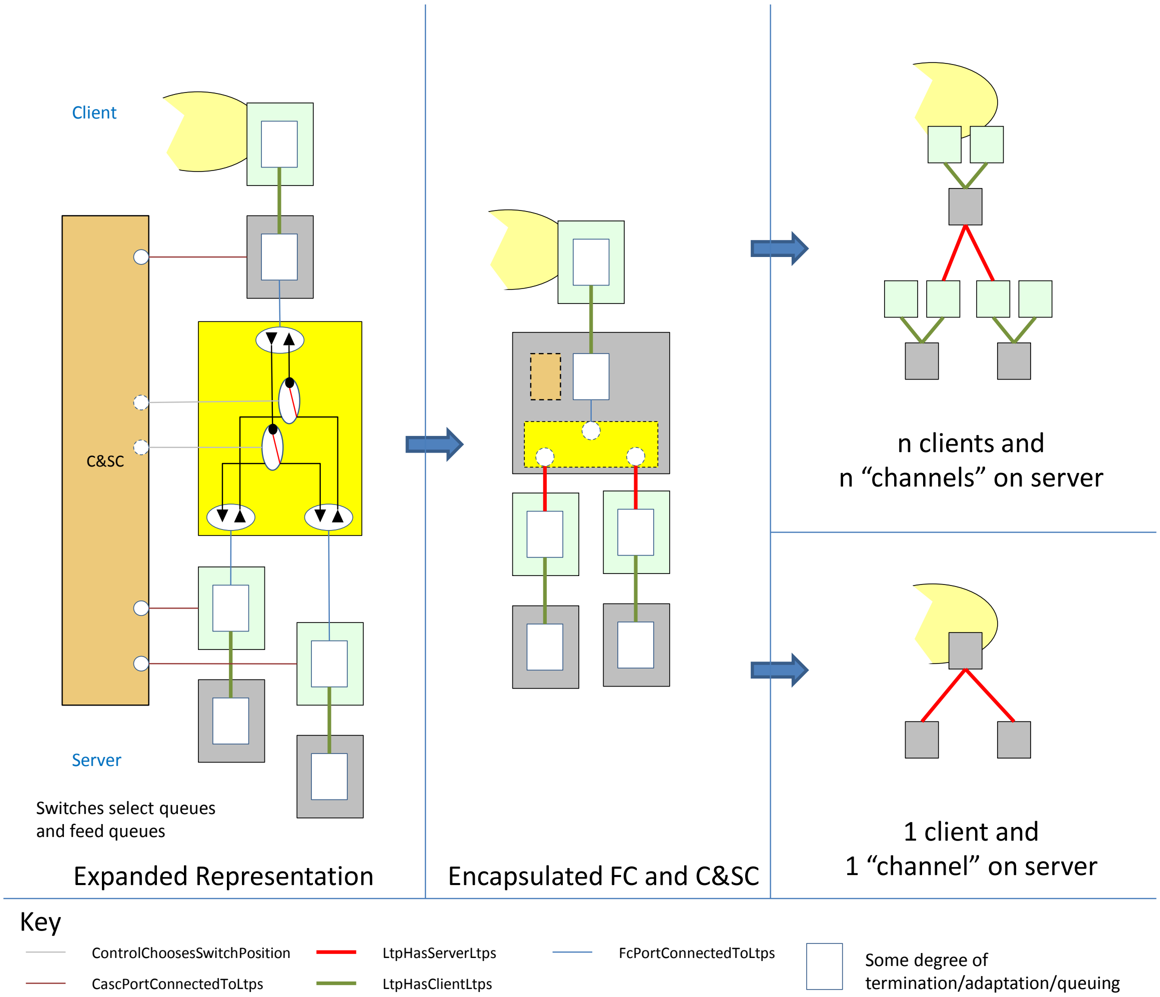


Figure 4-16 Representing Inverse Multiplexing

4.2.4 Clock, Timing and Synchronization

The following figure illustrates the essence of the timing synchronization flows in terms of the model classes and associations from a nodal perspective. The case shown is intentionally complex including clock protection.

