Use Case - 3

Multi-domain case

Xian Zhang
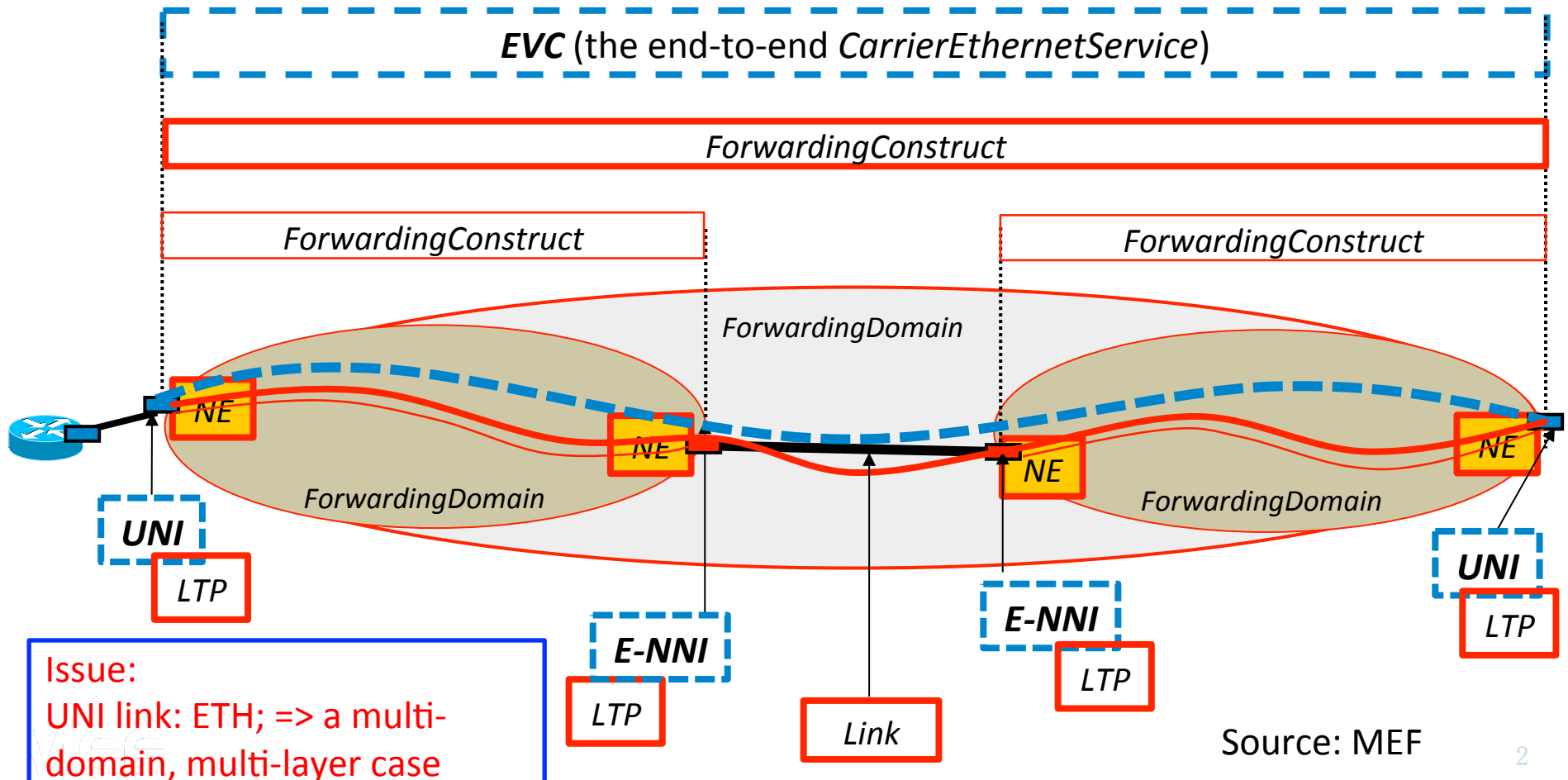
Huawei

Mapping from Service/EVC to ForwardingConstruct
Single Provider, single managed FD, partitioning

managed object classes at Service level
(potentially appearing at Legato, Interlude)

