Core Information Model (CoreModel)

TR-512.5

Resilience (Protection, Restoration and Recovery)

Version 1.5
September 2021

ONF Document Type: Technical Recommendation
ONF Document Name: Core Information Model version 1.5

Disclaimer

Open Networking Foundation
1000 El Camino Real, Suite 100, Menlo Park, CA 94025
www.opennetworking.org

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

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
1.2	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.
1.4	November 2018	No changes.
1.5	September 2021	Enhancements to model structure.

1 Introduction to the document suite

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%), 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 as well as 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 Resilience Model

The focus of this document is the modeling of resilience in the ONF-CIM.

This document:

- Introduces the resilience model structure
- Describes the key classes of the resilience model

- Explains the attributes of the resilience model
- Shows how the model deals with various resilience schemes
- Explains how the specification model describes resilience schemes (protection etc.)
- Highlights work in progress to further advance the resilience model

The resilience model builds on aspects of the Core Network Model related to Termination and Forwarding described in TR-512.2 and related to Topology TR-512.4. Resilience capability and other specification considerations are covered in TR-512.7.

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

2.1 Some working definitions

Resilience: A mechanism that ensures greater availability of a provided capability than could be achieved by use of a minimal set of dedicated resources. The mechanism uses additional resources to ensure that the provided capability continues to be provided even when one or more resource(s) originally used to provide the capability fails. The degree of failure supported depends upon the scheme. The time taken to recover depends upon the scheme. Several schemes may be used together.

Protection: A resilience mechanism where the resources used to achieve resilience against failure are in place and running ready to be selected so as to rapidly recover the service. The resources may be shared by several services such that under certain failure conditions one service may take the resilience resources from another causing the other to fail.

Restoration: A resilience mechanism where there are no additional resources over and above those needed to provide the capability in place and running but there is either a plan for resources to be used and/or there is a control capability that can determine which resources can be used to recover a failed service.

3 Resilience model detail

3.1 Resilience Pattern

The resilience model unifies a number of apparently different traditional model approaches that are used for various different resilience schemes (see [ITU-T 808.1]). The resilience model focus is the FcSwitch which represents the forwarding selector[1] and which enables changes of forwarding to achieve resilience. The model also represents the control element of the resilience control loop that monitors behavior, assesses that behavior identifying necessary configuration changes and applies those configuration changes to make the required adjustments to Forwarding so as to achieve the intended resilience.

[1] In this release only a selector that makes an absolute choice is explicitly supported. The switch can select multiple FcPorts at the same time and the assumption is currently that all selected ports are equally weighted. Through this the selector concept extends to cover high rate (e.g. packet rate selection from queues). An extension to this could allow a weighted sharing capability where the weighting may be based upon some algorithm. This would also apply to analogue cases (e.g. for power). Such an extension will be considered for a later release of the model.

Some resilience schemes require combinations of control elements and switches. A particular pattern of combination of control elements and switches along with forwarding of control messages fully describe each scheme. This single uniform approach replaces the various traditional approaches (e.g. in some traditional representations a protection group is used, the protection group is replaced by one or more control elements in the new model).

3.1.1 Resilience model in the context of other model additions to V1.3

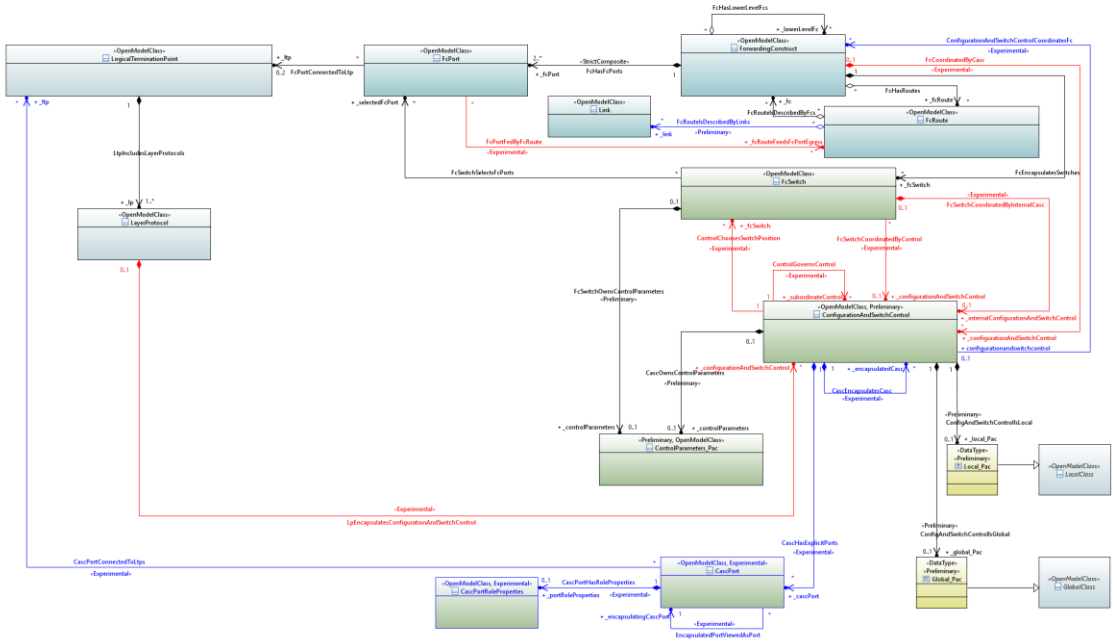
The resilience model is a specialist model that represents the components of resilience at a similar level of abstraction to the LTP and FC. This release of the ONF CIM includes a number of new classes and has further explanation of patterns that related to this model.

As will be observed in the model described below, a key consideration is that of control of the resilience scheme. In V1.3 a generalized Control model has been added (see TR-512.8). The control elements described in this document, the ConfigurationAndSwitchControl, can be seen as a specialist form of ControlComponent described in the generalized control model. The ControlComponent itself can be seen as a specialist form of ProcessingConstruct (described in TR-512.11) and ProcessingConstruct can be seen as a specialist form of Component (described in TR-512.A.2). If the entire model was represented as Component this would be particularly opaque.

As a consequence it has been chosen to represent explicit classes to describe the specialist roles. The specialist roles are clearly still generalized forms. An implementer may choose to represent everything as component with layers of spec or use the explicit classes of the Core model. It is recommended to not specialize any further.

3.2 Resilience Model

The figure below shows the key classes involved in modeling resilience and the associations between them.



CoreModel diagram: Resilience-Pattern

Figure 3-1 Basic resilience pattern

In the diagram:

- The classes shaded green are the classes that are solely present to support resilience
- The associations and classes shown in blue are new in this release.
- The blue and red associations are experimental.

The key classes present in the model that specifically support resilience are described in the following sections. The naming/identification classes, Local_Pac and Global_Pac, are discussed in section 3.4.8 – Naming the ConfigurationAndSwitchControl (C&SC) on page 29. The FcSpec class is included as it will be used to express the structure of the resilience scheme of the ForwardingConstructs, this is described in detail in TR-512.7. See also TR-512.2 for an explanation of some key classes in the figure.

In this release:

- The C&SC has been extended with ports (reflecting the ComponentSystem pattern detailed in TR-512.A.2). The port capability allows for representation of detailed signalling relationships and of asymmetric control (see 3.4.5 Symmetric and asymmetric C&SC on page 25). In basic control cases and for abstract representations the CaSC ports do not need to be expressed and can be omitted (being optional).
- The C&SC can be composed of C&SCs allowing for expression of complex control structures (see 3.4.1 Encapsulation of the ConfigurationAndSwitchControl (C&SC) on page 14). This decomposition is also reflected at the C&SC port.
- The C&SC can provide a list of references to controlled FCs (see 3.4.6 C&SC Coordinates FC on page 26) providing clarity as to which FCs are controlled when the C&SC deals with a subset of FCs in the FD and the C&SC is not related to the FC by

composition or via another subordinate C&SC. This allows aggregate statements expressed via attributes to be made at the C&SC and/or its ports about control effects on all FCs and/or their ports. As a result of the FC references and other rules the implications of the attributes can be interpreted and the effects on all related FCs can be determined.

- An FcPort can be internal (and hence not associate with an LTP) to allow chaining of switches (see 3.4.13 Overlaying and chaining switches on page 37).
- An FcPort can be associated with multiple switches to allow both chaining and other more complex arrangements (see 3.4.13 Overlaying and chaining switches on page 37).
- FcRoute is a GlobalClass.

The model in this release is a superset of that detailed in the previous release and all previous usages should be compatible with this release.

The following sections detail the key classes of the resilience model.

3.2.1 CascPort

Qualified Name: CoreModel::CoreNetworkModel::ConfigurationAndSwitchControl::CascPort

A port of a C&SC that can be used where there is significant asymmetry to be represented. This can represent any combination of:
- the conveying of messaging to/from the C&SC
- the conveying of control action
- the providing of indications of state etc.

This class is Experimental.

3.2.2 CascPortRoleProperties

Qualified Name: CoreModel::CoreNetworkModel::ConfigurationAndSwitchControl::CascPortRoleProperties

Container for properties associated with the port role as defined by the CascSpec.

This class is Experimental.

3.2.3 ConfigurationAndSwitchControl

Qualified Name: CoreModel::CoreNetworkModel::ConfigurationAndSwitchControl::ConfigurationAndSwitchControl

Represents the capability to control and coordinate switches, to add/delete/modify FCs and to add/delete/modify LTPs/LPs so as to realize a protection scheme.

This class is Preliminary.

3.2.4 ControlParameters_Pac

Qualified Name: CoreModel::CoreNetworkModel::NetworkCommon::ControlParameters_Pac

A list of control parameters to apply to a switch.

This class is Preliminary.

3.2.5 FcRoute

Qualified Name: CoreModel::CoreNetworkModel::FcRoute::FcRoute

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

Inherits properties from:

- GlobalClass

3.2.6 FcSwitch

Qualified Name: CoreModel::CoreNetworkModel::ForwardingConstruct::FcSwitch

The FcSwitch 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 or more 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 and essentially "decorates" FCs that are involved in resilience schemes that use switching in a protection mechanism.
Essentially FcSwitch performs one of the functions of the Protection Group in a traditional model. It associates 2 or more FcPorts each playing the role of a Protection Unit.
One or more protection, i.e. standby/backup, FcPorts provide protection for one or more working (i.e. regular/main/preferred) FcPorts where either protection or working can feed one or more protected FcPort.
The switch may be used in revertive or non-revertive (symmetric) mode. When in revertive mode it may define a waitToRestore time.
It may be used in one of several modes including source switched, destination switched, source and destination switched etc. (covering cases such as 1+1 and 1:1).
It may be locked out (prevented from switching), force switched or manual switched.
It will indicate switch state and change of state.
The switch can be switched away from all sources such that it becomes open and hence two coordinated switches can both feed the same LTP so long as at least one of the two is switched away from all sources (is "open").
The ability for a Switch to be "high impedance" allows bidirectional ForwardingConstructs to be overlaid on the same bidirectional LTP where the appropriate control is enabled to prevent signal

conflict.
This ability allows multiple alternate routes to be present that otherwise would be in conflict.

Inherits properties from:

- LocalClass

3.3 Key to diagrams

The following diagram highlights the symbols used in other diagrams in this document for various classes etc in the resilience model.