 © 2018 Open Networking Foundation

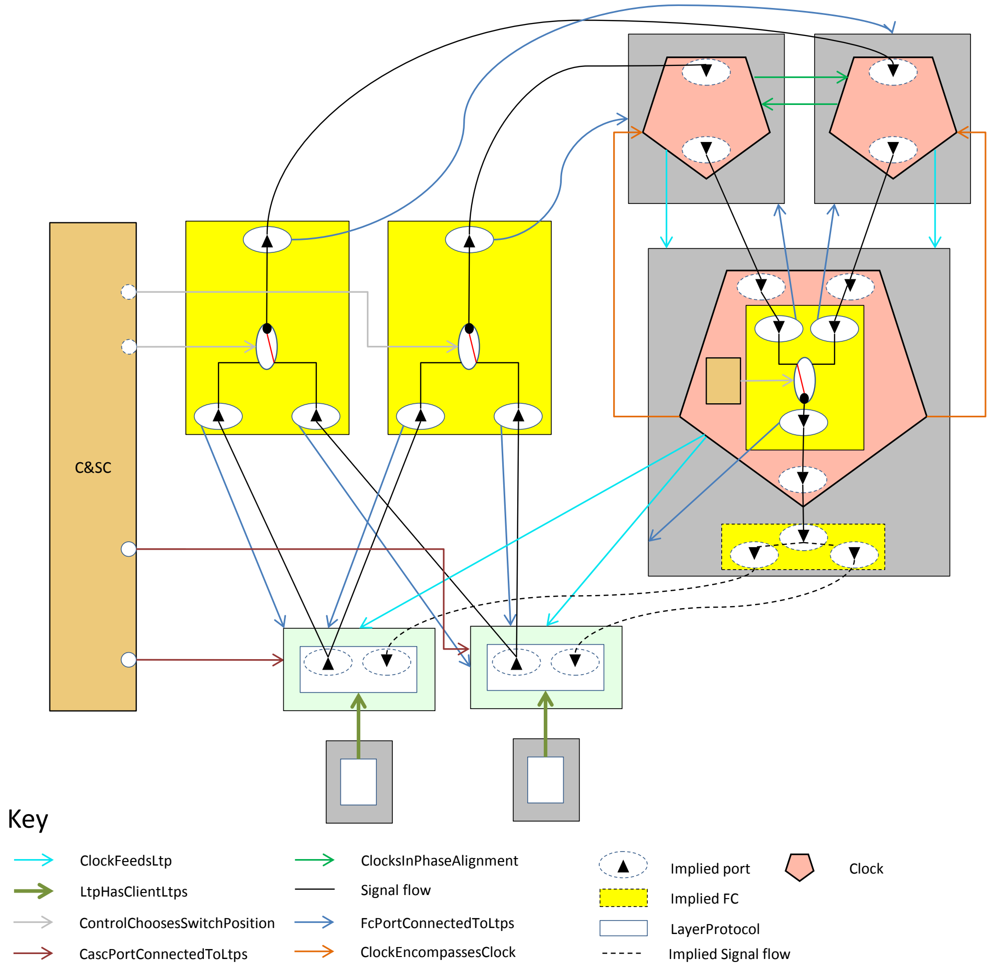


Figure 4-17 Clock and Timing Synchronization

The following figure shows only the model classes and associations (the signal flow is not shown).

 © 2018 Open Networking Foundation

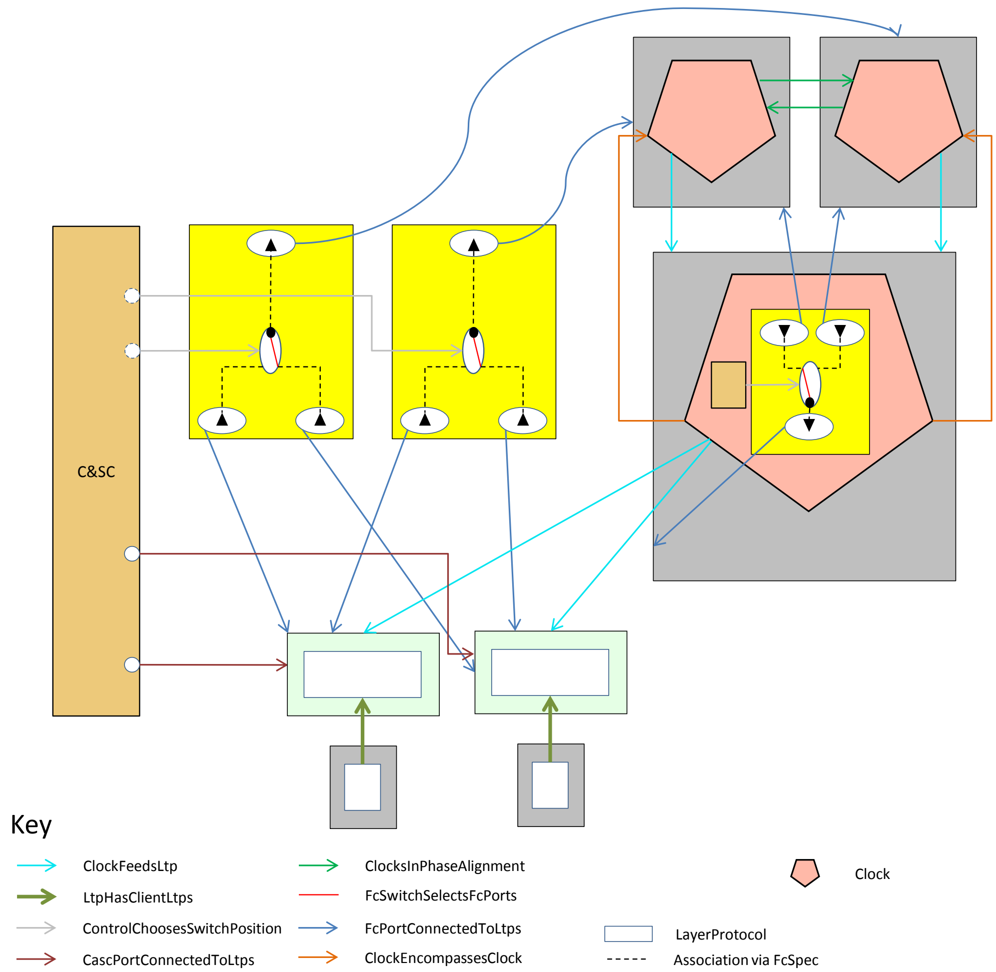


Figure 4-18 Clock and Timing Synchronization showing model only

The following figure shows a simpler case where there is only a single clock in the device.