managed object classes at Resource level
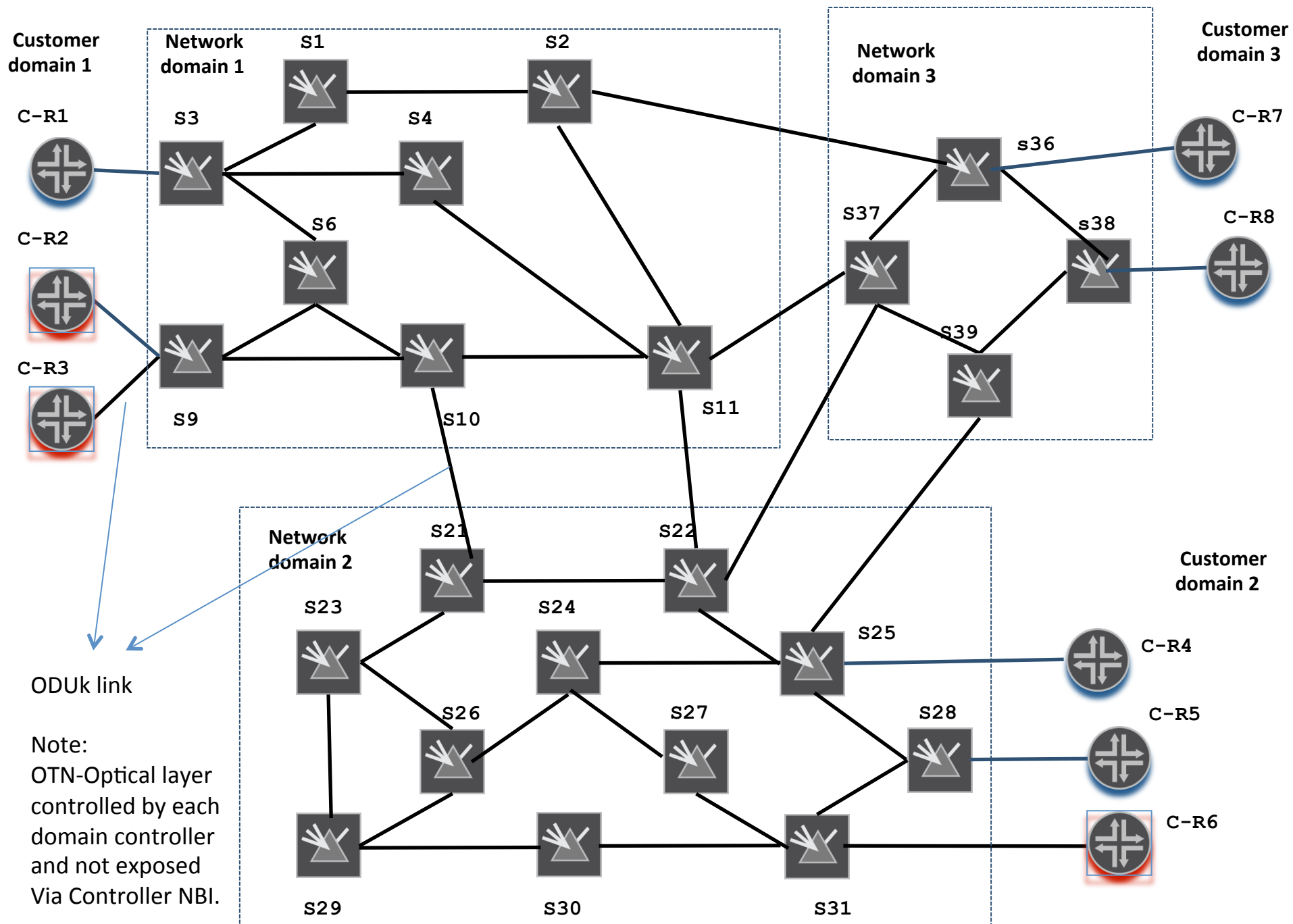(potentially appearing at Presto)

EVC (the end-to-end *CarrierEthernetService*)

ForwardingConstruct

ForwardingConstruct

ForwardingConstruct

ForwardingDomain

NE

NE

NE

NE

ForwardingDomain

ForwardingDomain

UNI

LTP

E-NNI

LTP

E-NNI

LTP

Link

UNI

LTP
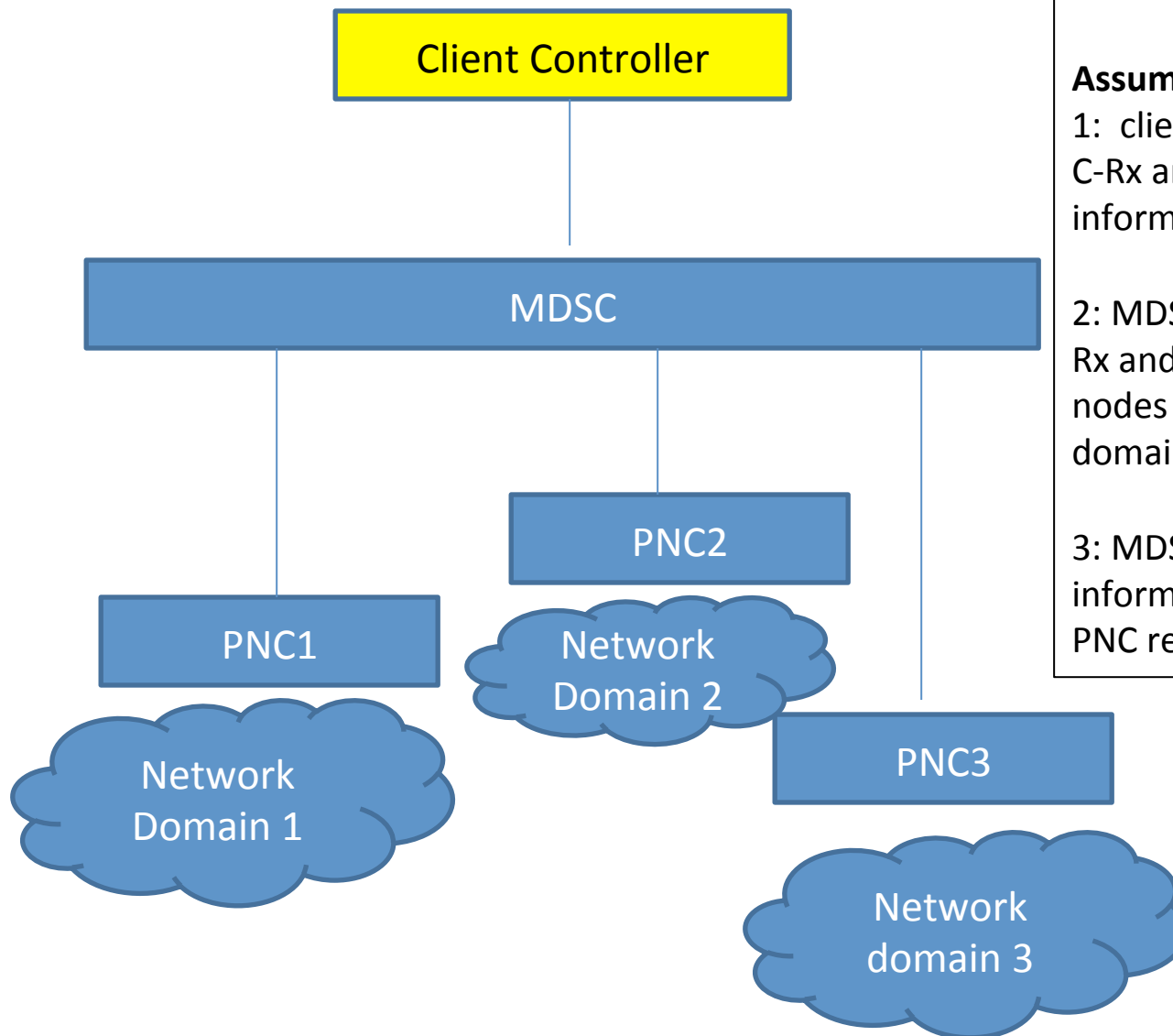
Issue:
UNI link: ETH; => a multi-domain, multi-layer case

