First Demonstration of Automated Updates of Disaggregate Blades in Multi-Domain/Layer Optical Path Network

Kiyo Ishii[1], Sugang Xu[2], Noboru Yoshikane[3], Atsuko Takefusa[4], Shigeyuki Yanagimachi[5], Takeshi Hoshida[6], Kohei Shiomoto[7], Tomohiro Kudoh[8], Takehiro Tsuritani[3], Yoshinari Awaji[2], Shu Namiki[1]

[1]*National Institute of Advanced Industrial Science and Technology (AIST), Japan, kiyo-ishii@aist.go.jp*
[2]*National Institute of Information and Communications Technology (NICT), Japan*
[3]*KDDI Research, [4]National Institute of Informatics, [5]NEC Corporation, [6]Fujitsu Limited, [7]Tokyo City University, [8]The University of Tokyo*

Abstract: Updating an OpenROADM node and subsequent re-routing were automated using a mathematical component-based model, triggered by the addition of node components. This process required only five minutes on an orchestrated testbed using SINET5 and a field optical network. © 2020 The Author(s)

1. Introduction

The technological migration of the network to 5G/Beyond-5G, and 6G will significantly increase its dependency on the optical layer, as ever high bandwidth, low latency, high reliability, and high security will be simultaneously demanded. To meet the various requirements expected to arise in diverse local places, and especially to address issues in case of emergency due to disasters, swift and dynamic reconfigurability of the network topology, even at the optical layer, must be ensured. In other words, optical fibers and components composing various networks must be easily installed, operated, and uninstalled by local operators. Owing to recent efforts on disaggregation in optical transport networks, various compact disaggregate blades (e.g., WSS or EDFA blades) are commercially available, and interoperability among multi-vendor transponders was enthusiastically demonstrated [1]. Based on vendor neutral disaggregate ROADM models, automatic optical path establishments that take physical impairment into account were demonstrated [2]. Such disaggregate ROADM models require a hardware abstraction layer (HAL) that maps between the actual components and the model. To the best of the authors' knowledge, such HALs are developed manually. Therefore, on every installation or uninstallation, operators have to modify the HAL to keep the consistency between the reality and the database in the control/management plane. Such operations including software reset/modification take at least hours or days [3]. These processes not only require high skill but also are prone to human errors. To overcome this problem, there must be a mechanism to bridge them automatically. In other words, for disaggregate ROADM systems consisting of diverse blades, there must be a mechanism to model the individual blades, and automatically generate the entire ROADM level model from the blade level model.

A new, human-agnostic model named functional block-based disaggregation (FBD) was proposed [4]. In this model, the switching functionalities (i.e. the constraints such as spectrum availability and input/output connectivity) of disaggregate blades such as WSSs or optical couplers are mathematically described by the integer linear programming (ILP) method using machine-readable GNU MathProg modeling language [5], and the fiber connections among the blades are specified. By computing the ILP formula, the switching functionalities of the whole nodes/ networks can be analyzed. Likewise, the FBD model providing a 1:1 correspondence between the model and the real hardware composition will obviate HAL for mapping. Moreover, FBD can be employed to model the individual blades at the blade level and form the ROADM level nodal model by aggregating the individual blade models.

Based on the FBD model, this study conducted automated network operations, including optical path establishment/removal and node structure updates with disaggregate ROADMs over a field testbed, for the first time. The node structure update was completed within five minutes without manual configurations. Multi-domain cooperative optical path recovery triggered by the update was also performed.