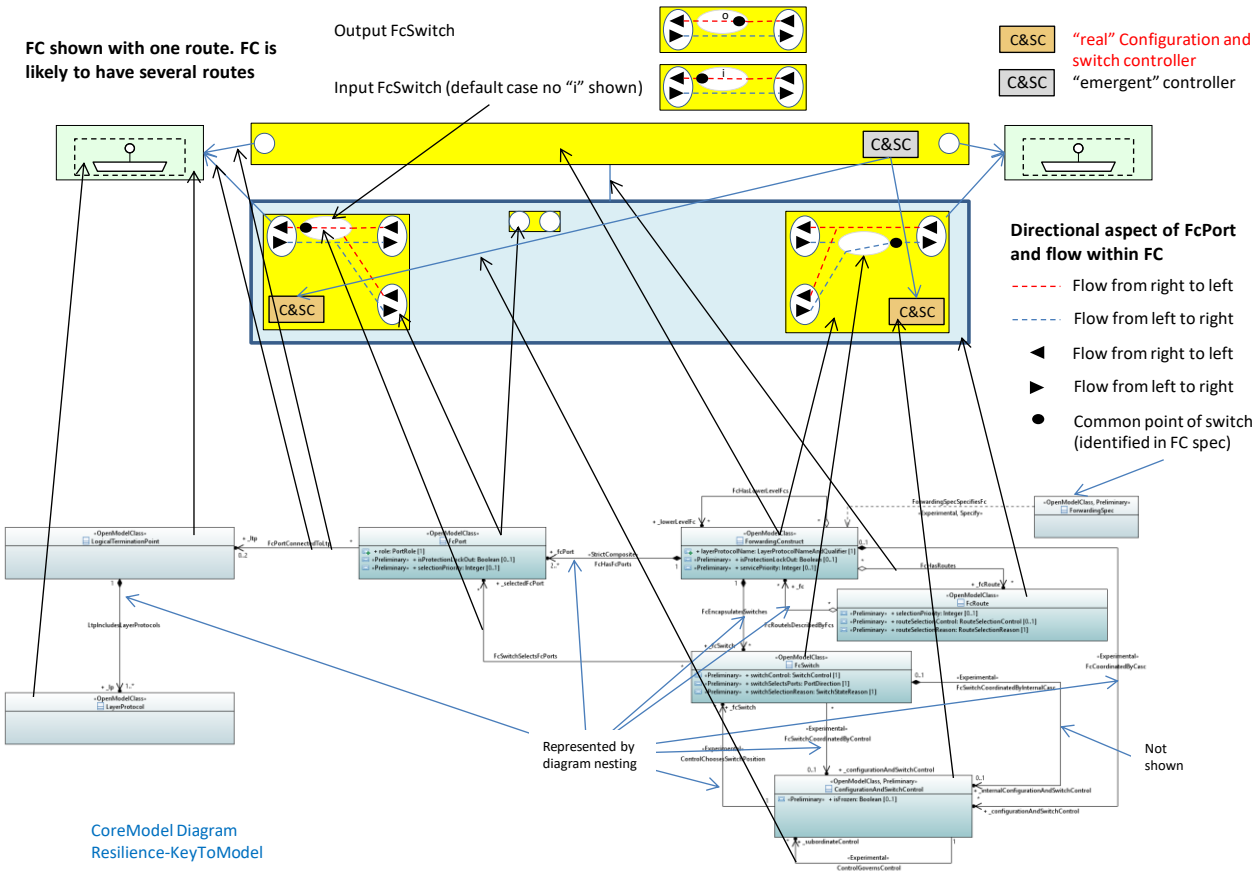


Figure 3-2 Instance diagram key

3.4 Further explanation of the model

3.4.1 Encapsulation of the ConfigurationAndSwitchControl (C&SC)

There are several degrees C&SC independence (see TR-512.A.11 for examples of each):

- C&SC encapsulated in a FcSwitch
  - Used where a C&SC has a control scope of a single switch and where there is benefit in exposing ControlParameters for a single switch
  - The C&SC may participate in scheme where only per-switch autonomous control is available or may be part of a broader scheme with a hierarchy/mesh of C&SCs
  - The C&SC need have no id as it is identified in the context of the switch and there can only be one C&SC per switch
- C&SC encapsulated in an FC
  - Used where the C&SC has a control scope across several switches in the FC and where there is a need to have consistent parameters across those switches
  - This approach could be used for an FC with a single switch instead of embedding the C&SC in the switch
  - The C&SC may participate in a scheme as part of a hierarchy/mesh of C&SCs
  - The C&SC has a local id in the context of the FC. There may be several C&SCs in the context of an FC
  - The arrangement of C&SCs in the FC is described by the FcSpec (see TR-512.7)
- C&SC encapsulated in a C&SC
  - Used where a complex control structure needs to be set out as a tightly coupled system of controllers
  - The C&SC has a local id in the context of the encapsulating C&SC
  - The usage if this is described in section 3.4.7 Relating the ProcessingConstruct , C&SC encapsulation and protection scheme on page 27
- C&SC encapsulated in an LTP
  - Similar to the FC cased but used where there is significant switching capability within the LTP
  - At this release there are no examples of usage for this capability
- C&SC stand-alone
  - Used where the C&SC coordinates switches and other configuration spread across multiple FCs etc
    - In this case it replaces the traditional protection group approach
  - There may be a hierarchy/mesh of C&SCs where a C&SC may govern others and may itself be governed
  - The C&SC may create/delete/adjust FCs as well as activate switches
  - The C&SC is part of the overall Management-Control Continuum (see TR-512.8)
  - The C&SC has a global id
  - The arrangement of C&SCs in a control scheme is described in a ControlSchemeSpec (see TR-512.7)

This model fragment offers flexibility in the way the FcSwitch gains its ControlParameters and provides an instantiable C&SC that can be positioned with an appropriate scope of control for any particular case.

The control parameters for a number of C&SCs/Switches could be provided by a profile (although this area of model is for further development).

The C&SC can be included in a ConstraintDomain (CD) which may define the scheme that the C&SC is part of or which may simply apply common constraints to a number of C&&SCs. (see TR-512.8)

3.4.2 An Open FcSwitch

The figure below (see section 3.3 Key to diagrams on page 13 for an explanation of the figure symbol set) shows an example of multiple open switches showing both legal and illegal settings.

The figure assumes a circuit switched technology and shows four cases of an NE[2] with a protected signal flow to one client LTP (green) supported by an LTP (purple) bound to a physical port (on the left of each diagram). The cases highlighted are the two normal states of switches in the upper two diagrams, a transient state in lower left and an illegal state in lower right where the Configuration and Switch Controller (C&SC) has failed.

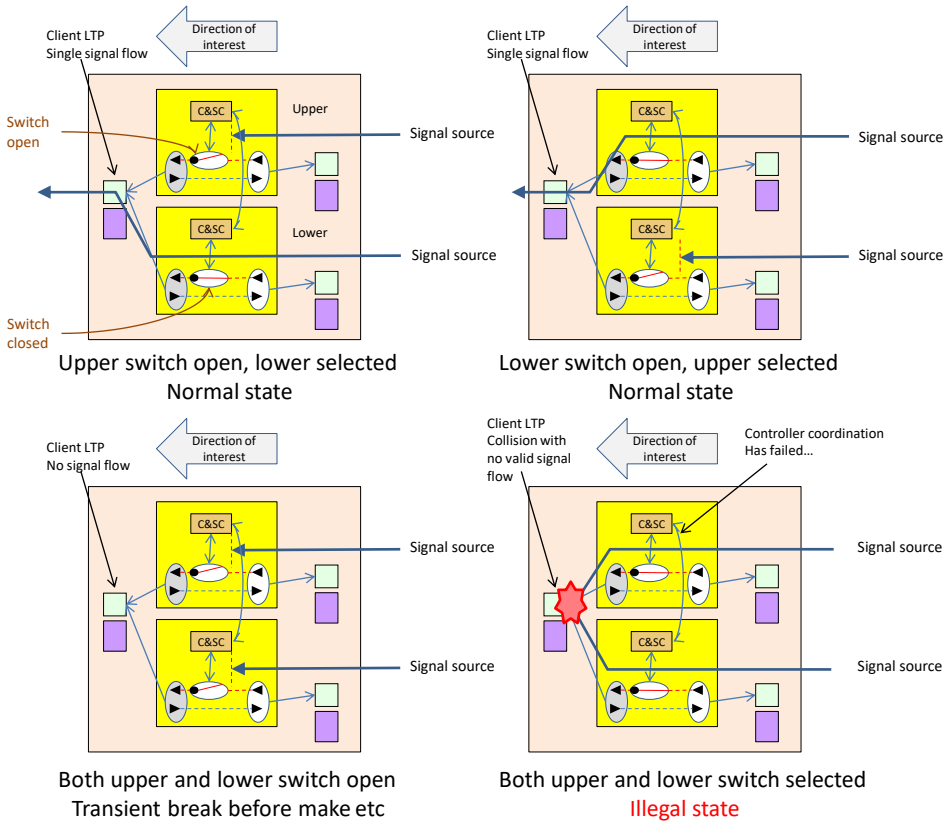


Figure 3-3 Multiple open switch case with one client LTP

[2] The NetworkElement class has been deprecated in this release (see TR-512.8). The term NE is used in a general sensein this document.

3.4.3 Sharing FcPorts and switch orientation convention

The diagrams in the figure below (in dotted red ellipses) illustrate usage of a mix of output and input switches (designated by "o" and "i" respectively). The modelling orientation convention is that the switch common is on the sharing FcPort if there is only one sharing FcPort (hence in some cases mixed ingress/egress switches are used). If there are two sharing FcPorts, or no sharing FcPorts the convention is that the input switch (default) is used unless there is specific complexity that can only be resolved with output switches.

See also figures in TR-512.A.11 for more details on the specific case of use.

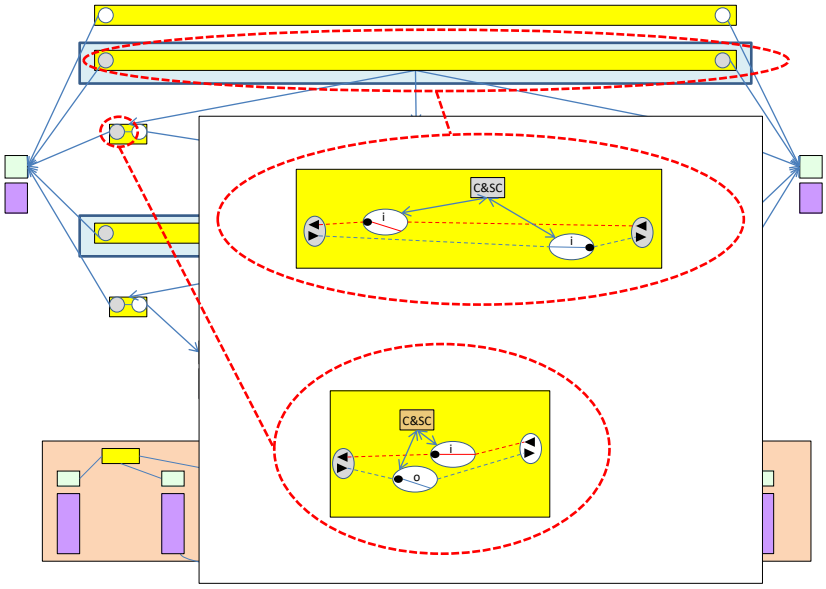
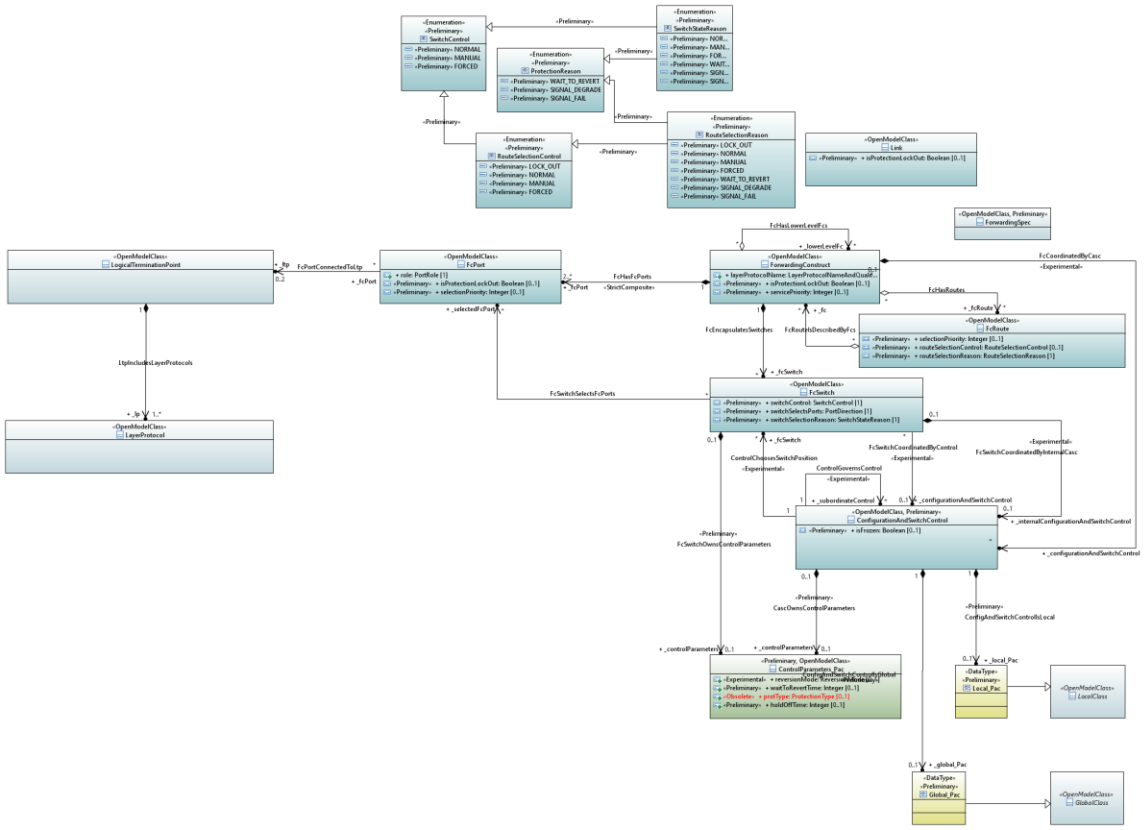


Figure 3-4 Sharing FcPorts and switch orientation convention

3.4.4 Resilience Attributes

The figure below highlights the key attributes of the resilience model.