Source: MEF

Figure provided by Igor

Controlling Hierarchy

Client Controller

MDSC

PNC2

PNC1

Network Domain 2

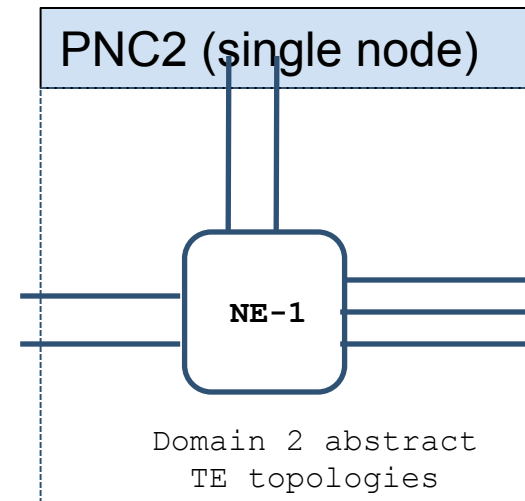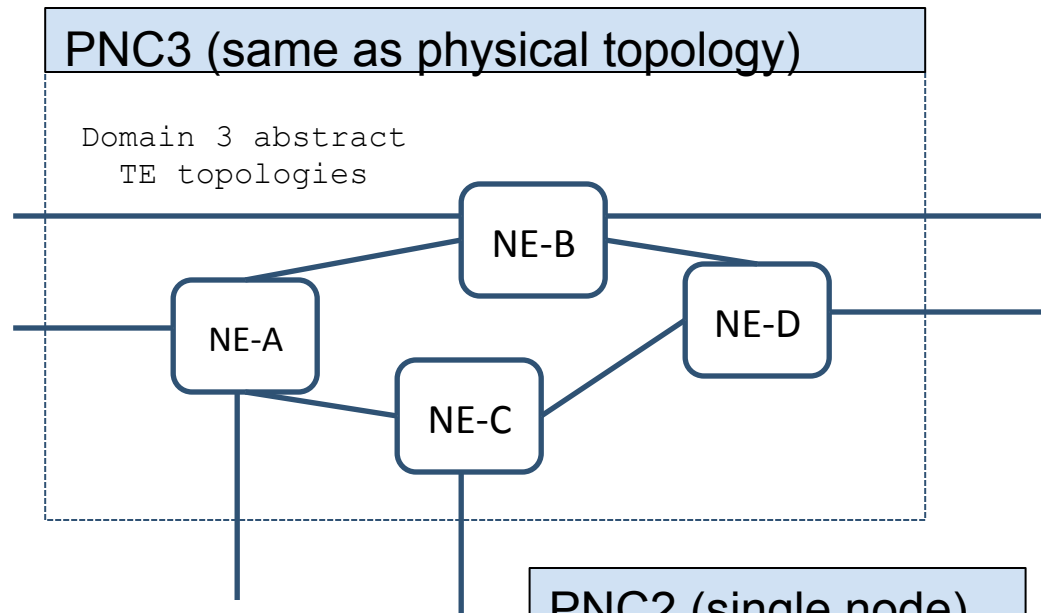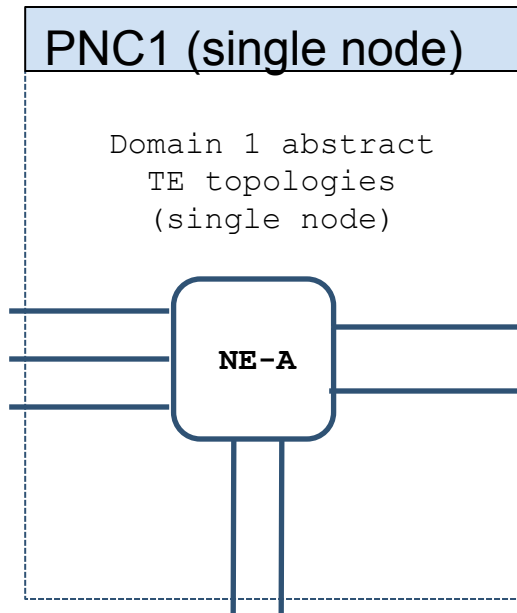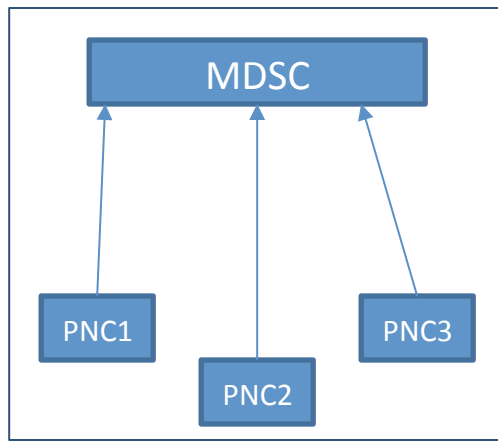Network Domain 1

