Spec model thoughts

Nigel Davis (Ciena)
WORK IN PROGRESS
Developed in collaboration with the modelling team

Version 20150306
(Updated to align with FD/FC)

Additions and changes in this version

- Updated to align with change from SN to FD and SNC to FC

Purpose

- To provide a model form to be conveyed as a specification of static aspects and ranges of dynamic aspects of classes that encapsulate significant complexity

Approach

- Information handled at several “degrees” of conceptual indirection
  - First degree is the instances of the actual classes representing elements of the real network and it provides attributes about flexible/dynamic aspects of the solution
  - Second degree is the specification instances representing the types of things in the problem space (e.g. protected FC – “protected” is the type of FC)
- At management-control system initialization, and then ongoing as newly invented capabilities appear, second “degree” information is gathered in the form of specifications of static aspects of types of structure in the solution
  - This will include gathering specifications for the variety internal structures of FCs and LTPs
  - This will also include specifications for distinct network structures arrangements and NE internal arrangements etc
  - This information may be gathered from the network, a central server or the manufacturer
  - This second “degree” information is necessary to interpret the first “degree” information fully
- Once all known specs are gathered first “degree” information about the network is gathered and alignment maintained
  - This will include the instances of LTP and FC that represent the current network state
  - Each item of first degree information has a type that references a spec
    - It may also have pointers to profiles etc that are not covered here

Consideration

- Two aspects to the approach
  - Information structure
  - Code
- Challenge is what balance
  - All code will make the solution opaque and not portable
  - All information structure will make the solution cumbersome
- The following proposes a balance

Essential specification classes

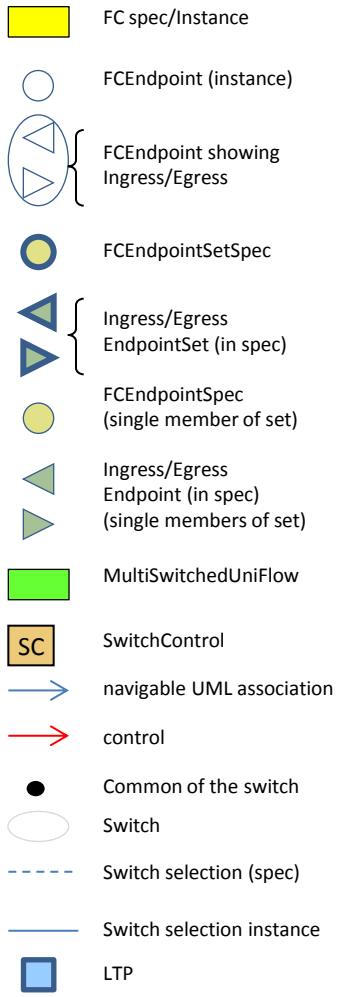
- Use classes very similar in structure to those in the existing model
- Key differences
  - Level of application is finer grain than the normal instance model
  - Some attributes have range values or abstract values rather than real values
- The spec form provides the arrangement of the internal parts of the instantiated classes
  - It essentially provides a description of a pattern of parts that can be reused

Describer this

- FC class
- FC instance
- FC spec class (format and rule for..)
- FC spec instance (profile of…)

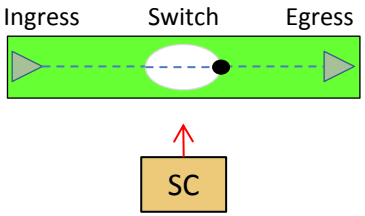
Show FC instance

Associations need to be reversed



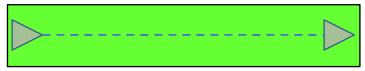
B	Bridge	P	Protecting	suf	Switched Unidirectional Flow	msmuf	Multi-ingress Switched Multi-cast Unidirectional Flow
E	Extra Traffic	R	Resilient	uf	Unidirectional Flow	uc	Unidirectional Connection?
W	Working	S	Standby	muf	Multi-cast Unidirectional Flow	usc	Unidirectional Switched Connection?
N	Normal						

FC spec considered

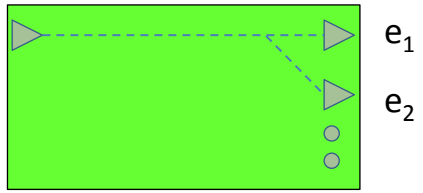


Switched Unidirectional Flow (this is the fundamental unit of specification of an FC) [suf]

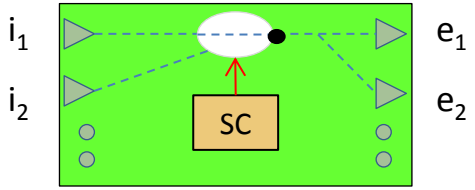
Switch Control rules and state machine



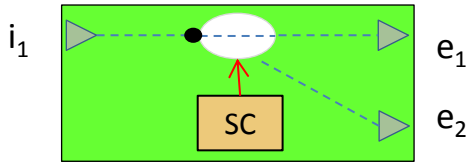
Unidirectional Flow (a switched flow with a rule on the switch = True) [uf]



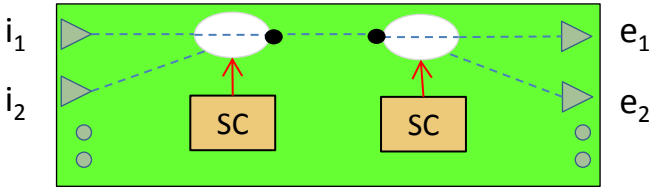
Multi-cast Unidirectional Flow (uf with multiple egress) [muf]



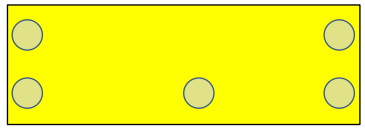
Multi-ingress Switched Multi-cast Unidirectional Flow (suf with multiple egress) [msmuf] Showing Switch Control embedded



Switched egress Unidirectional Flow [seuf]
Showing Switch Control embedded

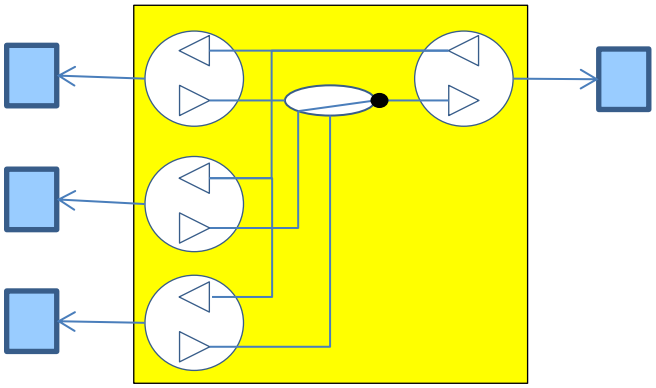


Switched ingress and egress Unidirectional Flow
Showing Switch Control embedded



FC spec macro (showing a random selection of endpoint roles)

Example for P-way protected

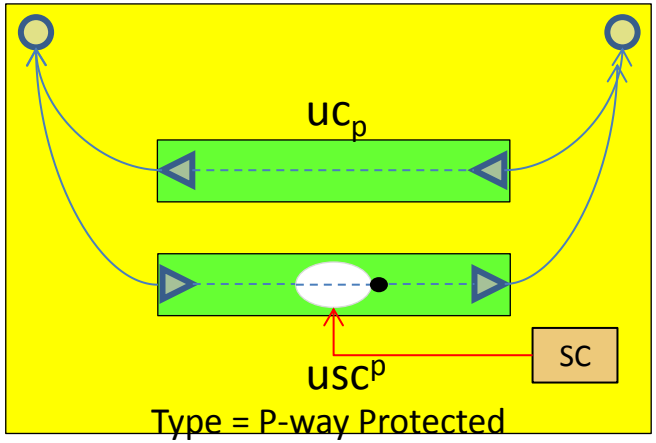


FC instance (showing LTPs)

P = protecting

P_p
p=2..n

$usc_p=P_p$
$uc_p=P_p$



R = Resilient

R_r
r=1

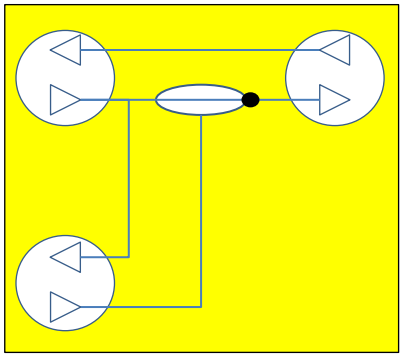
all = r^1

FC spec instance

Rule = switch^x= True Then switch$^{<>x}$ = False

Need to project to a single switch at top level (equivalent)