CoreModel diagram: Resilience-KeyAttributes

Figure 3-5 Key resilience attributes

The attributes are described in the following tables (which show all attributes of the classes, not just the attributes related to resilience).

3.4.4.1 CascPort

Table 1: Attributes for CascPort

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
portRole	Experimental	The role of the port of a C&SC. The interpretation of the role is provided by the C&SC spec. The C&SC spec will set out the role in the context of C&SC functions. The role will indicate how the port relates to the associated entity, e.g. is conveying messages.
isRelatedControlFlowDisabled	Experimental	If TRUE, then any Control signal flow related to this controller (to, from or drop-and-continue) is prevented from passing through the related LTP carrying the signaling for this controller. This can be considered as being realized using an FcSwitch in an FC embedded in the LP at the layer of signaling to disconnect the FcPort bidirectionally. This FcSwitch should be represented in the LTP spec. Note that the FcSwitch will be at the granularity of the relevant control signal and other flows may be passed uninterrupted.

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
isControlledFcPortDisabled	Experimental	If TRUE, then the related FcPort on the FC is disabled and hence signal will not flow through that FcPort. This is realized using an FcSwitch to disconnect the FcPort bidirectionally. Note that as the controller may control many FCs and may switch them all together as one, in an implementation the FcSwitch could be omitted from the FC instance model. Any omission should be explained by the FcSpec. This is equivalent to a blocked indication on the LTP used in other representations.
isProtectionLockOut		The resource is configured to temporarily not be available for use in the protection scheme(s) it is part of. This overrides all other protection control states including forced. If the item is locked out, then it cannot be used under any circumstances. This causes isRelatedControlFlowDisabled to become TRUE and isControlledFcPortDisabled to become TRUE.
_portRoleProperties	Experimental	A link to properties associated with the port role as defined by the CascSpec.
_ltp	Experimental	The LTP that conveys the messages related to the port and/or is subject to control action and/or provides indications of state etc. For direct association, there may be up to 2 LTPs (to account for directionality differences). In the specification representation, there may be a number rules that provide further LTP relationships that are implicit in the instantiated model.
_encapsulatingCascPort	Experimental	In a case where there is nested C&SC the ports are also nested and this references the superior port.

3.4.4.2 CascPortRoleProperties

Table 2: Attributes for CascPortRoleProperties

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
signallingFormat	Example Experimental	A reference to the definition of the signalling format used by the instance referenced by the related port. This is a placeholder for a more sophisticated capability.
monitoringDetails	Example Experimental	Information on what is being monitored in the instance referenced by the related port. This is a placeholder for a more sophisticated capability.
controlDetails	Example Experimental	Information on what is being controlled in the instance referenced by the related port. This is a placeholder for a more sophisticated capability.

3.4.4.3 ConfigurationAndSwitchControl

Table 3: Attributes for ConfigurationAndSwitchControl

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
switchRule	Experimental	A sketch of the presence of complex rules governing the switch behavior.
isFrozen	Preliminary	Temporarily prevents any switch action to be taken and, as such, freezes the current state. Until the freeze is cleared, additional near-end external commands are rejected and fault condition changes and received APS messages are ignored. All administrative controls of any aspect of protection are rejected.
isCoordinatedSwitchingBothEnds	Experimental	The C&SC is operating such that switching at both ends of each flow across the FC is coordinated at both ingress and egress ends.
resilienceControlStatus	Experimental	The state of the control process.
_fcSwitch	Experimental	The switch being controlled.
_controlParameters	Preliminary	The control parameters to be applied if local parameters are used rather than profiles.
_profileProxy	Experimental	Applied profiles.
_local_Pac	Preliminary	See referenced class
_global_Pac	Preliminary	See referenced class
_subordinateControl	Experimental	A C&SC that is fully or partially subordinate this C&SC. A peer is considered as partially subordinate in that the peer will respond to requests for action from this C&SC but will also make requests for action to be carried out by this C&SC. Where there is a peer relationship each controller in the peering will see the other controller as subordinate.
_cascSpec	Experimental	See referenced class
_encapsulatedCasc	Experimental	Where a C&SC is complex it may be decomposed into subordinate C&SC parts. The decomposition is described by the C&SC spec.
_cascPort	Experimental	A reference to ports of a C&SC that can be used where there is significant asymmetry to be represented. The C&SC need not have ports.
_coordinatedFc		See referenced class

3.4.4.4 ControlParameters_Pac

Table 4: Attributes for ControlParameters_Pac

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
reversionMode	Experimental	Indicates whether the protection scheme is revertive or non-revertive.
waitToRevertTime	Preliminary	If the protection system is revertive, this attribute specifies the time, in minutes, to wait after a fault clears on a higher priority (preferred) resource before switching to the preferred resource. If a further fault occurs on the preferred resource in the waitToRevertTime then the reversion attempt is cancelled. The WTR timer is overridden by the needs of a higher priority signal. Depending upon which resource is requested this may simply cancel the attempt to revert of may cause immediate reversion.
protType	Obsolete	Indicates the protection scheme that is used for the ProtectionGroup.
holdOffTime	Preliminary	This attribute indicates the time, in milliseconds, between declaration of a switch trigger condition (e.g. signal degrade or signal fail), and the initialization of the protection switching algorithm.
_networkSchemeSpecification	Experimental	See referenced class

3.4.4.5 FcPort

Table 5: Attributes for FcPort

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
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.
isProtectionLockOut	Preliminary	The resource is configured to temporarily not be available for use in the protection scheme(s) it is part of. This overrides all other protection control states including forced. If the item is locked out, then it cannot be used under any circumstances. Note: Only relevant when part of a protection scheme.

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
selectionPriority	Preliminary	The preference priority of the resource in the protection scheme for a particular FC. The lower the value the higher the priority. A lower value of selection priority is preferred If two resources have the same value they are of equal priority. There is no preference between equal priorities. If a resource with the lowest value selection priority fails,, then the next lowest value available (may be the same value) is picked. Hence on failure of the current resource the next best available will be selected. If there are several equal values, the choice is essentially arbitrary. If the scheme is revertive then when a resource of higher priority than the currently selected resource recovers it will be selected. This is equivalent to working/protection but allows for all static scheme types with n:m capability. In simple schemes 0 = working and 1 = protecting. If selection priority of an FcPort is increased in value and the FC is currently selecting this FcPort then if another FcPort of a lower selection priority value is available, the wait to restore process will come into action as if the other FcPort had just become available. If selection priority of a FcPort is changed and the FC is not currently selecting this FcPort but is selecting an item that is now of a higher numeric value than the changed FcPort then the wait to restore process will come into action as if the other FcPort had just become available.
isInternalPort	Experimental	The FcPort is not exposed and cannot have associated LTPs. This form of FcPort is used to enable chaining of FcSwitches or FcRoutes in complex network protection scenarios.
_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.
_fcRouteFeedsFcPortEgress	Experimental	Identifies which route(s) currently actively forward to the FcPort to exit the FC to an LTP (or for an internal FcPort to propagate to the next internal switch/route).
_fcPort	Experimental	An FcPort may have a direct association to another FcPort where there is a transition from one domain to another but where there has been no termination.
_portOfInternalFc	Experimental	See referenced class
_pin	Experimental	For media FCs, the name of the pin that terminates the media.

3.4.4.6 FcRoute

Table 6: Attributes for FcRoute

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
selectionPriority	Preliminary	The preference priority of the resource in the resilience scheme for a particular FC. The lower the value the higher the priority. A lower value of selection priority is preferred If two resources have the same value they are of equal priority. There is no preference between equal priorities. If a resource with the lowest value selection priority fails, then the next lowest value available (may be the same value) is picked. Hence on failure of the current resource the next best available will be selected. If there are several equal values, the choice is essentially arbitrary). If the scheme is revertive then when a resource of higher priority than the currently selected resource recovers it will be selected. This is equivalent to working/protection but allows for all static scheme types with n:m capability. In simple schemes 0 = working and 1 = protecting. If selection priority of a Route is increased in value and the Route is currently selecting this Route, then if another Route of a lower selection priority value is available the wait to restore process will come into action as if the other Route had just become available. If selection priority of a Route is changed and the FC is not currently selecting this Route but is selecting an item that is now of a higher numeric value than the changed Route, then the wait to restore process will come into action as if the other Route had just become available.
routeSelectionControl	Preliminary	Degree of administrative control applied to the route selection.
routeSelectionReason	Preliminary	The reason for the current route selection.
_fc		The list of FCs describing the route of an FC. In most cases the FcRoute has 2 or more FCs however there are some cases where a Route with one FC is valid.
_link		See referenced class

3.4.4.7 FcSwitch

Table 7: Attributes for FcSwitch

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
holdOffTime	Obsolete	Moved to ControlParameter_Pac. 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.
protType	Obsolete	Indicates the protection scheme that is used for the ProtectionGroup.
reversionMode	Obsolete	Moved to ControlParameter_Pac. This attribute whether or not the protection scheme is revertive or non-revertive.
switchControl	Preliminary	Degree of administrative control applied to the switch selection.

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
switchSelectsPorts	Preliminary	Indicates whether the switch selects from ingress to the FC or to egress of the FC, or both.
switchSelectionReason	Preliminary	The reason for the current switch selection.
waitToRestoreTime	Obsolete	Moved to ControlParameter_Pac and changed to waitToRevert. 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 operation (e.g. a fault has cleared) before restoring traffic to that preferred FcPort.
_selectedFcPort		Indicates which points are selected by the switch. Depending on the switch spec (via FcSpec) - more than one FcPort can be selected at any one time (e.g. egress switch, ingress packet switch) - zero FcPorts can be selected. For an ingress switch this indicates that the switch common (egress) is "high impedance" .
_profileProxy	Experimental	Provides a set of predefined values for switch control in place of the direct values available via the FcSwitch or via _configurationAndSwitchControl.
_configurationAndSwitchControl	Experimental	A ConfigurationAndSwitchController that is external to the switch (it is coordinating many switches and hence cannot be encapsulated in the FcSwitch.
_internalConfigurationAndSwitchControl	Experimental	A ConfigurationAndSwitchController encapsulated in the FcSwitch that controls the FcSwitch alone.
_controlParameters		See referenced class

3.4.4.8 ForwardingConstruct

Table 8: Attributes for ForwardingConstruct

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

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
isProtectionLockOut	Preliminary	The resource is configured to temporarily not be available for use in the protection scheme(s) it is part of. This overrides all other protection control states including forced. If the item is locked out then it cannot be used under any circumstances. Note: Only relevant when part of a protection scheme.
servicePriority	Preliminary	Relevant where "service" FCs are competing for server resources. Used to determine which signal FC is allocated resource. The priority of the "service" with respect to other "services". Lower numeric value means higher priority. Covers cases such as pre-emptible in a resilience solution.
_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. For example a low level FC could represent 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.
_fcRoute		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.
_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.
_fcSwitch		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.
_configurationAndSwitchControl	Experimental	Reference to a ConfigurationAndSwitchController that coordinates switches encapsulated in the FC. The controller coordinates multiple switches in the same FC.
_fcSpecReference:ClassRef	Experimental SpecReference	Reference to the specific FcSpec class that defines the properties that augment the instance of FC.
_supportedLink	Preliminary	An FC that spans between LTPs that terminate the LayerProtocol usually supports one or more links in the client layer.
_multipleStrandSpan	Experimental	See referenced class
_supportingPc	Experimental	The functionality supporting this entity.

3.4.5 Symmetric and asymmetric C&SC

In release 1.2 all C&SC usages were essentially symmetric (see TR-512.11 for explanation of symmetric and asymmetric) and hence the C&SC did not need ports. In release V1.3 the C&SC may optionally have ports and hence a number of cases that require asymmetric treatment of control are now supported. Like the FC, a C&SC port is associated with up to two LTPs (to allow for directionality differences)[3].

The C&SC port to LTP association is used to represent several distinct flows:

- The flow of C&SC signaling information to/from the LTP where at the LTP it is propagated with the traffic and hence is to/from the adapter in the spec of the LTP
- The flow of control information to be applied to the LTP (e.g. disable traffic flow)
- The flow of monitoring information from the LTP to be used by the C&SC

The purpose of the port with respect to the flows covered is expressed via the ports role. A C&SC port can have a composite role and it may deal with several of the above flows if appropriate and where the same LTP is involved in all aspects described. The C&SC port role is described in the C&SC spec.

