Automatic Resource Mapping Using Functional Block Based Disaggregation Model for ROADM Networks

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: Automated mapping of real hardware composition onto a ROADM-based model is demonstrated. The functional-block-based model precisely describing the physical layer structures can act as a hardware abstraction layer for more abstracted models like OpenROADM. © 2020 The Author(s)

1. Introduction

Optical network technologies are becoming increasingly important for future information services not only in conventional core/metro/access networks but also in inter-/intra-DC and the 5G mobile Xhaul networks [1, 2]. To cover the various requirements of such diverse networks efficiently, disaggregation and automation technologies are extensively studied [3]. Disaggregation approaches decouple transport equipment, sub-systems, and optical devices, so that high applicability can be achieved by customization or optimization of hardware and node structure assemblies. In the disaggregation context, vendor neutral models have been developed in working groups such as OpenROADM and OpenConfig [4-6]. These models regard a ROADM node as the unit of disaggregation in the network architecture. However, the real hardware of ROADM nodes is composed of various optical functional blocks such as wavelength selective switches (WSSs) and multicast switches (MCSs). Therefore, for true automation, hardware abstraction layers (HAL) that map the real hardware resources onto the ROADM-based model description are required. To the best of our knowledge, automatic HAL generation has not been proposed so far and the generation of HAL requires manual procedures to maintain consistency, since the ROADM-based models are more abstracted so as to interface with software-defined network controllers (SDN-C).

A model that can describe the individual functional blocks and the detailed intra-node structures has been proposed in [7]. Hereinafter, we call it the Functional Block based Disaggregation (FBD) model. In the FBD model, each real hardware has a one-to-one correspondence with the FBD model description, and therefore no additional information is required for mapping. Furthermore, the FBD model can be directly utilized for path computation because various switching functionalities of individual optical components are represented in a common mathematical format using integer linear programming (ILP) formulas. Using the FBD model, optical network management for multi-granular hierarchical and heterogeneous nodes has been demonstrated, and open-source-based test systems for drawing the topology, generation of XML-encoded topology description, and optical path computations have been reported [7].

This demonstration shows that the FBD model includes sufficient information to generate a ROADM-based model description. In addition, a software solution for the automatic mapping of an FBD-based real hardware composition description onto a ROADM-based model description is shown. In this study, the OpenROADM device model is employed as a typical ROADM-based model.

2. Innovation and Demonstration Overview

Fig. 1 (left) illustrates the relationships among models. The real hardware is shown at the bottom. Above the hardware, the FBD model precisely describes the real hardware composition without abstraction. Though the FBD model can directly interface with a proprietary optical network resource manager as demonstrated in [7], the FBD model cannot be directly used with standard SDN-C such as ONOS or OpenDaylight. To generate descriptions for the existing abstracted models such as OpenROADM or OpenConfig, we developed an automatic hardware mapper software, which we call "HW Mapper". It maps (translates) an FBD model description on to an OpenROADM model description. With this HW Mapper, we can achieve automated generation of OpenROADM device model descriptions, and thereby significantly reduce the burden and mistakes on device model processing, which had to be performed manually.

Fig. 1 (right) shows a detailed optical node structure, which is drawn with green and red lines, generated by the previously developed topology data generation system for the FBD model. The node consists of optical dividers, WSSs, EDFAs, a transponder aggregator, and terminal devices. For example, just by glancing Fig. 1 (right), optical network experts can recognize the entire node functionality as indicated in blue. That is, the optical node has three degrees and two add/drop groups (cf. SRG in the OpenROADM device model), where one has colorless, directionless, and contentionless (C/D/C) add/drop capability, and the other has directioned, colorless, and contentioned add/drop restrictions (cf. in the OpenROADM model, the add/drop contention constraint is defined using the wavelength duplication attribute). Our HW Mapper can automatically analyze the node switching capability with the FBD model and then translate it to the OpenROADM device model, in the same way as experts can. To achieve automated node topology analysis, optimization or constraint satisfaction problems with ILP formulations are utilized.