Example



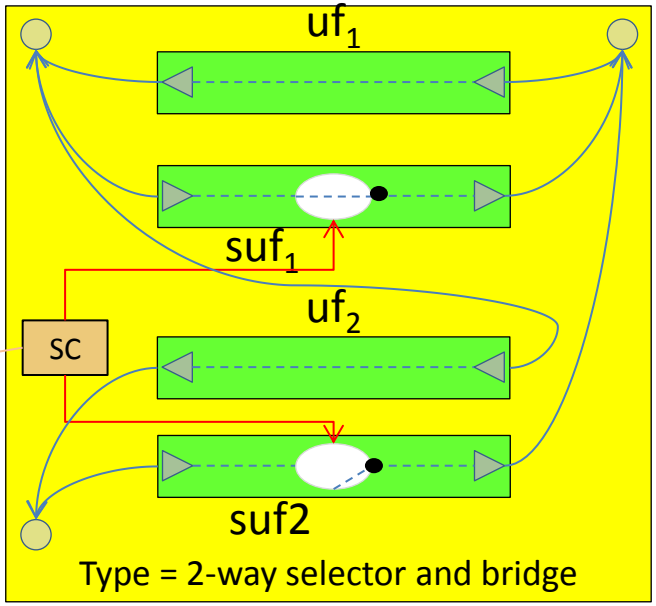
P = protecting

R = Resilient

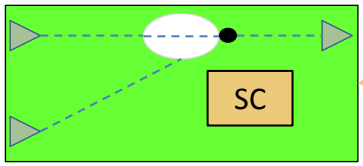
P_p
p=1

R_r
r=1

all = R_1



P_p



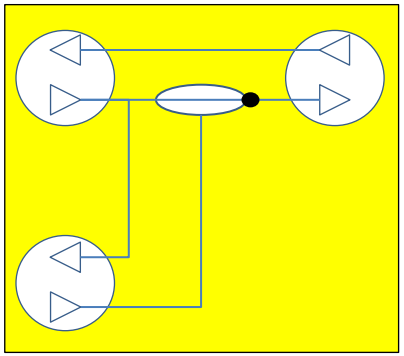
R_r

B_b

B_b
b=1

B = Bridge

Example



Actual FC

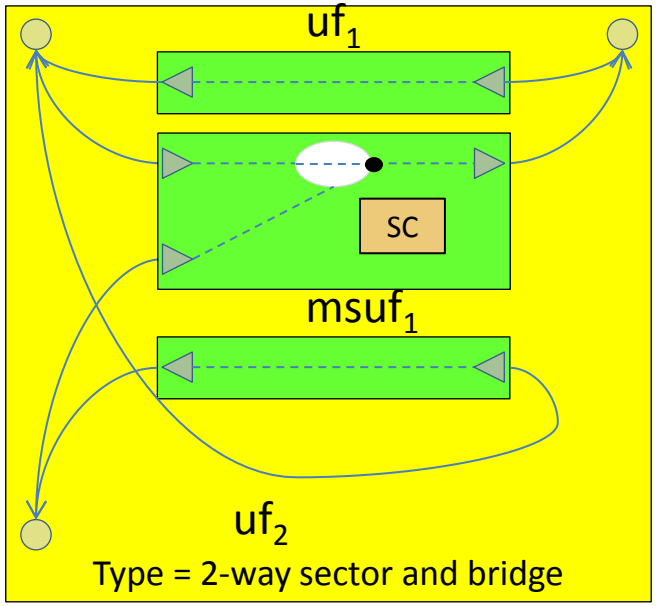
P = protecting

R = Resilient

P_p
p=1

R_r
r=1

all = R_1

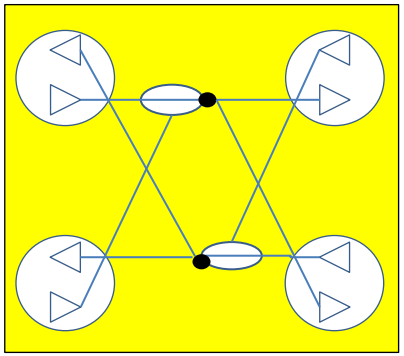


R_r

B_b
b=1

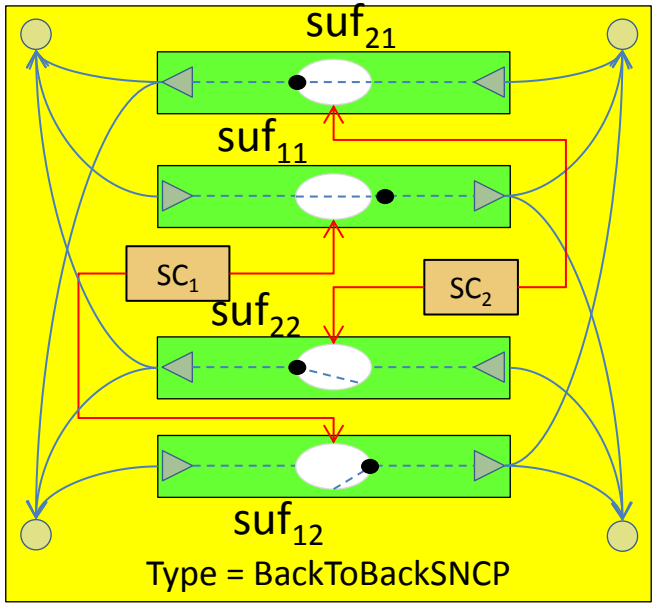
B = Bridge

Example

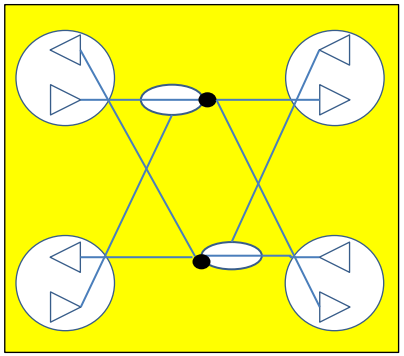


P = protecting

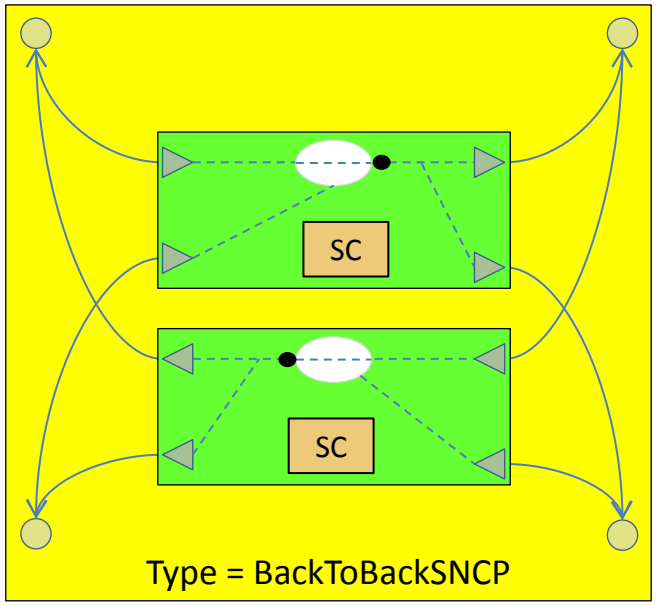
R = Resilient



Example



P = protecting
R = Resilient



PR_{11}

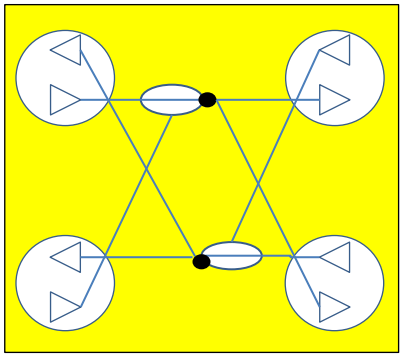
PR_{21}

PR_{12}

PR_{22}

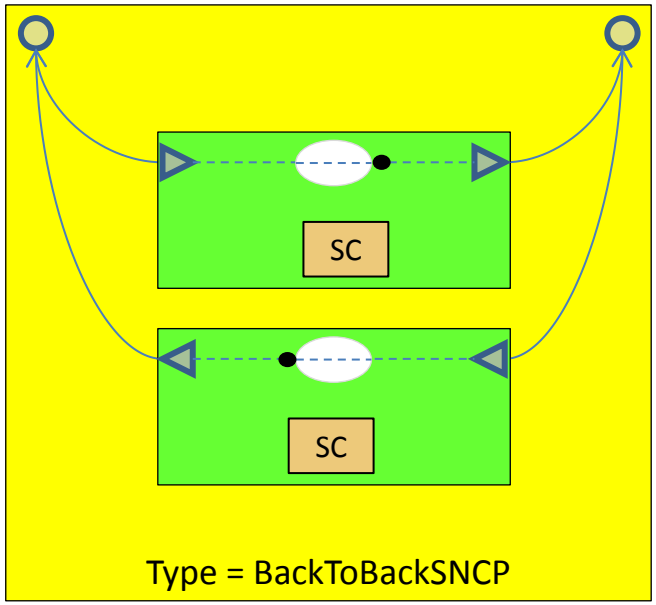
These are NOT endpoint set specs

Example



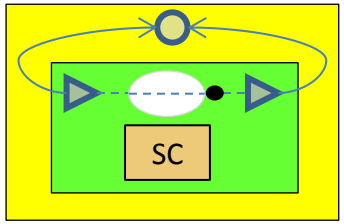
P = protecting
R = Resilient

PR_{1i}

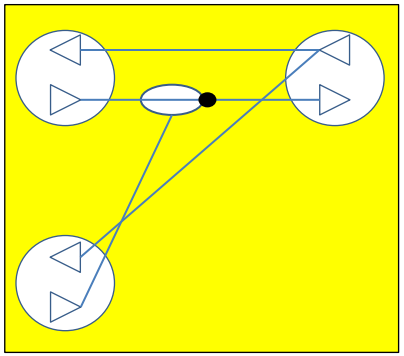


PR_{2j}

PR_{ij}

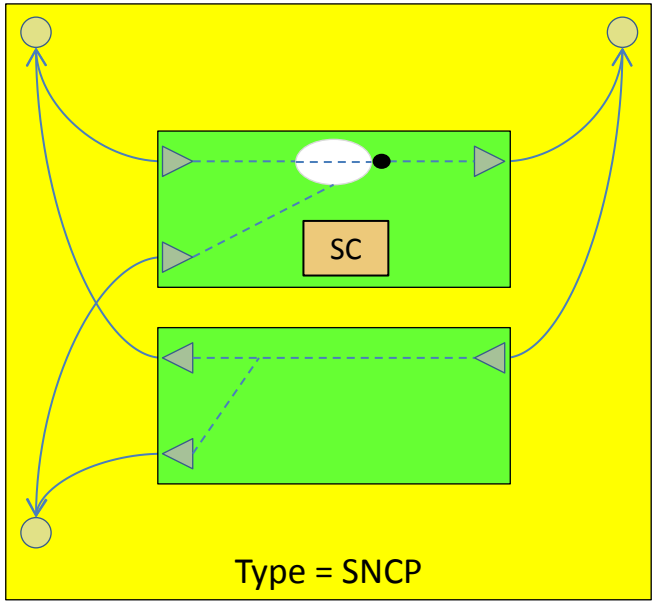


Example



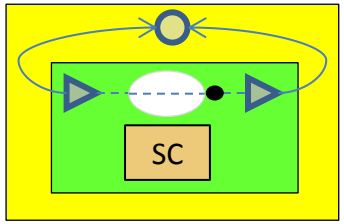
P = protecting
R = Resilient

P_1



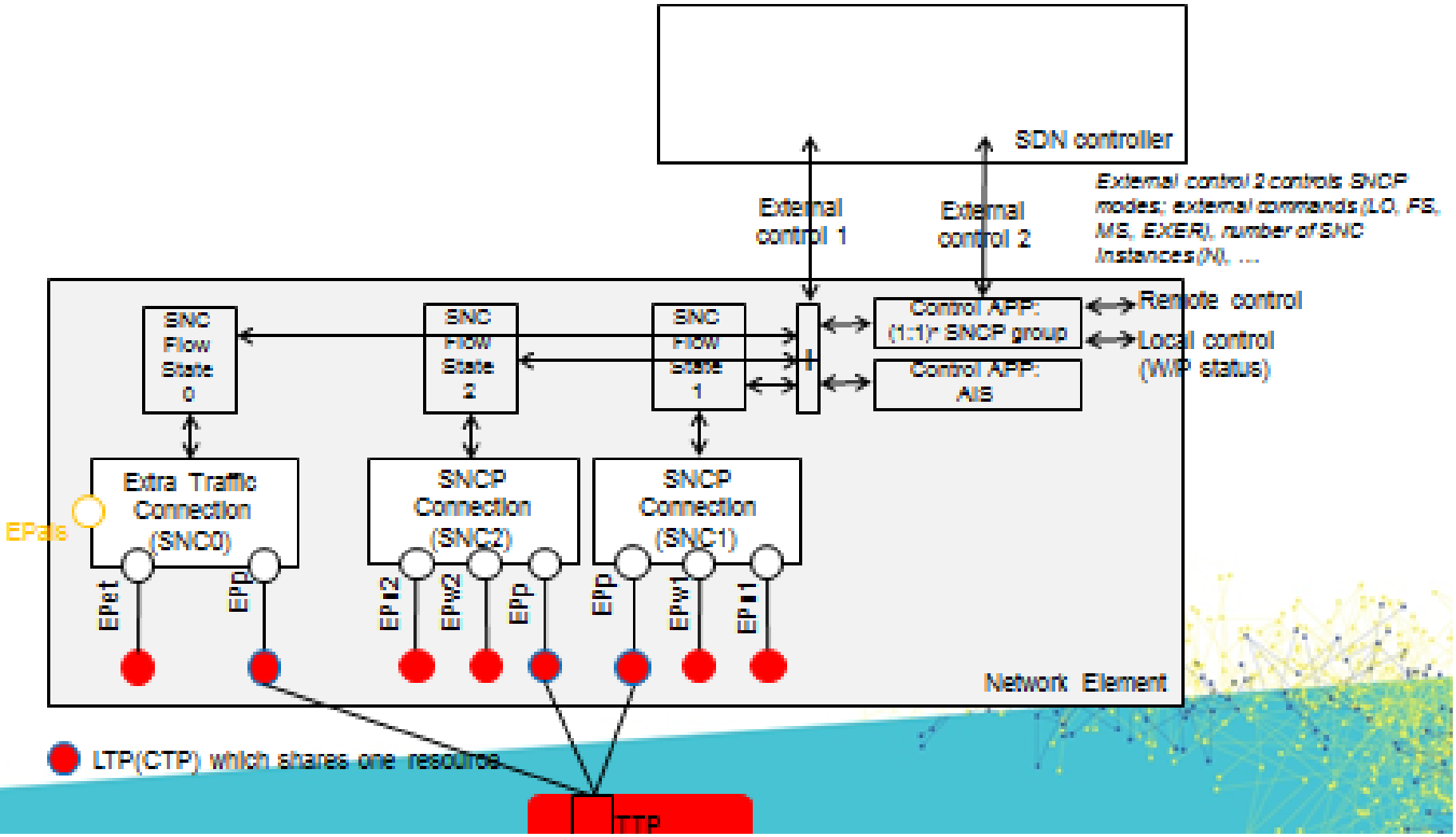
R

PR_{ij}

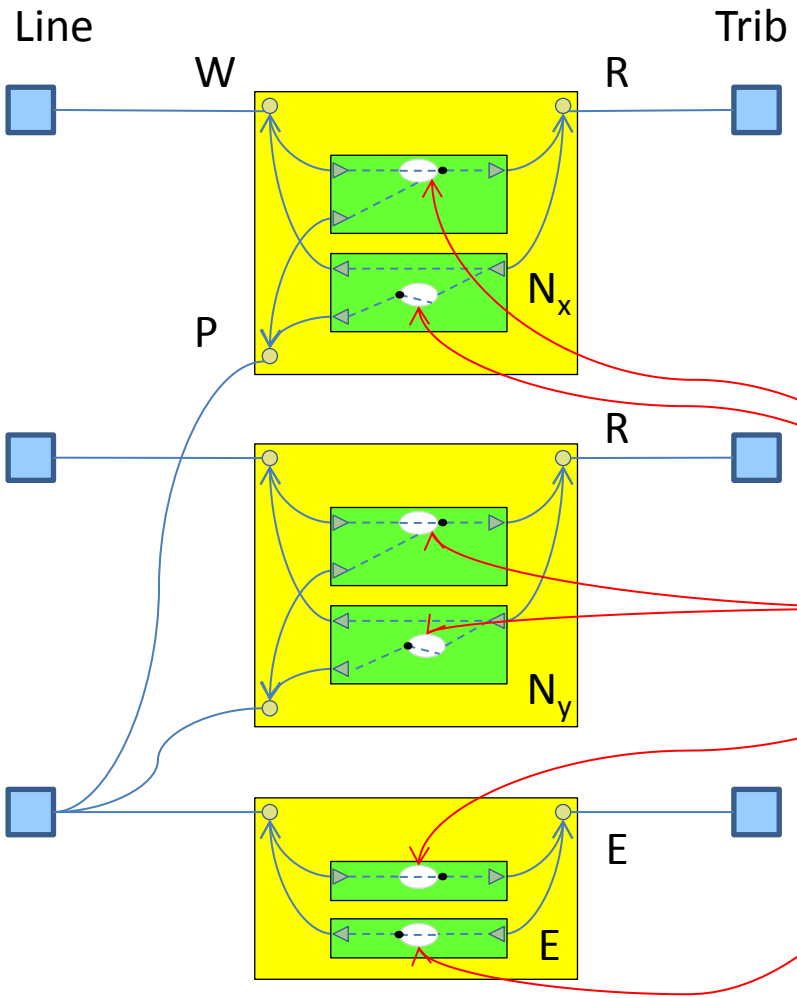


P_2

Architectural elements and overview of potential flows for connections in $(1:1)^N$ SNCP group



Semi Instance view



Commands

- Switch,Force -> E, N_n
  - Select E Then E=True, N_n=W
  - Select $N_{n=x}$ Then $N_{n=x}$=S, $N_{n<>x}$=W, E=False
- Lockout -> E, $N_{n=x}$
  - Select <> Lockout

Kam: Need better format for the equations. E.g., use the symbols such as ∀ (for all), ∃ (there exists), ≠, ∈, ∉, etc.
Let I = {i: i=1, ..., n}.
If select E, then E = True and N_i=M, ∀ i, i ∈ I
If select N_i, then E=False and Ni=M, for ∀j, j ∈ I and j ≠ i.

SC

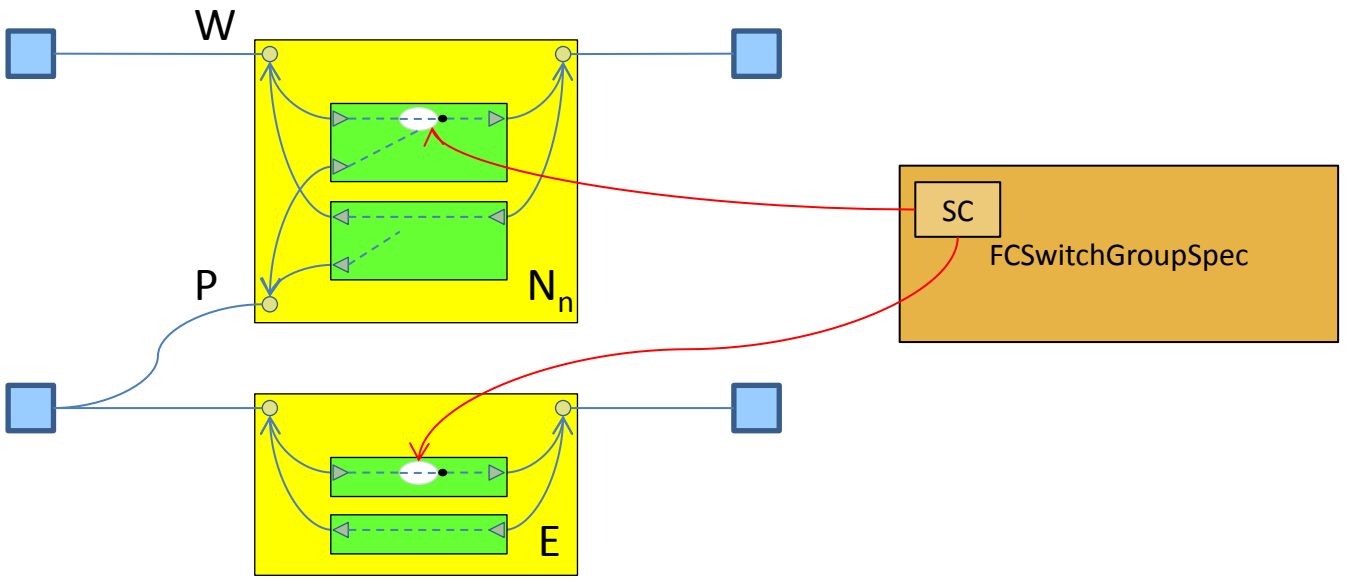
FCSwitchGroupSpec

Spec view

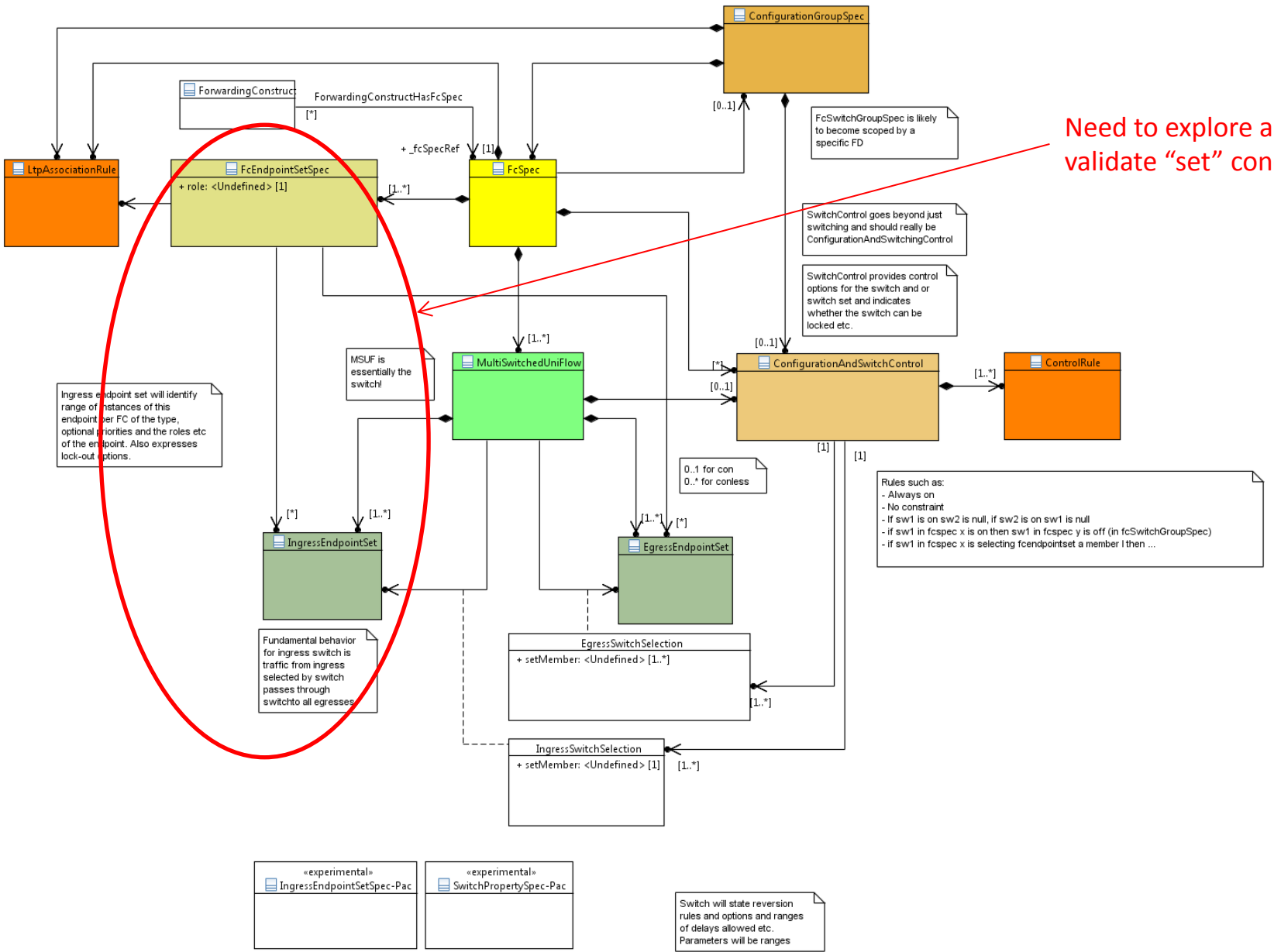
Commands

- Switch,Force -> E, $N_{n=x}$
  - Select E Then E=True, N_n=M
  - Select $N_{n=x}$ Then $N_{n=x}$=S, $N_{n<>x}$=M, E=False

- Lockout -> E, $N_{n=x}$
  - Select <> Lockout



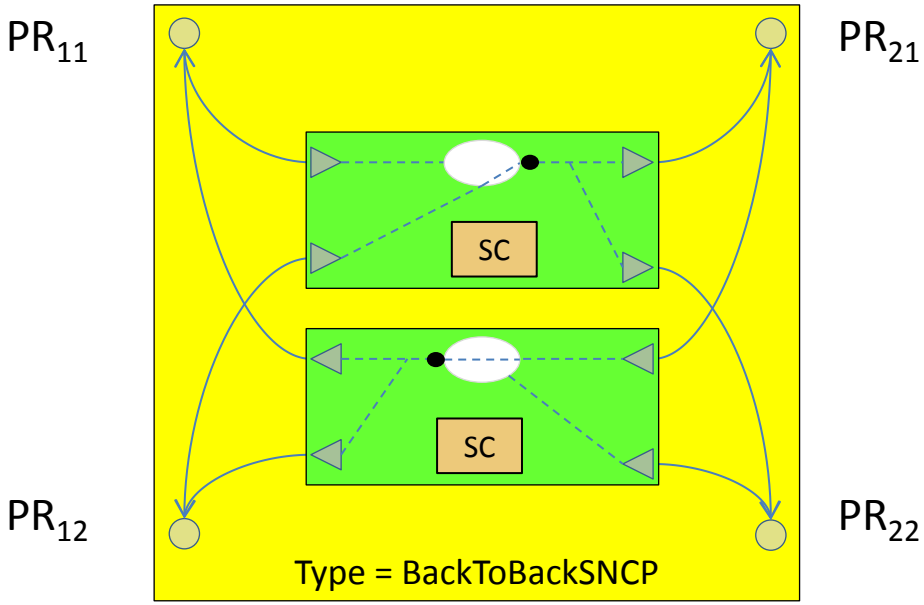
Snapshot view of Papyrus model (work in progress)



Abstraction simplifications

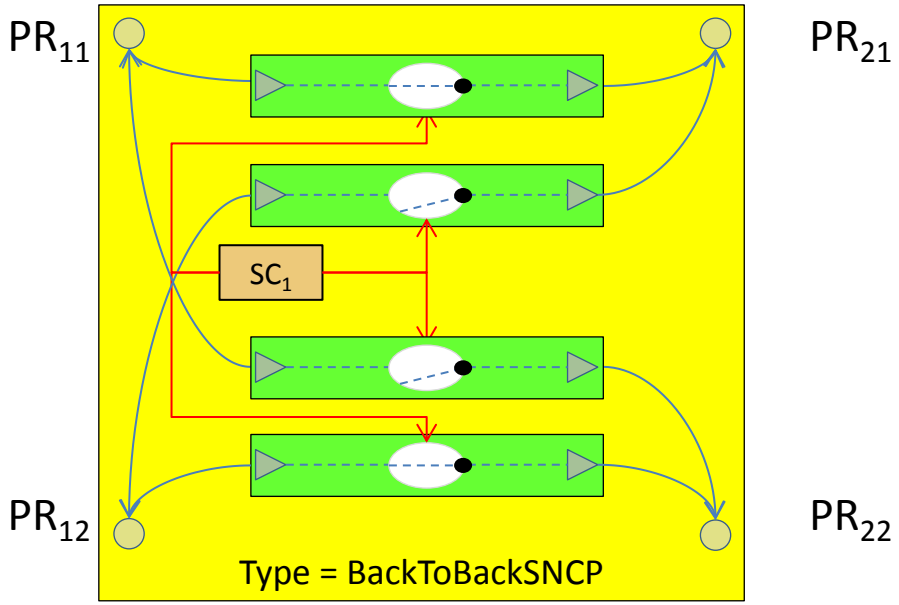
- The abstract view of an FC can be offered to a service oriented client
  - However there may be issues with validity of the view due to multiple failures

Expression of intended simple behavior



Kam: Is this the abstract view?

