# Two-Level Abstraction Approach for SDN-based Service Provisioning in Open Line Systems Featuring TAPI Externalized Path Computation

K. Ishii<sup>(1)</sup>, R. Casellas<sup>(2)</sup>, Fco. J. Vílchez <sup>(2)</sup>, R. Vilalta<sup>(2)</sup>, R. Martínez<sup>(2)</sup>, J.Ma.Fàbrega<sup>(2)</sup>, L. Nadal<sup>(2)</sup>, M. Svaluto Moreolo<sup>(2)</sup>, R. Muñoz<sup>(2)</sup>, S. Namiki<sup>(1)</sup>

<sup>(1)</sup> National Institute of Advanced Industrial Science and Technology (AIST), Japan, kiyo-ishii@aist.go.jp <sup>(2)</sup> CTTC/CERCA, Castelldefels, Spain, ramon.casellas@cttc.es

**Abstract** Co-employment of node level and component level model is essential to achieve both SDN controller scalability and accurate path computation for functional block-based disaggregation networks. A proposed control system is prototyped and optical path provisioning within 30 seconds is demonstrated on an optical network testbed.

## Introduction

In the evolution towards cloud-native photonic networks and beyond 5G, the requirements to optical networks are increasingly demanding especially in terms of flexibility and agility. To realize such future optical networks, disaggregation technologies have been widely studied, including partial disaggregation, full disaggregation, and component level disaggregation<sup>[1]</sup>. The latter considers finer management granularity and imposes a larger number of devices to manage on an SDN controller. Partial disaggregation decouples optical transponders from the line equipment and employs a dedicated open line system (OLS) controller for provisioning network media channels. It has attracted much attention due to advantages such as timely transponder upgrades, clear design of transmission lines, and SDN controller scalability. Yet, component level disaggregation is of increasing importance since having full visibility of the optical components is needed for an accurate path computation reflecting the actual node structures. Recently, a functional block-based disaggregation (FBD)<sup>[2]</sup> model that describes precise network topology including intra-node structures and mathematically defines the routing constraints has been developed.

In order to address both SDN controller scalability issues and to allow for an accurate path computation service, this paper demonstrates an automated optical network control/management system employing two abstraction levels, i.e., partial disaggregation and component level disaggregation. In the developed system, the WDM orchestrator and the OLS controller (OLS-C) operate at a "node" level abstraction, while the path computation function works at the "device" or "cir-



Fig. 1: Example of FBD model-based node topology and its relationship to the TAPI topology context.

cuit pack" level. The path computation function is externalized in a dedicated server, relying on the ONF TAPI model<sup>[3]</sup> with RESTCONF interface between the OLS-C and the path computation server (PCS).

### FBD model and TAPI topology context

In the FBD model, individual optical components are modeled and the optical fiber connections among the ports of the individual components are described. Furthermore, the switching/filtering functionalities (i.e., constraints such as input/output connectivity and spectrum availability) of the individual optical components are mathematically defined in a machine-readable format using ILP formulation methods. By computing the ILP formula, the switching functionalities for whole nodes or networks can be automatically analyzed<sup>[2]</sup>. On the other hand, in the TAPI topology context, the detailed intra-node structures are abstracted and attributes such as links, nodes, node edge points (NEP), and service interface points (SIP) are described. The photonic media model augments e.g., core components with information such as supportable frequency ranges.

Fig. 1 illustrates the schematic mappings between the FBD model (indicated by green and red lines) and TAPI topology context (indicated by orange circles and letters). Since the FBD model provides the precise topology information, the TAPI topology context can be generated from the FBD model-based topology description by extracting the information of nodes, links, NEPs, and SIPs. In addition, the supportable frequency at each NEP or SIP can be calculated with the ILP formula defined in the FBD model. We have developed a mapping algorithm which automatically analyzes an FBD model-based topology description and outputs a TAPI topology context.

