POLITECNICO DI TORINO
Repository ISTITUZIONALE

Open Line Controller Architecture in Partially Disaggregated Optical Networks

Original

Open Line Controller Architecture in Partially Disaggregated Optical Networks / Ambrosone, Renato; Borraccini, Giacomo; D'Amico, Andrea; Straullu, Stefano; Aquilino, Francesco; Breuer, Dirk; Schatzmayr, Rainer; Grammel, Gert; Curri, Vittorio. - ELETTRONICO. - (2023), pp. 1-3. (2023 International Conference on Photonics in Switching and Computing (PSC) Mantova, Italy 26-29 September 2023) [10.1109/PSC57974.2023.10297237].

Availability:
This version is available at: 11583/2983884 since: 2023-11-20T17:36:07Z

Publisher:
IEEE

Published
DOI:10.1109/PSC57974.2023.10297237

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

07 May 2026

Open Line Controller Architecture in Partially Disaggregated Optical Networks

1st Renato Ambrosone
Politecnico di Torino, Italy
renato.ambrosone@polito.it

2nd Giacomo Borraccini
Politecnico di Torino, Italy

3rd Andrea D'Amico
Politecnico di Torino, Italy

4th Stefano Straullu
LINKS Foundation, Italy

5th Francesco Aquilino
LINKS Foundation, Italy

6th Dirk Breuer
Deutsche Telekom AG, Germany

7th Rainer Schatzmayr
Deutsche Telekom AG, Germany

8th Gert Grammel
Juniper Networks Inc., California USA

9th Vittorio Curri
Politecnico di Torino, Italy

***Abstract*—A line controller architecture for partially disaggregated optical networks is presented, providing automation and interoperability in the use of multi-vendor equipment. Downstream of the architectural definition, the results obtained in an implementation operated on experimental equipment are shown regarding the functioning of the designed interfaces.**

I. INTRODUCTION

The ever-increasing demand for high-speed and reliable data transfer has resulted in a surge in network traffic on optical networks. Cloud computing, video streaming, and IoT devices created exponential increases in bandwidth usage demands that traditional network infrastructure was not ready to handle [15]. In this context, disaggregated optical networks and software-defined networking (SDN) [2, 14] have been identified as promising solutions to address these challenges and optimize the efficiency and scalability of optical networks. Disaggregated optical networks involve the separation of optical components into modular and interchangeable schemes, allowing network operators to combine multi-vendor equipment to meet specific requirements and minimize costs [6]. On the other hand, SDN involves separating the optical control plane from the data plane, allowing for more centralized control and management of network data traffic. The combination of these two approaches provides a powerful solution for addressing the challenges posed by the increasing network traffic demand, necessarily requiring the introduction of some degrees of openness. Consequently, many consortia have been created to allow operators to achieve this goal through the achievement of widely shared agreements [13]. For instance, OpenConfig [8] and Open ROADM MSA [7] proposed several Yet Another Next Generation (YANG) models to enable the control of different network elements (NEs).

However, actual implementations leverage partially disaggregated network models [9], especially in the case of optical line systems (OLSs). Following this approach, portions of a given network can be managed by a single controller in charge of managing a set of NEs through south bound interfaces (SBI). The mentioned controller also interfaces with the central SD network controller via north bound interface (NBI). The authors demonstrated the operation of an open and partially disaggregated optical network architecture, emphasizing on the orchestration of the system relying on the use of the physical layer (PHY) digital twin (DT) for the implementation of the optical control plane and lightpath computation engine [4]. Starting from the latter, this work focuses on the optical line controller (OLC) module, showing the potential of an open architecture in terms of telemetry management and capability of configuring optical amplifiers in a modular and interoperable approach, leaving the possibility of promptly updating or integrating the functioning of the system both in terms of software and hardware. All these are shown in the light of the proposed implementation on a laboratory setup demonstrating the effectiveness of the designed architecture. Furthermore, a possible implementation for an open-OLC is shown and proved via an experimental demonstration.

II. OPTICAL LINE CONTROLLER ARCHITECTURE

This work follows the network architecture proposed in [3]. In particular, the optical nodes are supposed to be controlled in a centralized manner via the SD network controller, whereas OLCs handle all the OLS components, i.e. booster (BST), pre-amplifier (PRE) and optical in-line amplifiers (ILAs). As described in Fig. 1, the arrangement of SDN and partial disaggregation enables the operation of the optical infrastructure in multi-vendor scenarios [1]. Thanks to this combination, each line can be composed by closed single vendor solution having proprietary-code implementation for the SBIs (solution A in Fig. 1). However, following the partially disaggregated network model, all the OLCs should be able to retrieve telemetry information [11] and configure the controlled NEs via common interfaces, suggesting the possibility of an open OLS management architecture. In this perspective, closed solutions do not allow a prompt scalability (e.g. update to the use of multi-band systems) and integration (e.g. introduction of new telemetry devices or sensors) of the infrastructure.

On the other hand, an open OLC can be developed following two different approaches. The first one is the most ecstatic solution exploiting standard data models for the amplifiers and leading to achieve a whole compatibility and compliance between the OLC and NEs. This solution requires vendors sharing common models, resulting in a current non-availability of any product considering the case of optical amplifiers.