 © 2018 Open Networking Foundation

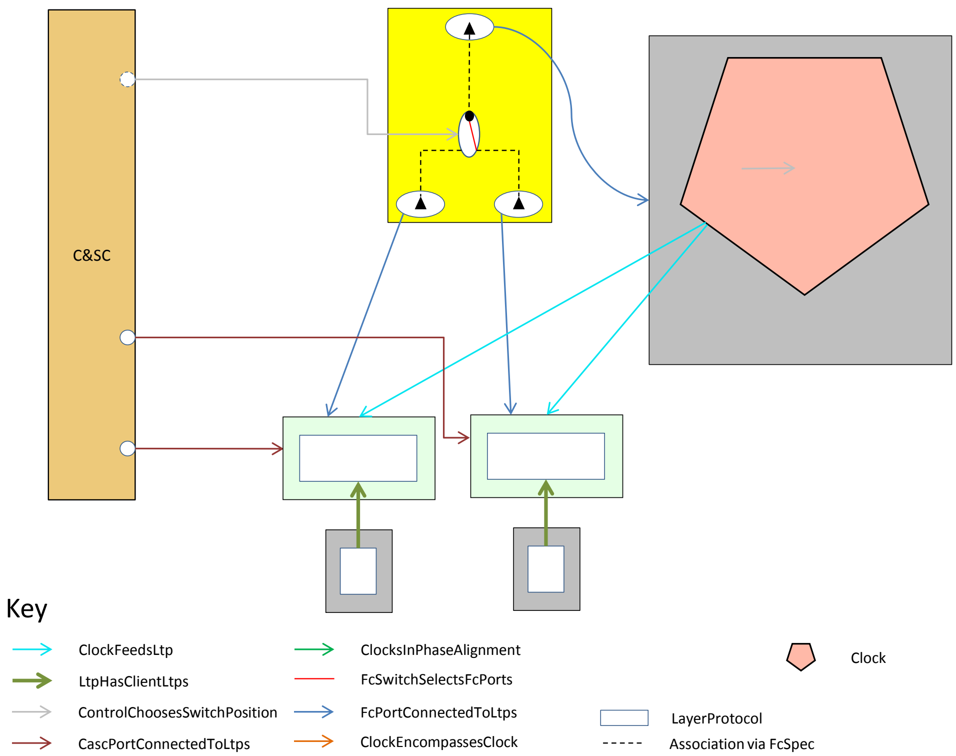


Figure 4-19 Clock and Timing Synchronization with a single clock showing model only

The detailed representation shown above is somewhat cumbersome. A more compact approach is to use a combination of SchemeSpec (see TR-512.7) and ProcessingConstruct (see TR-512.11). The SchemeSpec is used to represent the pattern detail of the scheme and then a ProcessingConstruct form of the scheme can be related to the scheme providing fewer instances in a simpler form (a detailed view of a use of this technique is provided in TR-512.5 (for G.8032 protection).

Examples of use of the model are provided in TR-512.A.8 and TR-512.A.9.

4.2.5 Termination in the media layer

The following figure shows a laser with a back diode allowing direct measurement of the light from the laser. The actual lasing element is represented by an FC as discussed in an earlier section. The electronic functions are represented as terminations. The overall effect of the electronic to photonic transition is termination.

The electronic to photonic and photonic to electronic transitions bound the photonic media layer.

 © 2018 Open Networking Foundation

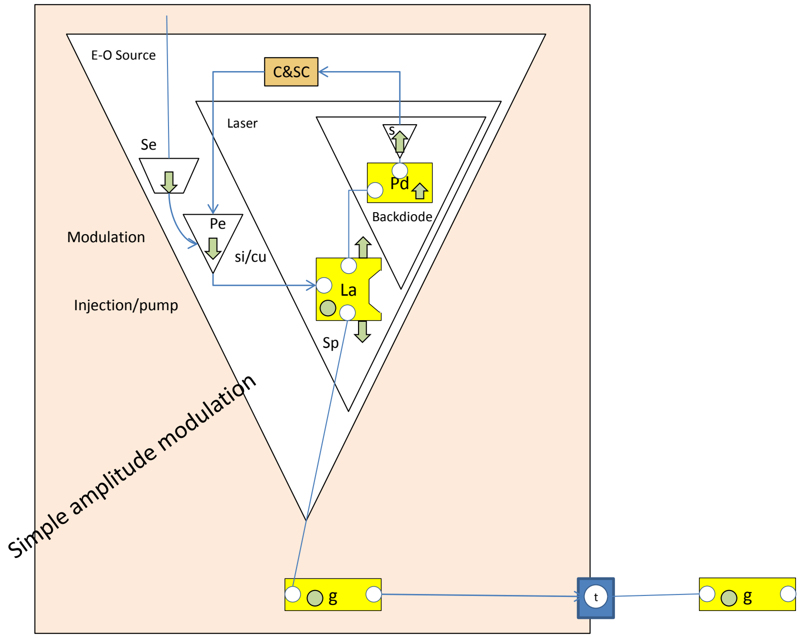


Figure 4-20 Laser as an active element (showing media)

The complex termination structures within the single LayerProtocol (outer triangle) of the LTP shown in the figure above are represented using the spec model. The inner triangles represent groupings of functionality that are more related to the physical structures. In an instance of LTP, the complexity is removed and the key properties are represented with attributes grouped to subordinate parts directed by the spec. The spec model is described in detail in TR-512.7. The application of the model to photonic media is explained in TR-512.A.4.

4.3 Network Considerations

This section highlights modeling of some simple network structures using LTPs and FCs in combinations. More complex network structures are covered in TR-512.4.

4.3.1 Media network

The figure below shows a simplified end to end view of a photonic network showing a single flow direction from left to right.