Effect of actual realization within distributed solution
(showing one direction only)

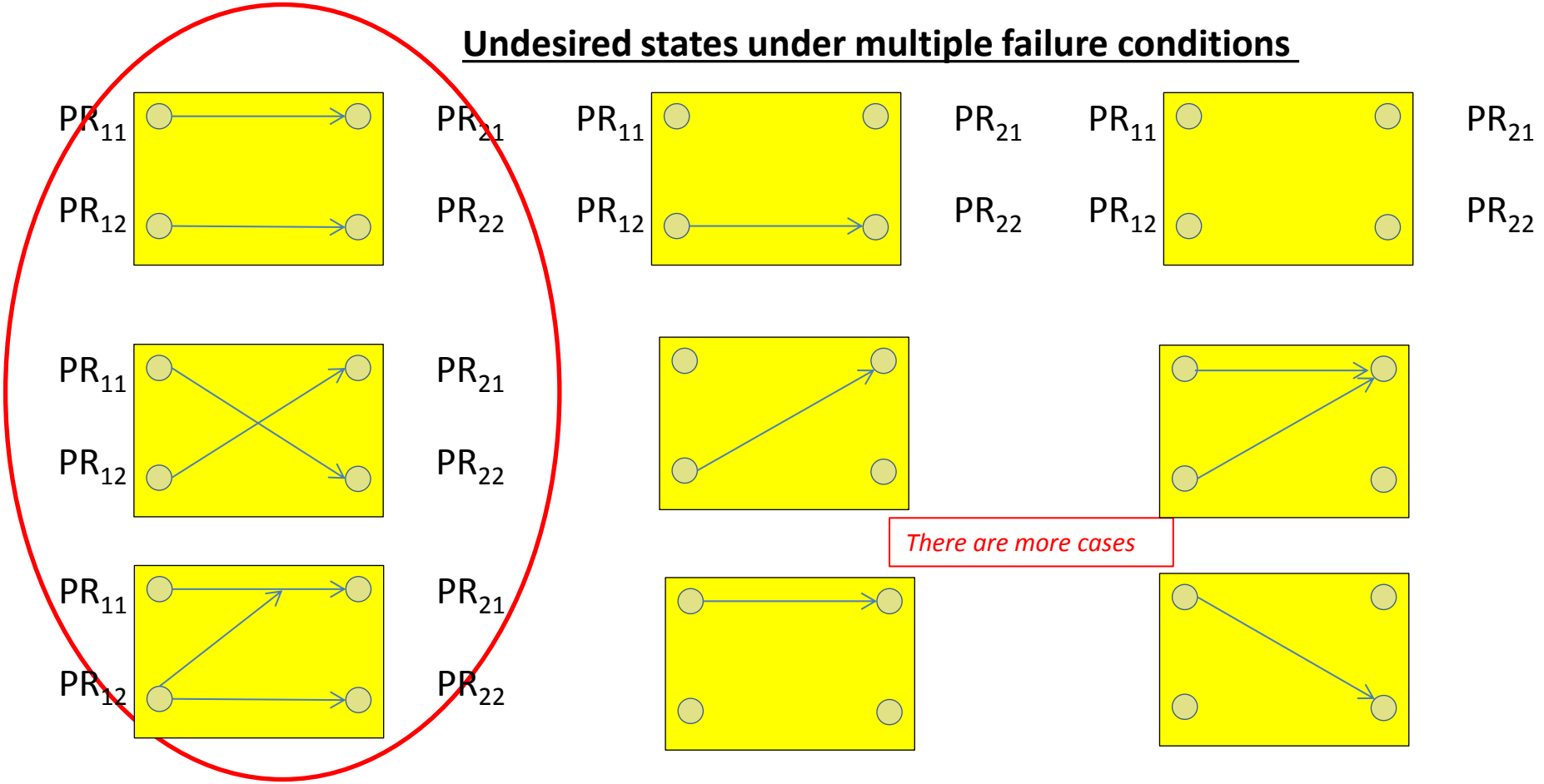


Kam: What do we call this view? Atomic view?

Potential states (one direction only)

Desired states

Undesired states under multiple failure conditions

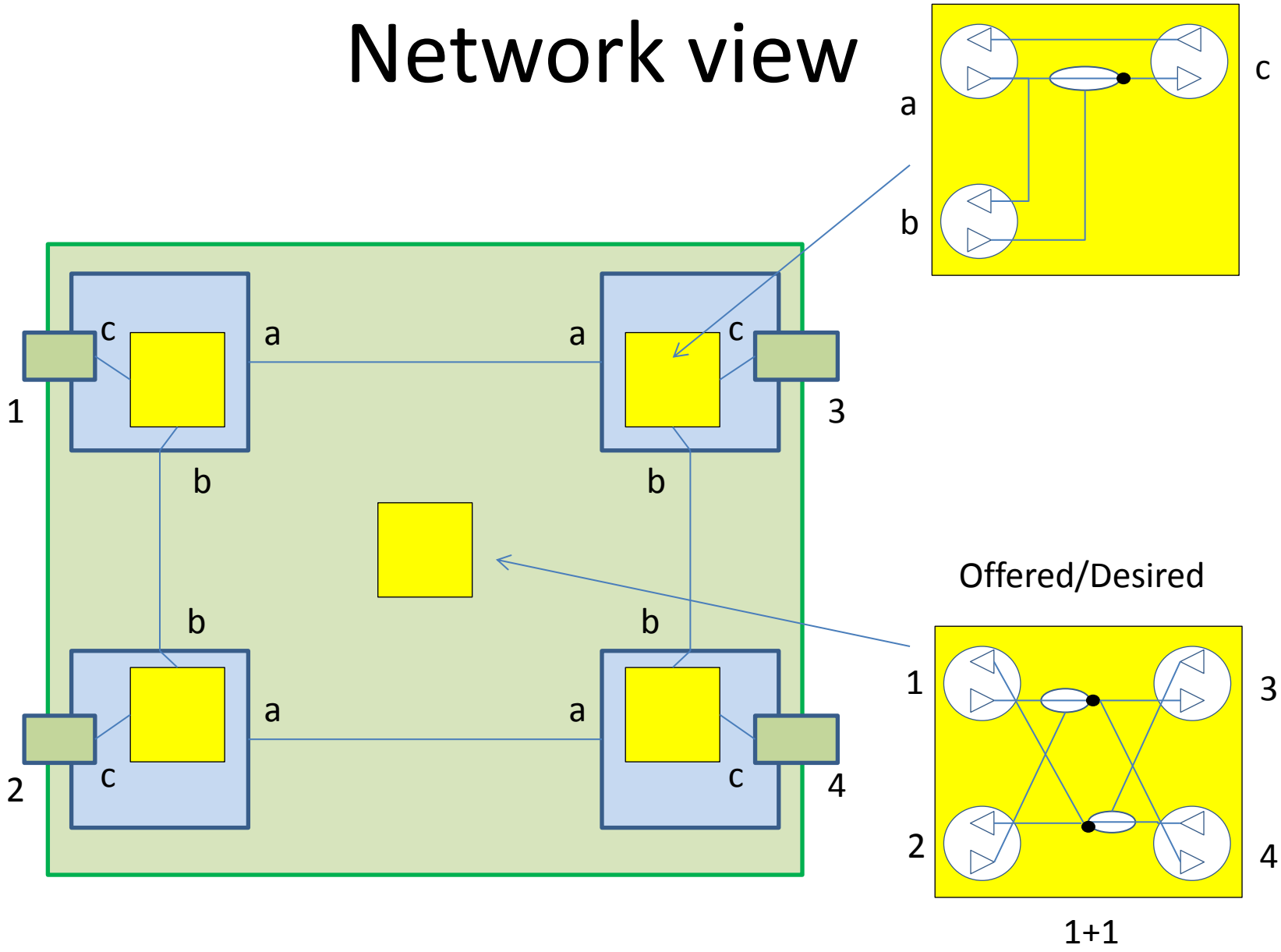


Potential flow query

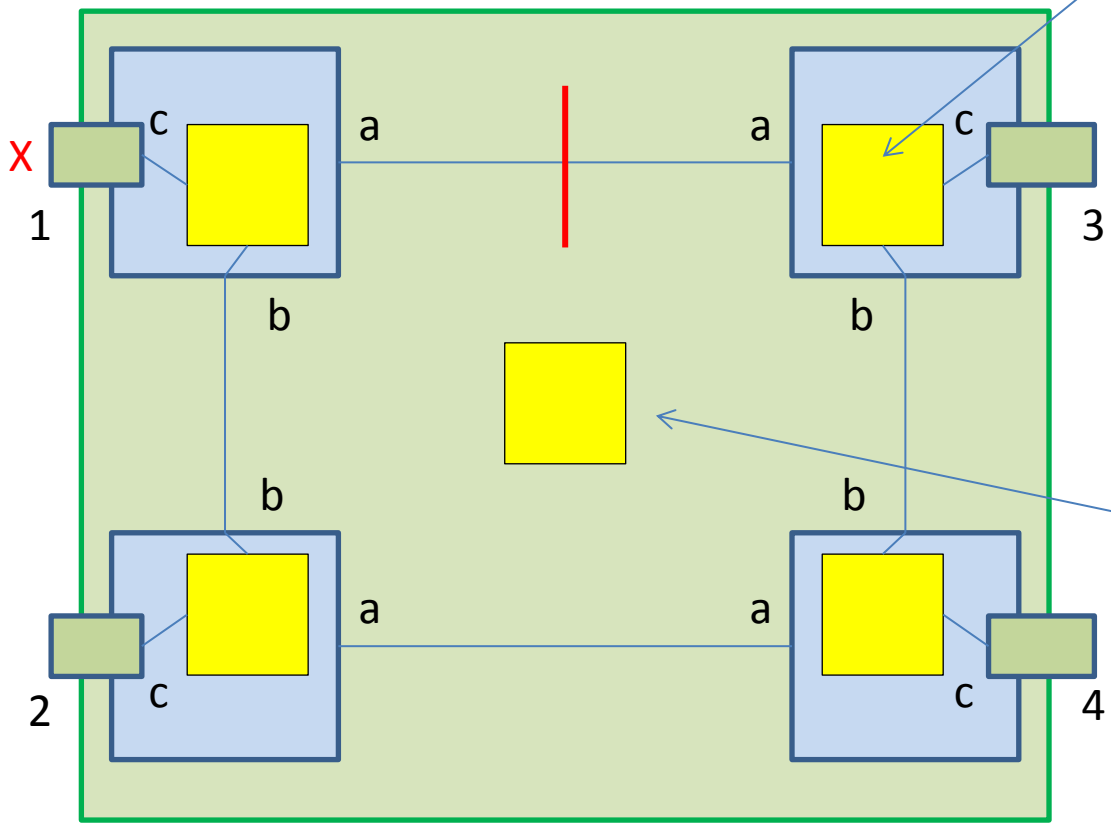
- To account for the issue highlighted on the previous slide a potential flow query should be provided (along perhaps with an alert of non-normal internal flow state)
  - The flow query would return multiple specs that described the behaviour of the current snapshot of disjoint structures
    - Suspect that raw SUFs would be sufficient (with no SCs)
- A normal internal potential flow state would be one that simply reflected the spec (that allows for simple absence of output)
- The same approach could be used to depict actual flow
- The coded form could be different between the actual flow and the spec

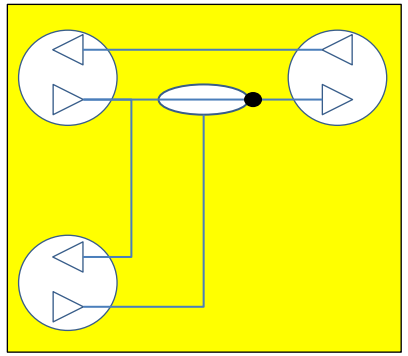
Kam: In DOC5, the ConnectionManager operational interface has, among other,
- *retrieveFCFlow ()*
- *retrieveAllFCFlowNames ()*

Network view

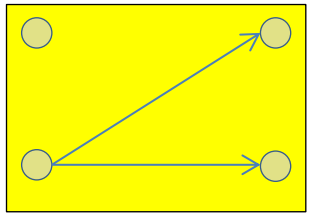


Network view (failures shown)

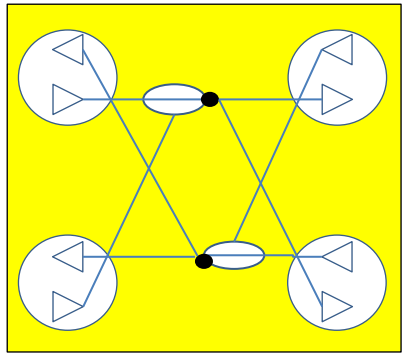




Kam: Should we show the following desired one instead?

