MDONS

Implementation of T-API topology in TransportPCE

Gilles THOUENON, Christophe BETOULE
OLN/WNI/AOT/SOAN

4th of November 2020

Context

1st implementation

Two different kinds of topology abstractions were exposed in T-API modelling:

- **One for 100GE client ports of Transponder when their respective network port were attached to the ROADM infrastructure**
  - under a single ETH node with as many onep ports as 100GE client ports
- **One for OTN nodes (SWITCH and MXPDR) coming from TransportPCE's OTN topology**
  - under a multi-layer tapi topology (DSR/ODU and photonics nodes) and transitional links

Recent evolution added otn links (OCH-OTU4 and ODU4) from TransportPCE's OTN topology

- two unidirectional otn links from otn-topology under a single bidirectional otn link in tapi topology
- cf. change https://git.opendaylight.org/gerrit/c/transportpce/+/92949

TAPI

Compliant with §5 of « TAPI v2.1.2 Reference Implementation » document (August 2019)

- focus on the topology service of TAPI

New implementation updates

Rework the multi-layer tapi topology

« T0 - Multi-layer topology »

- **give a multi-layer abstraction of otn nodes whose network ports are connected to the ROADM infrastructure through xponder-input/output links**
- **Add also in this multi-layer abstraction 100GE Transponder nodes whose associated network ports are connected to the ROADM infrastructure through xponder-input/output links**
- **Add a new Photonic TAPI node to also abstract the ROADM infrastructure**
- **Add in this multi-layer abstraction xponder-input/output links from openroadm-topology through TAPI OMS links**

Rework the 100G Transponder topology

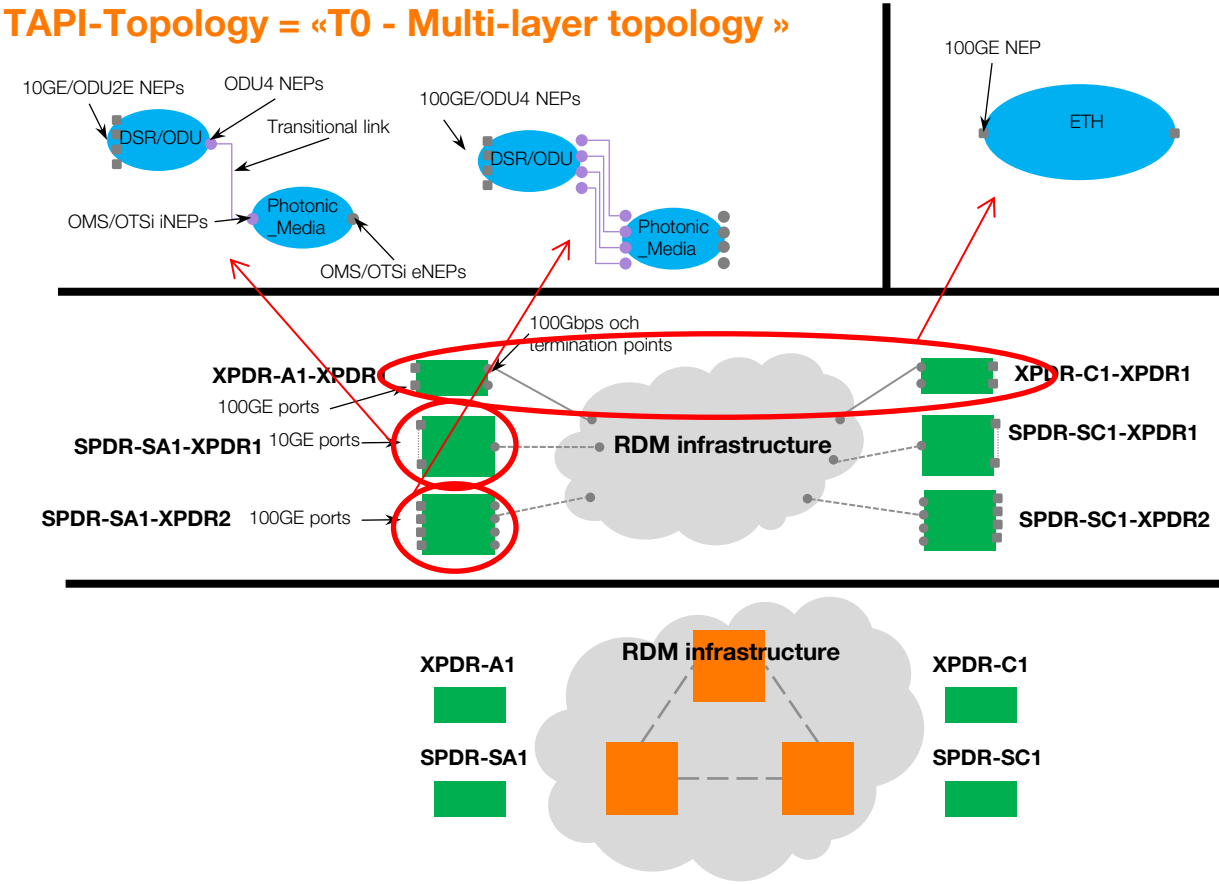
« Ethernet Topology » => « Transponder 100GE »