 © 2018 Open Networking Foundation

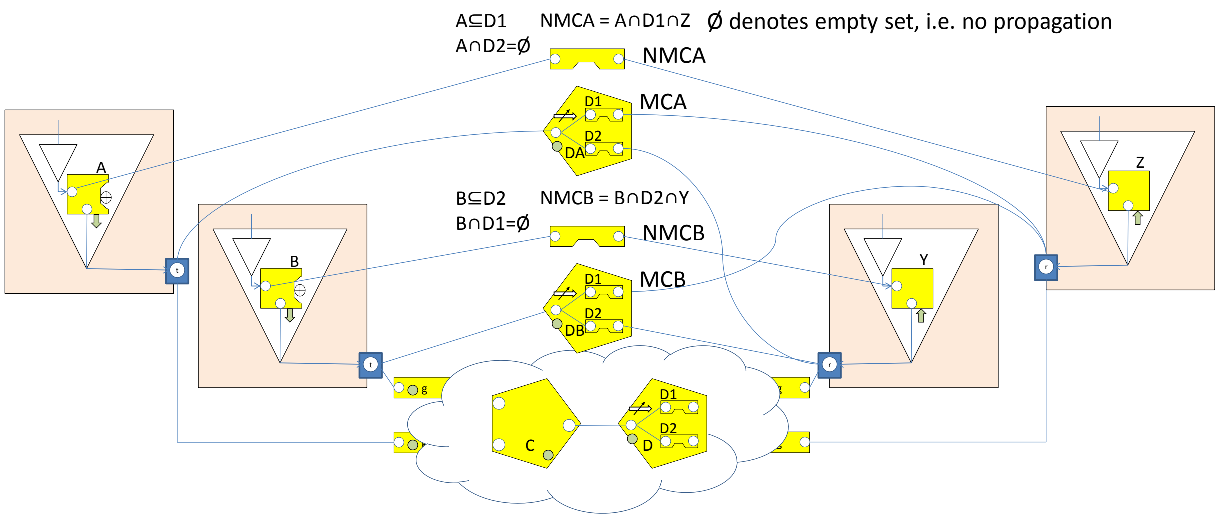


Figure 4-21 Network Media Channel formed from Media Channels

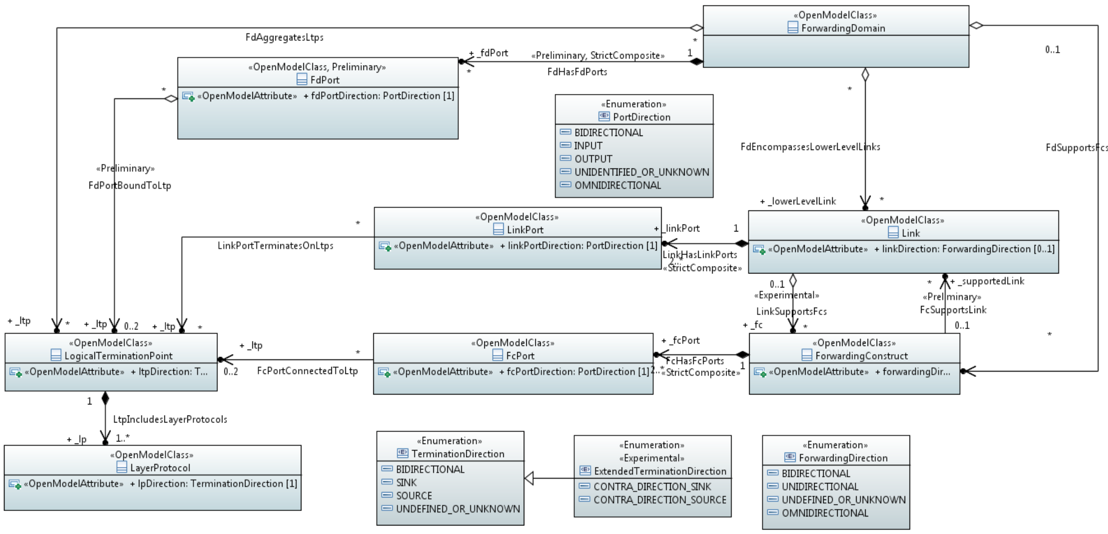
The Network Media Channel (the FCs NMCA and NMCB) is from the point of injection of electrons into the laser medium to the point of emergence of electrons from the photodiode[15]. The NMCs shown are formed as a result of the effects of the filters in the coupler C and splitter D which are reflected in the Media Channels MCA and MCB (both of which are FCs with three FcPorts). It is not until the lasers A and B are applied to the MCA and MCB that the effective NMCs can be determined. In the figure, Y and Z are wide band receivers. If A and B were tuned such that A⊆D2 (and hence A⋂D1=Ø) and B⊆D1(and hence B⋂D2=Ø), then NMCA would go from A to Y and NMCB from B to Z.

4.4 Directionality

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

[15] The case where an external modulator is used with the laser or a coherent receiver is used is described in TR512.A.4