# TAPI path computation service based on the FBD model

Fig. 2 shows an overview of the developed PCS based on FBD model which consists of two parts: the swagger-server and the FBD-based calculator module. The former provides a TAPI compatible interface where the server stub is generated by the open-source yang2swagger and necessary functions are implemented. In the latter, an FBD model-based topology description, a calculation model file describing constraint and objective functions, and calculation data files describing variable and set definitions are stored. The model file, data files, and the TAPI topology context file are automatically generated from the FBD model-based topology description file; here, the generation process is performed offline. In addition to the TAPI topology context, the PCS stores detailed resource usage to resolve the network resource contention at the component level.

For a path computation, the server receives a path computation request via TAPI, obtains the SIPs corresponding to the end points, and



Fig. 2: Path computation server and testbed architecture.

maps them to the component port names defined in the FBD model-based topology description. The calculator module adds the end-points to the data files for specifying the path computation source/destination points and then sends the model and data files to the open-source GLPK optimization solver<sup>[4]</sup>. Here, the objective function is simply set to minimize the required network resources at the component level. After receiving the computation results from the GLPK solver, the server returns the computation results to client and updates the resource usage and the TAPI topology context accordingly. Note that this computation service can compute not only the inter-node paths (visible links) but also the intranode detailed connections, i.e., the intra-node contention constraints are fully considered.

The PCS can universally handle any optical components or node structures without software updates since the routing constraints are defined externally in the topology description files at the component level with a general way (i.e., ILP formula) instead of dedicated definitions such as colored or directioned. Though a path computation is formulated in a single optimization problem here, the problem can be divided into several sub problems such as inter-node and intra-node routing, which can enhance the PCS scalability. Furthermore, the ILP formula are described in a standard machine readable format, GNU MathProg modeling language<sup>[4]</sup>, so that the external optimization solver can be easily changed/upgraded. The time of path computation based on the FBD model is discussed in<sup>[2]</sup>.

## **Experimental setup**

The validation of the approach is carried out in the ADRENALINE testbed<sup>[5]</sup>, shown in the Fig. 3. As per Fig. 2, the WDM orchestrator controller (192.168.0.82) follows a hierarchical partial disaggregation and is responsible coordinating the functions of the OLS-C and the Transponders. The PCS (IP address 172.50.0.1) starts and constructs a per-device FBD-based node model of the optical network and abstracts it into a TAPI 2.1 photonic-media model and context. The OLS-C, responsible for setting up Media Channel (MC) connectivity services, has two IP addresses: 172.50.0.254 within the same sub-net as the path computation server in order to retrieve the TAPI context, topology and to request path computations and 192.168.169.100, which is used to connect to the agents running in the nodes (notably



Fig. 3: ADRENALINE Testbed Open Line System (left) and sample Path Computation Reply (right)

node 2 and node 4, with IP addresses 10.1.1.112 and 10.1.1.114 are the agents at the nodes.

The experiment is as follows (within brackets, the corresponding item in Fig. 4): At time 0, [1] the OLS-C starts and requests the TAPI context from the PCS. Since the OLS-C needs to provision the path once computed, the IP addresses and ports of the node agents are included in the topology objects using the "value-name" and "value" mechanism of TAPI to encode additional node attributes. In [19] the PCS replies. At [20], the WDM controller gets the whole context from the OLS and maps transceivers line ports with OLS client ports. At [56], the WDM controller requests a TAPI photonic media channel connectivity service between 2 OLS client ports, using the create-connectivity-service remote procedure call (RPC). The OLS-C [57] retrieves the most up-to-date context (this is done on a per request basis) and requests a path computation service [79]. Approximately 20 seconds later ([84]) the PCS replies with the path (see Fig. 3 for an example encoding). The OLS-C instantiates the crossconnections to nodes 2 and 4, [86, 87]. The RPC reply is sent to the WDM controller orchestrator [94]. As shown, the process takes approximately 30 s, being the path computation the most important factor. Note that this process only involves the OLS domain and does not include the additional time required by the transceivers. A similar process [95-135] is done to release the connectivity service, which takes less time (approx. 7 s) since there is no path computation involved.