- **Abstracts at a higher level the « T0 - Multi-layer topology » to expose only 100GE ports of 100G Transponder under a single DSR node composed of as many onep as 100GE ports clients ports connectable between them**

Strengthens, improves and optimizes tests

Old implementation

TAPI-Topology = «T0 - Multi-layer topology »



TAPI-Topology = « topo ethernet »

- **single node** ETH with as many « owned-node-edge-point as 100GE port whose their associated network port is connected to the RDM infrastructure
- per default, full forwarding between NEPs in TAPI node

openroadm-topology / otn-topology

- node view with TP corresponding to physical ports
- XPONDER-INPUT/OUTPUT links to connect XPDR network ports to RDM infrastructure

openroadm-network

- NETCONF / per device view

New tapi topology implementation

TAPI « Transponder 100GE »

- empty

TAPI «T0 - Multi-layer topology »

- If at least one roadm contains one SRG node
- else empty

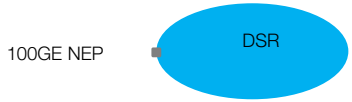
RDM infra

openroadm-topology / otn-topology

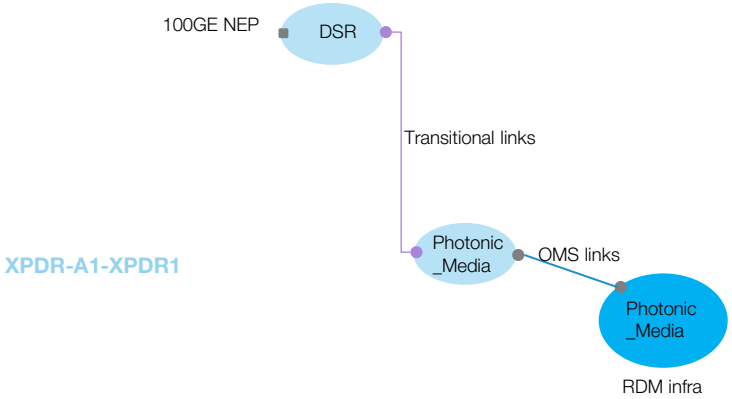
- no xponder device attached to the ROADM infra

New tapi topology implementation

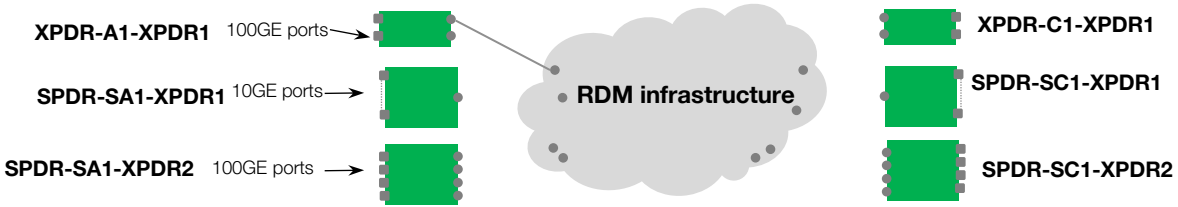
TAPI « Transponder 100GE »



TAPI «T0 - Multi-layer topology »