 © 2018 Open Networking Foundation



CoreModel diagram: ForwardingConnectivityFragmentWithLtAndDirection

Figure 4-22 Model highlighting directionality

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

 © 2018 Open Networking Foundation

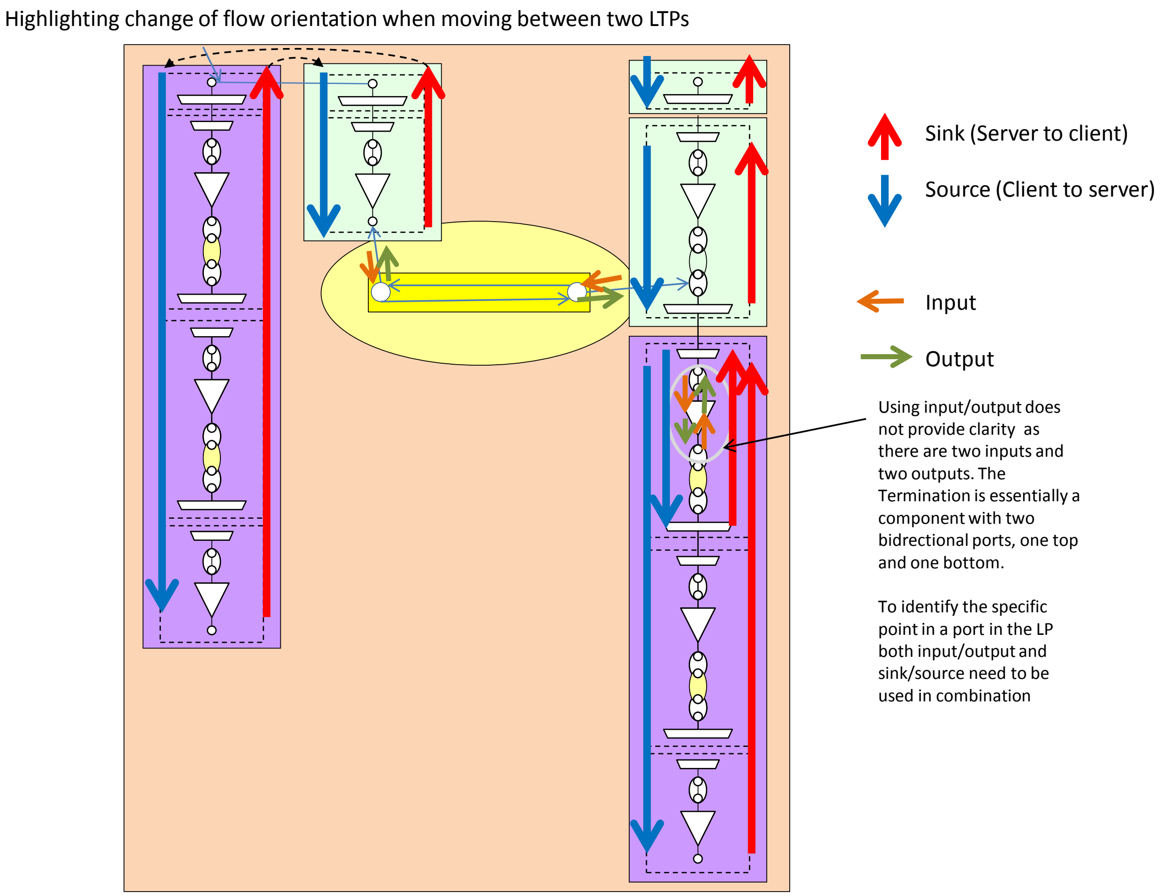


Figure 4-23 Interpreting the direction attributes

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.

 © 2018 Open Networking Foundation

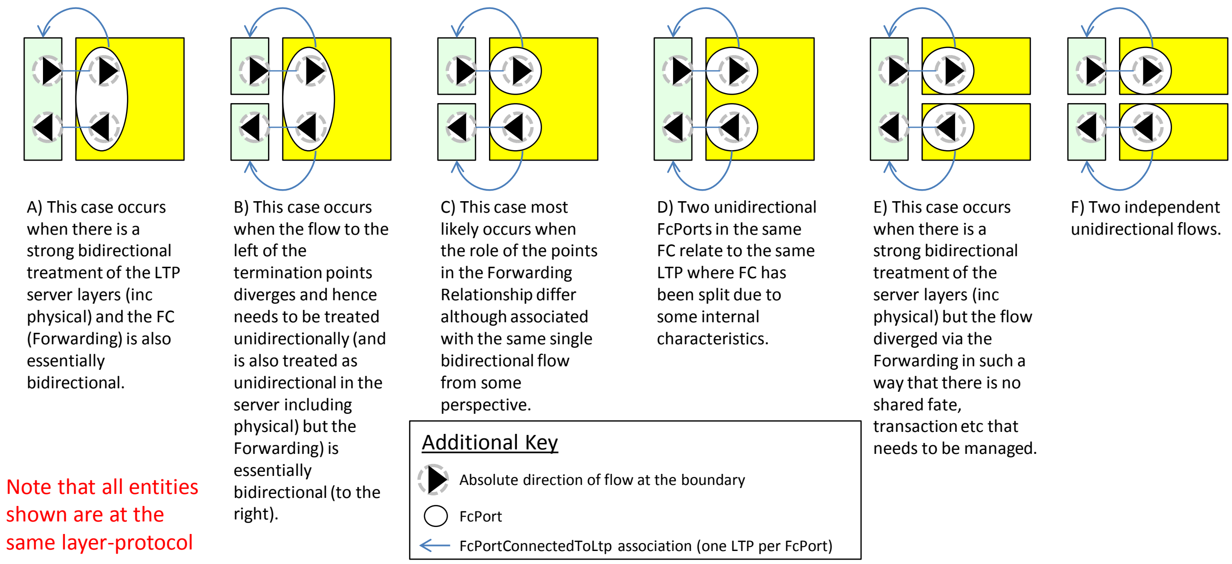


Figure 4-24 Various mixed directionality forms

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.

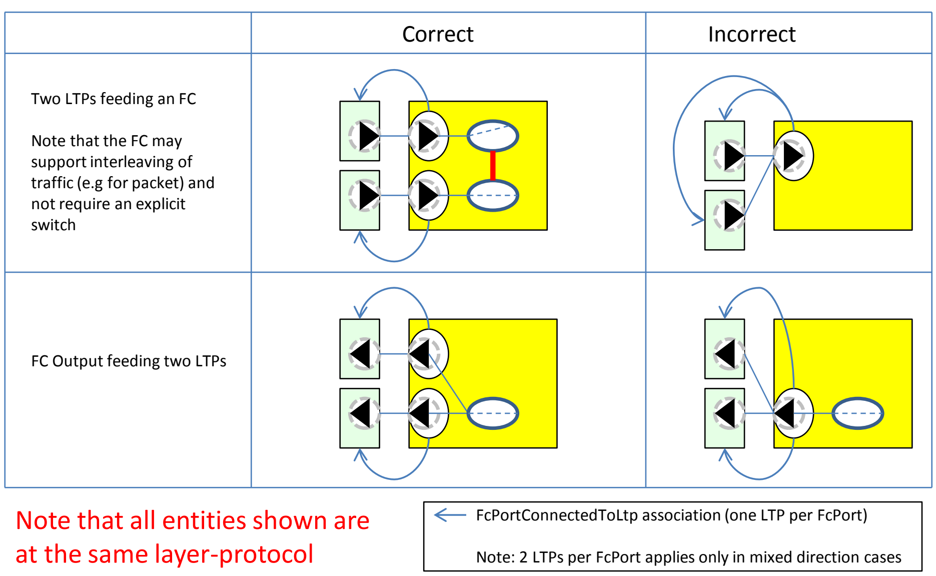
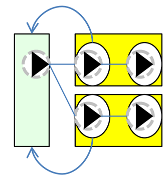
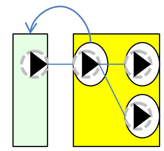
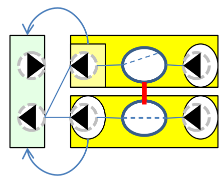
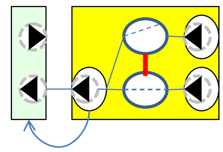


Figure 4-25 Interrelationship between a pair of unidirectional LTPs and a unidirectional FC

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.