### Conclusions

A control architecture co-employing partial and component level disaggregation models was experimentally demonstrated. The experiment

18.8888888888		
190.027513748	172.50.0.1 172.50.0.254	13038 HTTP/1.0 200 OK (application/json)
2011.370799603	192.168.0172.50.0.254	284 GET /restconf/data/tapi-common:context HTTP/1.1
54 11.378387852	172.50.0.2. 172.50.0.254	4113 HTTP/1.1 200 OK (application/json)
56 11.381115582	192.168.0172.50.0.254	1055 POST /restconf/operations/tapi-connectivity:create-connectiv:
57 11.382598549	172.50.0.2. 172.50.0.1	220 GET /data/tapi-common:context/ HTTP/1.1
77 11.403961858	172.50.0.1 172.50.0.254	21694 HTTP/1.0 200 OK (application/json)
7911.426642873	172.50.0.2. 172.50.0.1	723 POST /operations/tapi-path-computation:compute-p-2-p-path/ H
84 31.851096193	172.50.0.1 172.50.0.254	1526 HTTP/1.0 200 OK (application/json)
86 31.890885290	192.168.1610.1.1.114	222 POST /passion/sbi/opticalSwitch/connections HTTP/1.1 (applie
88 31.890104149	192.168.1610.1.1.112	223 POST /passion/sbi/opticalSwitch/connections HTTP/1.1 (applie
90 31.935796894	10.1.1.112 192.168.169.100	217 HTTP/1.0 200 OK (application/json)
92 31.935872852	10.1.1.114 192.168.169.100	217 HTTP/1.0 200 OK (application/json)
94 31.937780355	172.50.0.2. 172.50.0.254	1124 HTTP/1.1 200 OK (application/json)
95 43.325416679	192.168.0172.58.0.254	284 GET /restconf/data/tapi-common:context HTTP/1.1
121 43.332975512	172.58.8.2. 172.58.8.254	32398 HTTP/1.1 200 OK (application/json)
123 43.335584815	192.168.0172.50.0.254	165 POST /restconf/operations/tapi-connectivity:delete-connectiv:
126 43.374622389	192.168.1610.1.1.112	143 DELETE /passion/sbi/opticalSwitch/connections HTTP/1.1 (app)
127 43.374627446	192.168.16. 10.1.1.114	143 DELETE /passion/sbi/opticalSwitch/connections HTTP/1.1 (app)
129 43.420818473	10.1.1.112 192.168.169.100	217 HTTP/1.0 200 OK (application/json)
131 43.420135848	10.1.1.114 192.168.169.100	217 HTTP/1.0 200 OK (application/json)
133 43.421502888	172.50.0.2. 172.50.0.254	99 HTTP/1.1 200 OK (application/json)
135 43.422791001	172.50.0.2. 172.50.0.1	147 POST /operations/tapi-path-computation:delete-p-2-p-path/ HT

Fig. 4: Wireshark capture of the control plane workflow including externalized TAPI path computation and service provisioning.

shows the applicability of the TAPI models (i.e., the photonic-media layer) for provisioning and path computation. Results are in line with industry practice and validate the approach: latency and overhead are acceptable, with setup delays in the order of seconds, significantly lower than typical hardware configuration delays. Path computations at the component level remain as concern about computation time for large scale networks, however, the component model described in a machine-readable way will be beneficial to the future disaggregation era; since it can address any node structures and make externalization of the path computation function simpler.

#### Acknowledgements

Work partially supported by the Spanish AURO-RAS (RTI2018-099178-B-I00) project and JSPS KAKENHI (JP19H02164).

#### References

- R. Keys, "Sftware defined open programmable packet optical networking", in OECC/PSC, IEEE, 2019.
- [2] K. Ishii et al., "Optical network resource management supporting physical layer reconfiguration", J. Lightw. Technol., vol. 37, Aug. 2020.
- [3] https://www.opennetworking.org/open-transport/.
- [4] https://www.gnu.org/software/glpk/.
- [5] R. Casellas *et al.*, "An OLS Controller for Hybrid Fixed / Flexi Grid Disaggregated Networks with Open Interfaces", in OFC2020, IEEE/OSA, 2020.