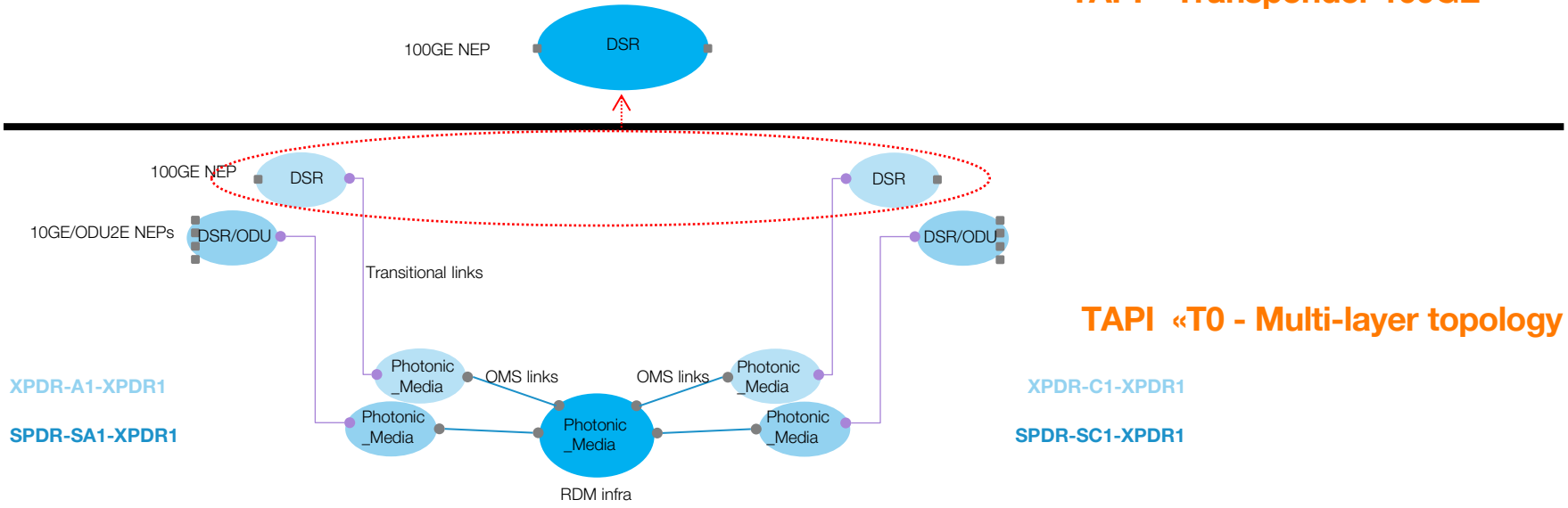


openroadm-topology / otn-topology

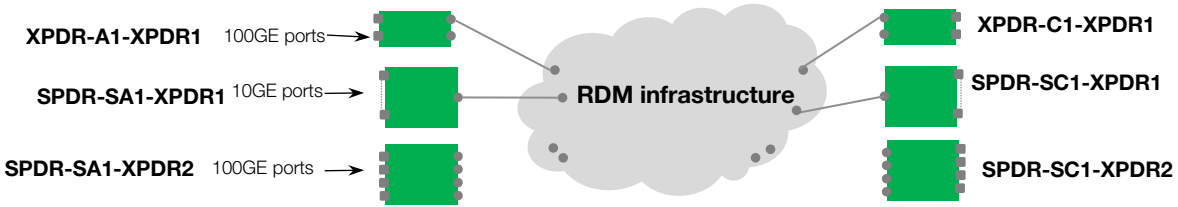


New tapi topology implementation

TAPI « Transponder 100GE »



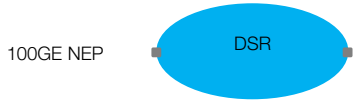
TAPI «T0 - Multi-layer topology »