PNC3

Network domain 3

Assumption:
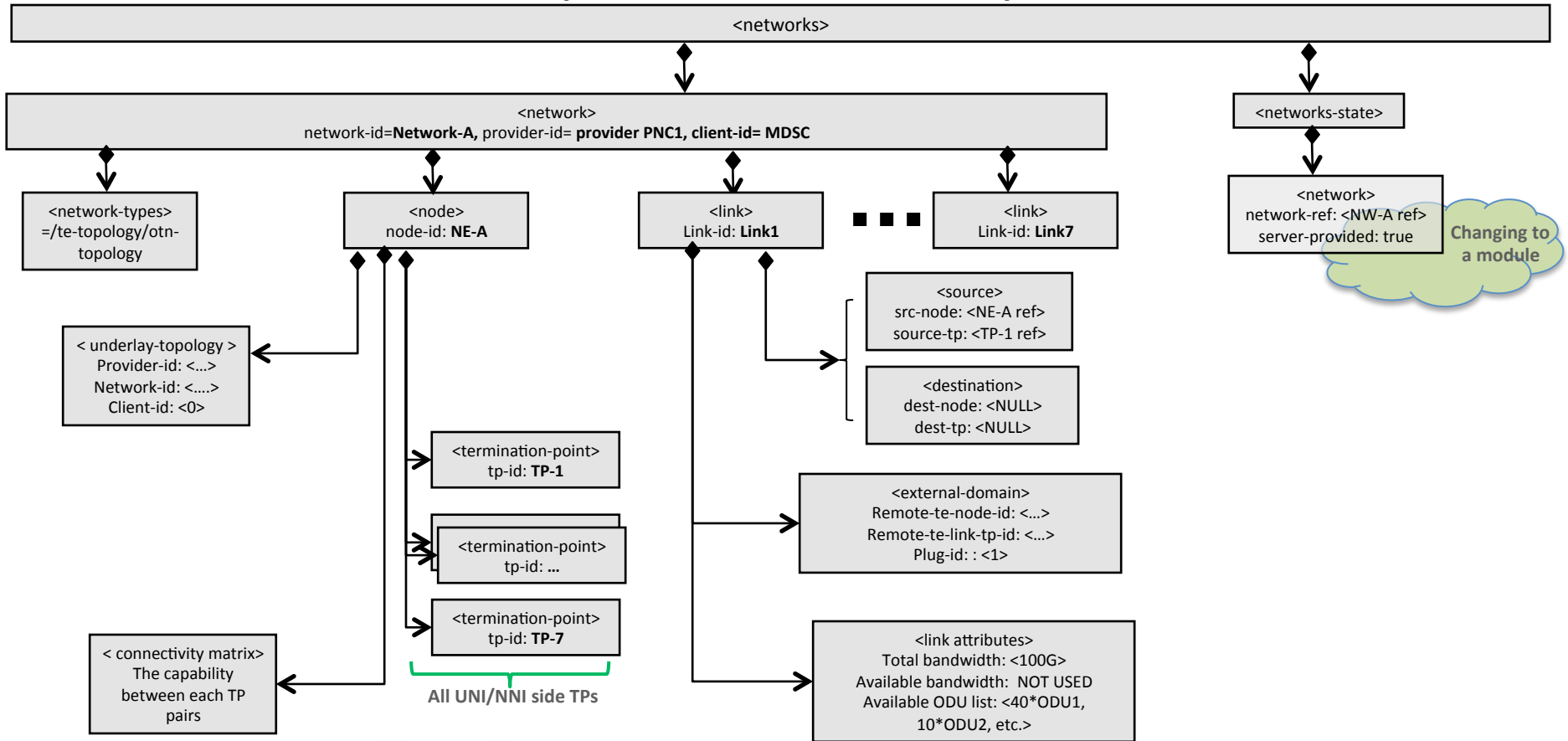1: client controller knows the C-Rx and its access link information.