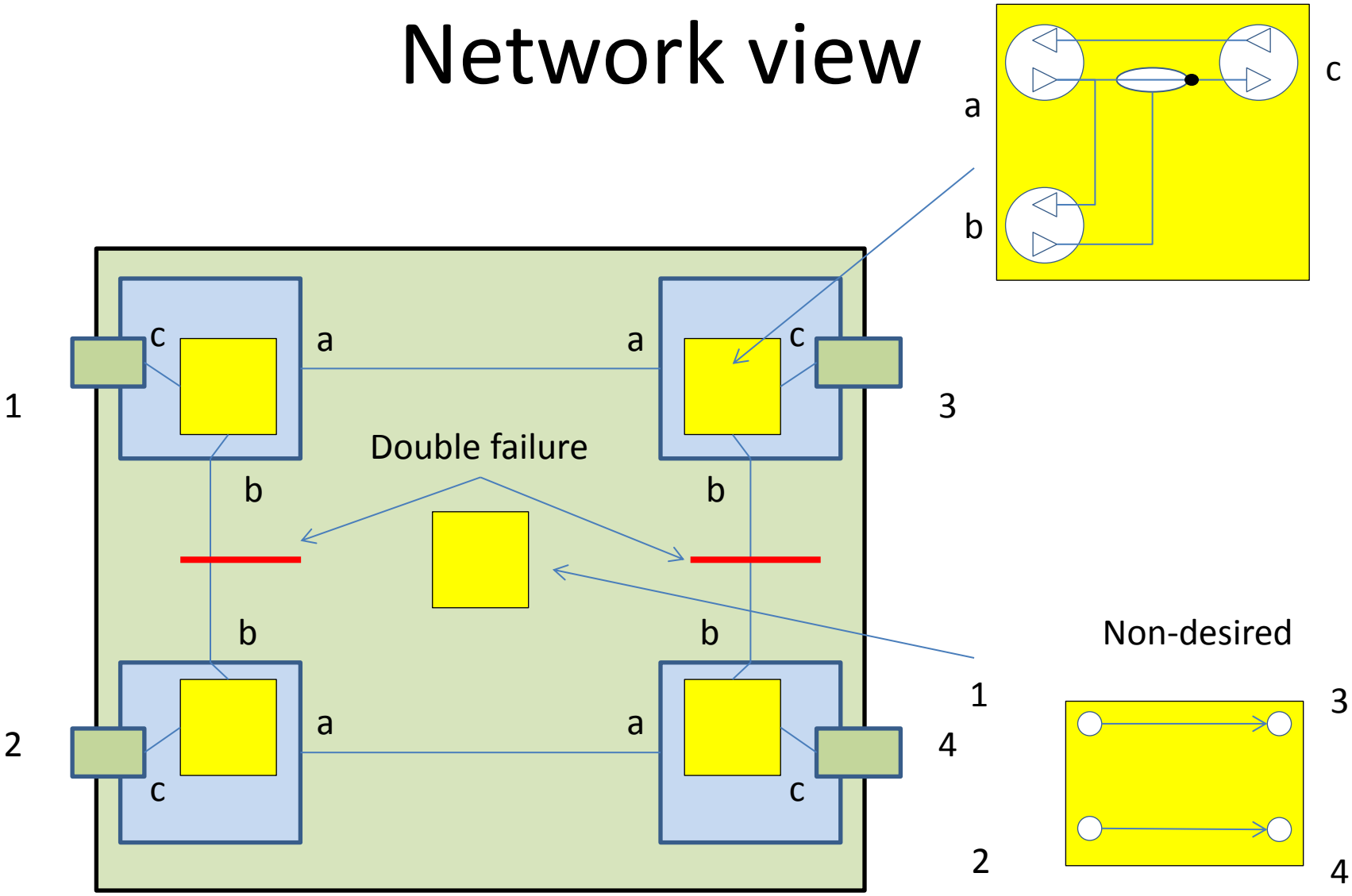


Desired

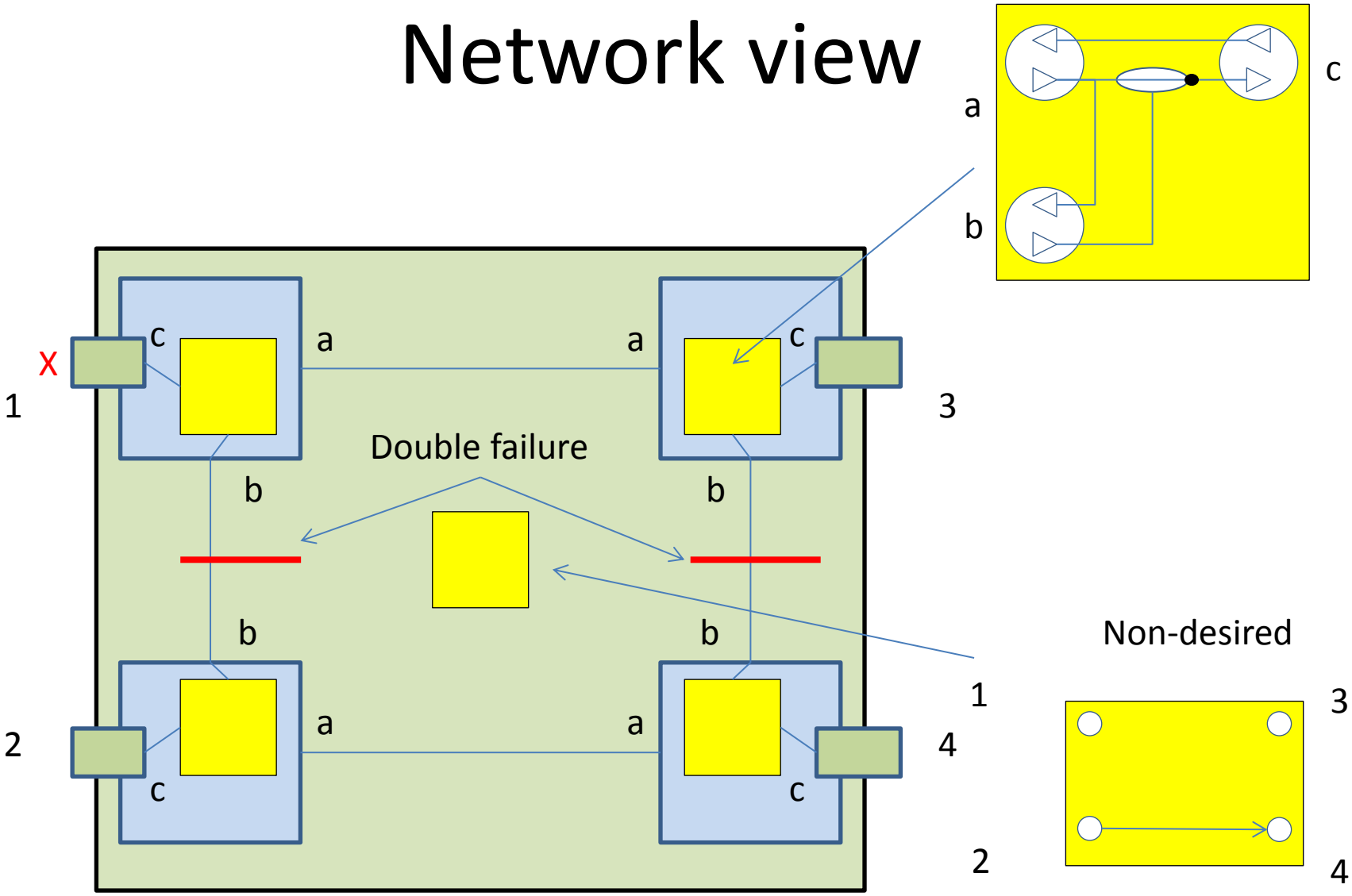


1+1

Network view



Network view

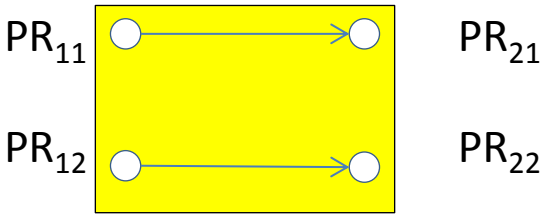


Failure case Potential shown

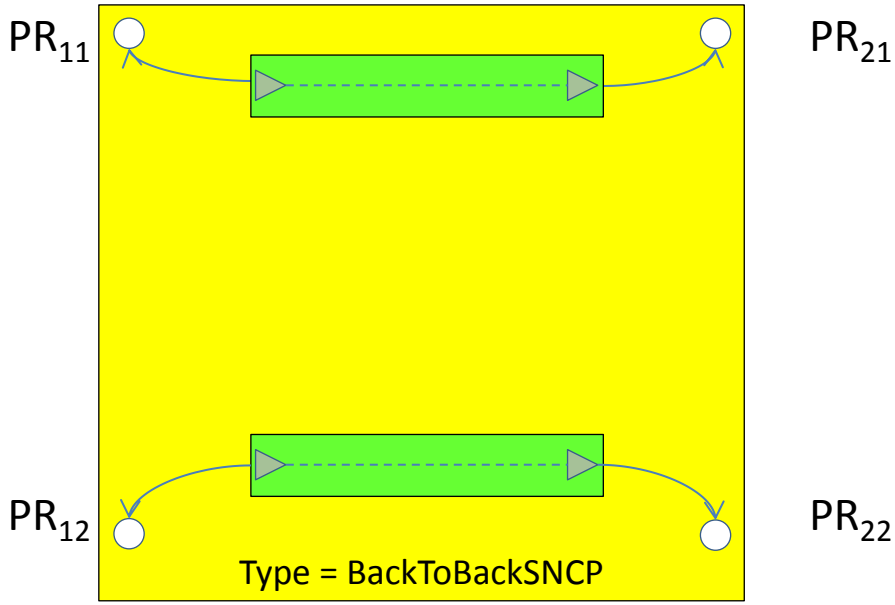
Kam: Sorry, what do you mean by "Coded form" actually? Improve terminology and explain more!!

Coded form is just a simple structure

- Ingress point goes to egress point list



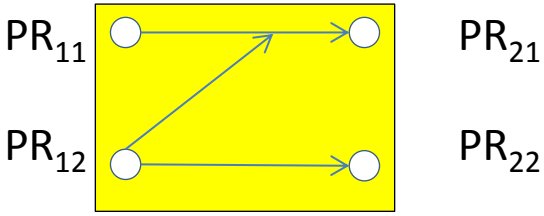
Effect of actual realization within distributed solution
(showing one direction only)

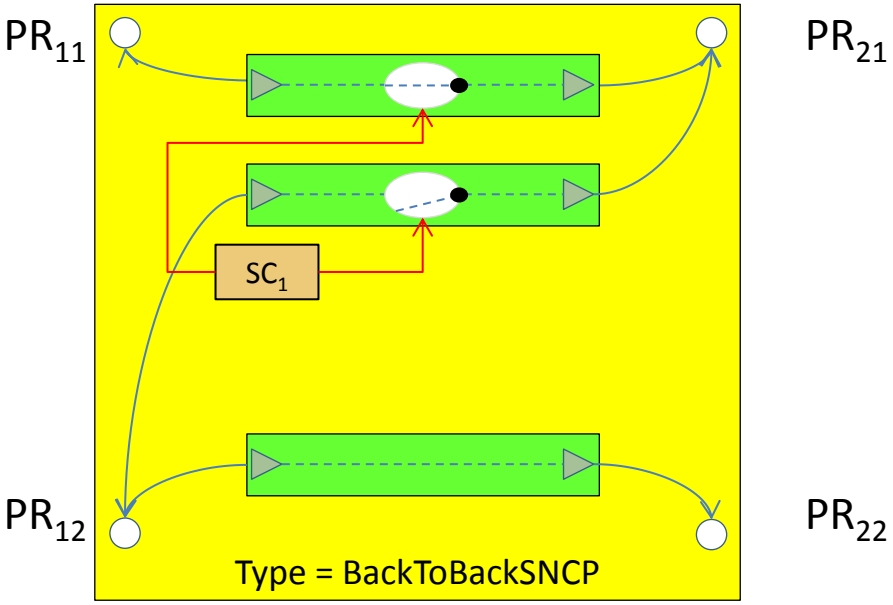


Failure case Potential shown

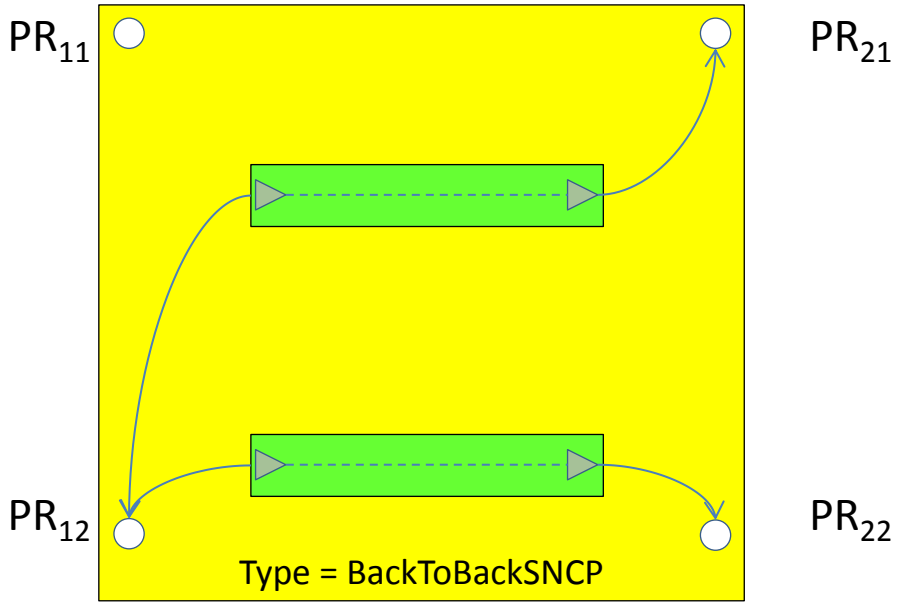
- SC provide diagnosed failure rules

- May also be due to engineering works

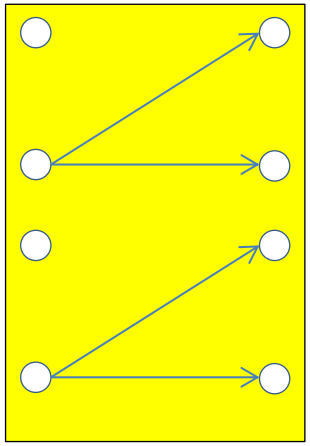


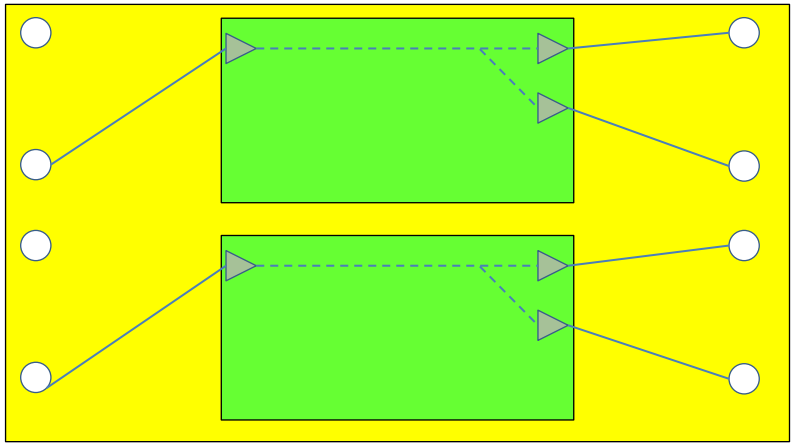


Effect of actual realization within distributed solution
(showing one direction only)



Flow macro in an instance (random example)

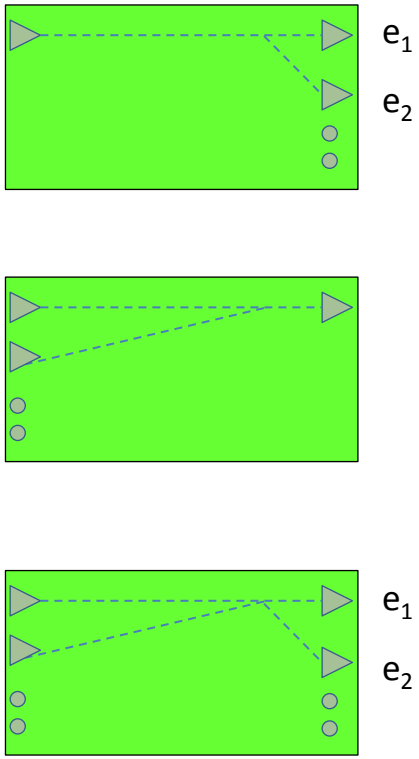




Coded form is just a simple structure

- Ingress point goes to egress point list

Flow macros and role rough notes

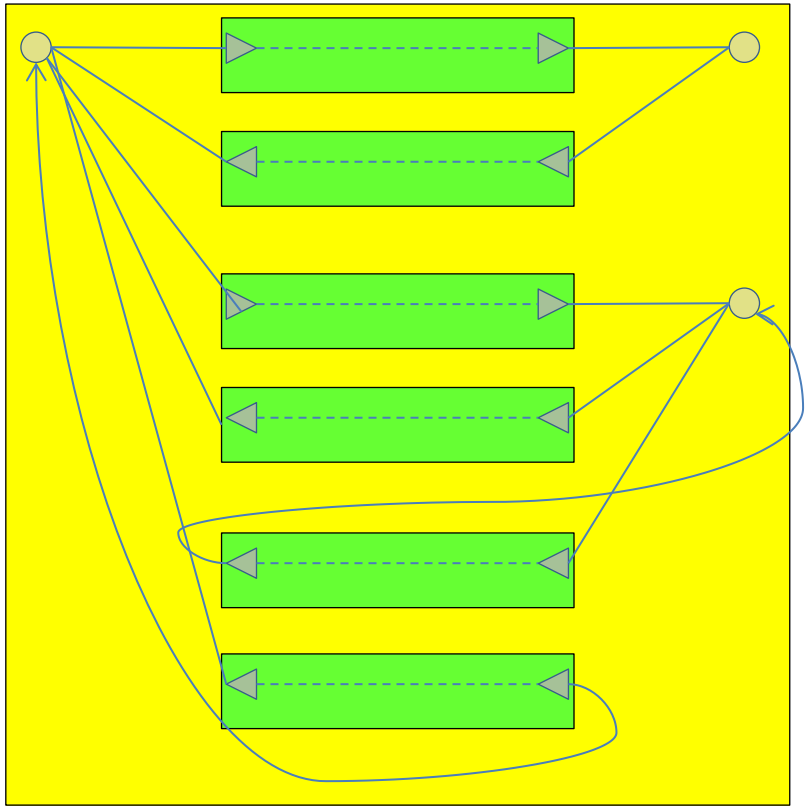


Roles

- Primary
- Secondary
- Alternate
- Resilient
- Common
- Active
- Standby
- Shared
- Balanced
- Distributed

Rooted multi-point (Hub/Spoke)

Root



Hub = Root
Spoke = Leaf

Add Leaf groups

More complex cases

- Multi-hub and spoke changes to dual hub and spoke

Need to cover

- Multi-layer ring case
  - May require understanding of the spec model for LTPs as layer transition handled by LayerProtocol stack
- Internal points…
  - Find a case that indicates we need them (e.g. separate control domains)

Key


NE


Actual LTP instance that is attached to physical port (TM Forum PTP)


Actual LTP instances carrying a unit of payload (TM Forum CTP)


Potential LTP instance


Potential LTP instance that currently cannot be made actual due to other configuration


Actual FC instances


Potential FC instance


Potential FC instance that currently cannot be made actual due to other configuration

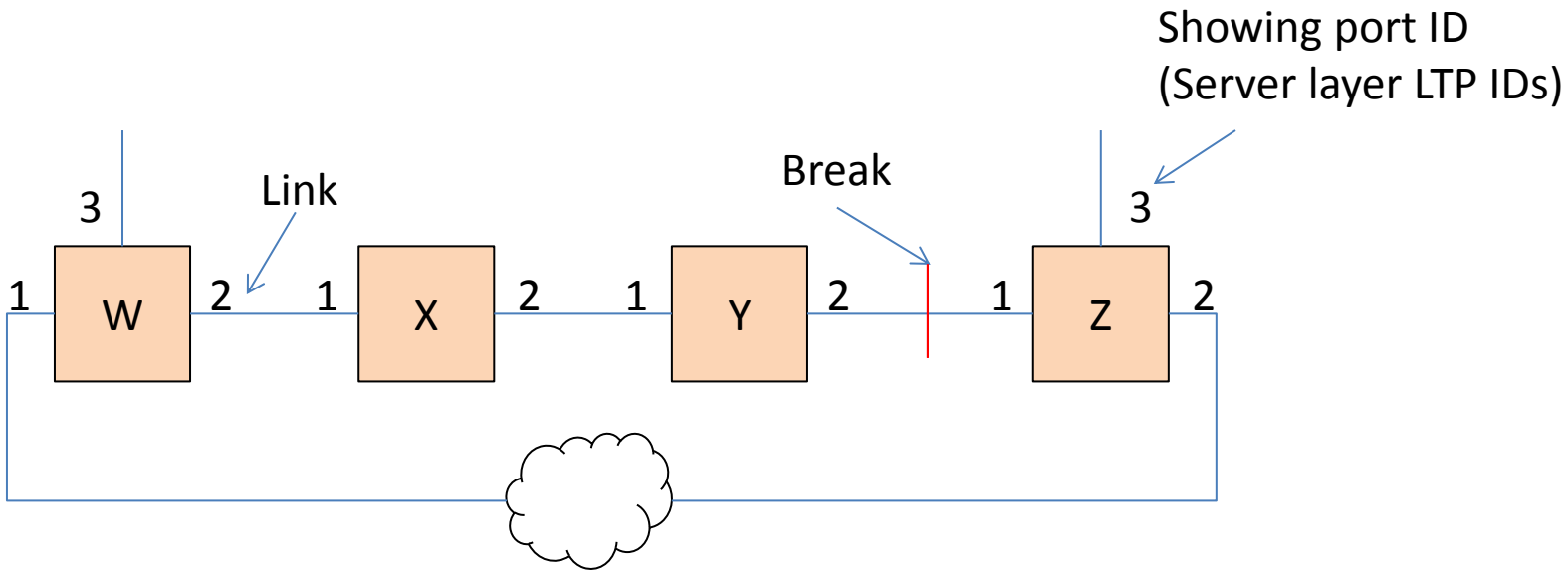
Physical Link

Failure in physical Link


Other NEs not drawn

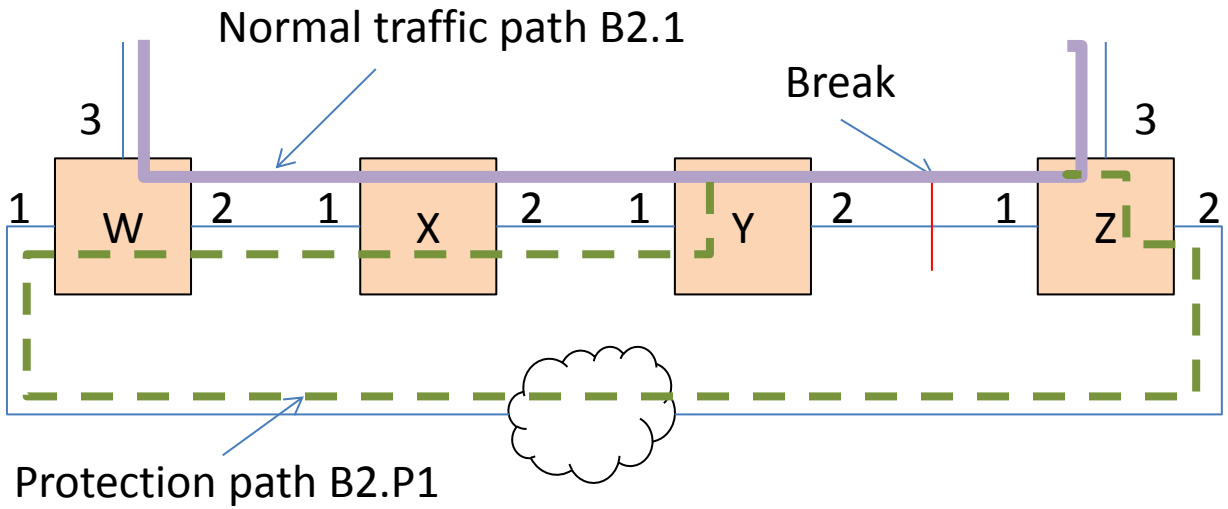
Other symbols as on earlier key

Network

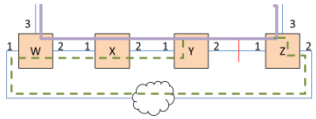


- The network technology is such that there are 8 channels of capacity on each link where 4 channels are available for traffic and 4 for protection.
- A single traffic signal could use just a single channel, could use two channels or could use all four channels
  - In the two channel case any available channels from the 4 can be used to make the capacity, i.e. the channels do not need to be adjacent
  - Different channels can be used on different links in the ring
  - Hence blocking is simply on capacity not channel
- The signals are numbered 1-4 for the single channel signal (B1) and 1-2 for the two channel signal (B2)

Network showing wrapping



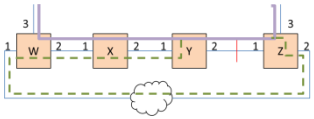
- A signal is passing from port 3 node W to port 3 node Z
- When a link Y-Z fails the traffic is routed back round the ring from the break on the corresponding protection capacity B2.P1
- Traffic can be monitored at intermediate points
- The following figures only show the 2 channel and 4 channel traffic (B2 and B4 respectively)
- To simplify the figures:
  - The same channel is maintained throughout the ring for both normal path and protection such that B2.1 must use B2.P1
  - No extra traffic is shown


3
3
1
2
1
2
1
2
1
2
W
X
Y
Z

Wrapping – “services” in the ring

- In general B1.n and B1.Pn is not shown. There is no B1 traffic in the ring
- Somewhere in the cloud there is a B2.2 service and a B4.1 service that require protection hence in all NEs shown there will be a B2.P2 and B4.P1 opportunity enabled.
  - If there was no B2.2 connection anywhere in the ring the B2.P2 would not be required.
  - If there was no B4.1 connection anywhere in the ring B4.P1 would not be required.