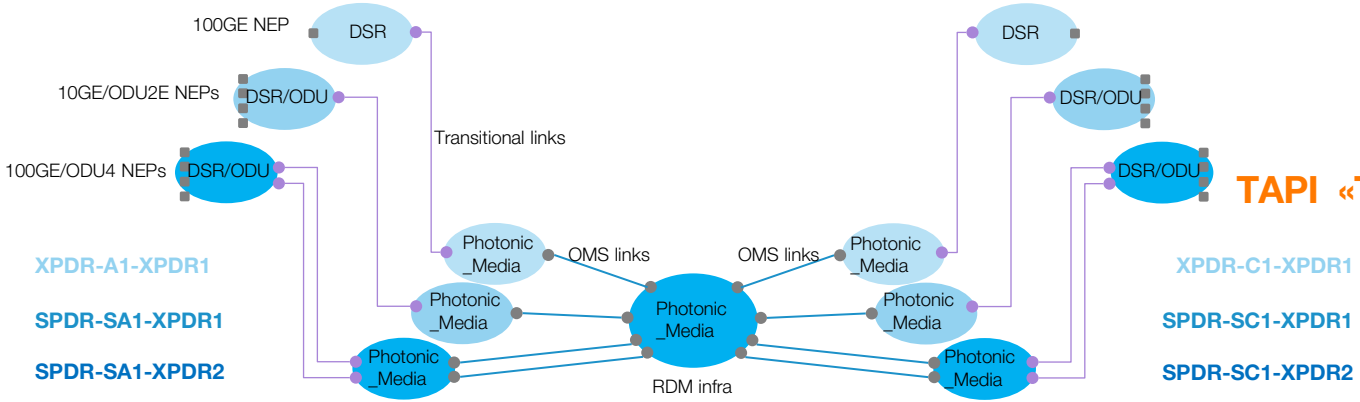
openroadm-topology / otn-topology

New tapi topology implementation

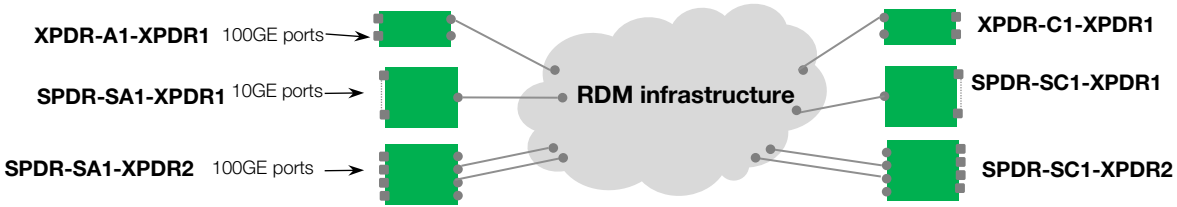
TAPI « Transponder 100GE »



TAPI «T0 - Multi-layer topology »



openroadm-topology / otn-topology



Conversion of node forwarding rules (not changed but also applied to the Tpdr case)

otn-openroadm topo

node contains 1 odu-switching-pool, with a list of non-blocking-list

TPDR

as many list as client/network ports

- {[C1, N1], [C2, N2], …}
- Or {[C1, N1]}, {[C2, N2]}…

MXPDR

as many list as client port number

- {[C1, N1], [C2, N1], …, [Cn, N1]}

SWITCH

a single list with all client/network ports

- {[C1, C2, C3, C4, N1, N2, N3, N4]}

TAPI - T0 – Multi-layer topology

a global « MAY_FORWARD_ACROSS_GROUP » rule defined for all nodes

MXPDR / TPRD

DSR/ODU node: as many node-rule-group as client port number

- {[C1, N1], [C2, N1], …, [Cn, N1]}
- {[C1, N1], [C2, N2], …}

Photonic_Media node: a single node-rule-group

- {[iNEP1, eNEP1]}

SWITCH

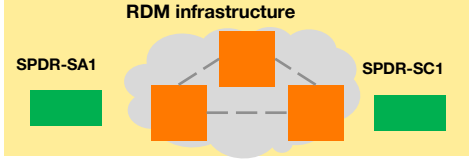
DSR/ODU node: a single node-rule-group with all NEPs (client and network sides)