2: MDSC knows how to map C-Rx and its network side of nodes within its network domain.

3: MDSC has no topology information at all before each PNC reports its topology.

Domain controller reporting

Network Topology (teas) Model Instantiation (PNC1 to MDSC)

<networks>

<network>
network-id=**Network-A,** provider-id= **provider PNC1, client-id= MDSC**

<networks-state>

<network>
network-ref: <NW-A ref>
server-provided: true

Changing to a module

<network-types>
=/te-topology/otn-topology

<node>
node-id: **NE-A**

<link>
Link-id: **Link1**

<link>
Link-id: **Link7**

< underlay-topology >
Provider-id: <…>
Network-id: <….>
Client-id: <0>

<source>
src-node: <NE-A ref>
source-tp: <TP-1 ref>

<destination>
dest-node: <NULL>
dest-tp: <NULL>

<termination-point>
tp-id: **TP-1**

<termination-point>
tp-id: **…**

<external-domain>
Remote-te-node-id: <…>
Remote-te-link-tp-id: <…>
Plug-id: : <1>

<termination-point>
tp-id: **TP-7**

All UNI/NNI side TPs

< connectivity matrix>
The capability between each TP pairs

<link attributes>
Total bandwidth: <100G>
Available bandwidth: NOT USED
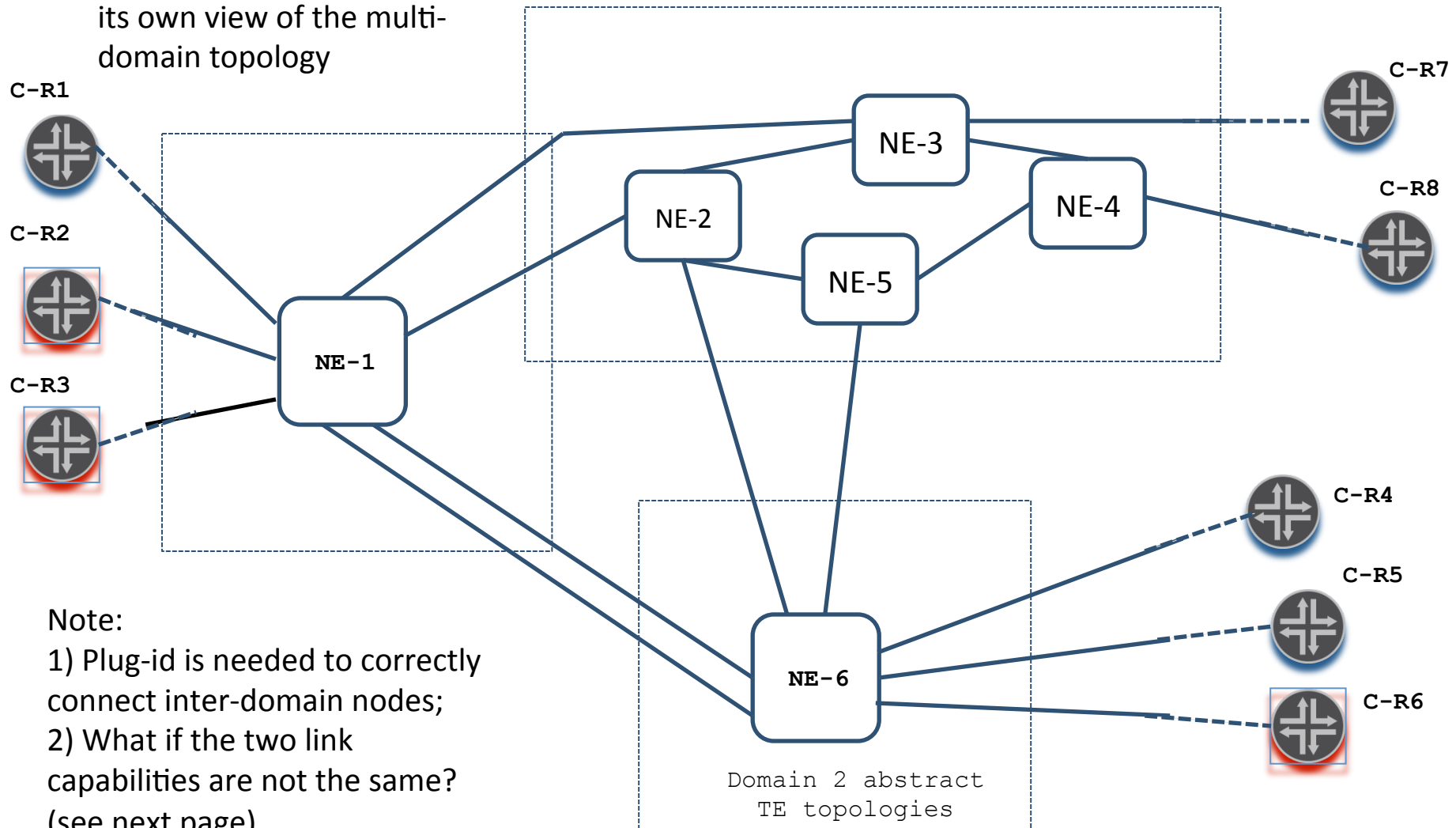Available ODU list: <40*ODU1, 10*ODU2, etc.>