For rigid invariant patterned cases the relationship between C&SC port and LTP may be covered fully in the spec allowing a symmetric C&SC instance to be used. For more flexible cases explicit layout of instances of asymmetric C&SC port to LTP associations will be required.

The figure below, which is a sketch of an aspect of the [ITU-T G.8032] solution (see also TR-512.A.11 for more details) shows a C&SC with ports related to an LTP for various purposes. It also includes associations described in the next section.

[3] An association to the FcPort was also explored but not added in this release.

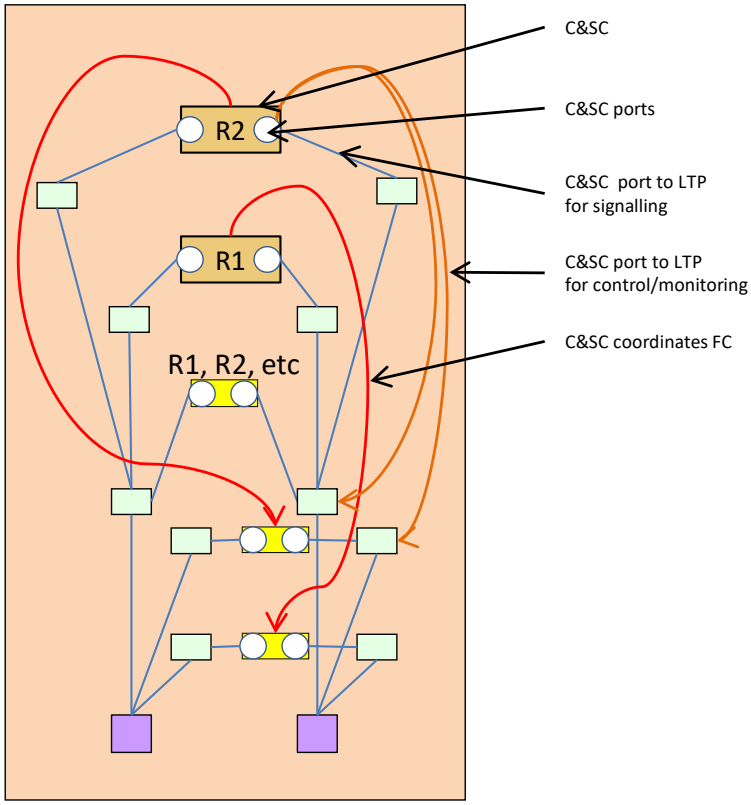


Figure 3-6 Figure showing C&SC with ports and association to FC

3.4.6 C&SC Coordinates FC

In some schemes the C&SC may control FC flow indirectly via manipulation of an LTP associated to an FcPort and manipulation of switches in the FC associated with that FcPort. In these complex cases it is not sufficient to embed C&SCs in the FC. The focus of the control action is the combination of FcPort and LTP and it is based on the C&SC asymmetry exposed via the CascPort. The CascPort to LTP association carries some of this information[4]. Whilst it is possible to determined indirectly the C&SC association to the FC by examining the C&SC port to LTP associations and identifying the corresponding FCs, to cover these cases more explicitly a direct association between the C&SC and the FC is used. The figure above shows this association in use.

This association indicates which FCs a C&SC coordinates. Where there is a C&SC embedded in the FC (FcCoordinatedByCasc association) that is governed by a superior C&SC, that superior C&SC does not need to reference the coordinated FCs directly as the ControlGovernsControl association provides all necessary information.

[4] The LTP association can mean “controls”, “monitors”, “signals via” or some combination of the three.

3.4.7 Relating the ProcessingConstruct , C&SC encapsulation and protection schemes

Where there is a complex behavior that does not fit the definition of one of the functional classes such as LTP and FC the ProcessingConstruct (PC) is used. The PC is described in more detail in TR-512.11. The C&SC is essentially a PC and a ControlComponent (as noted in 3.1.1 Resilience model in the context of other model additions to V1.3 on page 9), but considering importance of network application resilience in SDN, the choice has been made to define a specific class[5] to represent this behaviour instead of using the more general PC class.. The C&SC represents complex behavior of an assembly of parts where the emergent effect is that of Configuration and Switch Control[6]. Where a resilience scheme has a specific repetitive structure that is complex it may be beneficial to encapsulate the detail of the various C&SCs etc that enable the scheme in a superior C&SC. The scheme, including encapsulated C&SC and associations, is then describe in a spec structure (see TR-512.7 for more detail). The complex structure may be summarized as defined in the spec (including key parameters) and may be exposed as a constrained hierarchy.

The figure below shows some alternative encapsulations of C&SC explored for the [G.8032] solution[7]. The actual [G.8032] solution is in TR-512.A.11.

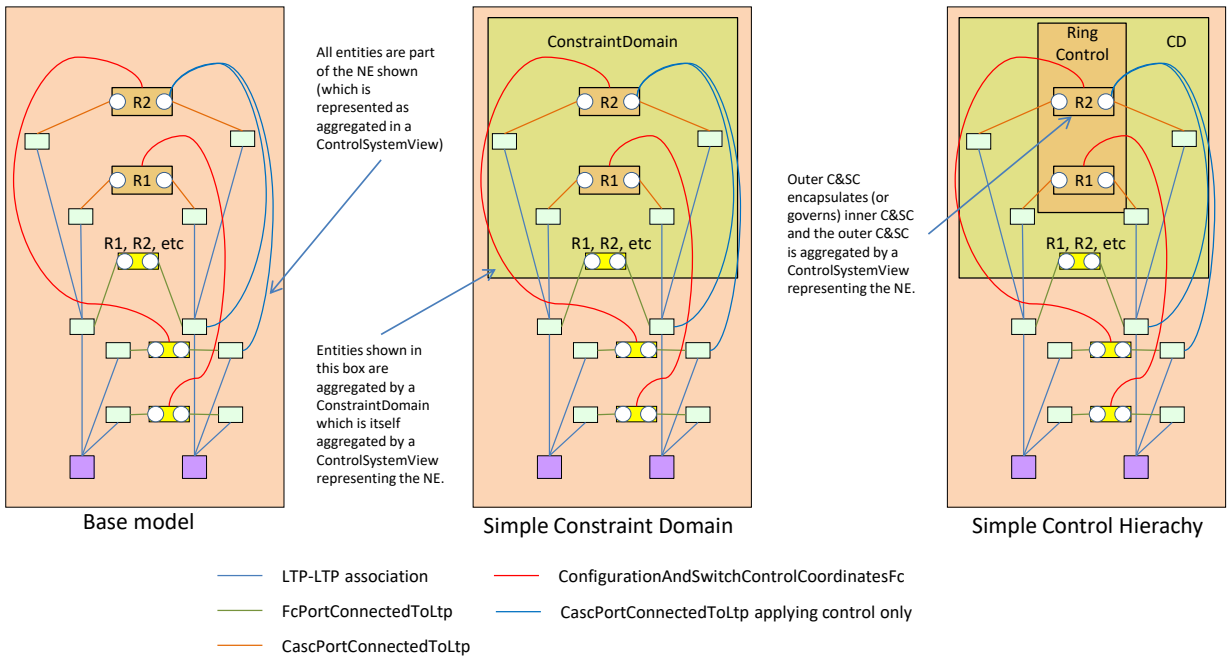


Figure 3-7 Figure showing basic groupings in CD and in C&SC

The figure above shows a Constraint Domain grouping LTPs, FCs and C&SCs in the "Simple Constraint Domain" diagram and shows the outer C&SC governing the inner C&SCs (via ControlGovernsControl) in the "Simple Control Hierarchy" diagram.

[5] This is true also for the LTP etc.
[6] This is true for all classes in the model as explained in TR-512.11 and TR-512.A.2.
[7] Overlay of diagram entities represents an explicit relationship between the entities where the inferior entity is shown to the front.

The figure below examines a more sophisticated encapsulation:

- The base model is as above
- The diagram of "The spec pattern (n cases of each item)" which identifies what to encapsulate in the spec, i.e. what would need to be specified for a single C&SC unit
- The diagram of "The spec pattern showing refactoring" highlighting an alternative structure of encapsulated C&SC
- The diagram of "The spec pattern…" shows the key parts and associations in the spec where there can be n of each item in the spec and each item will carry key attribute definitions

An approach that sets out explicit instances for each LTP, FC, C&SC etc as per the "Base model" would be reasonable as would an approach with a single instance of C&SC abiding by the spec shown in the diagram as "The spec pattern showing refactoring". The spec pattern would need to be related to base model as shown in the "The spec pattern (n cases of each item)" such that if desired a recipient of the information could expand the model.

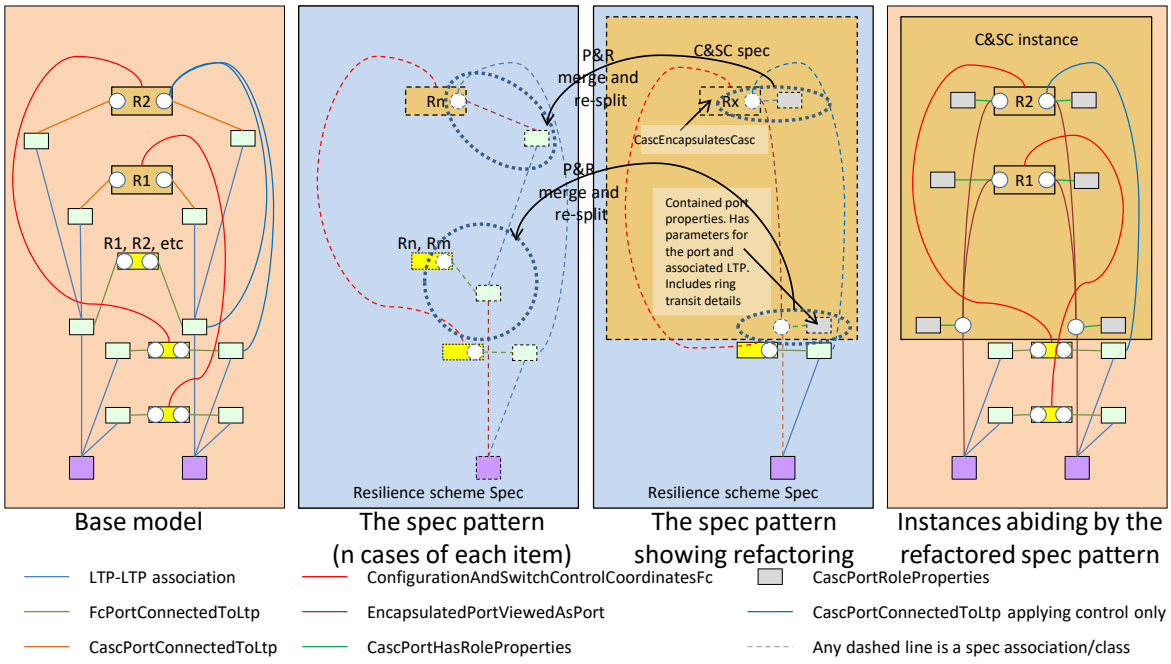


Figure 3-8 Figure showing a single C&SC encapsulating C&SCs defined by spec

Clearly it is preferable to have a single solution for each protection scheme. However, it is recognized that under particular constrained circumstances a representation of a fully capable realization may be verbose (i.e. expose details that are not of interest for the particular purpose) and hence there may be a desire to compact the representation. On this basis only recommendations are made for representation of particular schemes. These recommendations are in TR-512.A.11. Clearly, any alternative representation of a scheme should be justified by there being some need to simplify the model due to limitations in capability and any representation of a scheme should abide by the model in terms of entities used and should provide full details in

the spec model supporting scheme. This approach offering a variety of interpretable encapsulations is essentially the same as the approach used for the LTP etc (see TR-512.A.2).

3.4.8 Foldaway of complexity – Naming the ConfigurationAndSwitchControl

The ConfigurationAndSwitchControl can be:

- Embedded in an FcSwitch, a local class, essentially as a _PAC with no need for ids etc.
- Embedded in an FC, a global class, essentially as a local class with need for only relative ids etc.
- Stand alone as a global class with need for a UUID

Where there is one switch controller in a context (e.g. a switch or an FC) and where the controller relates to the context entity by composition it is reasonable to fold the controller into the context entity.

- The context entity gains the controller attributes
- Any reference to the controller becomes a reference to the context entity

Where there are several switch controllers in a context but where those controllers do NOT need to be referenced in any way from outside the context entity it is reasonable to fold the controllers into a data structure within the context entity

- The context entity gains a structure of multiple controller attribute blocks
- The controller "instance" is resolved by position in the structure
- It is NOT POSSIBLE to reference a controller from outside the context entity

Where there are several switch controllers in a context and/or where those controllers need to be referenced from outside the context it is not possible to fold the controllers into the context entities but the entities representing the controllers can have a relative identification (localId) within the scope of the identifier for the context

- References are via an address with contextId and localId as elements

Where the switch controller is not in any stable (long lived) context then it must have a UUID and can be directly referenced via that UUID.