 © 2018 Open Networking Foundation

Shows access to FC	Overlay form	Encapsulated form
One LTP feeding two FC that have different remote destinations (or the same remote via different routes)		
Two FCs feeding one LTP where the FCs have the same remote sources (only legal if flows switched or interleaved)		

Note that all entities shown are at the same layer-protocol

Figure 4-26 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)[16]. 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.

[16] The measures for the non-intrusive monitor are no different from the measures for the corresponding Termination. The Termination is embedded in the LP… hence so is the non-intrusive monitoring.

 © 2018 Open Networking Foundation

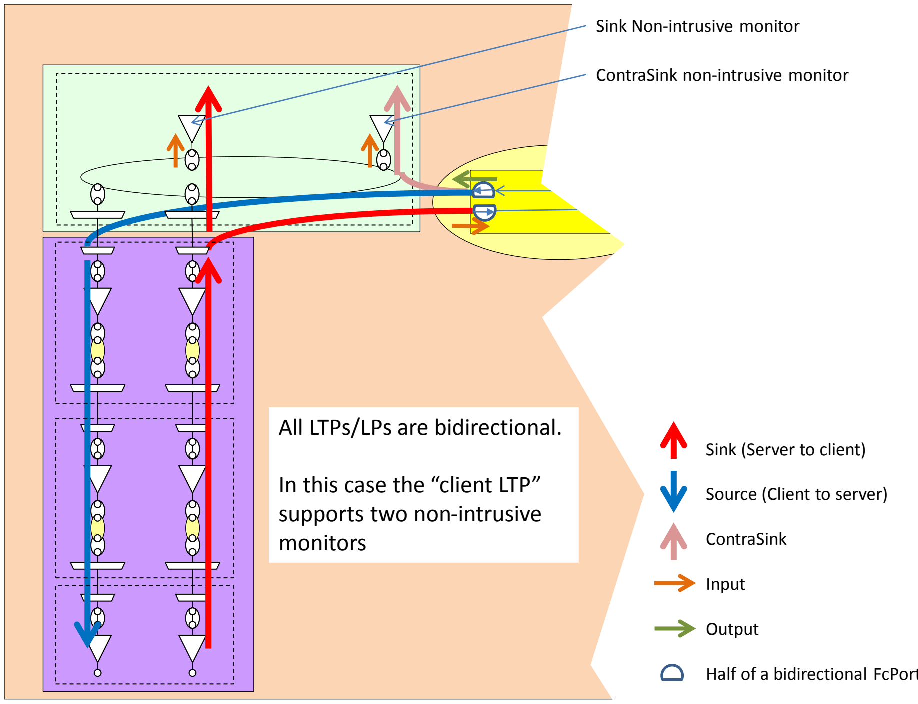


Figure 4-27 Contra-directionality showing monitors

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)

 © 2018 Open Networking Foundation

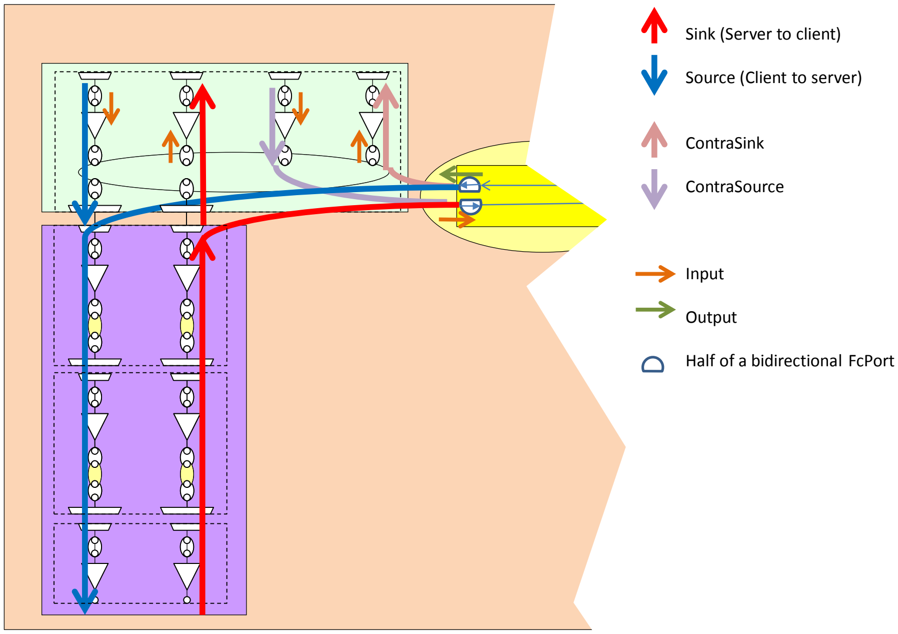


Figure 4-28 Contra-directionality showing monitors and signal sources

The Client LTP has one LP (which is considered simply as Bidirectional) which has four termination functions (where two are contra-directional). As a consequence there are four inputs to the termination functions, these are distinguished as follows:

- Source Input
- ContraSource Input
- Sink Input
- ContraSink Input

It is expected that the LP directly include the Source and Sink attributes and a composed part of the LP would include the ContraSource and ContraSink attributes (this is for further study[17]).

In the following example, there is a deep inspection capability dealing with two layers of inspection. It is assumed that the forwarding technology is such that the server layer supports only one client. Although the LTPs are bidirectional, the upper LP of the green LTP is a unidirectional Sink. This illustrates one case where an LTP directionality is different from the directionality of an included LP.