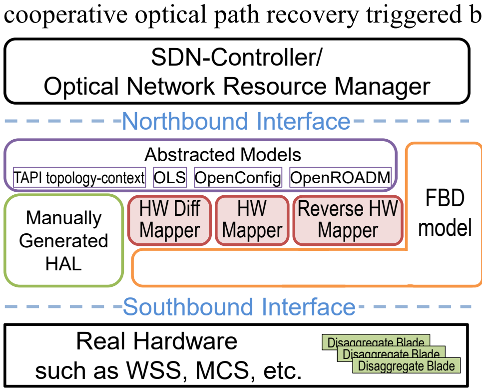

Fig. 1. Hierarchical relationships among the models

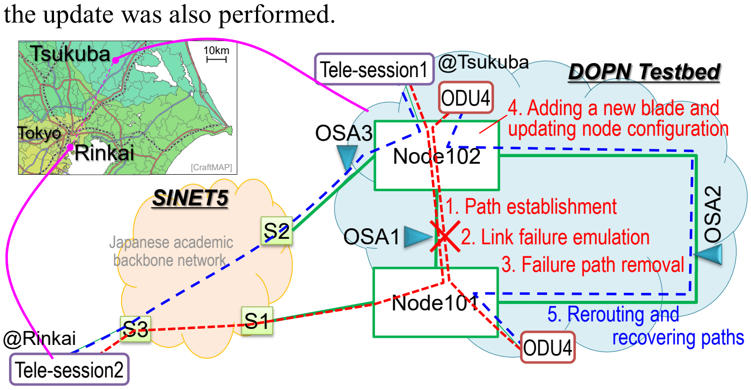

Fig. 2. Testbed architecture and five-step demonstration scenario

2. FBD model

The FBD model allows automatic generation of HAL in a machine-processable manner as illustrated in [4]. The hierarchical relationships among the model layers are shown in Fig. 1. Above the hardware shown at the bottom, the FBD model precisely describes the real hardware composition with minimal abstraction. As the FBD model cannot be directly used with standard SDN controllers such as ONOS or OpenDaylight, we developed three Mappers to bridge the real hardware and the abstracted models via the FBD model. The OpenROADM model [6] was employed as a typical ROADM-based abstracted model in this paper. HW Mapper translates an FBD model description onto an OpenROADM model description, as demonstrated in [7]. Reverse HW Mapper and HW Diff Mapper were newly developed. Reverse HW Mapper translates OpenROADM-based optical-path-configuration RPC commands into FBD model-based commands specifying individual blade configuration. HW Diff Mapper automatically compares two FBD topology descriptions (i.e. the ones before and after system changes), extracts the differences, and generates OpenROADM-based *edit-config* RPC commands to reflect changes in the physical layer topology in the databases of the control plane. The three Mappers can automatically generate OpenROADM device descriptions or RPC commands that used to be manually generated, thus significantly reducing the burden and errors in device model processing.