If nothing is provided, it can either be:
1) No constraints at all;
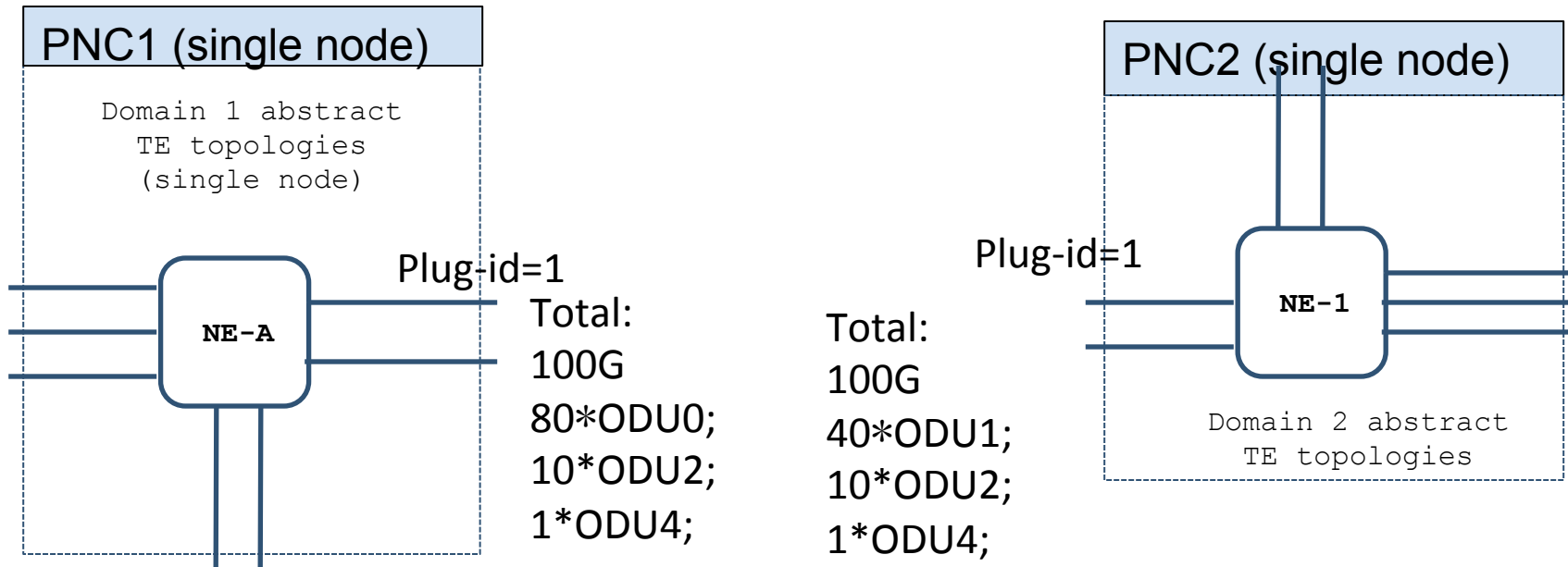2) Path computation function of Tunnel model is expected to be used.

Note:
1: PNC2 will report something similar;
2: PNC3 will report as described in use case 1(additional intra-domain links/attributes;

What does MDSC get?

MDCS merges what is gets from PNCs and generate its own view of the multi-domain topology

C-R1

C-R7

C-R2

NE-3

C-R8

C-R3

NE-2

NE-4

NE-1

NE-5

C-R4

C-R5

NE-6

C-R6

Domain 2 abstract
TE topologies

Note:
1) Plug-id is needed to correctly connect inter-domain nodes;
2) What if the two link capabilities are not the same? (see next page)

Non-identical Link attributes

PNC1 (single node)

Domain 1 abstract
TE topologies
(single node)

NE-A

Plug-id=1

Total:
100G
80*ODU0;
10*ODU2;
1*ODU4;

Plug-id=1

Total:
100G
40*ODU1;
10*ODU2;
1*ODU4;

PNC2 (single node)