Hence it is necessary to use a mechanism that allows the class to have a variable id strategy. This is achieved using conditional composition rather than inheritance (this approach has only been applied here but may be relevant for other cases in the model).

3.4.9 FcRoute has FCs and/or Links

There are two methods of describing the forwarding resources used by an FC to achieve forwarding across the network:

- Direct aggregation of FCs via FcHasLowerLevelFcs association where each FC exists in an FD/Link. The aggregation may be:
    - Single layer

    - Multiple layer where some of the FCs represent "Trails"[8]
- Indirect aggregation of FCs and/or Links via the Route. Where the route is described by FC those FCs need not exist in an FD but instead may stand-alone describing some arbitrary fragment of the flow[9].

The direct aggregation approach is the normal approach. The FCs in the Links are omitted and only the FCs in the FDs are provided.

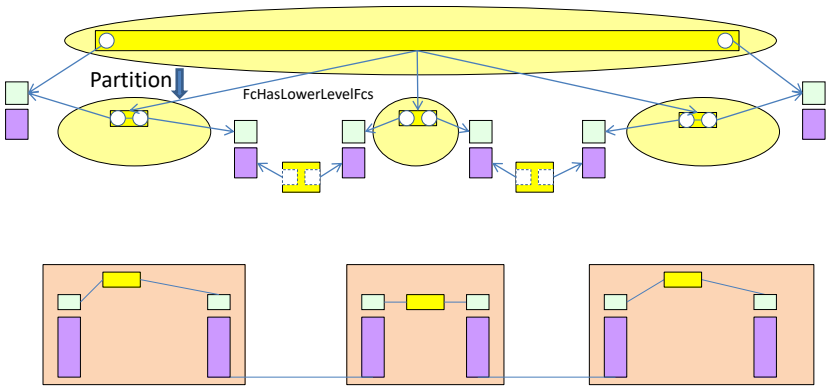


Figure 3-9 Forwarding detail represented via direct aggregation (or partition)

In some cases the direct aggregation is not sufficient and a route mechanism is used.

[8] A Trail is a forwarding relationship between Access Points (as per [ITU-T G.805]. In the ONF CIM it is represented by a ForwardingConstruct

[9] The FC may only exist in the context of the Route and have a lifecycle dependent upon the existence of the route or may exist in several routes.

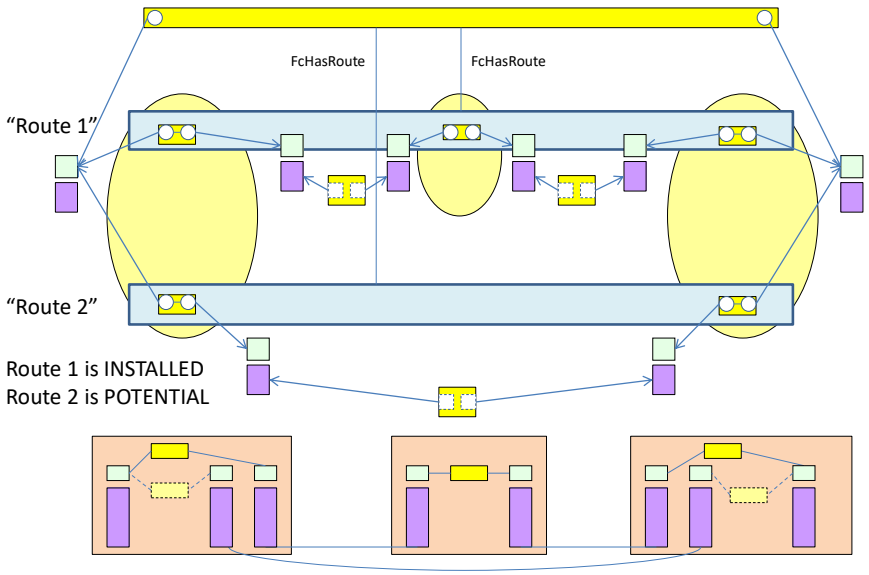


Figure 3-10 Showing a basic route based representation using FCs

The FcRoute has several potential uses:

- As part of the constraint model related to routing an FC
- As a description of future alternative ways through the network to cover variability in the service need or some other where only one is active at any one time (as depicted above)
- To represent each of a number of alternative ways through the network for a particular FC to provide resilience
- As a description of the current way through the network for a particular FC (current route)

The FcRoute may be fully detailed or quite abstract in terms of constraints.

The key focus in this document is the use of FcRoute for resilience. The actual instantiated active route across the network, i.e. the actual configuration of the real devices, must necessarily be fully detailed (otherwise information could not flow). But the definition of the desired route can be just a set of constraints that the actual route simply needs to satisify. Similarly the alternative routes may be simply constraints.

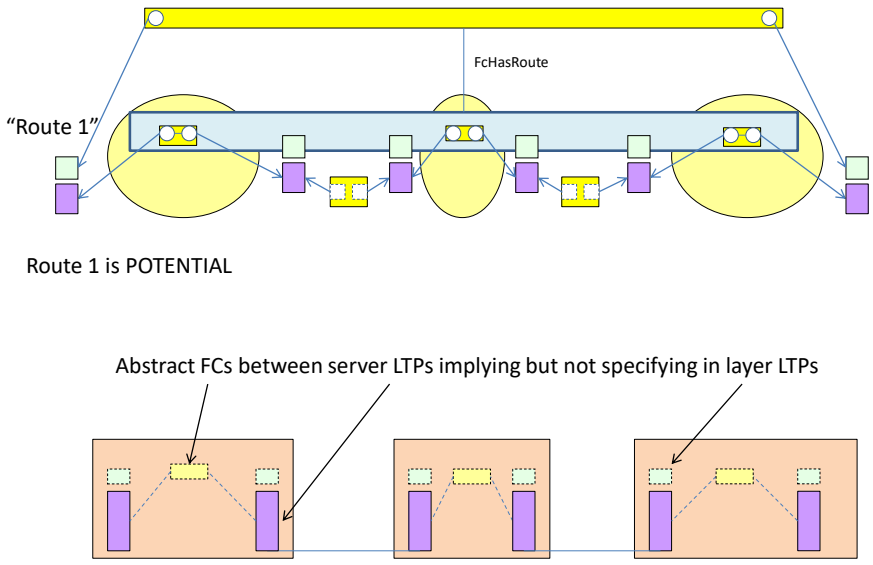


Figure 3-11 Showing a basic route based representation using abstract FCs

Essentially the route constraints used in conjunction with knowledge of the necessary layout constraints for the type of FC should be sufficient to allow an instance to be created.

The degree of detail available in a route definition depends upon a number of factors including design philosophy and level of risk tolerated. For example, to minimize the risk of the route not being successful when application is requested full detail will be required and the resources will need to be dedicated.

If the route is itself complex including combinations of switched segments etc then FC orientation may be critical and hence will need to minimally include abstract FCs whereas if the route is a trivial point to point structure with no embedded protection and the network technology does not restricted channelization and there is no committed bandwidth etc then the route can suitable be described in terms of just links.

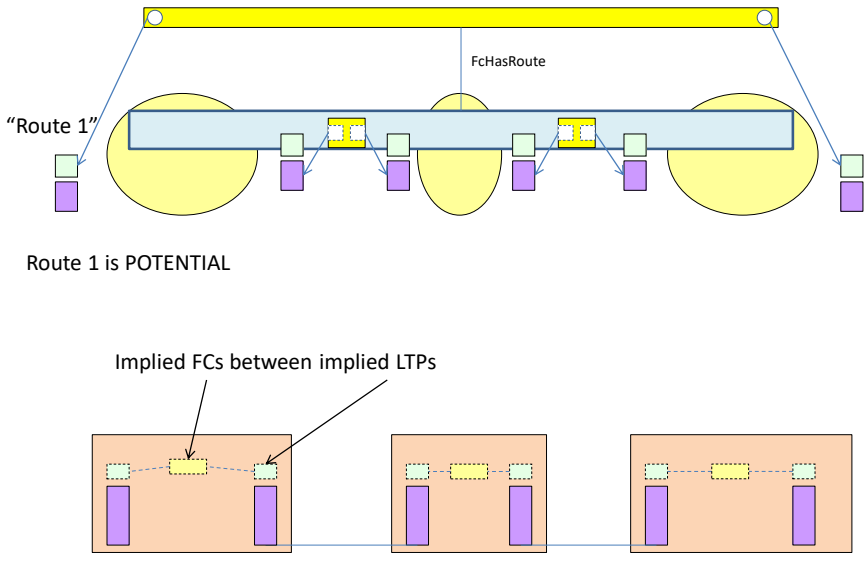


Figure 3-12 Showing a basic route based representation using Links

If there is committed bandwidth and there are restrictions in the FDs transited by the route then the route should be in terms of suitably detailed (partially detailed) FCs[10].

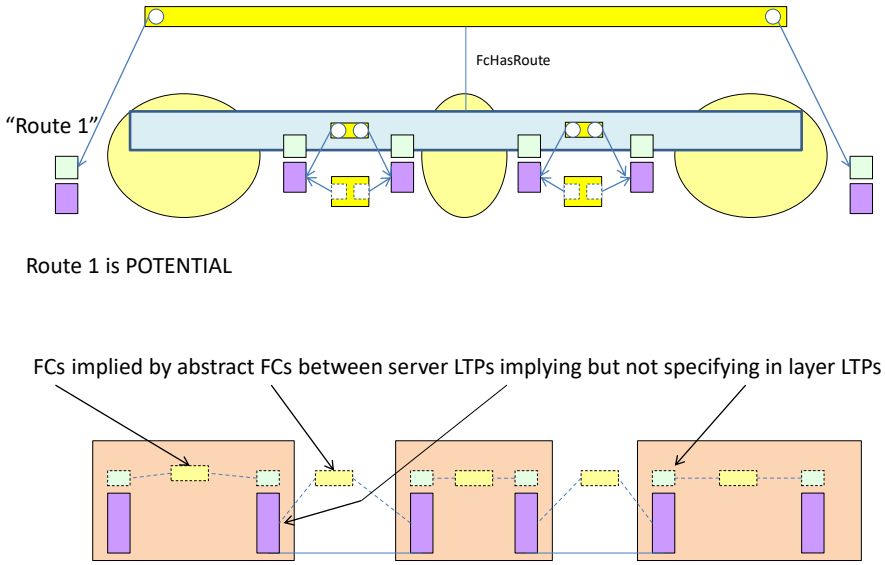


Figure 3-13 Showing a basic route based representation using abstract Link FCs

[10] It can be argued that the route should always minimally in terms of abstract FCs in either a Link or FD context so that properties such as bandwidth commitment etc can be recorded. This will be explored further in subsequent releases to help normalize route expression.

3.4.10 FcRoute LifecycleState

As implied in the previous section the FcRoute inherits the Lifecycle state attribute.

3.4.10.1 General considerations

The LifecycleStates of POTENTIAL and INSTALLED allow alternative routes reflect the selection state of a resilience scheme where INSTALLED means selected and POTENTIAL means deselected. When a route is selected it is available for use.

The LifecycleState of an FcRoute is:

- PLANNED: If the resources are not present in the network
- POTENTIAL_BUSY: If the resources are present in the network but are shared with other FCs and are currently used by those FCs
- POTENTIAL_AVAILABLE: If the resources are present in the network and are shared with other FCs but are not currently used by any FCs
- INSTALLED: If the resources are present and allocated to this FC (whether shared or not)
- PENDING_REMOVAL: If the FcRoute is INSTALLED and the intention is to remove the FcRoute

3.4.10.2 Protection

From the perspective of a protection scheme it is usual for all resources for routes to be present in the network.

3.4.10.3 1+1 Protection

In a 1+1 protection scheme, both of the worker (main) FcRoute and protection (standby) FcRoute have resources active in the network such that the LifecycleState of both will be INSTALLED (even if the route is not selected and there is no continuous traffic path as a result of switch states etc). The switch states are changed to select the route.

3.4.10.4 X:Y Protection

In a X:Y (X<= Y) scheme, although resources are present in the network but are shared and hence not necessarily available to protect a failure in a worker (main). A route may be POTENTIAL_AVAILABLE if the resources are not currently used by any FCs and POTENTIAL_BUSY when some or all of the resources necessary for protection are used by one or more other FCs.

When a route is "POTENTIAL_AVAILBLE" then some other process is required to configure and activate the resources of the route before it can be used by a protection scheme[11]. It is possible that even if the state of the FcRoute is POTENTIAL_BUSY a control process could have the authority to preempt and remove the blocking FcRoute.