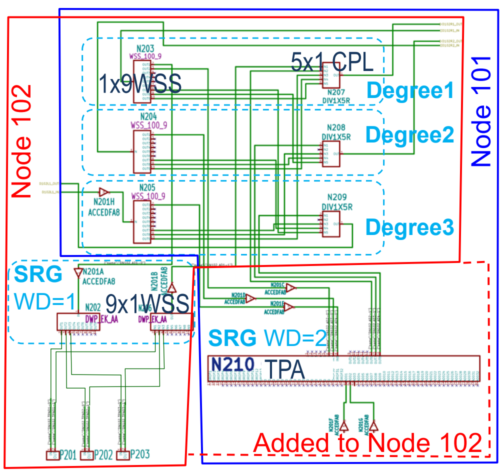

Fig. 3. Detailed node architecture

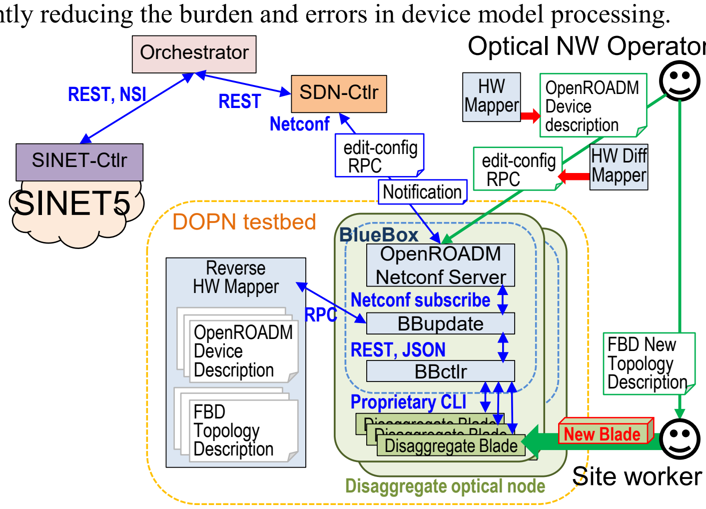

Fig. 4. Implemented control-plane architecture

3. Testbed architecture, demonstration scenario, and experimental results

Fig. 2 shows the testbed configuration and demonstration scenarios. The testbed consisted of two network domains: the dynamic optical path network (DOPN) testbed and SINET5. The DOPN testbed is an optical network domain consisting of two disaggregate ROADM nodes. SINET5 is a Japanese academic backbone network used by more than 800 universities and research institutes owned by NII [8]. In SINET5, layer-2 VPN paths can be established with an L2 on-demand service via the network service interface (NSI) API [9]. In other words, SINET5 dynamically manages L2 paths and the DOPN testbed manages L1/L0 optical paths. The links between SINET5 and the DOPN testbed are field optical fibers, and ports S1 – S3 are the L2 ports equipped with media converters. The demonstration performed the five steps shown in Fig. 2: **1.** establishing two paths (red broken lines), namely a 100-Gbps ODU4 path and a 10-Gbps 4K uncompressed tele-session (i.e. an application) path; **2.** link failure emulation between nodes 101 and 102; **3.** removal of failure-affected paths; **4.** adding a blade to node 102; and **5.** path recovery with multi-domain cooperation (blue broken lines). The two tele-session terminals were located more than 60 km apart via SINET5. Fig. 3 shows the detailed architecture of nodes 101 and 102 consisting of multi-vendor disaggregate blades or modules. Node 101 was a CDC ROADM consisting of WSSs, optical couplers, EDFAs, and a transponder aggregator (TPA) [10], as indicated in blue. First, node 102 was a colorless but directional and contention ROADM node, as indicated in red. Then, at the 4^{th} step, a TPA was added forming a CDC feature to reroute the failure-affected paths as indicated in red broken line. The optical functional blocks (dark red rectangles) and the optical fiber connections (dark green lines) were specified in the FBD model. The node functionalities such as degree and SRG were automatically analyzed by HW Mapper.

Fig. 4 shows the developed control plane. The orchestrator was interconnected with the SDN controller (Ctlr) and the SINET-Ctlr via the REST APIs. It managed paths crossing the two domains. SDN-Ctlr was interconnected with the intermediate controller [11] named BlueBox, which aggregated the disaggregate blades. SDN-Ctlr automatically loaded OpenROADM device descriptions from BlueBox via Netconf API (c.f., *get-config* RPC) and controlled the DOPN testbed. It also performed automatic control of the testbed according to the five-step demonstration scenario, including normal and failure situations, in cooperation with the orchestrator. BlueBox consisted of three software blocks: BBctlr interpreted the configuration commands from the FBD model-based commands into the blade native proprietary ones; Netconf Server, which was developed with the ConfD software [12], was connected to SDN-Ctlr via Netconf API; BBupdate interconnected BBctlr and Netconf Server. In the experiments, the OpenROADM device description generated by the HW Mapper was first loaded into Netconf Server.

To establish an optical path, SDN-Ctlr sent an *edit-config* RPC to BlueBox to add a *roadm-connection* entry (i.e. in OpenROADM device model) in Netconf Server. BBUpdate recognized the update with the subscribe function provided by ConfD and sent the updated entry to Reverse HW Mapper, which computed the intra-node path based on the OpenROADM device description and the FBD topology description, and then responded with the necessary configurations of individual blades to BBUpdate. Next, BBUpdate sent the configurations to BBCtlr, which then sent the blade native commands to the blades. For the node update at the 4th step, a site worker added a new blade to the node firstly. Given that the FBD topology description of the hardware composition was comprehensive, it could be directly used as an instruction document. Then, the optical network operator sent *edit-config* RPCs to Netconf Server to reflect the change of the hardware composition onto the database. The RPC commands were automatically generated by HW Diff Mapper. BBUpdate recognized the change through the subscribe function and then sent the topology change notification to SDN-Ctlr as well as to BBctlr and Reverse HW Mapper. Because BlueBox, FBD model, and the three developed Mappers could wrap the real hardware composition, SDN-Ctlr could in turn control the disaggregate blades in the DOPN testbed as standard OpenROADM nodes without manual engineering.

Fig. 5 shows the measured optical spectrogram at OSA1 between the 1st and 3rd steps of the demonstration scenario and at OSA2 and OSA3 between the 4th and 5th steps. The link-up states were confirmed with LED indicators for the ODU4 paths and video images on the display for the tele-session paths. Fig. 5 indicates that the optical paths were successfully established and removed according to the demonstration scenario. Fig. 6 shows the captured messages in the control plane. The 1st step of the demonstration scenario is depicted in (I), the 2nd and 3rd steps are shown in (II), and the 4th and 5th are displayed in (III), confirming that the OpenROADM-based commands and orchestration between the two network domains were appropriately performed. In this experiment, it took only five minutes to upgrade node 102 to install a disaggregate blade, connecting optical fiber cords, and reflecting the update in Netconf Server; it was performed during the time between Fig. 6 (II) and (III).