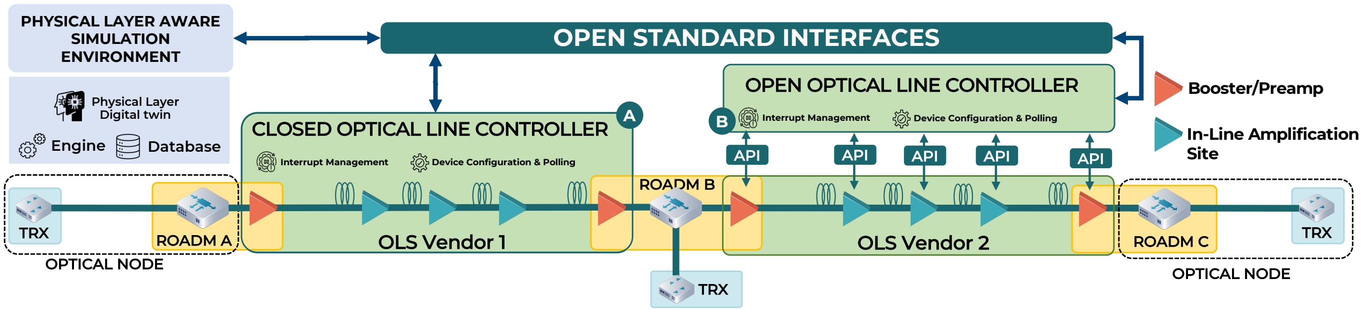


Fig. 1. Example of partially disaggregated optical network composed of two OLSs. Solution A describes a closed optical line controller with proprietary access to devices. Solution B shows optical line controller exploiting open SBIs.

For this reason, a second approach is followed in this work, involving the development of a framework implementing both NBI and SBI. The developed open OLC exploits NBI for communication with physical layer aware simulation environment (PLASE) [4]. In particular, the proposed controller architecture homogeneously manages all NEs with four main functions, which are: establishing/closing a connection between the controller and the specific NE, and get/set operations on the status and NE configuration. For operational reasons, the only significant distinction is to classify *amplifiers* and *telemetry instruments*. Consequently, every instance recognized by the controller needs an SBI implemented by a specific driver (solution B in Fig. 1). This solution is decidedly more verbose in terms of amount of needed code and does not ensure full support for possible hardware functions, but it makes possible the interaction with the amplifiers exploiting their low-level interfaces.

From an operational point of view to perform the operations of the optical control plane, the OLC goes through an automatic fine-tuning procedure after power up and installation. This process starts by creating a stable connection with all the NEs. Then, the framework aims to acquire telemetry information from the field in order to probe the PHY. This information is then forwarded to the PLASE that can collect and store data in a database and apply the required optimization strategy exploiting an updated model of the PHY-DT.

III. EXPERIMENTAL SETUP

The testbed setup is depicted in Fig. 2. The setup is composed of a single OLS of 7 commercial optical ILAs connected through 6 standard single mode fiber (SSFM) spans with a nominal length of 65 km each. At Node A, a C-band wavelength division multiplexing (WDM) comb is generated with a 193.75 THz center, 96 channels separated at 50 GHz, and each modulated at 32 GBd. Four modulated channels, centered at 192, 193, 194, and 195 THz, respectively, are produced using CFP2-ACO/DCO coherent pluggable transceivers offered by Lumentum and plugged into two Cassini AS7716-24SC white-boxes, used as source and destination points. The other 92 channels are generated by shaping the output of an amplified spontaneous emission (ASE) noise source using a commercial waveshaper filter. The multiplexer (MUX) filter of a commercial wavelength selective switching (WSS) is used at the Node A to combine the ASE channels with the four modulated channel under test (CUTs), achieving a full C-band spectral load; similarly, a demultiplexer (DMX) filter is used at Node B to separate each CUT before its detection. Thanks to the optical spectrum analyzer (OSA) placed at the output of the MUX and the DMX filters, the WDM spectra shown in Fig. 2 as been captured. Each amplifier can be controlled via SSH, enabling the development of custom drivers for the OLC. Likewise, Cassini white-boxes, via OcNOS operating system from IP Infusion, offers a NETCONF interface, which allows to obtain telemetry data and to set the CUTs working parameters (i.e. center frequency, modulation format, transmitted power, etc.).

IV. CONTROLLER IMPLEMENTATION & RESULTS

The proposed OLC architecture is implemented as a Python software framework. As previously discussed, the chosen model requires the development of drivers to translate high-level calls into SSH commands. However, the software has a modular design so that it supports effortless extensions to different vendor equipment with different protocols and data structures, requiring only the implementation of compatible driver modules. The communication between the OLC and the PLASE is achieved through representational state transfer (REST) interfaces, while data durability is achieved using MongoDB [10], a No-SQL database.