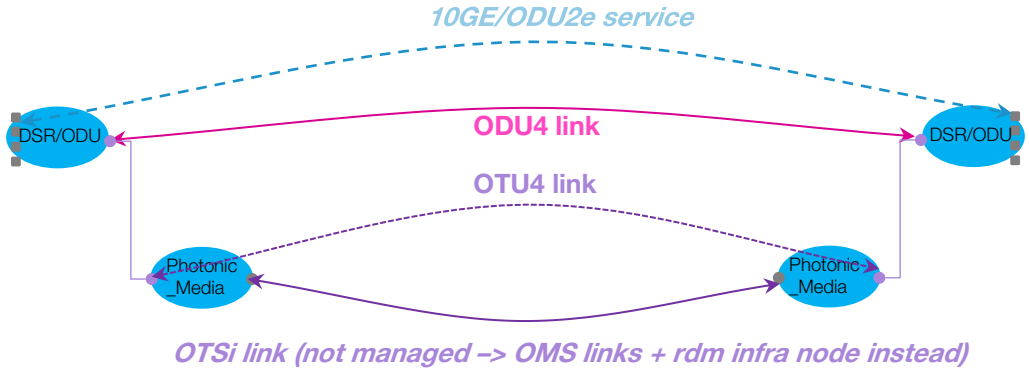
- {[C1, C2, C3, C4, N1, N2, N3, N4]}

Photonic_Media node: as many node-rule-group as network port number

- {[iNEP1, eNEP1], [iNEP2, eNEP2], …, [iNEPn, eNEPn]}

End-to-end connectivity service



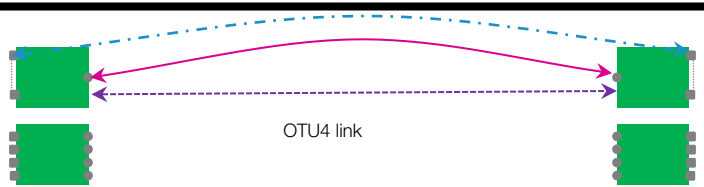


TAPI-topology

OTSi link (not managed –> OMS links + rdm infra node instead)

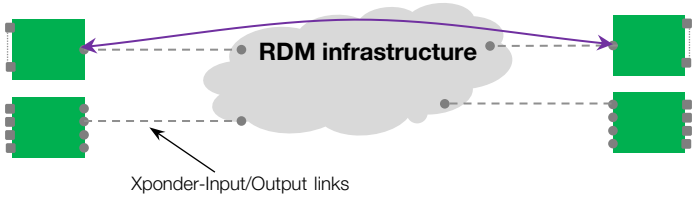
10GE/ODU2e service creation

ODU4 service creation



otn-topology

OCH/OTU4 service creation



openroadm-topology

Conclusion

This new implementation

- **homogenizes tapi multi-layer topology abstraction whatever the xponder device type is (Tp100G, otn Mxpdr, otn switch)**
- **Gives a more relevant abstraction taking into acount data from both otn-topology and openroadm-topology**
  - only xponder node connected to the roadm infrastructure
- **Gives an abstraction of the ROADM infrastructure**

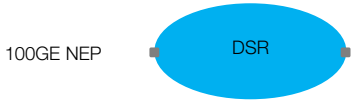
To be noticed

TAPI topology exposed by TransportPCE is not saved in ODL Datastore

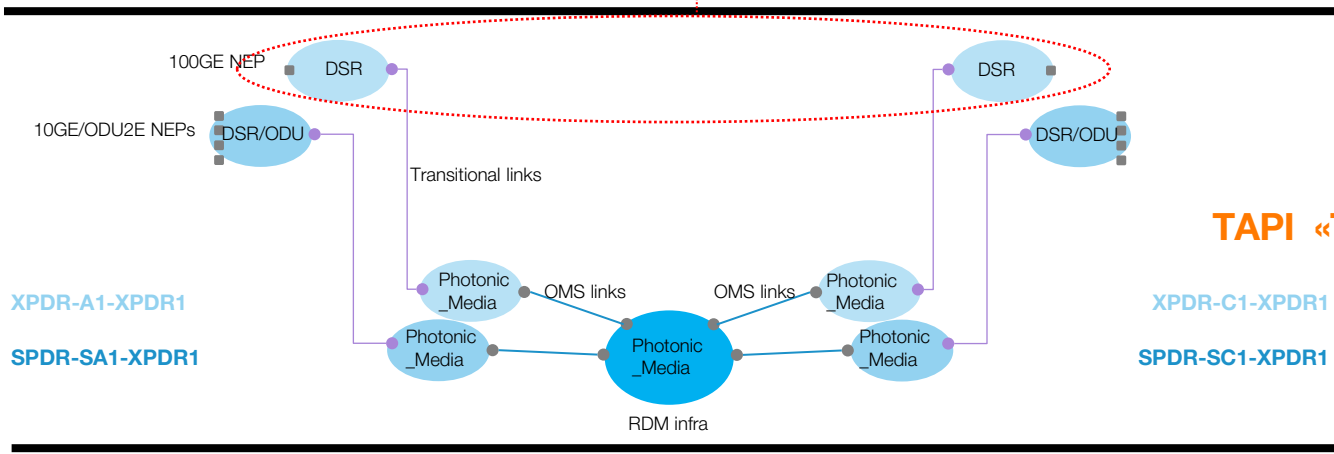
- **calculated on the fly at each TAPI:get-topology-details request**
- **…but with deterministic and stable uuids…**

Thank You

TAPI « Transponder 100GE »



TAPI «T0 - Multi-layer topology »



openroadm-topology
otn-topology