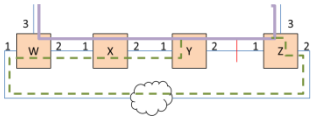
Wrapping: NE Y and NE X (no failure in ring)

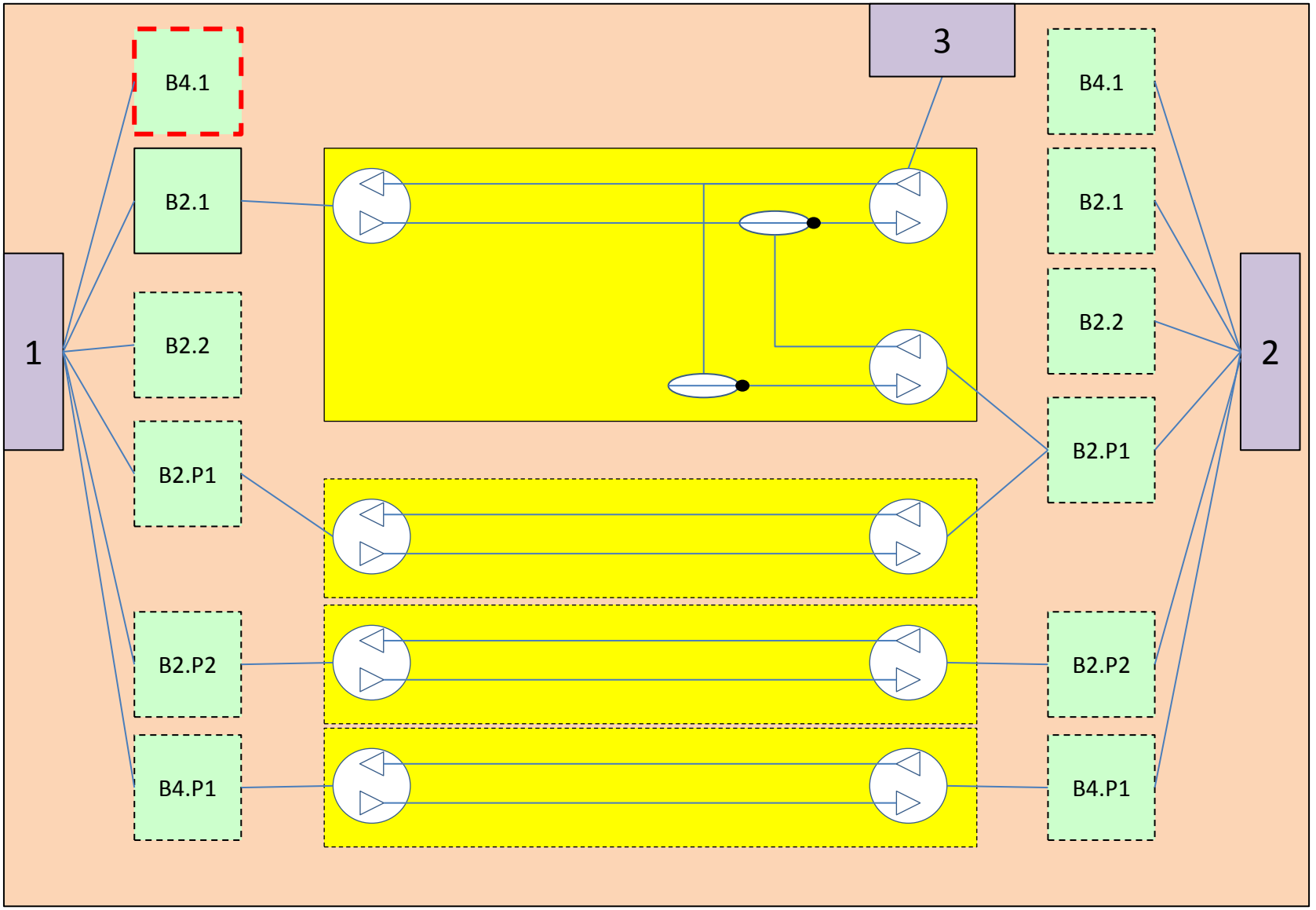


B4.1
B2.1
B2.2
B2.P1
B2.P2
B4.P1
1
B4.1
B2.1
B2.2
B2.P1
B2.P2
B4.P1
2

Wrapping: NE Z (no failure in ring)

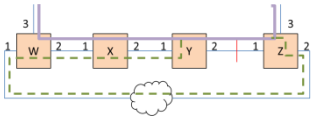
Note that the protection FCs do NOT protect the service shown

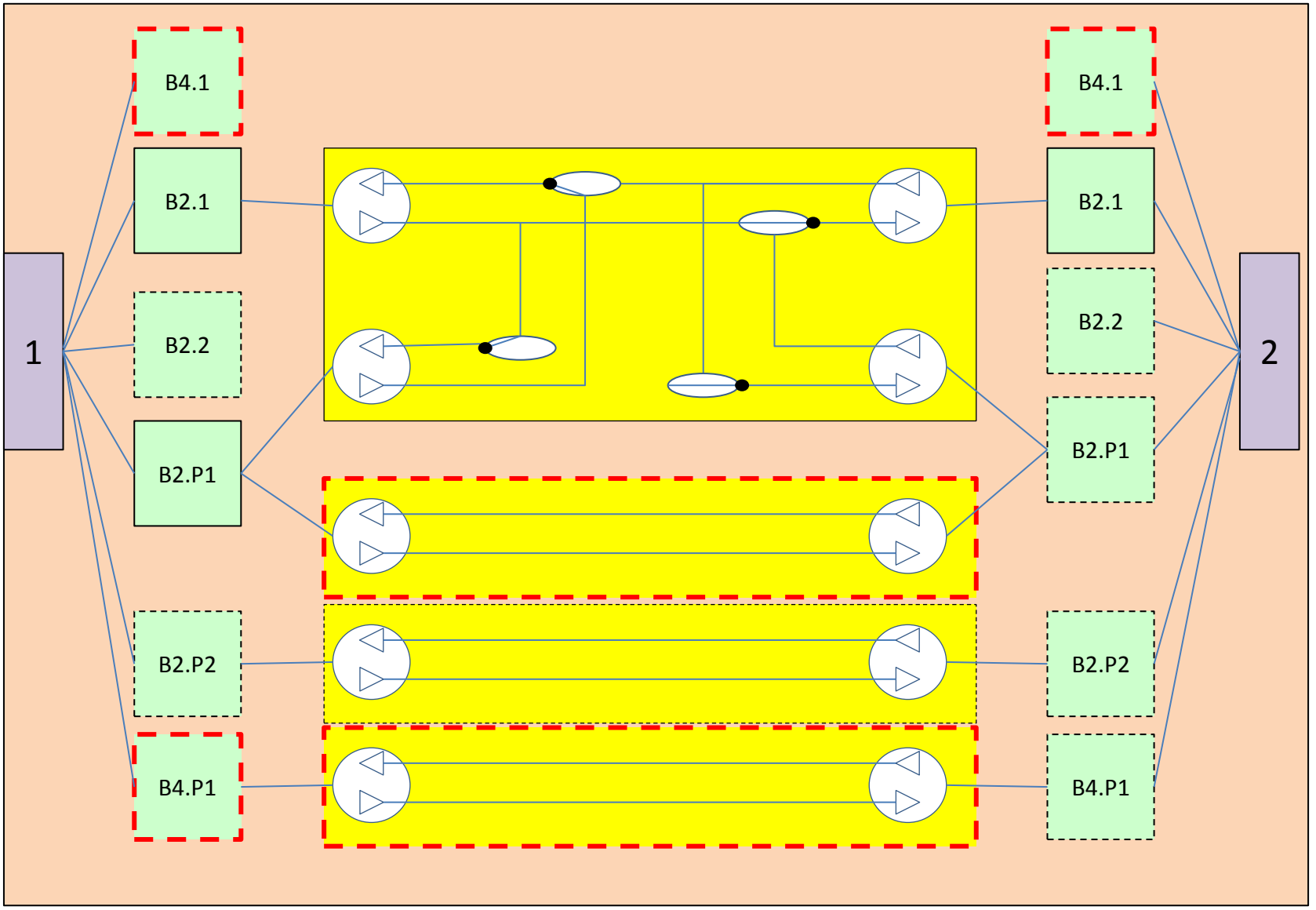




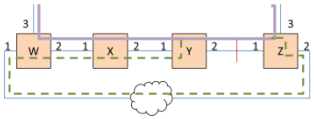
Wrapping: NE Y with failure on port 2

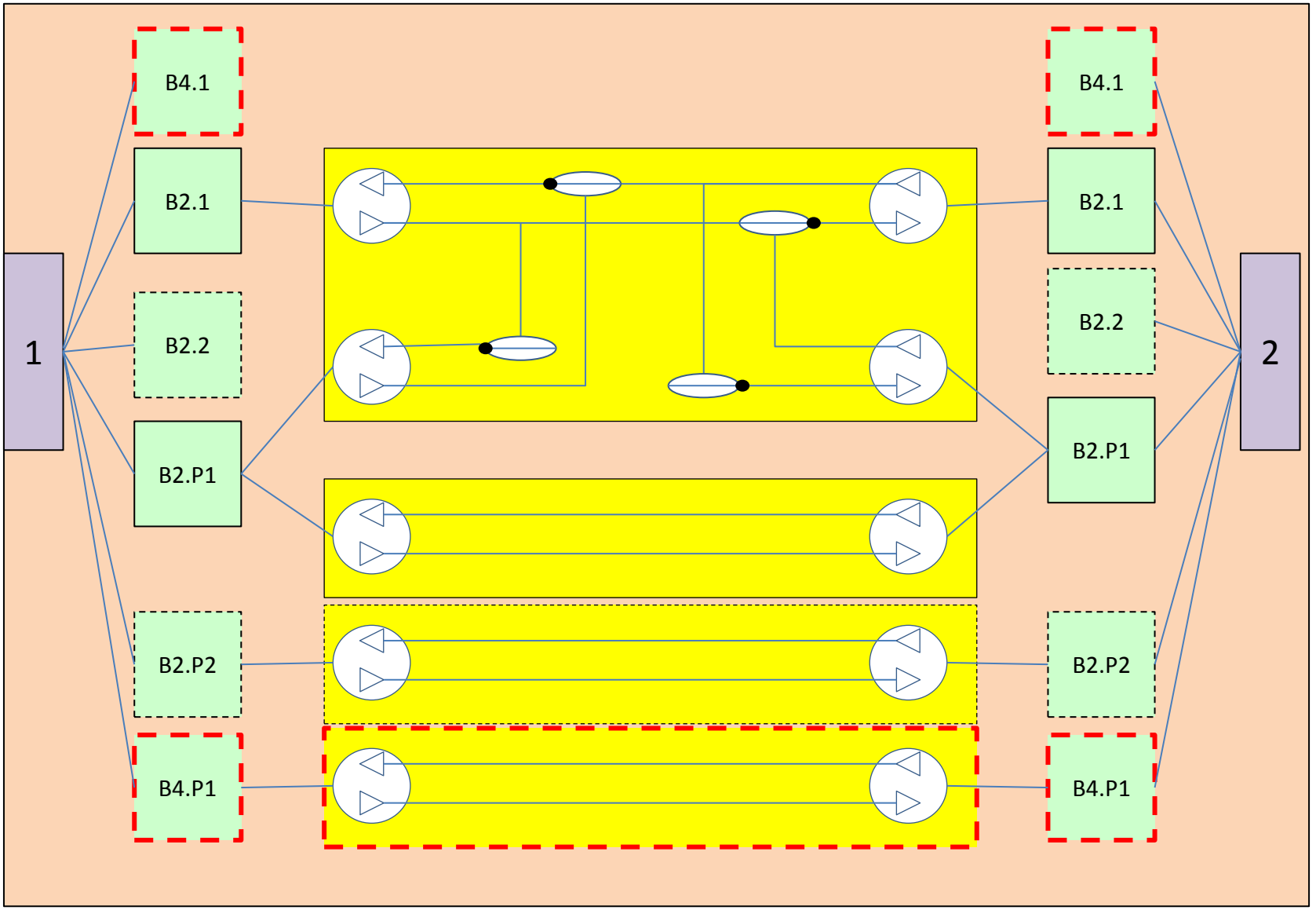
FC between port 1 and 2 on B2.p1 is no longer available due to the FC to B2.1



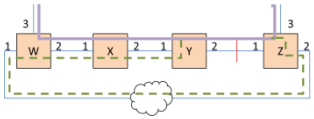


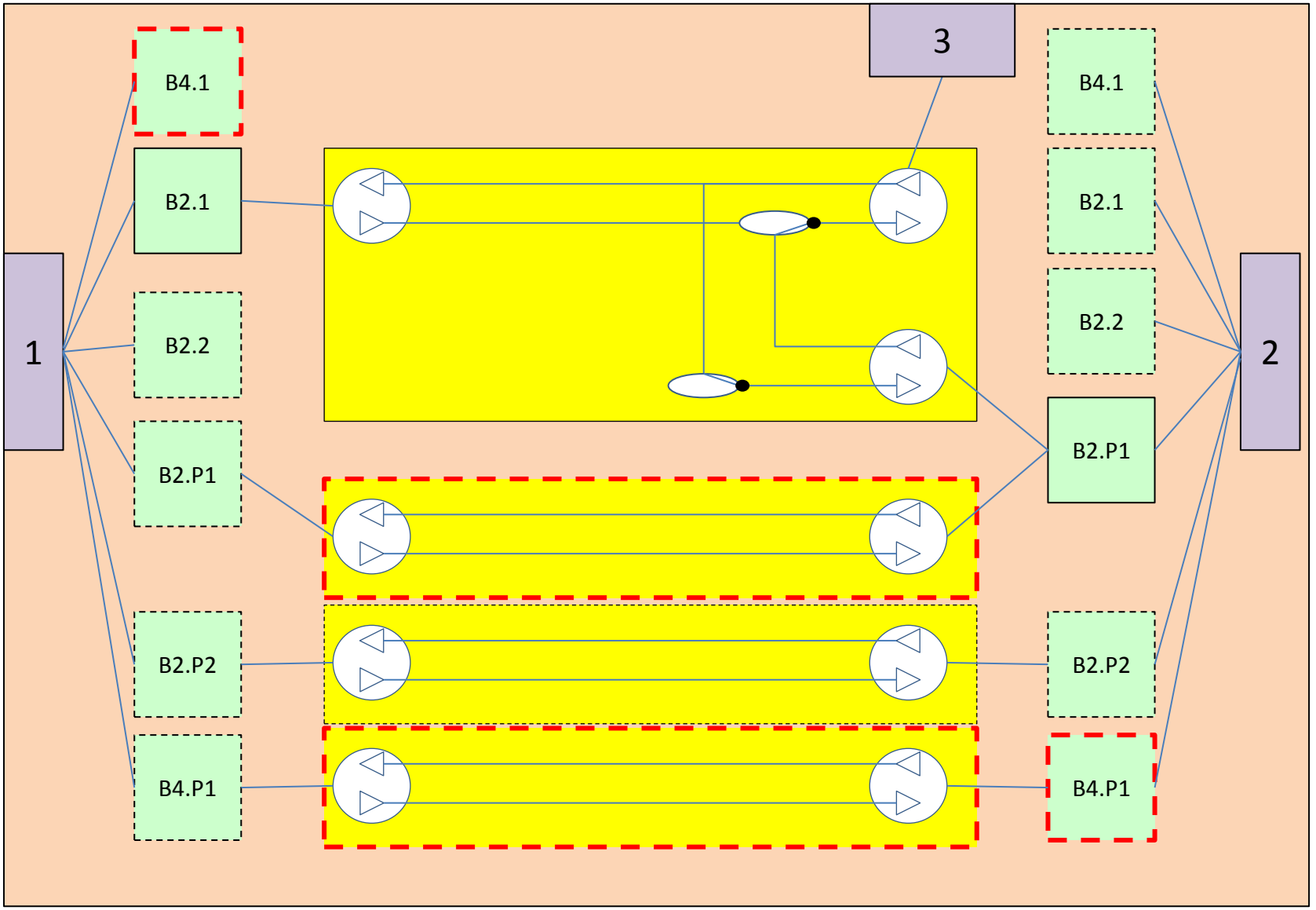
Wrapping: NE X with failure on NE Y port 2