NE-1
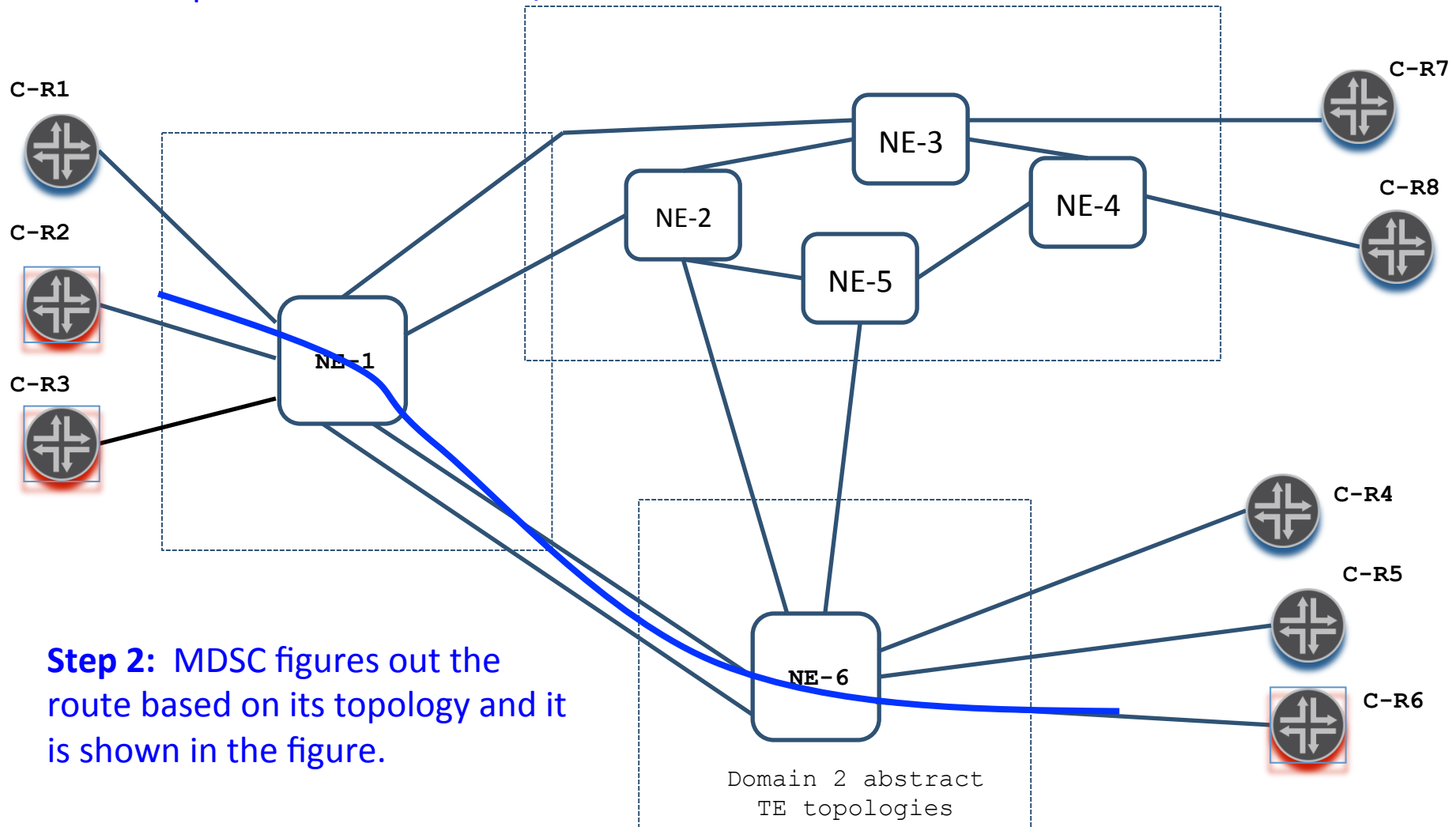
Domain 2 abstract
TE topologies

Issue: what to do when the two links reported by different PNCs do not have identical attributes?
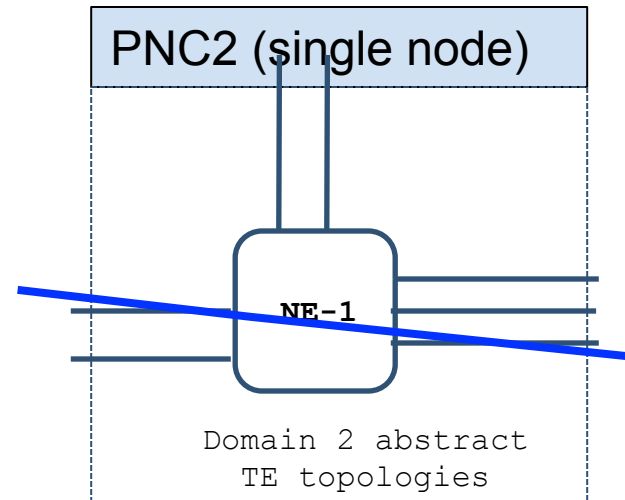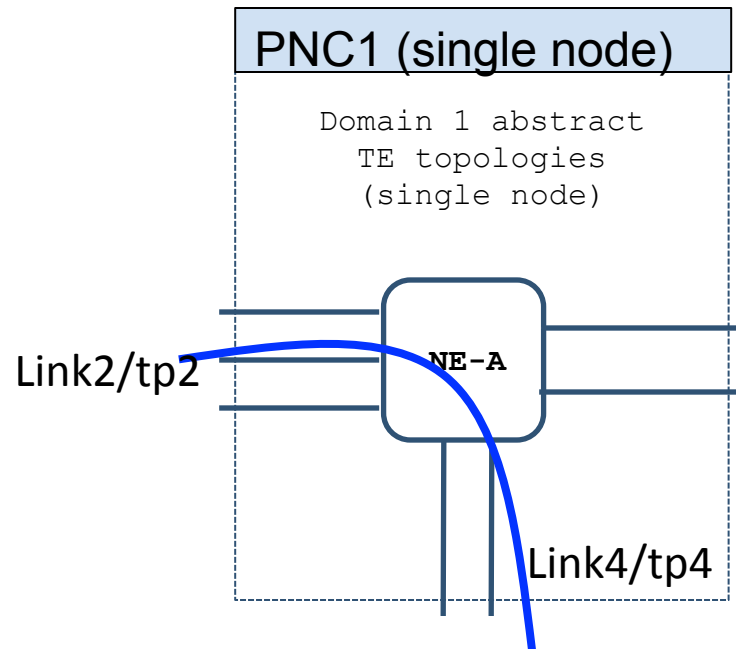
Solution: MDSC can keep what is mutual on both links in its own TED. In this case, only setting up ODU2 or ODU4 LSP/tunnel.