3.4.10.5 Restoration schemes

In a restoration scheme there may be a number of alternative routes. At most one of those reoute will be INSTALLED. The other routes will be PLANNED. In some revertive schemes a preferred route (often called the home route) is remembered (pinned, retained) and the resources retained when an alternative is being used (due perhaps to failure of the preferred route). When the preferred route recovers the FC is caused to revert to it. When an alternative is being used the

[11] This is a typical case for ASON restoration where signaling is used to trigger distributed control components to activate the standby path.

preferred route is POTENTIAL_AVAILABLE (as it is not shared) and when it is used it is INSTALLED.

If the route is only described in abstract constraints then when it is INSTALLED the actual FCs abiding by the abstract constraints will be created. This actual FC will be added to the actual route and may become part of the route description for later re-instantiation if the policy for that FC indicates this should be the case.

3.4.10.6 Further considerations of state

It is currently not possible to distinguish, using states, between the following cases:

- All resources have been specified but not reserved
- Some resources have been specified but not reserved
- No resources have been specified

3.4.11 Route Feeds FcPort

In some views it is possible that the detail below the route is not accessible but it is still clear to the viewer that there are multiple alternative routes. In these views it is beneficial to indicate to the viewer that a particular route is being used to feed the output from a particular FcPort.

The FcPortFedByFcRoute association reflected in the _fcRouteFeedsFcPortEgress of the FcPort identifies which route feeds the FcPort.

The figure below shows a case where there are two routes. From the switch detail it is clear that the upper route is feeding to the right and the lower route is feeding to the left. However, the switch detail of the FC in the route, shown in the figure slightly greyed out, is not visible to the viewer. The _fcRouteFeedsFcPortEgress attribute in each FcPort of the visible FC references the relevant visible route objects.

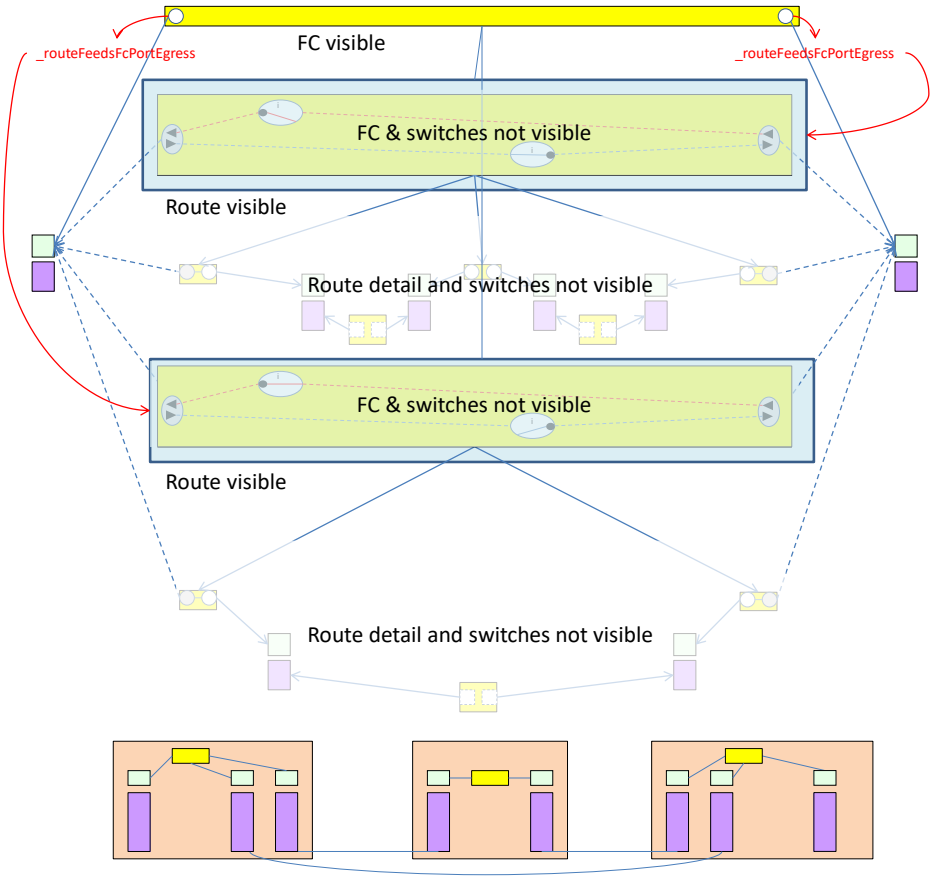


Figure 3-14 Understanding the active route in an opaque view

In more complex contexts there may be many routes and/or the upper-most FC may have multiple FcPorts. In these cases it is possible that multiple routes are actively feeding an FcPort such that information is flowing from at least one other FcPort on each of the routes[12]. In these cases the _routeFeedsFcPortEgress in each FcPort simply list all routes that are actively feeding that FcPort.

3.4.12 Abstraction of resilience viewed through the supported Link

The Link is an abstract view of underlying resources. It exposes the effects of the technology specific aspects and of the protection of the underlying network. The abstract view is in terms of the characteristics of forwarding and is covered in TR-512.4.

The request for service will be in terms of the essential characteristics of forwarding e.g. timing, integrity etc. The need for a resilience mechanism will be interpreted by considering these characteristics. If the distance between the points of delivery is very short the characteristics may be achieved with unprotected resources, if the delivery points are very distant then the same characteristics may require some form of resilience. The resilience chosen will depend upon the

[12] Clearly this is only applicable to some network technologies.

specifics of the characteristics. Certain combinations of characteristics will not be achievable beyond a certain distance.

Because the resilience scheme is chosen to enable particular characteristics to be achieved then the abstraction of the scheme as viewed at the links should be in terms of those characteristics. It is not meaningful to express in the Link the type of scheme used in the underlying network because:

- This violates the intended opaqueness (as the user of the link abstraction would need to understand the meaning of a scheme in a foreign layer or domain)
- It does not provide the user with sufficient information to derive the characteristics of the scheme since these will depend upon many factors including length and these cannot be interpreted without deep knowledge of the underlying network

On this basis the expression of the underlying resilience viewed through the link should simply be in terms of generalized properties of forwarding.

Clearly, if the network is not hidden the user can navigate from the Link to the underlying network structures. The opportunity to navigate to the server FC and from that to the resilience scheme details is fully supported by the model[13].

The figure above highlights two key relationships in red. These provide the most direct inter-layer resilience navigation. If an FcRoute is described in terms of Links the underlying FC in the server layer (that represents the Trail as per [ITU-T G.805]) can be identified by reverse navigation of the FcSupportsLink association. This will arrive at the supporting FC and all relevant resilience details. It is also possible to navigate via the Linkport to the LTP and then from the LTP to the supporting FC and all resilience details/

3.4.13 Overlaying and chaining switches

The following figure shows an abstract example of an FC with chained switches using the internal FcPort where some FcPorts are fed by more than one switch. As it notes in the figure the layout is arbitrary, solely for the illustration of the model capability and is not a representation of any particular switching scheme. As usual the FC spec would explain, in terms of switched flows (which represent FcSwitches), the range of switching opportunities and distinguish the port roles.

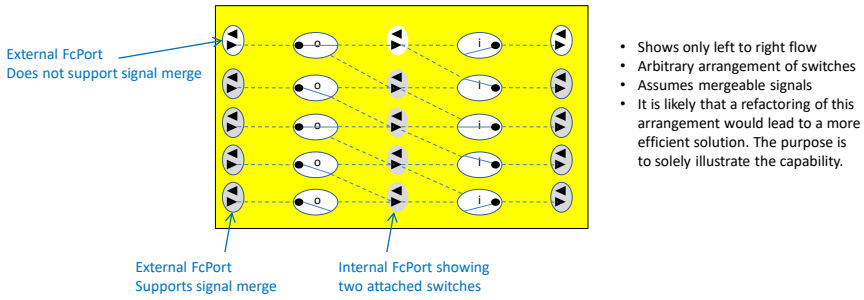


Figure 3-15 Internal FcPorts and Ports fed by several switches

[13] In many cases it is not possible/allowed to represent full server details, hence an abstraction would be used.

3.4.14 Controls from CascPort

The CascPort supports both the sending of signaling through and/or application of control to the associated LTP and/or the gathering of monitoring information from the associated LTP. The controls can be applied directly to the associated LTP and/or indirectly to an appropriately deterministically related LTP peer or server to the associated LTP as described by the scheme spec (see TR-512.7). The same applies to the gathering of monitoring information.

Considering [ITU-T G.8032] protection as an example the control parameter related to the "isRelatedControlFlowDisabled" property of the port applies also to the indirectly related LTP dealing with the control signal and the "isControlledFcPortDisabled" property of the port applies specifically to the port of the controlled FC as explained by the scheme spec.

In addition the scheme spec will indicate whether the actual state of each individual controlled FC can be determined directly from the FC or whether only the aggregate state is available. Clearly the former may cause performance issues in an implementation if hundreds of FCs are controlled and switched together especially if notifications are sent for changes in every one independently.

3.4.15 Use of FcSpec to explain unexpected flow through a protection scheme

The FcSpec is used to state rules for and constraints on flows through the FC so as to define the FC internal interconnectivity. The normal usage is to provide an FcSpec per type of FC. Clearly the intended flows in a protection scheme can be stated in terms of an FcSpec.

Under some failure conditions the flows in a protection scheme may not reflect the expected flows. Under these circumstances it is possible to use an FcSpec structure to describe the unexpected flows. Such an FcSpec could be made available as part of the description of the protection scheme if the failure modes are deterministic and the range of different flow patterns were limited.

In the case where the failure patterns are extensive, rare and not readily pre-calculable on occurrence of an unexpected flow state a temporary FcSpec could be constructed to express the current flow case.

The following basic network can be used to illustrate the complex behavior. The network includes four NEs. Each NE is configured as shown for the upper right NE with Interconnect-Protection[14]. The external view of the effect of the configuration of FCs in the NEs is Back-to-Back-Protection (depicted as Offered/Desired).

[14] This is an illustrative FcSpecName.

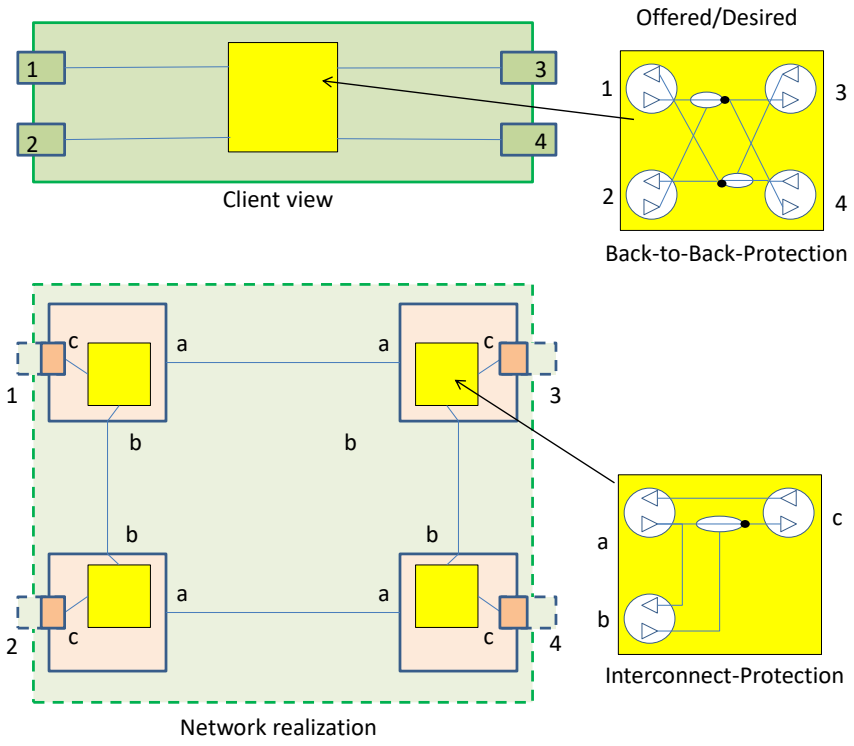


Figure 3-16 Basic network showing back to back protection abstraction of underlying protection

Under single failure conditions the external effect is still Back-to-Back-Protection

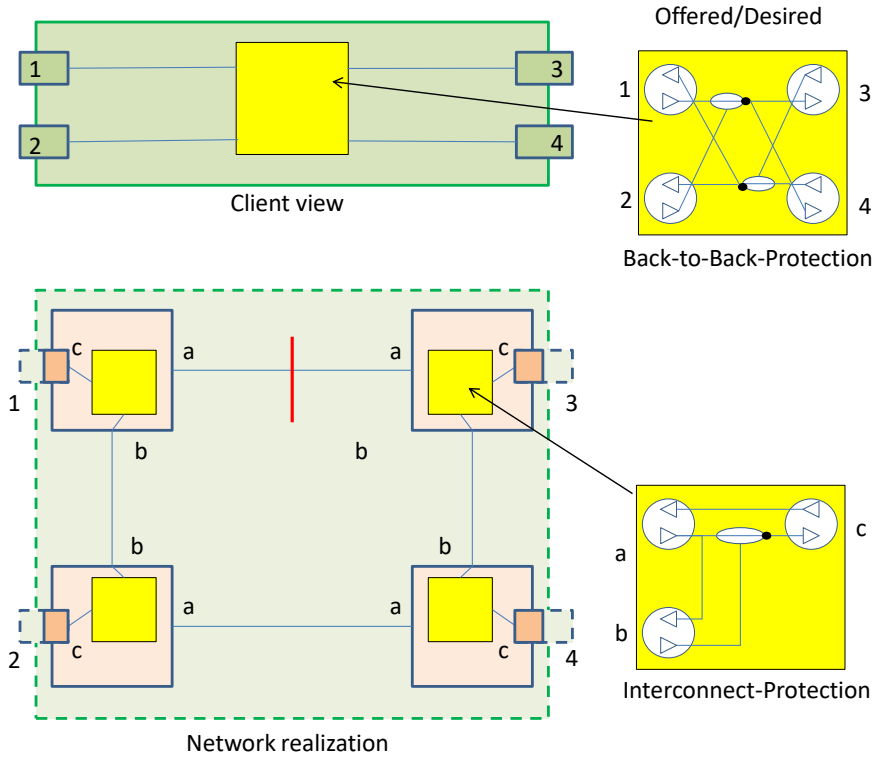


Figure 3-17 Single failure in network

When there is a failure of the input to the protection scheme operates as expected by the Dual 1+1 definition.as shown below.