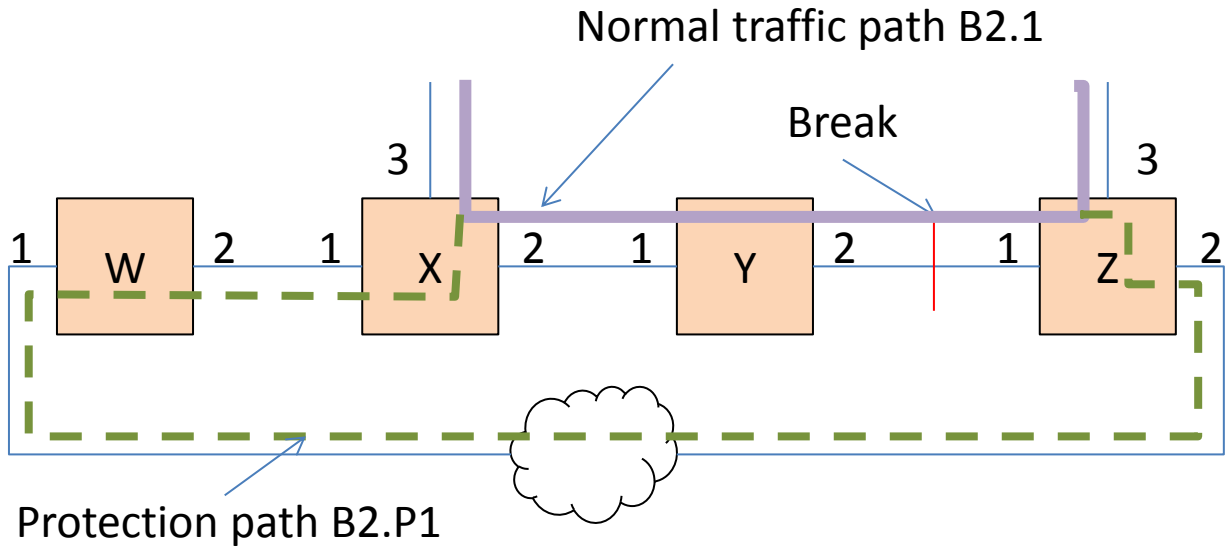


Wrapping: NE Z with failure on NE Y port 2



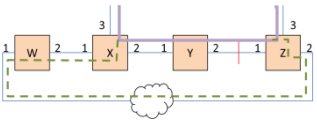


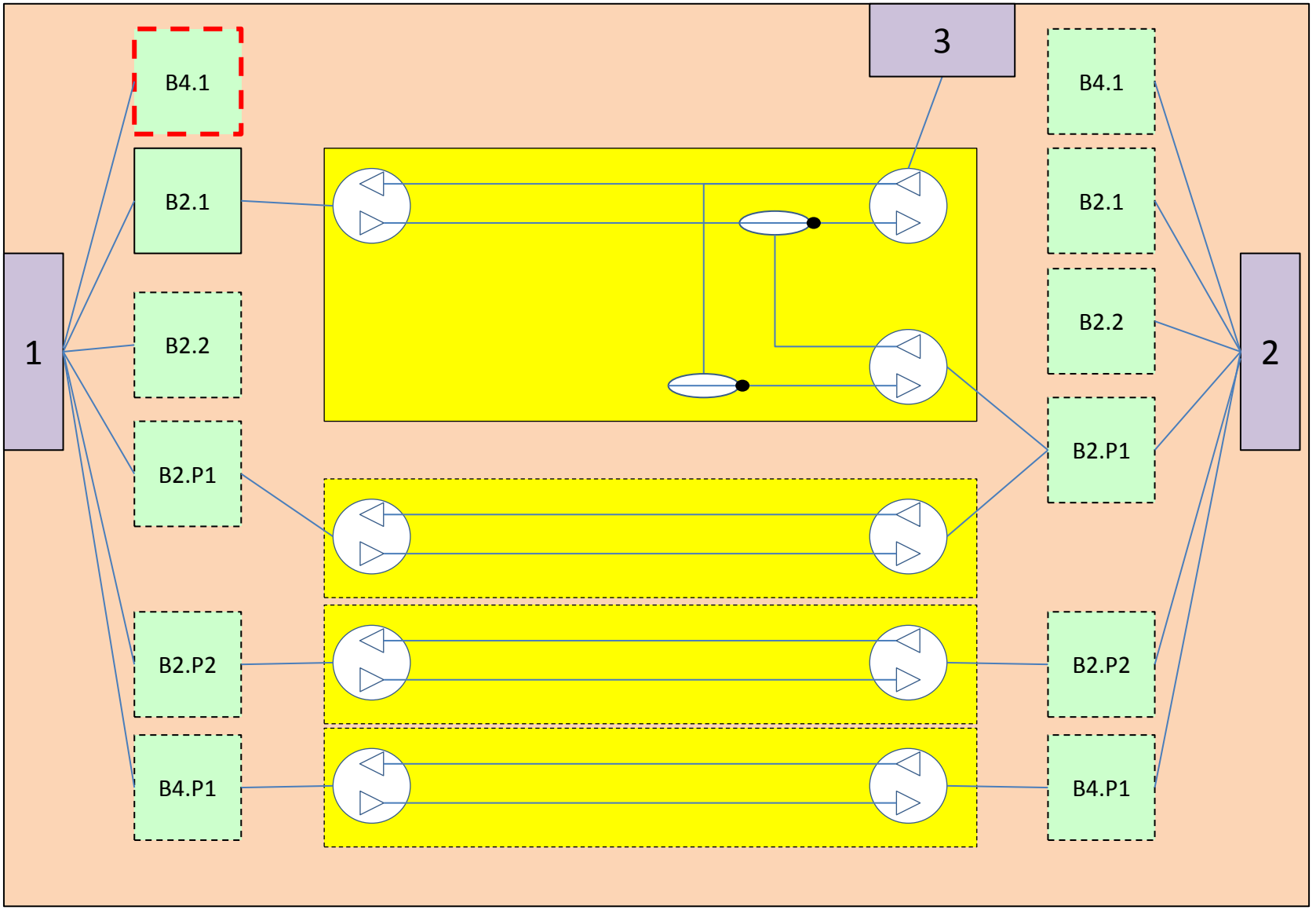
Network showing steering



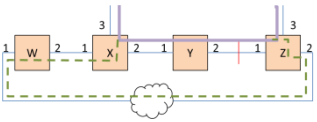
- A signal is passing from port 3 node X to port 3 node Z
- When a link Y-Z fails the traffic is routed back round the ring from origin on corresponding protection capacity B2.P1
- Traffic can be monitored at intermediate points
- The following figures only show the 2 channel and 4 channel traffic (B2 and B4 respectively)

Steering: NE Z (no failure in ring)





Steering: NE X (no failure in ring)



3

1

2

B4.1

B2.1

B2.2

B2.P1

B2.P2

B4.P1

B4.1

B2.1

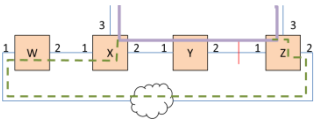
B2.2

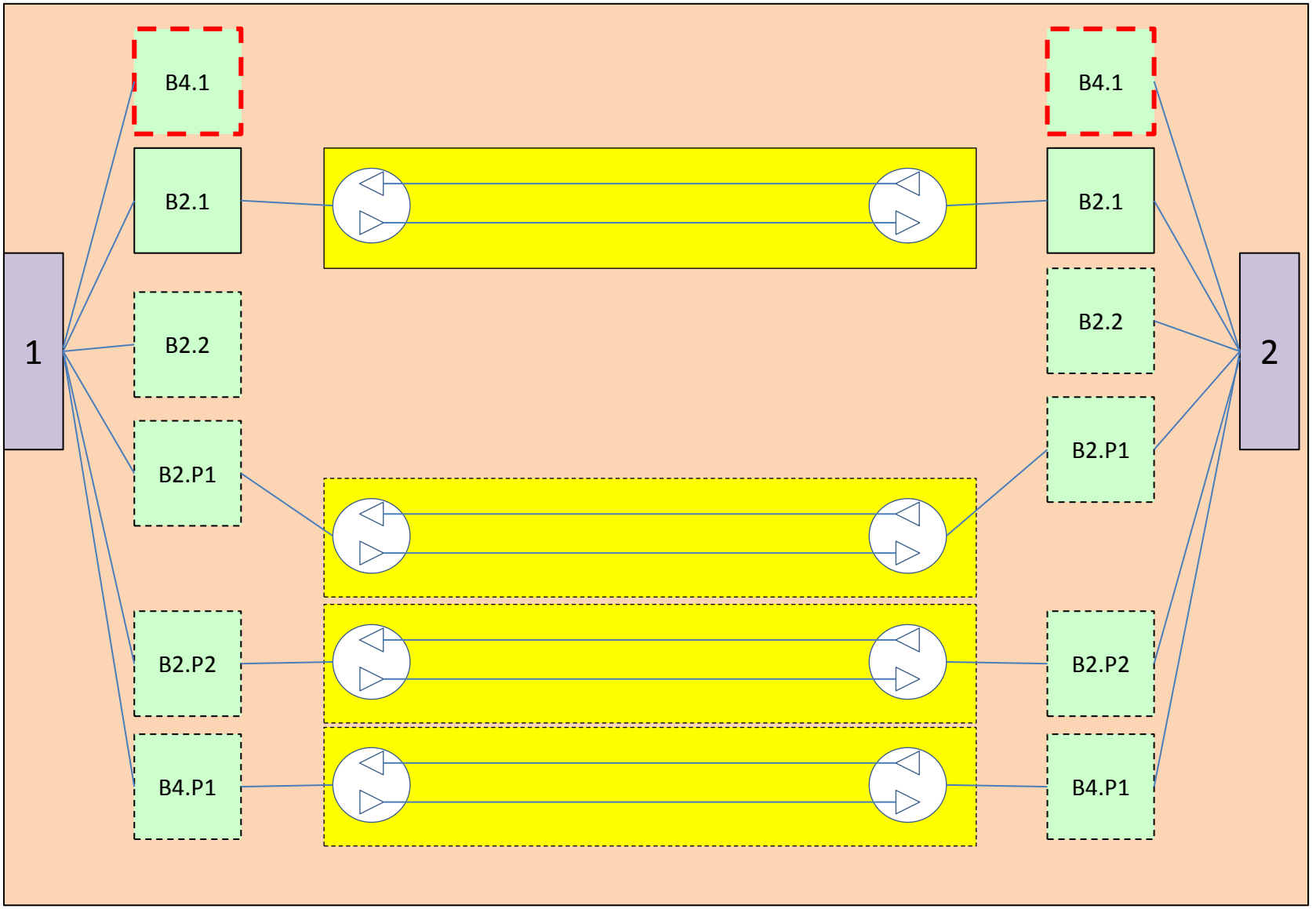
B2.P1

B2.P2

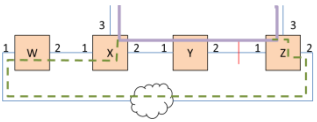
B4.P1

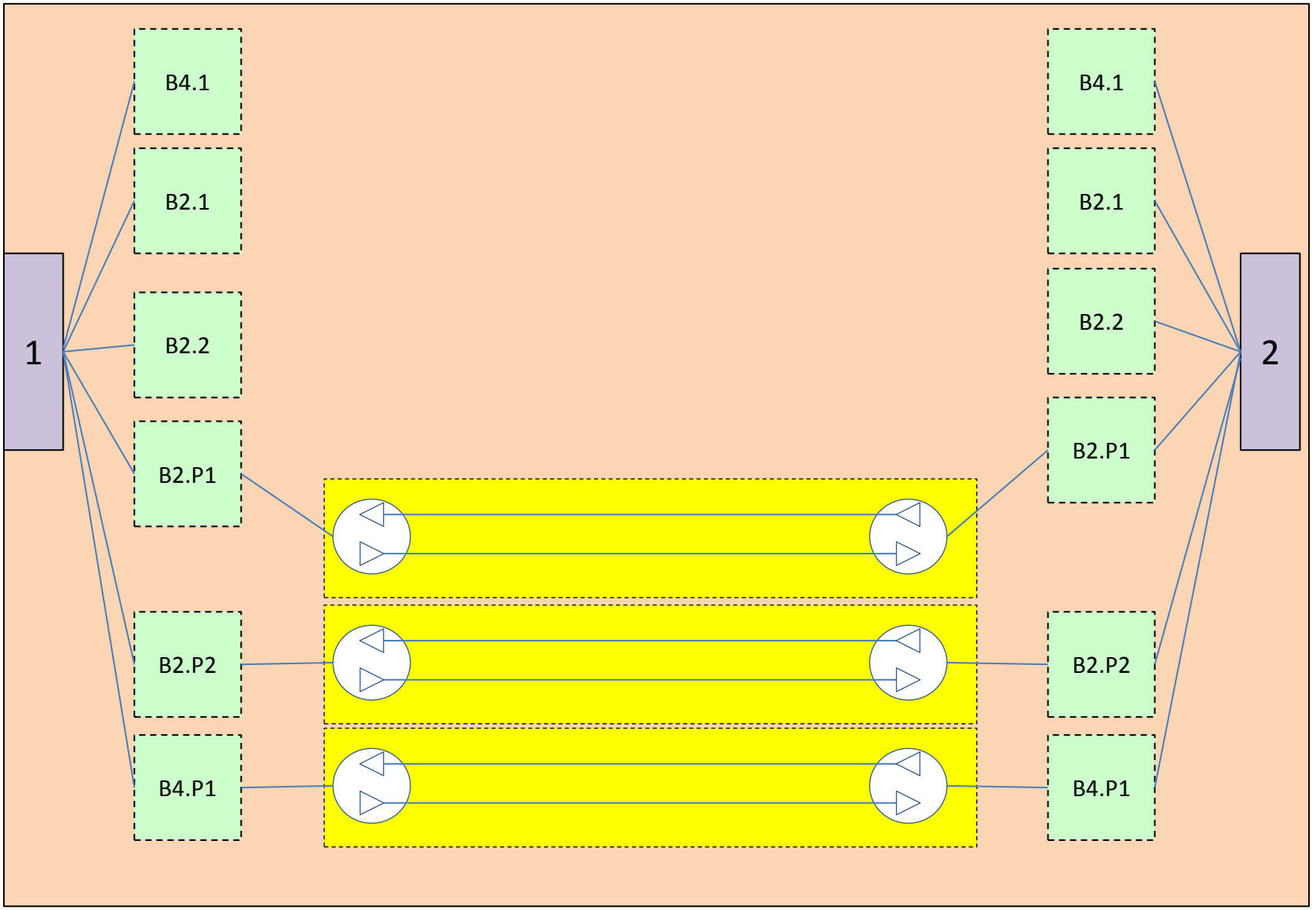
Steering: NE Y (no failure in ring)



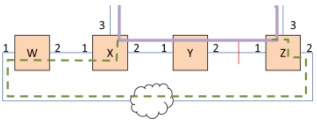


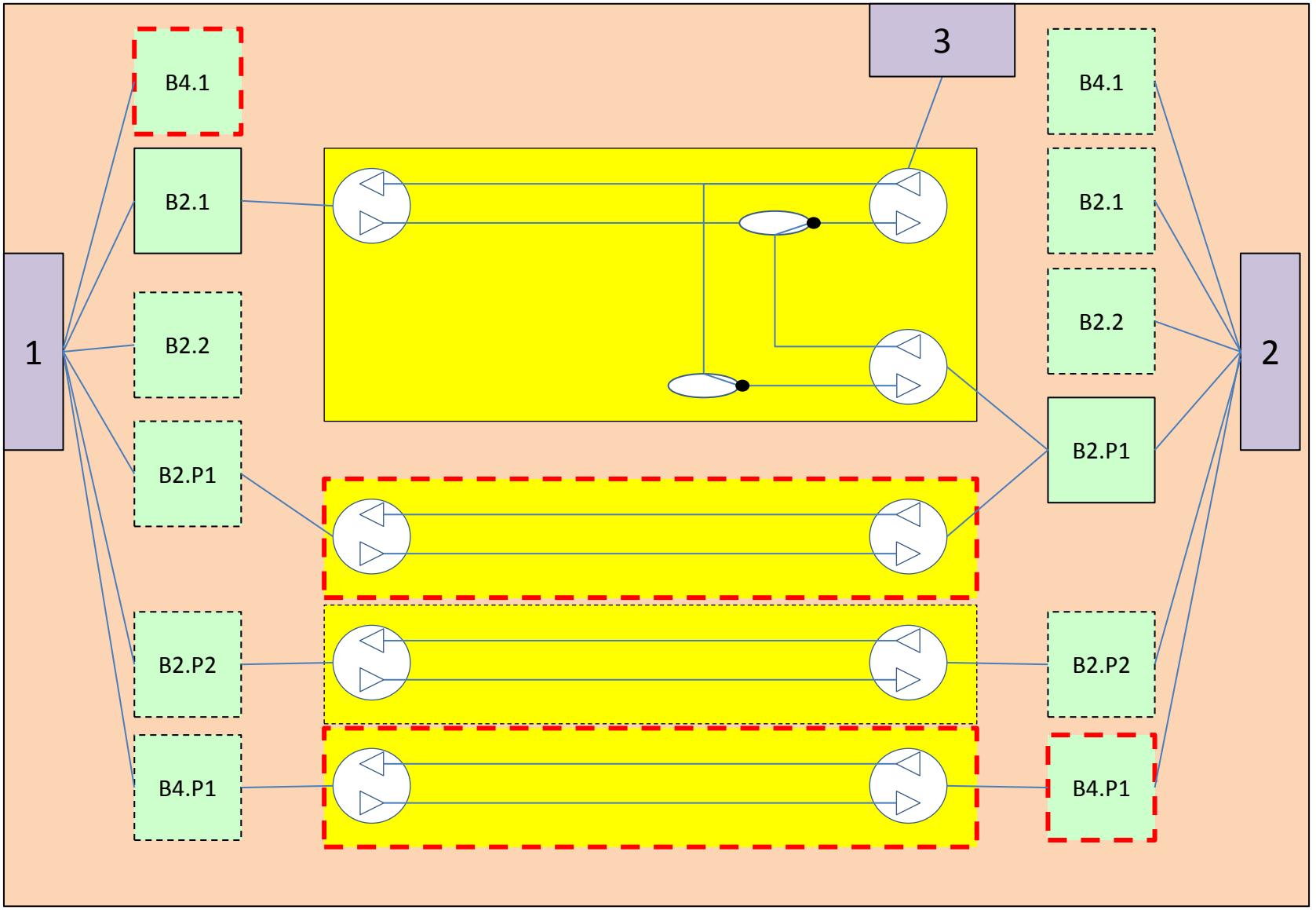
Steering: NE W (no failure in ring)



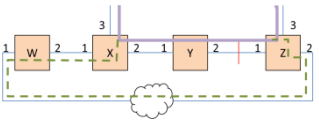


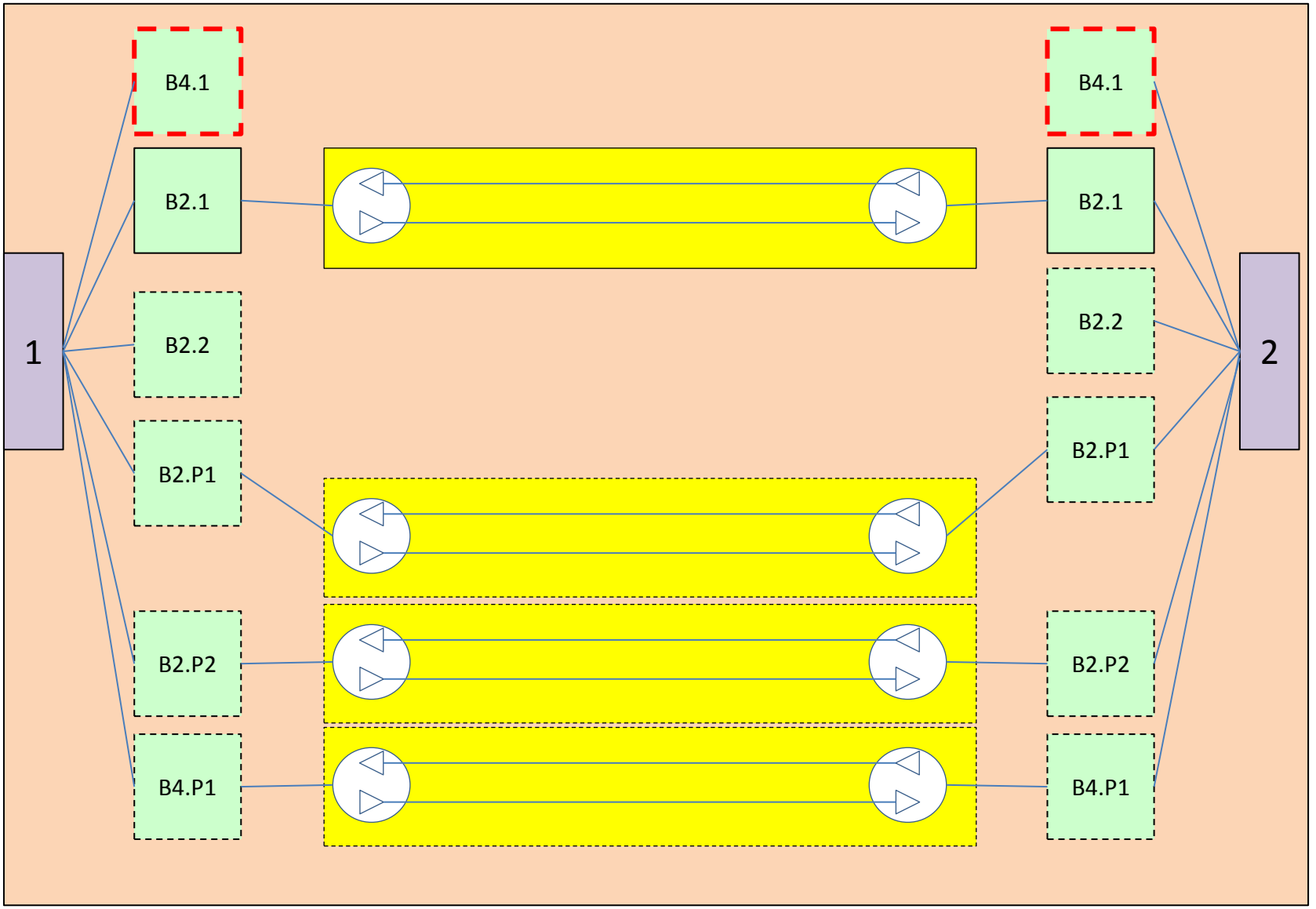
Steering: NE Z with failure on NE Y port 2