MDSC – Path Computation

Step 2: MDSC figures out the route based on its topology and it is shown in the figure.

Domain 2 abstract
TE topologies

MDSC=>PNC

MDSC

PNC1

PNC2

PNC3

PNC2 (single node)

NE-1

Domain 2 abstract
TE topologies

PNC1 (single node)

Domain 1 abstract
TE topologies
(single node)

Link2/tp2

NE-A

Link4/tp4
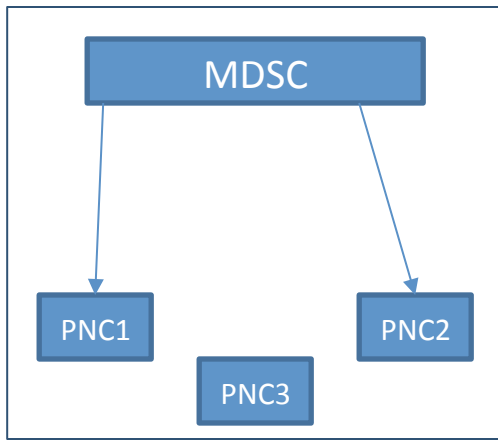
TEAS Tunnel Model Instantiation (client => MDSC)

MDSC

PNC1 PNC2 PNC3

```
<te>
   |
   v
<tunnels>
   |
   v
<tunnel>
name: <Service-1 ref>
type: <p2p ref>
tunnel-id: <tunnel-1-id ref>
```

<config>
name: **Service-1**
tunnel-id: **tunnel-1-id**
type: p2p
bandwidth: 10Gbps
lsp-prot-type: unprotected
admin-status: up

<primary-paths>
Name: <path A-ref>

<secondary-paths>
NULL

<config>
type: explicit

<source>
Source: NE-A
<destination>
Destination: NE-A
<src-tp-id>
Source-ttp: NULL
<dest-tp-id>
Destination-ttp: NULL

<explicit-route-objects>
index: 0
e-r-usage: route-include-ero
type: link
link-index: <link2>
Index=1
e-r-usage: route-include-ero
Type=label
Label: <...>
Index: 2
e-r-usage: route-include-ero
type: link
link-index: <link4>
...
Label: <...>

Issues:
1) If there are multiple-topologies, should the MDSC specify on which topology?
2) Link-index in ERO missing!
3) How to specify ODU label? (in ODU tunnel)
4) ODU tunnel: how to augment from TE Tunnel?
5) Is unnumbered link same as TP? (router-id + interface-id)

Another issue: TP directionality?

Case A

Node A

Src

TP1 0 | 0 TP2
         0 TP3

Dst
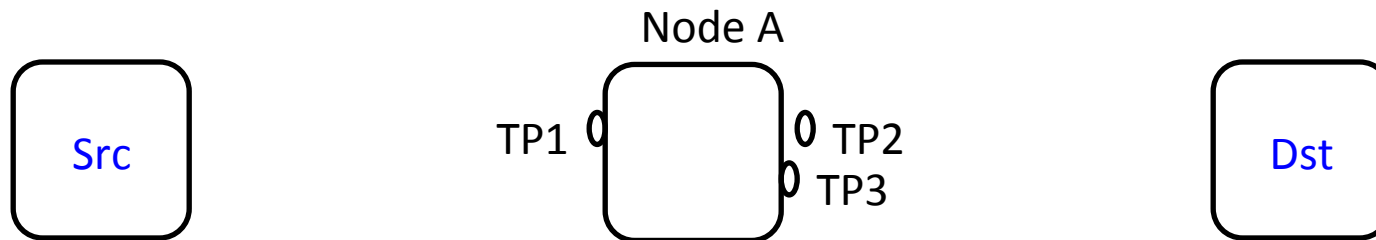
Loose ERO as: NodeA/TP1 (ingress)
It will be like: Src-NodeA/TP1-NodeA/TP2-Dst

Case B

Node A

Src

TP1 0 | 0 TP2
         0 TP3

Dst
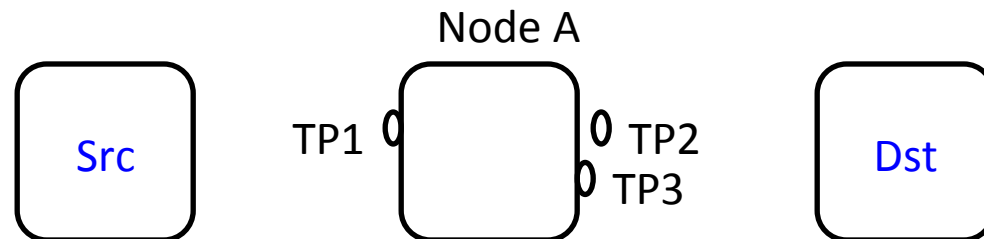
Loose ERO as: NodeA/TP1 (egress)

It will be like: Src-NodeA/TP2-NodeA/TP1-Dst