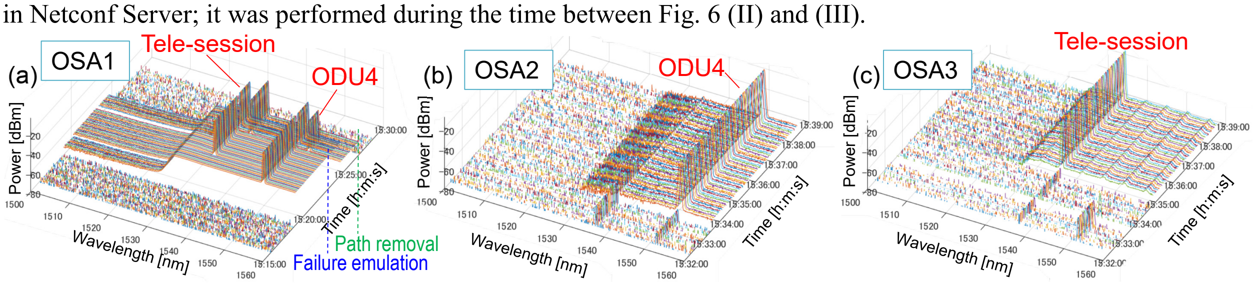

Fig. 5. Measured optical spectrogram

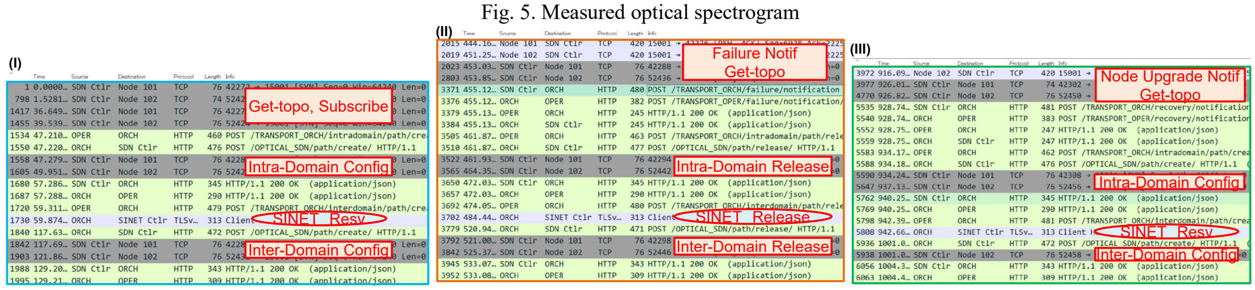

Fig. 6. Captured messages for network configuration and release

4. Conclusion

This study demonstrated automated operations of full disaggregated optical networks for the first time; where disaggregate blades were wrapped as OpenROADM nodes by BlueBox and three Mappers. The node update was completed within five minutes. These demonstrations ensure that the significant expansion of the automated operation ranges from path establishment to physical topology updates. The enhanced automation will deskill network operations free from human errors, leading to the grass-roots dissemination of optical networking technologies.

Acknowledgement: This work was partly supported by JSPS KAKENHI JP19H02164. The authors thank Toshiyuki Shimizu for his technical assistance.

[1] M. Filer, et al., JOCN, vol. 10, no. 2, 2018.
[2] S. Oda, et al., JLT, vol. 35, no. 8, 2017.
[3] Y. Uematsu et al., IEICE, (2017): 2016EBN0013
[4] K. Ishii, et al., JLT vol. 37, no. 21, 2019.
[5] https://www.gnu.org/software/glpk/
[6] http://openroadm.org/
[7] K. Ishii, et al., in Proc OFC2020, M3Z.4
[8] https://www.sinet.ad.jp/en/aboutsinet-en
[9] G. Roberts, et al., GFD-R-P.212, OGF2014.
[10] S. Nakamura, et al., JSTQE, vol. 22, no. 6, 2016.
[11] IEC TR 62343-6-10:2017(E)
[12] https://www.tail-f.com/confd-basic/