The Client LTP (which is considered simply as Bidirectional) has two terminations in the layer-protocol of the FC (where one is contra-directional). As a consequence there are two inputs to the termination function block (that includes both terminations). These are distinguished as follows:

- Sink Input

[17] The measures etc for the SINK and CONTRA_DIRECTION_SINK are likely to be the same hence the need to partition the CONTRA_DIRECTION_SINK measure etc into a composed part (to avoid name clashes)

 © 2018 Open Networking Foundation

- ContraSink Input

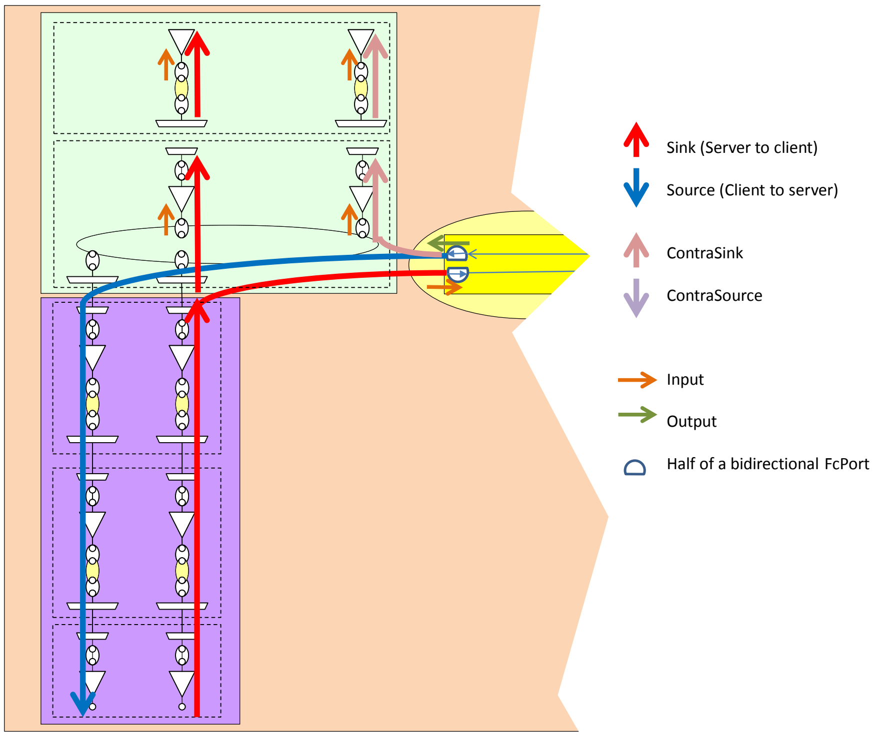


Figure 4-29 Contra-directionality showing deep inspection

4.4.1 Inherent and assigned directionality

A photonic filter is inherently omni-directional. A device may be constructed that encapsulates a pair of identical filters and administratively designates one filter to be used in the reverse direction to the other. However, the directionality of the representation of the parts is still omni-directional. There is no imposed directionality knowledge. Only if there is a constraint within the scope of the things that a thing oversees, do we allow the directionality to be exposed.

If there is a circulator that makes an element unidirectional, then that is exposed at the ports of the overall representation (FC). Likewise the end-end FC from laser to receiver is unidirectional, but it is possible that the components between are omni-directional. Where there is a dark fibre offered to a customer it is omni-directional.

The operator may choose to put detectors on the fibre to make the service unidirectional or simply offer it as unidirectional but the unidirectional property then only appears at the highest level.

The abstraction of intention will convey the assigned directionality, the representation of the realization will expose the inherent directionality.

A signal has an inherent direction of propagation. Consider the case of a 3 port filter with a media channel between ports 1, 2 with a relative central frequency of -100 and a width of 100 and another media channel between port 1 and 3 with a relative central frequency of +100 and a

 © 2018 Open Networking Foundation

width of 100. The administrator could designate the 1-2 media channel to be used only for signals propagating in the 1-2 direction and the 1-3 media channel to be only used for signals propagating in the 3-1 direction. This is a simple example of single fibre working. Whilst this is not a common mode of operation[18], it is covered by the model.

It is reasonable to administratively assign directionality to the media channels for port 2 and 3 but port 1 must remain omni-directional.

4.5 Relationships to the physical port

Complex relationships between the physical port and the LTP are discussed in TR-512.6 (a symbols set is described in that document for physical connectors and pins which is used below).

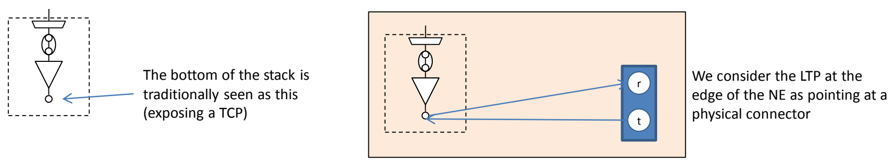


Figure 4-30 Basic association between LTP and Physical Connector

As explained earlier, the effect of the media, e.g. a fiber, is represented by an FC.

The model also allows FC port to directly associate with a connector pin to allow the representation of the fiber with connectors.

The following sequence of figures shows that the fundamental relationship is that between the FC and the pin and that this is the basis for the LTP to pin (and hence port).

In the figure below, assume that the LP shown is an OE capability operating in the photonic domain to the physical connector (where the LP represents a laser/receiver pair). There is necessarily a fiber present between the laser and the physical connector, and a fiber present between the connector and the receiver (assuming that this is not a single fiber working case).

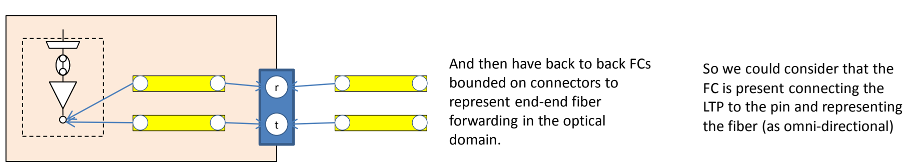


Figure 4-31 FC and Physical Connector

But considering the LTP model, the FC can be encapsulated in the LTP.