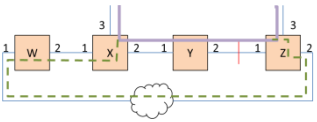


Steering: NE Y with failure on port 2 (same as no failure)





Steering: NE X with failure on NE Y port 2



3

1

2

B4.1

B2.1

B2.2

B2.P1

B2.P2

B4.P1

B4.1

B2.1

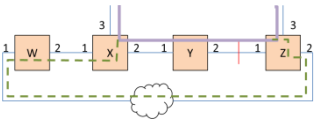
B2.2

B2.P1

B2.P2

B4.P1

Steering: NE W with failure on NE Y port 2



B4.1
B2.1
B2.2
B2.P1
B2.P2
B4.P1
1
2
B4.1
B2.1
B2.2
B2.P1
B2.P2
B4.P1

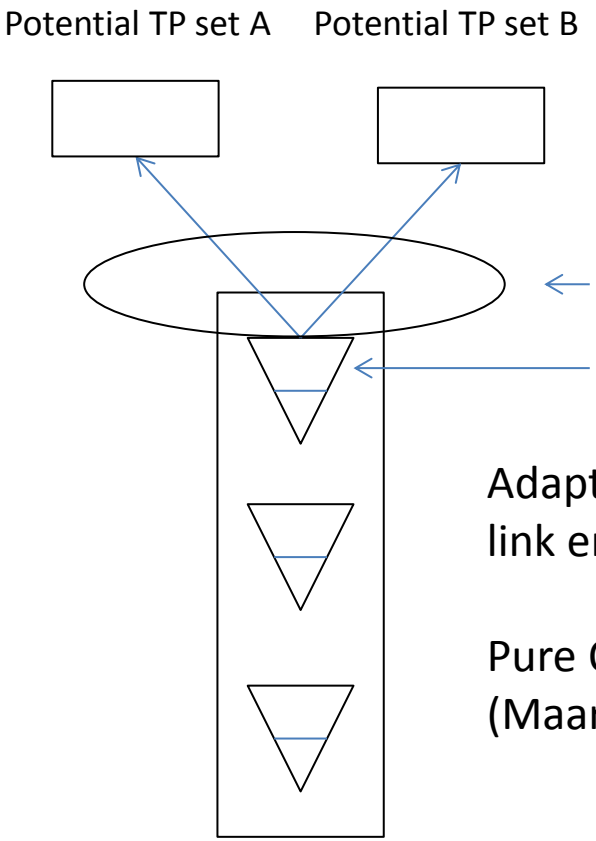
Various approaches and considerations

- Three options… FCs are
  1. "created" as potential and then activated when protection requires (like the CTPs in TMF)
  2. not present until protection requires but are known to be potential through a specification
  3. created as actual (rather than potential) with a switch disabling them and are switched on when protection requires
- The CTPs approach could be the same as the FC approach but there are some hybrids possible
  1. The CTPs could be not present even if the FC is until selected by a switch
- Interpreting the variety above aiming for a solution
  - We need a spec to explain what can exist and what needs to be created when to form the correct behaviour
  - The spec can remove the need to report/notify entities
    - A composite notification could be designed to inform of a complex configuration change if defined in a spec
  - Potential CTPs and FCs can be considered as "partially created" in that a query on the live system could return them as instances and when they become active this could be considered as a state change rather than a creation (as the rule is indeed known by the NE)
  - A hybrid (with a switch) of potential+off and actual+on could be considered
    - When a CTP is disabled it is potential and when enabled it is actual
    - When CTPs and FCs are gathered /notifiedonly enabled FCs and CTPs would be reported
  - States may be
    - Actual
    - Potential
    - Potential – disallowed
  - Creation v state change when spec is provided…
    - Seems that much of the CTP/FC would be known from the spec so a simple state change is all that is required
  - Spec could identify group notifications that indicate a change of state of many classes or a batch notification could be provided
  - Challenges: Potential misalignment between spec and reality
  - Note that the SC is really a configuration controller
  - The actual state MUST be available, the question is how much potential should be reported and how much should be in the spec. The feeling at this point is that the potentials should NOT be reported other than via the spec.

Specs for Network Constructs

- Network Constructs include
  - NE, Protected ring, ForwardingDomain and associated TP
    - Is a protected ring as ForwardingDomain (think not as the TPs have to be included not just referenced)
  - Specs identify restrictions and capabilities of ForwardingDomains etc in the network construct
- So an NE may allow various types of FC in its ForwardingDomain in various arrangements
  - But once the ForwardingDomain is part of a ring the set of allowed FCs is changed (I was thinking reduced but perhaps some FC types only make sense in the ring)

LTP spec considerations



Provides naming rules etc

Pooling

Adaptation rule set belongs to server side

Adaptation rules explain the pool interaction for client layer link ends

Pure OTN example and Hybrid packet example (Maarten/Nigel)

Short form sketch

LTP Spec

- Includes:
  - Layer structure for entire LTP stacks
  - Client pool s
    - Mapping rules for clients and client mapping interactions
    - Naming rules for clients
    - Note that this is the essence of an LTPP so an LTPP is subsumed into an LTP Spec
  - LTP and layer protocol Attribute options and ranges
- Is referenced from LTP and Layer Protocol

Generalizing the spec model

- Should we construct a generalized model (perhaps derived from the TM Forum work) that provides a specification capability for each component and assembly of components (System) we have
  - Entities such as LTP and FC are components (some have exposed ports and other just use specialized relationships in place of ports)
- The generalized spec form could be used directly where even the identifier of the class to which the spec applies would be data or could be specialized such that each major class has its own spec class
  - The former clearly provides most flexibility but is both more complex to formulate and opaque to use
  - Regardless of which approach is used the essential of the specification mechanism will need to be highly generalized and versitile to allow for new cases to be covered without the need to change the model

Actions

- List outstanding issues
- Develop a generalized specification model
- Document delegation of mac learning

Generalized spec model

- The component and its abstracted detail
  - The basic FC switch spec discussion has been on this
  - The spec could be generalized to cover any “component”
    - Complexities where components do not have modelled endpoints but instead use associations to imply endpoints
    - Many different encapsulated functions need to be considered not just switching
    - Encapsulated function assembly can be almost as complex as the system it abstracts
- The system assemblies
  - The generalized expression of rules/constrains for assembly of an arbitrary set of components whilst possible will clearly need significant sophistication
  - The specific need in ONF at this point is quite constrained
- Component and system models in ONF
  - It is proposed that initially per case spec models are developed for early deployments but that in parallel work is carried out to explore potential generalizations
  - It is suggested that the initial spec models are highlighted as preliminary and likely to change
  - It is recommended that the ONF work with other organisation such as TMF and NFV where similar problems also need to be tackled and some work has already been done

Interaction between specs for FC, LTP etc

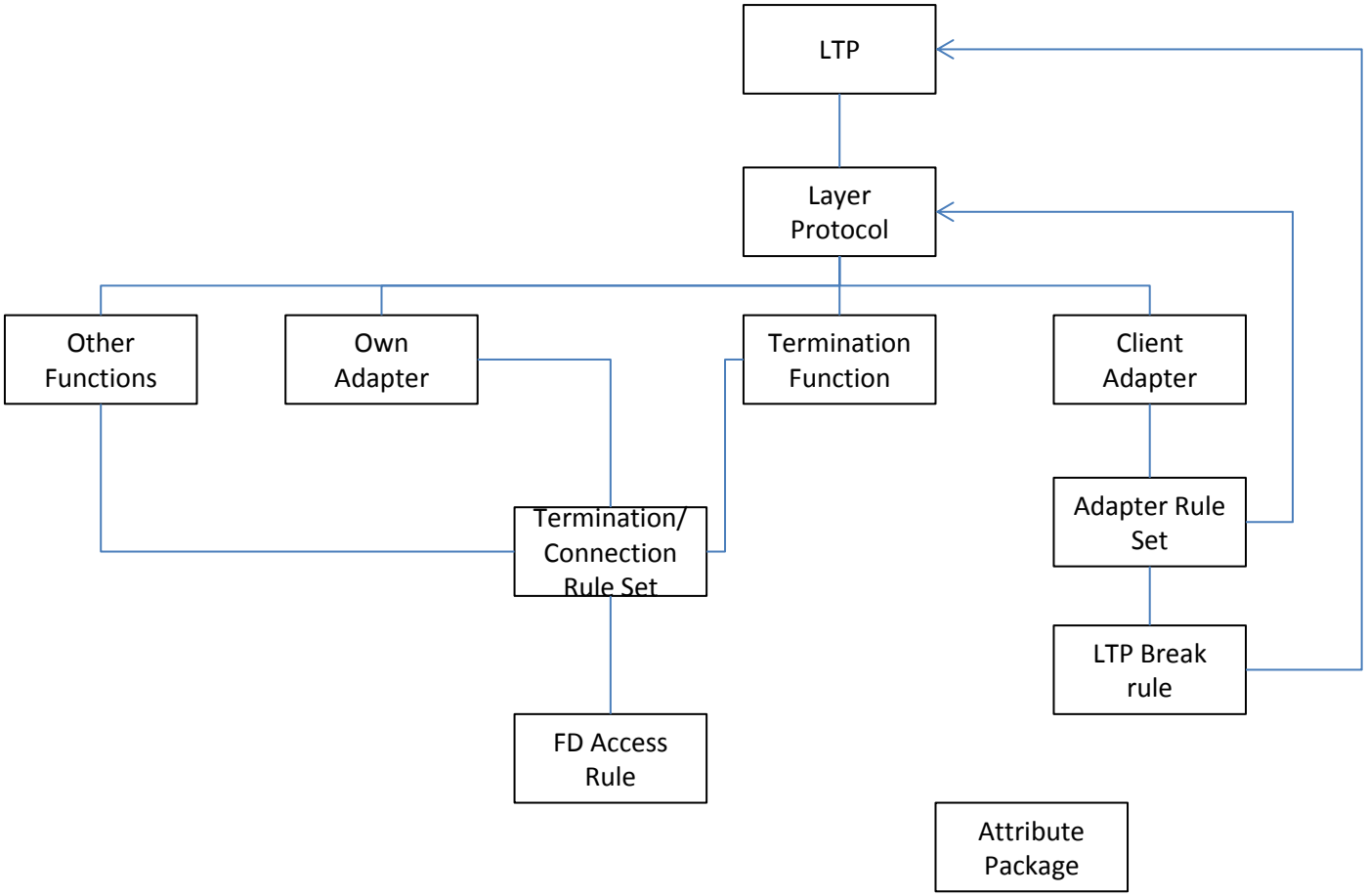
- General considerations
  - FC opportunities are enabled by ForwardingDomain and LTP capabilities
  - LTP spec “explains” Layers that can be terminated and the interaction between terminations
  - ForwardingDomain explains which LTPs bound it and ways in which those LTPs may interrelate
  - Combined specifications provide creation rules for FCs and LTPs in the context of server LTPs and the associated ForwardingDomain
- Specific case of ring protection
  - LTP spec explains client interactions indicating which LTPs can be created together
  - ForwardingDomain for the protection ring explains the legal arrangement of subordinate ForwardingDomains and how they must interconnect (i.e. in a ring)
  - ForwardingDomain for the protection ring identifies layout rules for FCs that travers it in terms of subordinate FC detail (i.e. main and protection paths)
  - SC → Configuration Coordinator: has high level rules for legal combinations of FCs and protection coordinating the creation and deletion of subordinate FCs in the ring
- Need to determine how much of this we need to state explicitly
  - Solution could clearly be private and coded simply allowing or blocking FC creation attempts for the entire ring based upon rule knowledge
  - If mixed rings are required or rings where a controller only controls part then a more explicit model may be necessary
- Remember the initial rationale for the spec work was simply to determine what needed to be exposed in each instance of FC and to allow us to avoid conveying common fixed data for every instance
  - This has extended to encompass switch control considerations and hence the additional complexity
  - There clearly is a need to consider switch control and expected behaviour from the ring etc

Routes (related point)

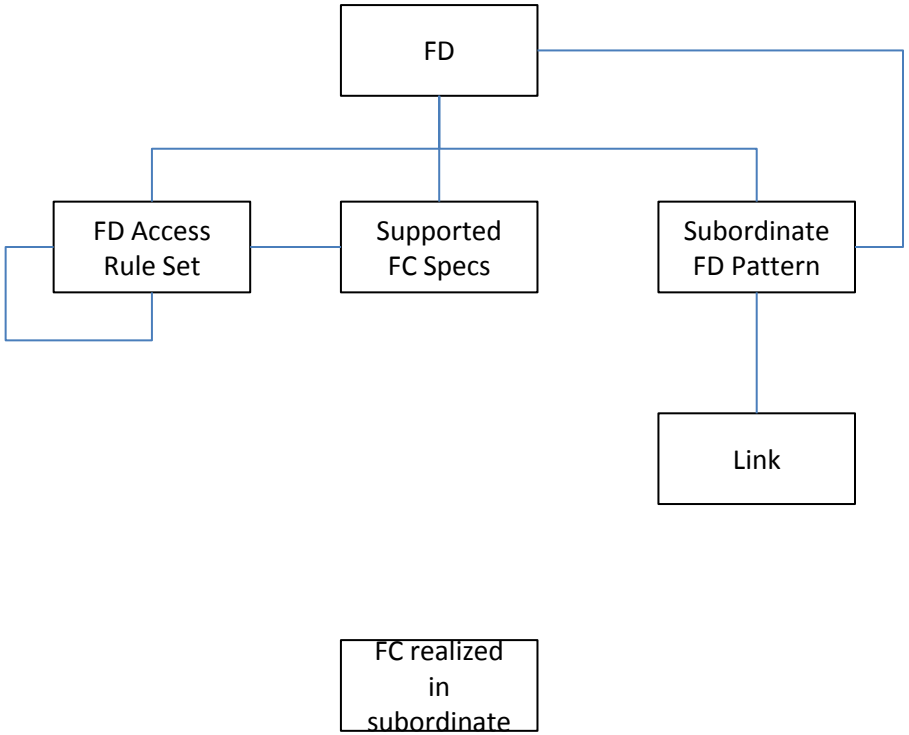
- Assuming a single controller manages the entire ring then limited exposure of capability is required but at this level of abstraction it may not be possible
  - To make a compact statement of capability
  - To provide a deterministic outcome (i.e. the determinism is at a level lower)
- When expressing the FC in the ring
  - Each FC could have a specific protected route (graph) exposed
  - OR: It would be possible to expose the working and protection paths as two separate routes and show interaction in the ring in a more simple fashion perhaps
    - Note that if there are two separate routes they conflict at the ends