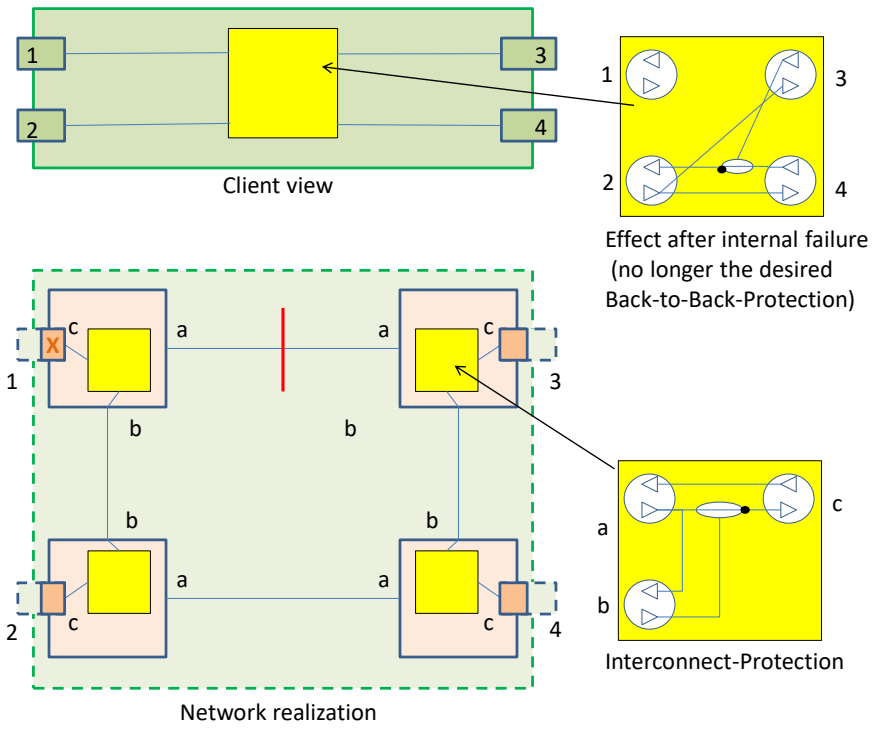


Figure 3-18 Failure at an input to the network

However under certain internal failure scenarios the network is split into two and there is a non-desired flow. Although from the external perspective traffic is being delivered at both ports 3 and 4, external failures will not give the desired Backto-Back-Protection characteristic.

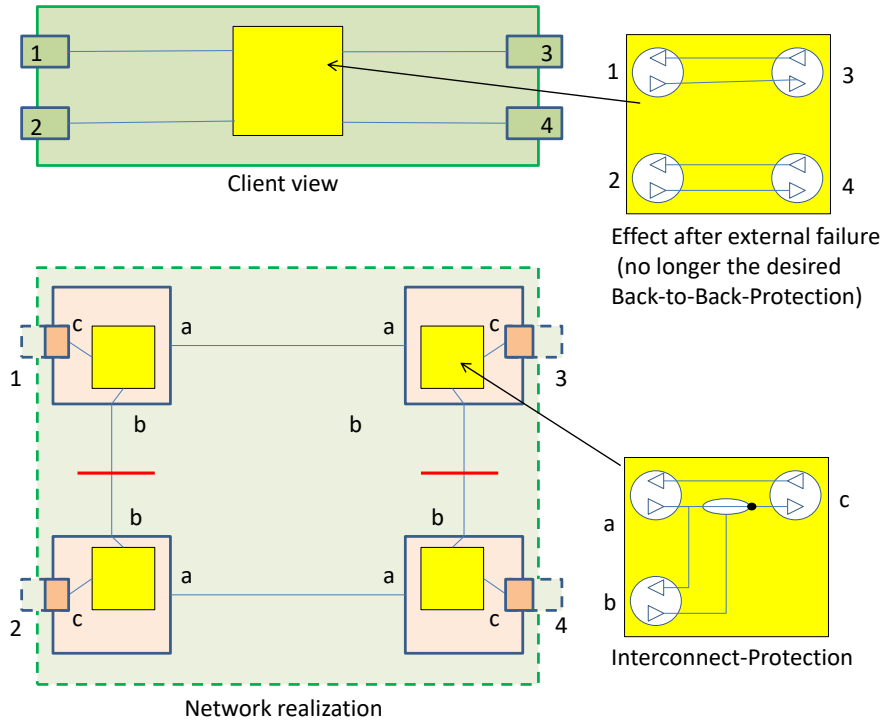


Figure 3-19 Two internal failures

A failure occurring on port 1 will, unexpectedly from the client's perspective, cause the output at port 3 to fail.

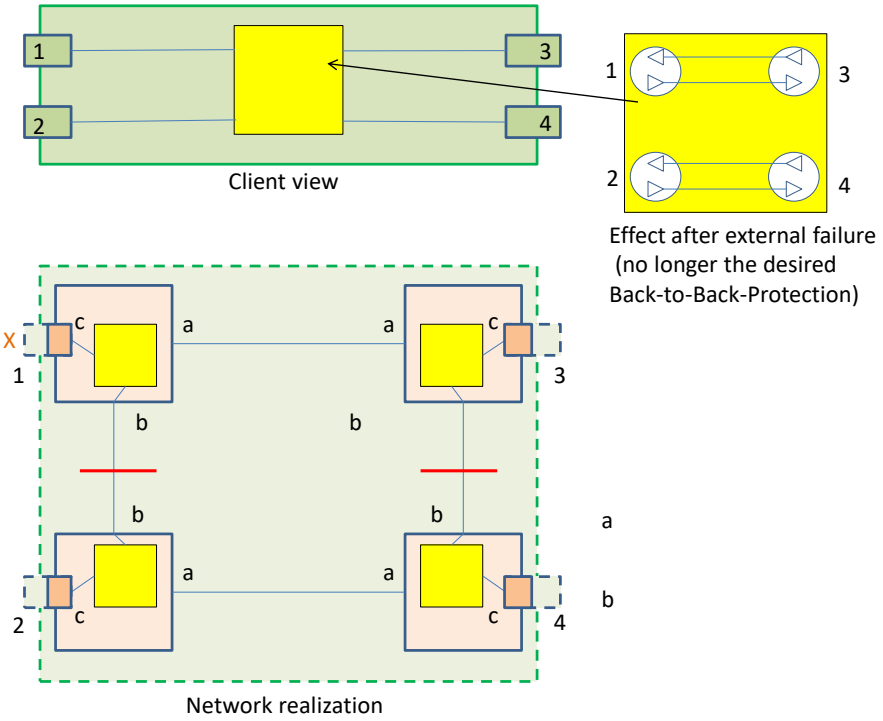


Figure 3-20 Two internal failures with external failure

The figure below shows the potential states of flow of a realization of Back-to-Back-Protection under internal failure modes (some are dependent on their being a more sophisticated underlying network than that shown in the figure above). The states highlighted in the red ellipse are not expected from the simple external presentation of Back-to-Back-Protection. Only one direction of flow is shown to reduce clutter, the diagrams have been simplified to show only the flow (not the specific switches and the port numbers have been generalized (maintaining the orientation as per the the Back-to-back-protection shown in the figures above). This is further explained later in this section.

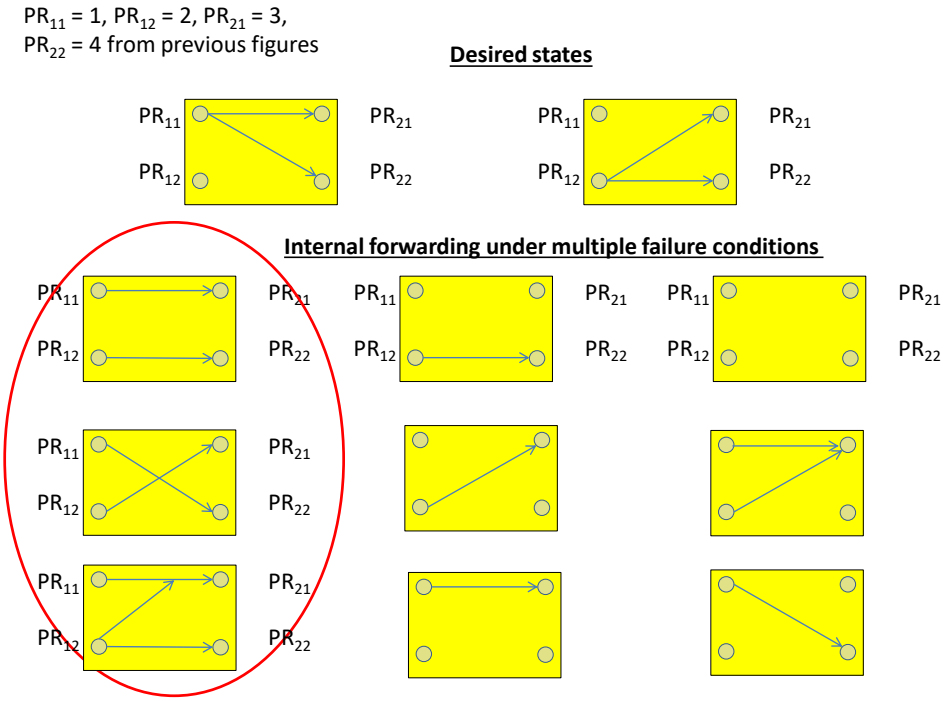


Figure 3-21 Representation of forwarding under normal and failure conditions

The FcSpec (see TR-512.7) can be used to represent the Back-to-Back-Protection scheme as shown in the figure below.

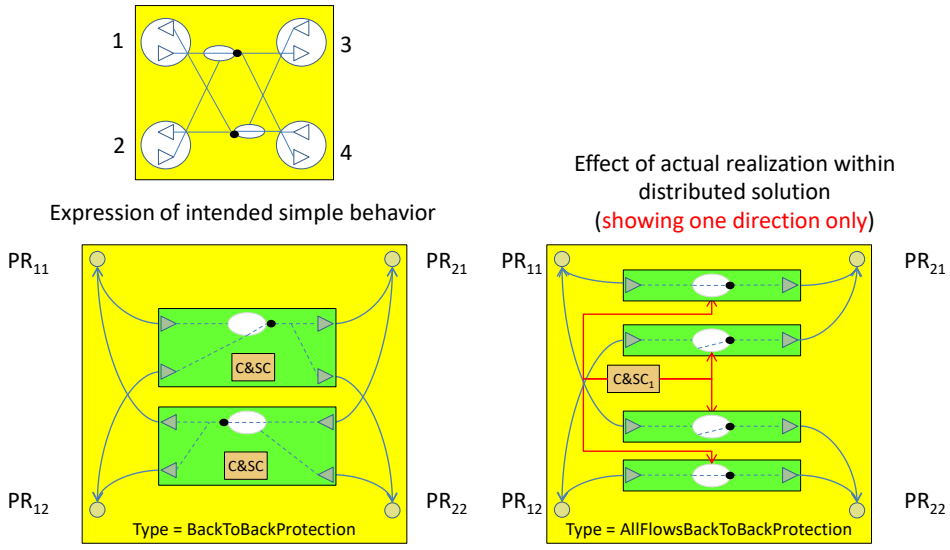


Figure 3-22 Spec for Back-to-BackProtection

The FcSpec can also be used to represent any of the desired and undesired forwarding patterns. An example is shown in the figure below.