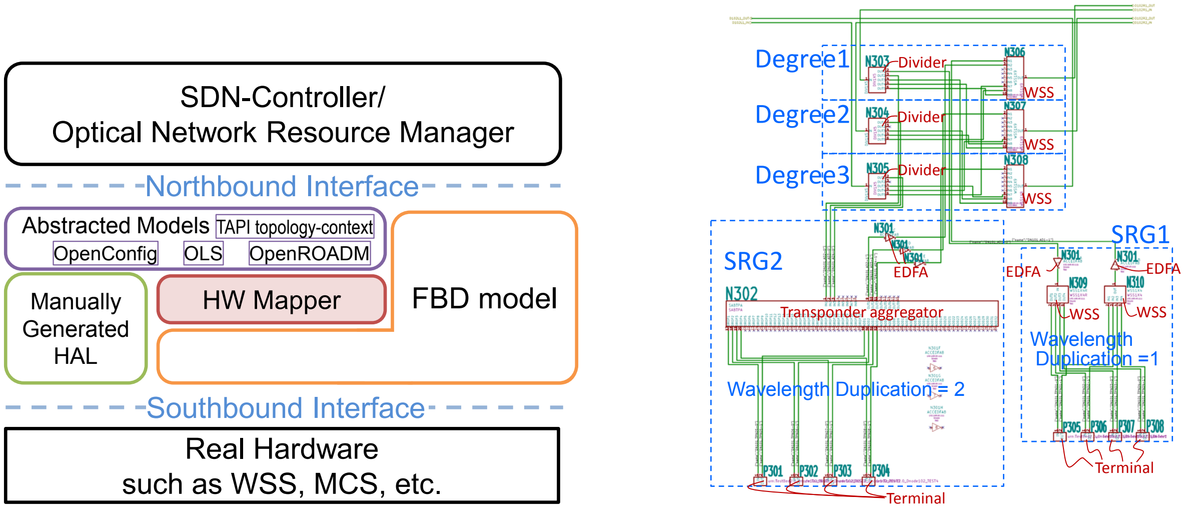
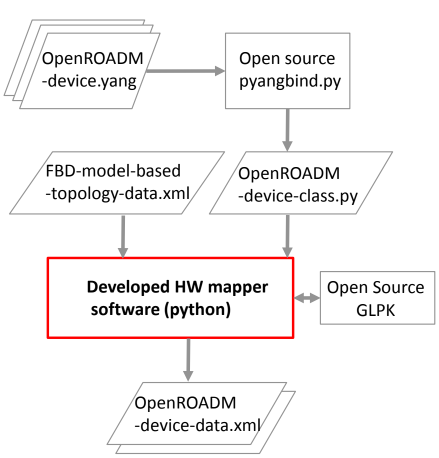


Fig. 1. Relationships among models (left), and an example of FBD model based node topology data drawn with KiCAD [10] (right).

Fig. 2 (left) shows an overview of the developed HW Mapper. First, with PyangBind [11], which is an open-source software for generating a Python class hierarchy from a YANG data model, the OpenROADM device YANG data model is converted into an OpenROADM device class in Python. Our HW Mapper first imports the OpenROADM device class and loads the FBD model-based topology description encoded in XML. Then, the HW Mapper analyzes the topology data using an open-source optimization engine named GLPK [8]. Then, based on the analysis results, the topology data are appended to the OpenROADM device class and exported as an OpenROADM device model encoded in XML. Fig. 2 (right) shows an excerpt from the analysis results for the node configuration shown in Fig. 1 (right). Line 11 indicates that the HW Mapper recognizes the node has three degrees, where each degree consists of the components N303 & N306, N304 & N307, and N305 & N308, respectively. Lines 2-4 show that the node has two add/drop groups where one add/drop group is C/D/C as in lines 5-6 and the other add/drop group has contention restriction as in lines 7-8. In addition, the HW Mapper generates logical connection point names in accordance with the OpenROADM device model, as shown in line 16. A connection map defining the connectivity among the degree ports and transponder ports is also generated (line 18-19). The connection map also indicates the directioned restriction for an add/drop group (i.e., SRG1 in Fig. 1 (right)).




```
1  /Dnode102/ has 6 degrees:N304_1,N303_1,N305_1,N306_1,N308_1,N307_1,
2  /Dnode102/ has 2 SRGs.
3  The SRG1 includes the following components:P304, P302, N302, P303, N301, P301,
4  The SRG2 includes the following components:N310, P307, N301, P306, P308, N309, P305,
5  SRG group (('P302_2', 'P302_1'), ('P301_2', 'P301_1'), ('P303_2', 'P303_1'), ('P304_2', 'P304_1')):
6        the wavelength-duplication value is 2
7  SRG group (('P307_2', 'P307_1'), ('P306_2', 'P306_1'), ('P305_2', 'P305_1'), ('P308_2', 'P308_1')):
8        the wavelength-duplication value is 1
9  Now gathering the topology information for transforming to OpenROADM device model.TPNDtoS
10 ...RDMextPorts['N304_1', 'N303_1', 'N305_1', 'N308_1', 'N306_1', 'N307_1', 'N310_4', 'N309_4',
11 ...idxDegree{1: ['N303', 'N306'], 2: ['N304', 'N307'], 3: ['N305', 'N308']}
12 ...DgrIntrnlPorts['N304_1', 'N308_1', 'N304_1', 'N306_1', 'N304_1', 'N307_1', 'N303_1', 'N308_1',
13 ...DgrExtrnlPorts['N304_1', 'N303_1', 'N305_1', 'N308_1', 'N306_1', 'N307_1']
14 ...SrgIntrnlPorts['N301_4', 'N301_2', 'N301_8', 'N302_20', 'N301_6', 'N302_20', 'N301_10', 'N302
15 ...SrgExtrnlPorts['N310_4', 'N309_4', 'N310_3', 'N309_3', 'N310_2', 'N309_2', 'N310_5', 'N309_5',
16 ...LCP{'N304_1': 'DEG2-TTP-RX', 'N308_1': 'DEG3-TTP-TX', 'N306_1': 'DEG1-TTP-TX', 'N307_1': 'D
17 ...
18 Connection Map is as follows
19 {'N304_1': ['N308_1', 'N306_1', 'N307_1', 'P302_1', 'P301_1', 'P303_1', 'P304_1'], 'N303_1': ['N30
```
snip