LTP AND FD SPEC MODEL

Developing the LTP spec model

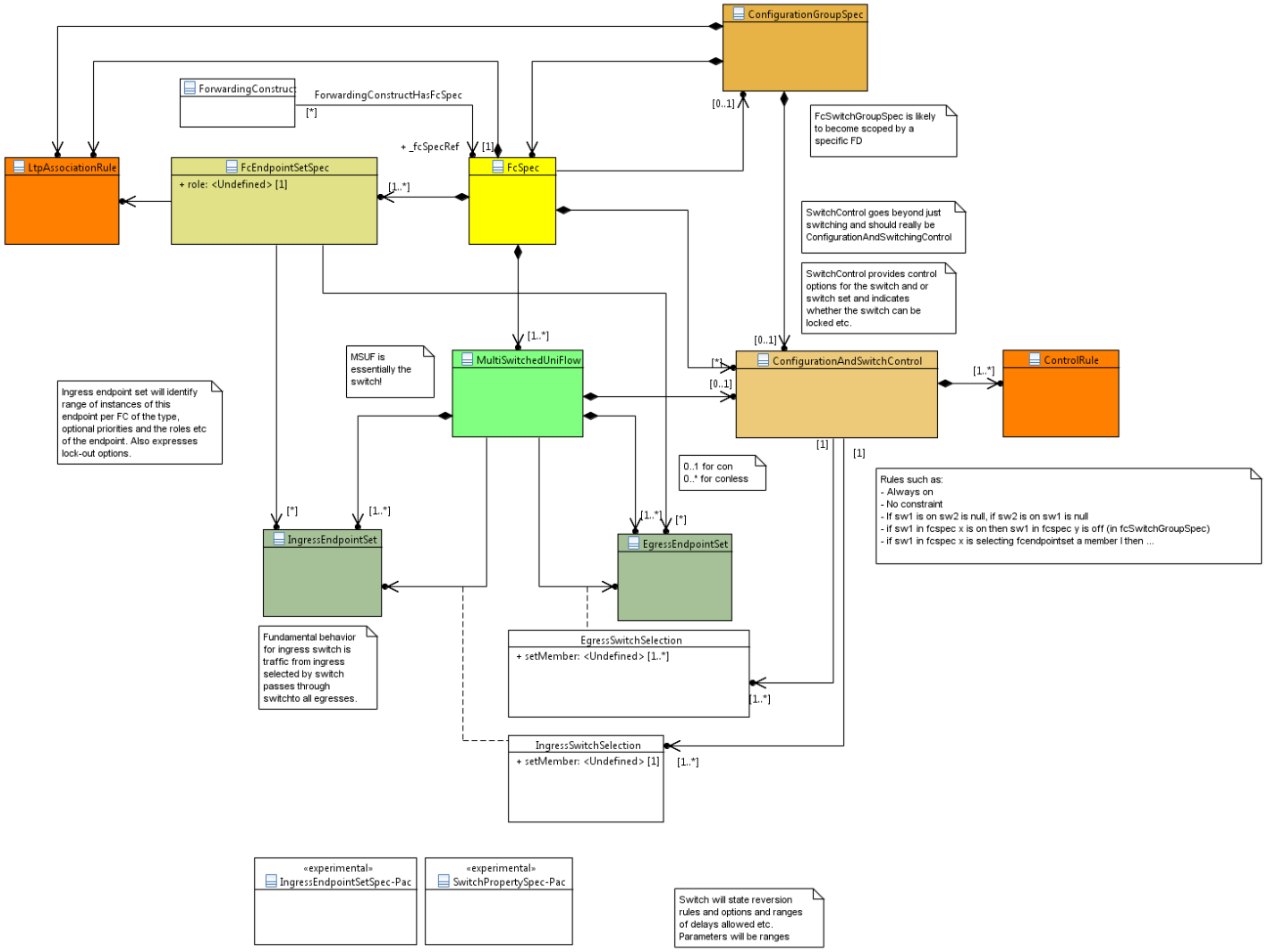


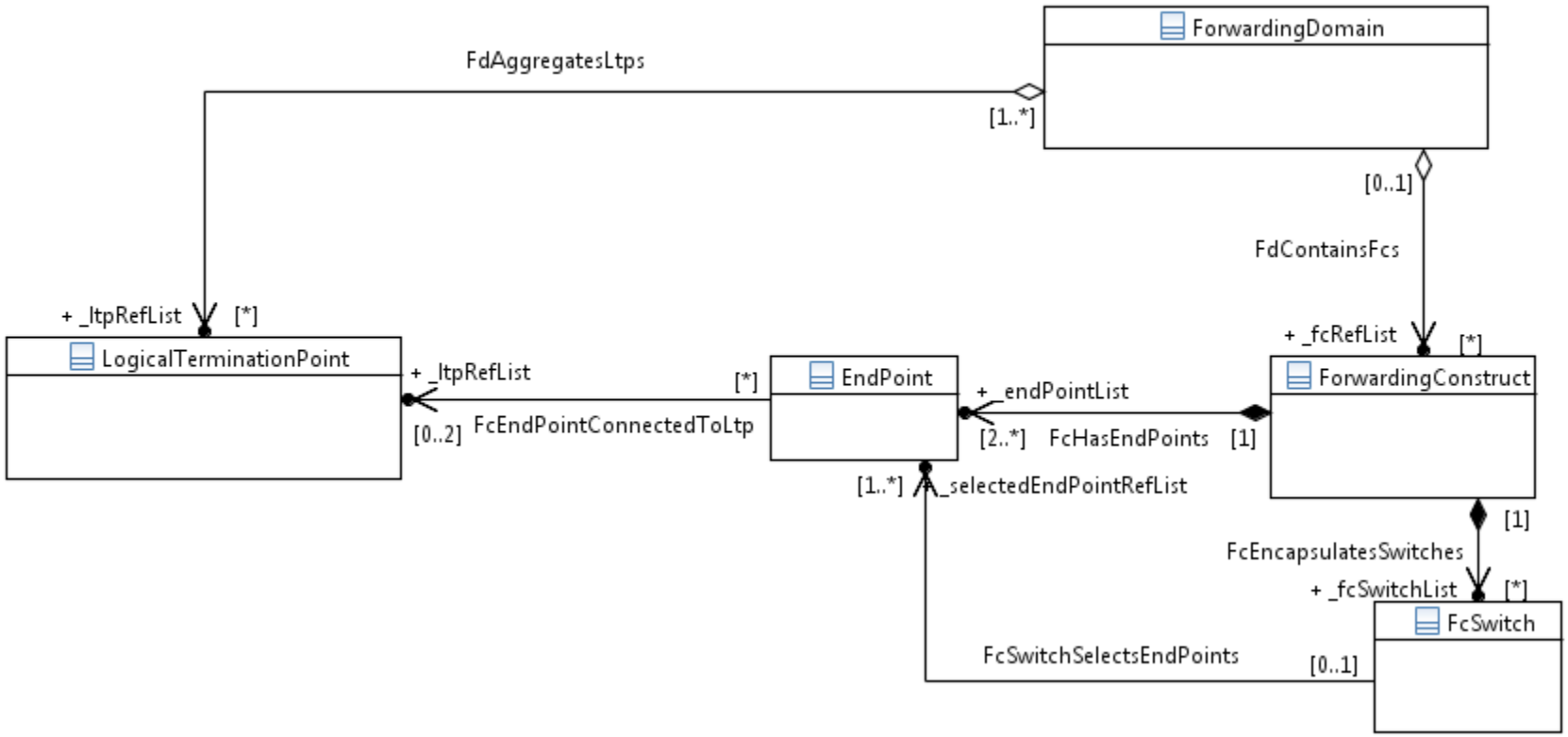
Developing the FD spec model



Spec to TTP mappings

Latest Model




ForwardingDomain
FdAggregatesLtps
[1..*]
[0..1]
FdContainsFcs
+ _ltpRefList
[*]
LogicalTerminationPoint
+ _ltpRefList
[*]
EndPoint
[0..2]
FcEndPointConnectedToLtp
+ _endPointList
[2..*]
FcHasEndPoints
[1]
+ _fcRefList
[*]
ForwardingConstruct
[1..*]
_selectedEndPointRefList
[1]
FcEncapsulatesSwitches
+ _fcSwitchList
[*]
FcSwitch
FcSwitchSelectsEndPoints
[0..1]

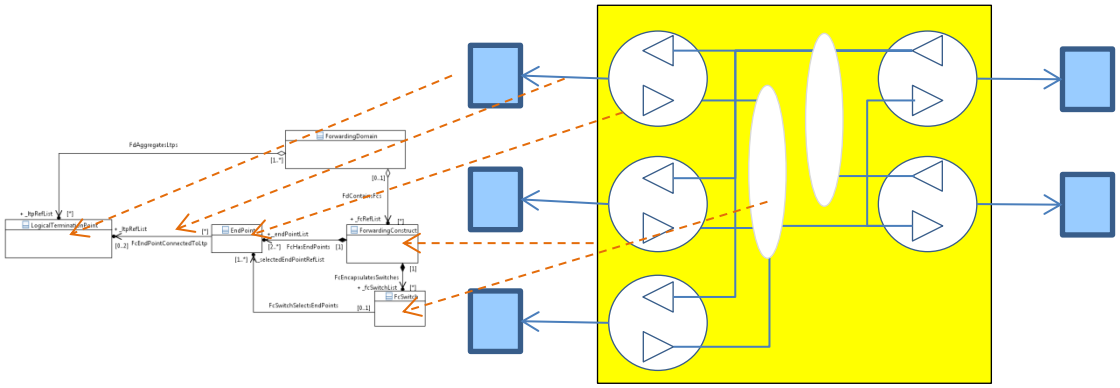
Material on TTPs

C:\Workspace\
Documents\OnfIn

Attached is a TTP model (text file) for a single flow table supporting configuration of EVP-Tree behavior using a pair of VIDs.

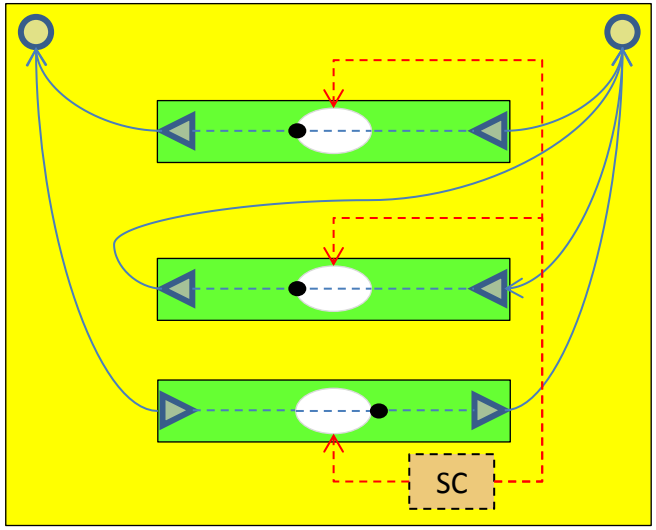
This is still a pattern (i.e., not fully reduced to a single instance with 4 ports that we discussed).

Example for Tree (or hub-spoke)



FC instance with a two roots (showing LTPs in blue)
Switches shown as present as there may be parameters on the queuing that may be relevant

Leaf



Root

FC spec instance

Switch is “abstract” packet rate queuing switch
There is no specific direct control

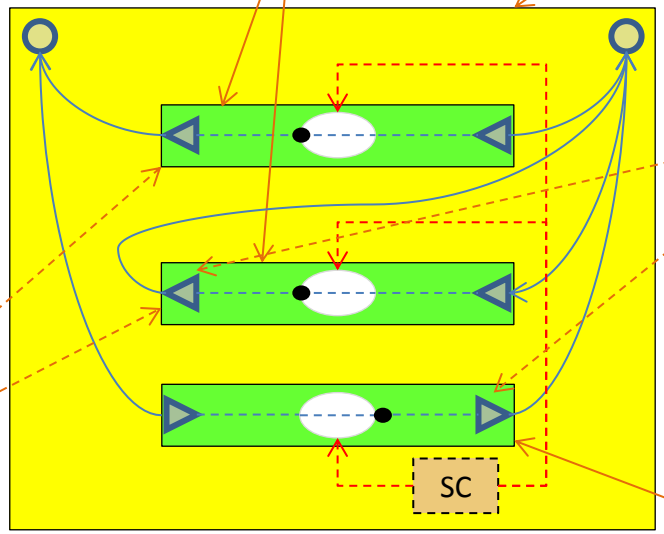
Example for Tree (or hub-spoke)

Add VID
Diagram for Leaf vid

```
"name": "root_flood",
"priority": 1,
"doc": "Flood frame from a root entity to all ports.",
"match_set": [
 {"field": "VLAN_VID", "fix_mask": "0x1000", "fix_value": "0x1000",
   "mask": "0x0fff", "value": "<root_VID>"}
],
"instruction_set": [
 {"instruction": "APPLY_ACTIONS",
  "actions": [
   {"action": "GROUP", "group_id": "<AllPorts>"}
  ]
```

```
"name": "Unicast",
"priority": 2,
"doc": Unicast forwarding entry, e.g. for learned MAC address.",
"match_set": [
 {"field": "VLAN_VID", "fix_mask": "0x1000", "fix_value": "0x1000",
   "mask": "0x0fff", "value": "<VID>"},
 {"field": "ETH_DST", "value": "<learned_MAC>"}
],
"instruction_set": [
 {"instruction": "APPLY_ACTIONS",
  "actions": [
   {"action": "OUTPUT", "port_id": "<port_n>"}
```

Leaf



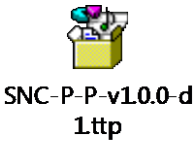
Root

```
"name": "RootPorts",
 "doc": ["Output to all root ports in a tree (except
IN_PORT if it is a root port).",
      "Entry per EVP-Tree containing a bucket for each
root port in the EVP-Tree."],
 "group_type": "ALL",
 "bucket_types": [
  {"name": "LeafFloodPort",
   "action_list": [{"action": "OUTPUT", "port_id":
"<port_n>"}]
  }
```

```
"name": "AllPorts",
 "doc": ["Output to all ports in a tree (except IN_PORT).",
      "Entry per EVP-Tree containing a bucket for each port in the EVP-Tree."],
 "group_type": "ALL",
 "bucket_types": [
  {"name": "RootFloodPort",
   "action_list": [{"action": "OUTPUT", "port_id": "<port_n>"}]
  }
 ]
},
```

```
"name": "leaf_flood",
"priority": 1,
"doc": "Flood frame from a leaf entity to root port(s).",
"match_set": [
 {"field": "VLAN_VID", "fix_mask": "0x1000", "fix_value": "0x1000",
   "mask": "0x0fff", "value": "<leaf_VID>"}
],
"instruction_set": [
 {"instruction": "APPLY_ACTIONS",
  "actions": [
   {"action": "GROUP", "group_id": "<RootPorts>"}
  ]
 }
```

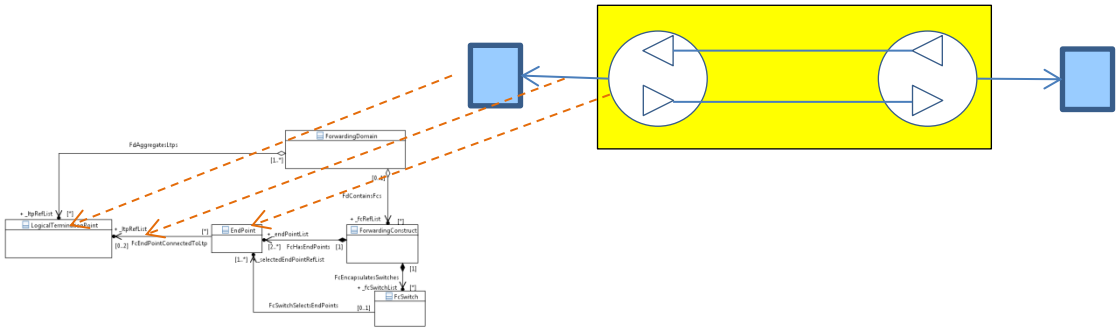
Material on TTPs



Attached is a TTP model (text file) for a single flow table supporting configuration of a point to point bi-directional capability.

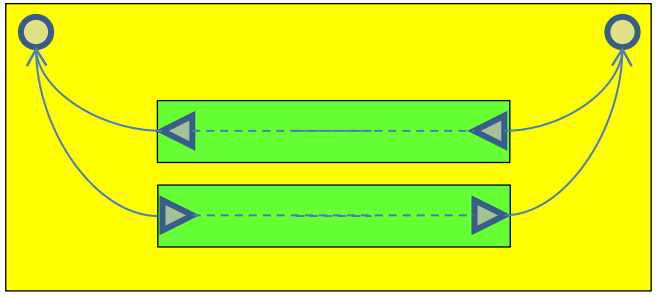
This is still a pattern (i.e., not fully reduced to a single instance with 4 ports that we discussed).

Example for point to point

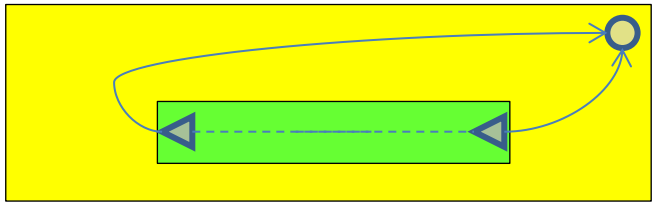


FC instance with a two points

A1



A2

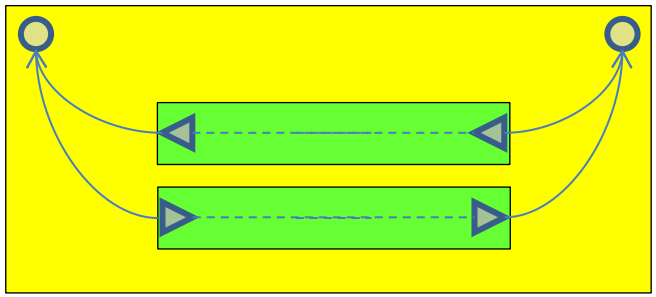


A

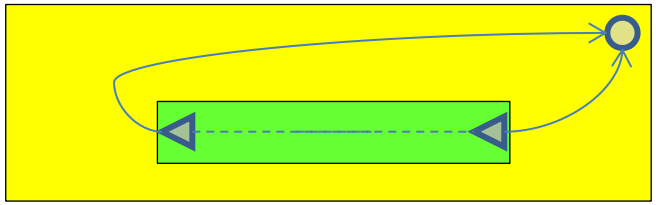
Not self

Example for point to point

A1



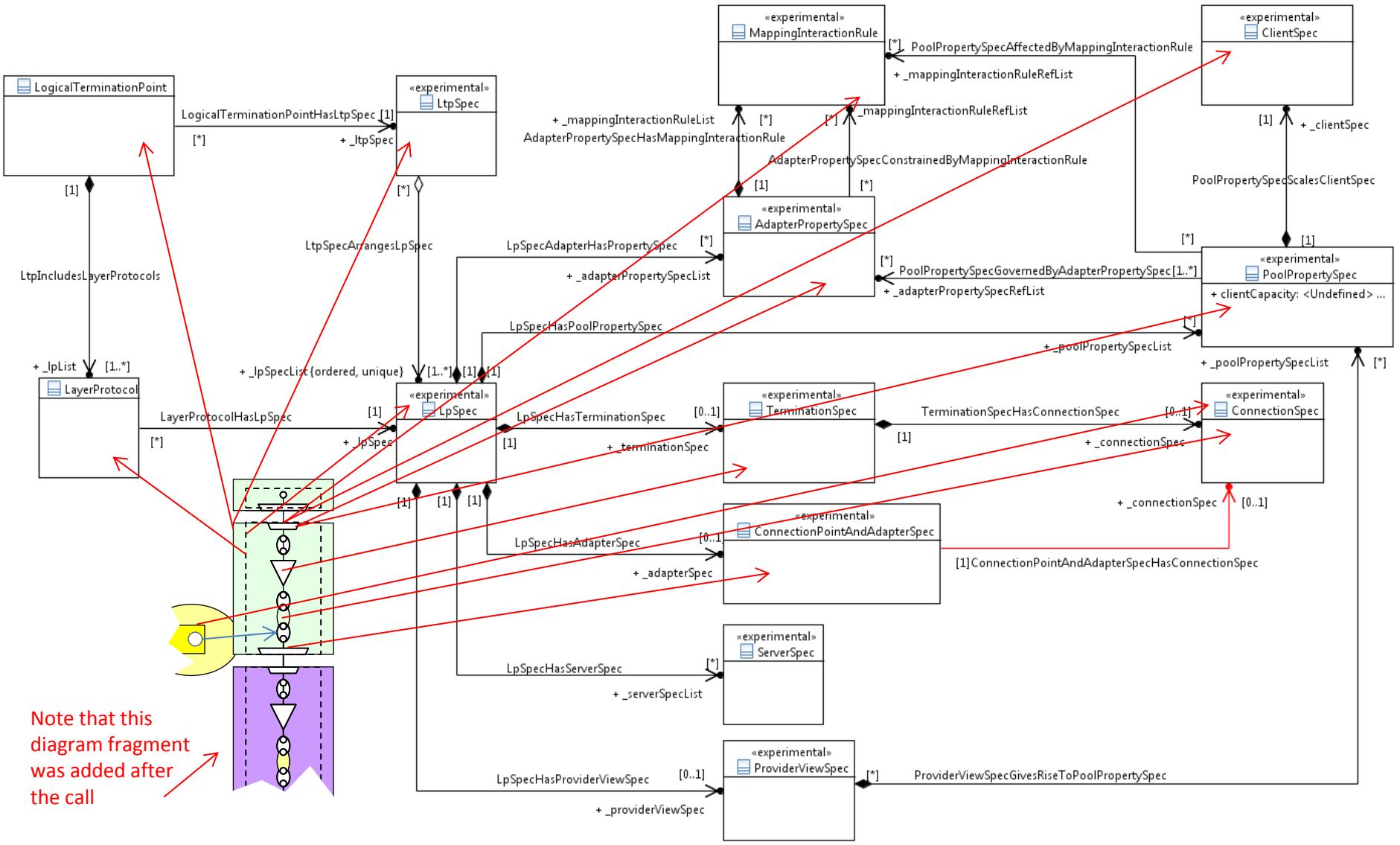
A2



A

Not self

Sketch of LTP Spec model (this work is experimental)



ConfiguredClientCapacity