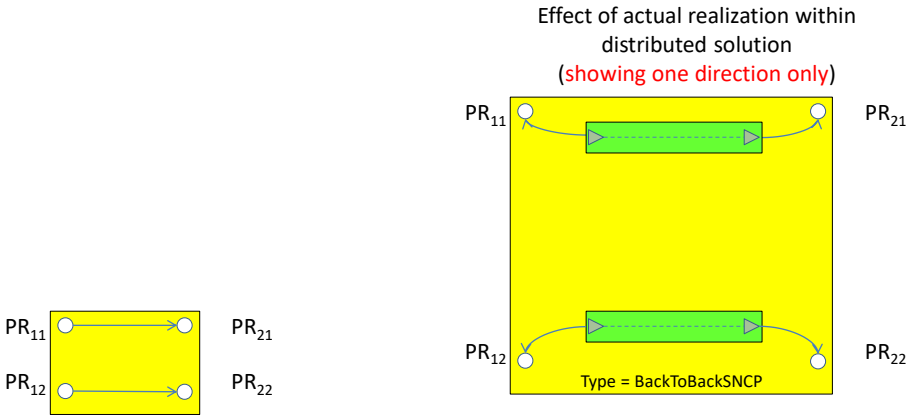


Figure 3-23 Spec representation of one of the undesired cases

The figure below shows another one of the failure cases.

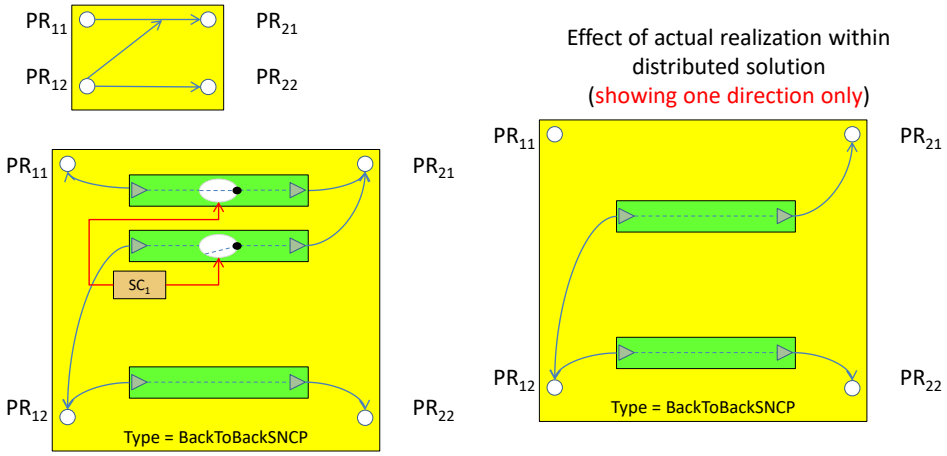


Figure 3-24 Spec representation of another of the undesired cases

3.4.16 Dealing with multiple control domains (this section requires further work)

There are many network cases where an NE participates in several resilience schemes where two or more scheme instances are responsible for the resilience of a single information flow that transit the NE. The model supports two ways of dealing with this:

- Single FC with split control
- Two FCs each with separate dedicated C&SCs where the two FCs are connected by a zero-length-link

Two options… Zero length link or single FC.

4 Protection schemes considered

The resilience model is designed to support standard resilience schemes in a consistent fashion. The model has been exercised for a number of schemes (see TR-512.A.11).

Resilience schemes considered in detail include:

- Linear protection including
  - 1+1
  - 1:1
  - 1:N
- Mesh protection including
  - N:1
- Ring protection including
  - [ITU-T G.8032]

The above schemes are protection schemes. Various restoration schemes are also supported by the model but these have not yet been covered in detail in the examples.

5 Protection of other functions of physical things

The Physical model covered by TR-512.6 which focuses on the modeling of Equipment. The Equipment is considered to be purely physical. The document also provides some modeling of functions that are emergent from a physical assembly when powered. Clearly, all functions including those encapsulated by LTPs and FCs are only realized in by a powered physical assembly.

The functions being supported by the equipment can be protected. This type of protection often goes under the name "Equipment Protection". This name has not been used as it blurs the intentional constraint that Equipment is purely physical (where a physical thing can be measured with a ruler). Physical things are not protected, the functions that they support are protected; it is functions supported by additional (redundant) physical things that give rise to resilient/protected functions.

The TR-512.6 document provides a sketch of how functional resilience could be represented. This aspect is for further work in the next release. The intention is to use a switch/controller based pattern to represent functional resilience/protection.

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

6.1 Signaling information flow

Some signaling flow considerations have been covered in this release but there is a need to cover the general case of inter-controller signaling. This will be tackled in a later release.

There are two distinct cases to consider:

- Closed: where the signaling/messaging is solely within the visible/controlled network
- Open: where the signaling/messaging emerges from the visible/controlled network

The open case occurs where, for example, there is an admin boundary the cuts a protection scheme and where the administrative entities have agreed to enable their management/control systems to exchange messages/signals to achieve inter-administration automation. This applies to B2B exchanged and E-NNI exchanges[15]

6.1.1 Closed case

- Current assumption is that a controller that uses signalling is identified in the appropriate spec
  - The model uses ControllerGovernsController in both directions to indicate a peer.
    - Attributes could be added to indicate whether the controller is signalling to a peer or not and that the signalling grouping is determined from the spec and switch orientation
- It is possible to show

[15] The intention in the long term is to unify these two currently distinct considerations under one single architecture.

  - Signalling flow through the network by associating the C&SC with an LTP as for the [ITU-T G.8032] model supported in this release
    - The LTP spec explains the adaptation and hence association with another C&SC can be derived
  - A direct peer association between C&SC with no view of the underlying mechanism
  - A full forwarding model for the signalling information flow
    - This could be in a referenced pattern that is the summarized rigorously in one of the above forms
    - The resilience scheme spec would explain the signalling flow alternatives
- Note that a full forwarding model is necessary when the signalling flow routing is not coincident with the traffic flow routing
  - An attribute could be added to indicate that signalling is co-routed with the traffic being controlled

6.1.2 Open case

- This case has an open signalling path so there needs to be an expression of the signalling where it will emerge explaining what it is etc. Signalling information is exposed at the edge of the network
  - Again current assumption is that a controller that uses signalling is identified in the appropriate spec
    - Also with the attribute to indicate whether the controller is signalling to a peer or not and that the signalling grouping is determined from the spec and switch orientation
  - The ControllerGovernsController cannot name peer as it is not within the view so an off-net form of foreign pointer is necessary (or there could be a dummy controller with a few parameters (perhaps discoverable, perhaps manually entered) as well as the name)
- Potentially more relevantly in this case we could show
  - Signalling flow through the network by associating the C&SC with an LTP via a new association that indicates that signalling information is sent through the adapter of the LTP
    - The LTP spec would explain the adaptation and hence association with another C&SC could be derived
  - A direct peer association between C&SC with no view of the underlying mechanism
  - A full forwarding model for the signalling information flow

- This could be in a referenced pattern that is the summarized rigorously in one of the above forms
- Note that a full forwarding model would appear to make sense when the signalling flow routing is not coincident with the traffic flow routing
  - An attribute could be added to indicate that signalling is co-routed with the traffic being controlled

6.1.3 Signaling control

- Need to identify parameters related to signalling and control that are independent of switching or only partly dependent on switching
  - Can timers be adjusted?
  - Can signalling be disabled?
  - Can aspects of signalling be disabled?
  - Can control be adjusted?

6.2 Additional considerations for FcRoute

An FcRoute may:

- Be provisioned in the network but turned off
- Have resources reserved but not provisioned in the network
- Have resources that are reserved but shared with one or more other routes (either in the same FC or a different FC)
- Have specified but not reserved resources
- Have partially specified resources
- Have no resources specified and hence no subordinate FC detail

This implies the need to add properties on LifeCycleState (reserved, provisioned etc for the route) and to support a route in terms of constraints

An FcRoute may have encapsulated protection or other complex nesting of resilience schemes. Whilst the model supports this it has not been exercised with any cases. The figure below has a sketch of two alternative routes both of which have internal protection

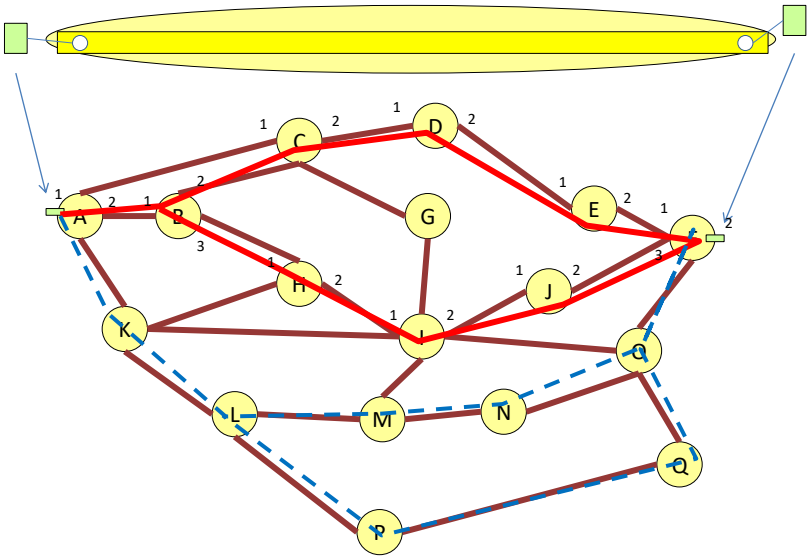


Figure 6-1 Sketch of two routes with internal resilience

6.3 Representation alternatives – Partition or Route

Consider the figure below of a simple network with relatively sophisticated switching scheme with a single FC spanning from A1 to C2.

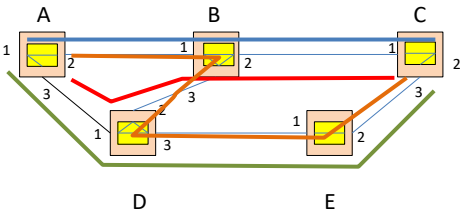


Figure 6-2 FcRoute in a complex network

A1-C2 has four routes each of which has one FC

- Blue: A1-A2, B1-B2, C1-C2
- Red: A1-A3, D1-D2, B3-B2, C1-C2
- Green: A1-A3, D1- D3, E1-E2, C3-C2
- Brown: A1-A3, B1- B3, D2-D3, E1-E2, C3-C2

In more complex cases there could be many potential routes for a sophisticated switch configuration where there are only a few well defined switches.

Adding two more nodes and two more switches would double the number of routes. Adding more ends would further multiply the number of routes.

For complex layouts the route approach is not an efficient way of expressing the layout and instead the FC partition should be used.

If there are alternative FC partitions as a result of their being a combination of protection and restoration each FC partition will be considered as a route (where each route is composed of FCs).

6.4 Relationship to the ProtectionGroup approach

The brief figure below sketches the relationship between a Protection Group approach and the FcSwitch. Further work is required to formalize the relationships.

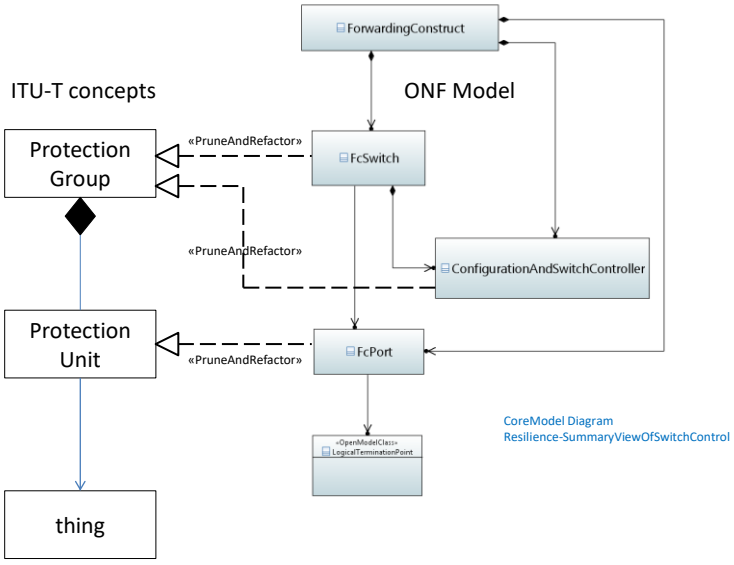


Figure 6-3 Relationship between FcSwitch approach and ProtectionGroup approach

End of document