[18] It may become popular for sync distribution since the differential delay between the directions can be computed very accurately (chromatic dispersion and length).

 © 2018 Open Networking Foundation

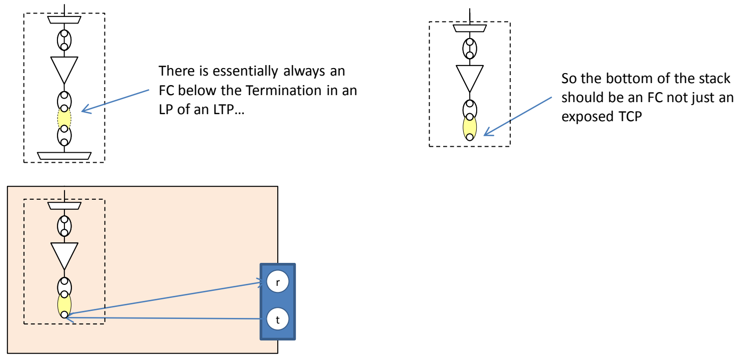
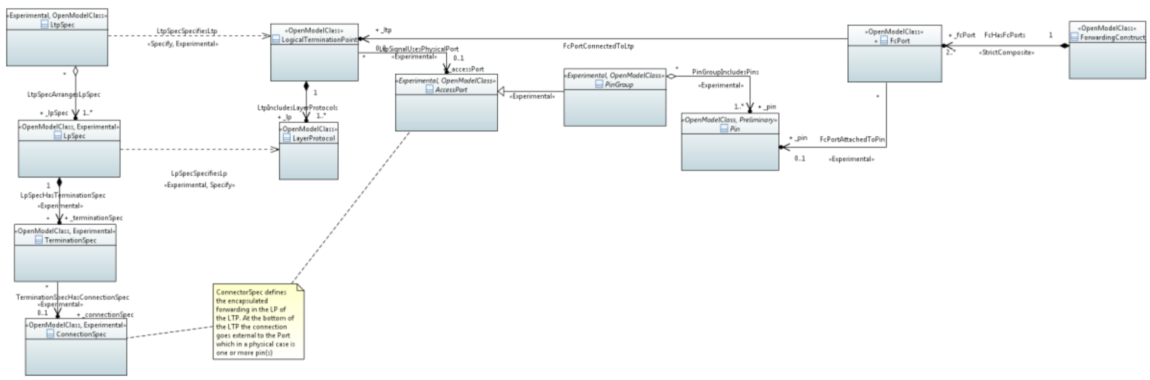


Figure 4-32 Clarified LTP to Physical Connector association

The LTP associating with the physical connector is really as shown above and hence the physical connector is always essentially associated with an FC.



CoreModel diagram: Forwarding-PhysicalPortAndFc

Figure 4-33 LTP and LP to Pin via ConnectionSpec

In a simple photonic couple/splitter the FC is directly connected to the pins.

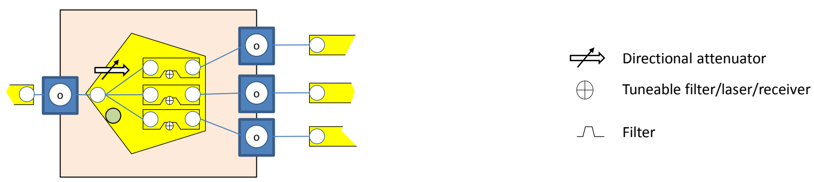


Figure 4-34 Broadband coupler/splitter with tuneable filter showing pins

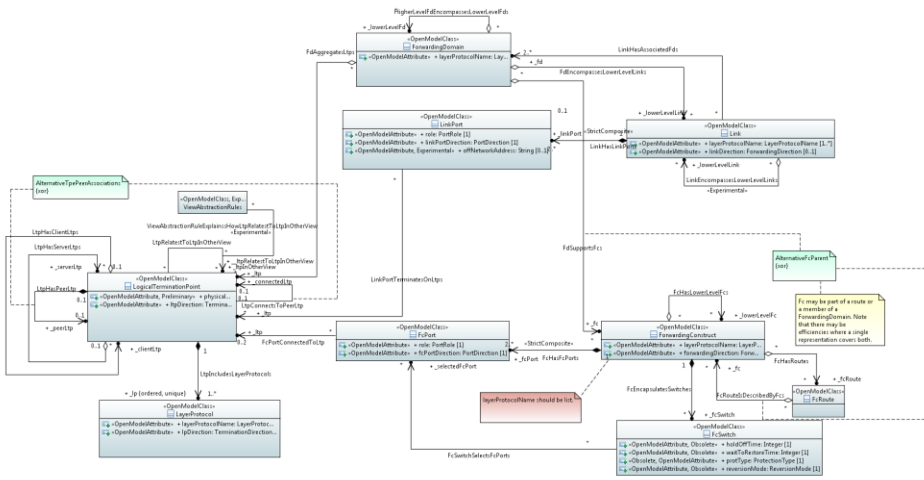
The relationship between an FcPort and a connector Pin allows only one pin per FcPort as the intention is that the photonic FC corresponds to a single unit of media and hence the FC cannot distribute across multiple pins (the pin also corresponds to a single unit of media). Where a signal is distributed across multiple pins it will necessarily be on several units of media and hence will have several media FCs.

 © 2018 Open Networking Foundation

Not all FcPorts have associated physical connectors that are visible in the view. The model allows FCs to be chained FcPort to FcPort with no need for a physical connector to be exposed.

5 Work in progress (see also TR-512.FE)

The figure below shows some constraints on the associations in the model. Further work is being carried out on how to most appropriately represent constraints. The figure also shows some classes related to other parts of the model covered in other documents (see TR-512.4 and TR-512.5).



CoreModel diagram: Forwarding-HighLevelDetail

Figure 5-1 Class Diagram of all key classes showing attributes and constraints

The above diagram shows owned attributes of the key 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.

End of Document

 © 2018 Open Networking Foundation