Fig. 2. Overview of the developed HW Mapper software (left), and an excerpt from the analysis results for the node configuration in Fig. 1 (right)

The flow of the developed topology analysis algorithm is shown in Table I. In the FBD model, the optical switching functionalities are defined in the ILP formulas described in the machine-readable GNU MathProg modeling language [8]. Therefore, any port to port routes such as inter-/intra-node connections or possible connections within individual components can be automatically computed as ILP constraint satisfaction problems (steps 2, 6, and 8). In addition, route restrictions that depend on whether multiple optical paths can be added/dropped from/to the same add/drop group with the same wavelength channel can also be systematically computed (step 10). The degree ports and the transponder ports can be recognized as follows (step 5). The port under consideration is one of the degree ports if the port at the other end of the link belongs to another node. If it belongs to a terminal device, the port is one of the transponder ports. Detailed algorithms of steps 7 and 9 in Table I are shown in Fig. 3. The algorithms in Fig. 3 are for the input degree port sides; the output degree port sides can also be computed in the same manner.

Live demonstrations of the HW Mapper, FBD model-based topology data files, exported OpenROADM model-based data files, and topology data generation for multiple node configurations will be presented at the demonstration floor.

Table I. Algorithm flow of topology analysis

Step	Description	Method
1	Load FBD based topology data file	
2	Compute internal connections for individual components	Use ILP formulas in the topology file and GLPK solver
3	Classify the intra- and inter- fiber links	Based on the link end point (port) locale
4	Recognize the nodes to which each component belongs	Based on the attribute parameters defined in the topology file
5	Recognize the degree ports and the transponder ports	
6	Compute the intra-node connectivity among the degree ports	Use ILP formulas in the topology file and GLPK solver
7	Recognize the components belonging to which degree	See Fig. 3(b)
8	Computer the intra-node connectivity among the degree ports and the transponder ports	Use ILP formulas in the topology file and GLPK solver
9	Recognize the components belonging to which add/drop (SRG) groups	See. Fig. 3(c)
10	Compute the contention restrictions of the add/drop routes for individual add/drop (SRG) groups	Use ILP formulas in the topology file and GLPK solver

(a)

Symbol	Description
D_{in}	A set of input degree ports
D_{out}	A set of output degree ports
T	A set of transponder ports
$C(i,j)$	A set of components included in a route from the port i to j
$P(c,i,j)$	A set of ports included in a route from the port i to j and in the component c

(b) **for d in D_{in}:** $s[d] = \bigcap_{o \in D_{out}} C(d,o)$

$s[d]$ indicates the components configuring the degree d

(c) **for t in T:** $x[t] = \bigcap_{d \in D_{in}} C(d,t)$

Transponders with ports $t, t' \in T$ are in the same add/drop group if and only if

$$\left\{\bigcup_{d \in D_{in}, c \in x[t]} P(c,d,t)\right\} \cap \left\{\bigcup_{d \in D_{in}, c \in x[t']} P(c,d,t')\right\} \neq \phi$$

Fig. 3. Detailed algorithm for steps 7 and 9 in Table I. (a) notation, (b) detailed algorithms for step 7 and (c) for step 9.

3. Relevance and Conclusion

The demonstration is particularly relevant to OFC delegates including operators, vendors, and academia interested in optical network automation. Our developed HW Mapper can automatically map the detailed hardware composition onto ROADM-based models (i.e., OpenROADM device model in this demonstration) via FBD model. The developed HW Mapper can be used for automated operations, e.g., node topology analysis, model generation at the time of node installation or upgrade. The tools for FBD model generation, HW Mapper, and OpenROADM model generation were developed on an open-source basis. By slightly modifying the developed algorithm, the FBD model can generate mappings between the real hardware composition and any other abstracted models since the FBD model contains a complete set of information that can reproduce the switching functionalities and node configurations. In other words, the FBD model is detailed enough to serve as a universal platform for more abstracted, ROADM-based disaggregated models such as OpenROADM.

Acknowledgement: This work was partially supported by JSPS KAKENHI Grant Number 19H02164.

[1] E. Breverman et al., "Data models for optical devices in data center operator networks," OFC2019.
[2] A. F. Beldachi, et al., "A novel programmable disaggregated edge node supporting heterogeneous 5G access technologies," ECOC2019.
[3] D. Kilper, "Optical disaggregation and the role of SDN in converging mobile, access and cloud networks," tutorial, ECOC2019.
[4] http://openroadm.org/home.html
[5] http://www.openconfig.net/
[6] S. Oda et al., "A learning living network with Open ROADMs," JLT, vol. 36, no. 8, 2017.
[7] K. Ishii et al., "Optical network resource management supporting physical layer reconfiguration," JLT, vol. 37, no. 21, 2019.
[8] https://www.gnu.org/software/glpk/
[9] https://wiki.onosproject.org/display/ONOS/ONOS
[10] http://kicad-pcb.org/
[11] https://github.com/robshakir/pyangbind