Exploiting the described optical equipment setup, an experimental proof-of-concept for the presented OLC architec-

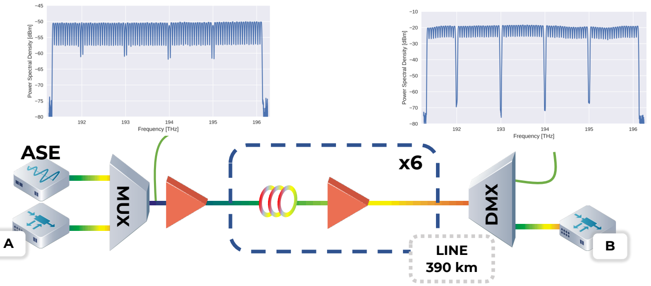
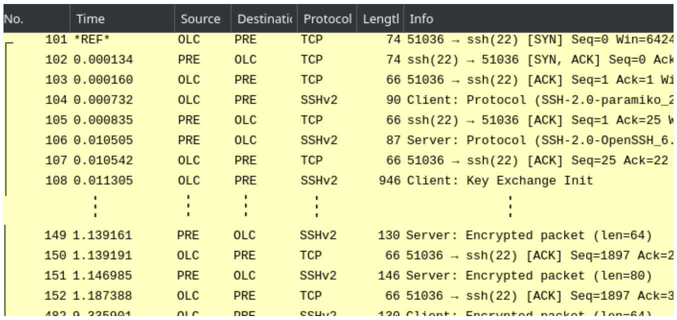
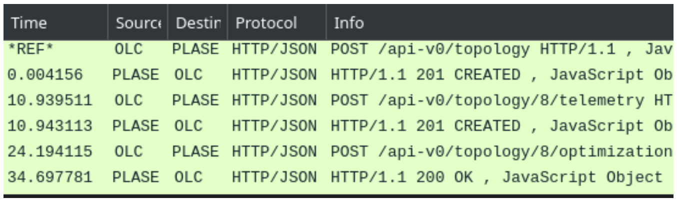


Fig. 2. Sketch for the optical laboratory setup reproducing an OLS composed of multi-vendor equipment.



```
No.   Time      Source Destinati Protocol Lengtl Info
  101 *REF*     OLC    PRE       TCP          74 51036 → ssh(22) [SYN] Seq=0 Win=6424
  102 0.000134  PRE    OLC       TCP          74 ssh(22) → 51036 [SYN, ACK] Seq=0 Ack
  103 0.000160  OLC    PRE       TCP          66 51036 → ssh(22) [ACK] Seq=1 Ack=1 Wi
  104 0.000732  OLC    PRE       SSHv2        90 Client: Protocol (SSH-2.0-paramiko_2
  105 0.000835  PRE    OLC       TCP          66 ssh(22) → 51036 [ACK] Seq=1 Ack=25 W
  106 0.010505  PRE    OLC       SSHv2        87 Server: Protocol (SSH-2.0-OpenSSH_6.
  107 0.010542  OLC    PRE       TCP          66 51036 → ssh(22) [ACK] Seq=25 Ack=22
  108 0.011305  OLC    PRE       SSHv2       946 Client: Key Exchange Init
        ⋮         ⋮      ⋮         ⋮                         ⋮
  149 1.139161  PRE    OLC       SSHv2       130 Server: Encrypted packet (len=64)
  150 1.139191  OLC    PRE       TCP          66 51036 → ssh(22) [ACK] Seq=1897 Ack=2
  151 1.146985  PRE    OLC       SSHv2       146 Server: Encrypted packet (len=80)
  152 1.187388  OLC    PRE       TCP          66 51036 → ssh(22) [ACK] Seq=1897 Ack=3
```

Fig. 3. Portion of messages exchanged during connection establishment between OLC and an amplifier.



```
Time       Source Destin Protocol  Info
*REF*      OLC    PLASE  HTTP/JSON POST /api-v0/topology HTTP/1.1 , Jav
0.004156   PLASE  OLC    HTTP/JSON HTTP/1.1 201 CREATED , JavaScript Ob
10.939511  OLC    PLASE  HTTP/JSON POST /api-v0/topology/8/telemetry HT
10.943113  PLASE  OLC    HTTP/JSON HTTP/1.1 201 CREATED , JavaScript Ob
24.194115  OLC    PLASE  HTTP/JSON POST /api-v0/topology/8/optimization
34.697781  PLASE  OLC    HTTP/JSON HTTP/1.1 200 OK , JavaScript Object
```

Fig. 4. Set of REST requests exchanged between OLC and PLASE.

ture has been conducted. At the OLC boot phase, a list of ILAs with their credentials and network references are kept within a database. The sequence of messages exchanged by the controller and the PRE are shown in Fig. 3. The total lapse time to establish a connection between the OLC and the PRE is approximately 1.2 seconds. Since this operation can be parallelized and the number of controlled ILAs can hardly exceed the number of threads of a modern processor, this operation can be considered with complexity of $O(1)$. After the boot is completed, the OLC shares the topology description with the PLASE, saving them into the database and returning to the OLC an unique identifier which can be used for future requests (first couple of requests in Fig. 4). Then, the controller starts a probing procedure in order to retrieve telemetry data needed by PLASE (second couple of request in Fig. 4) to properly feed the PHY model [12]. It is worth noting that this operation can also be implemented using a watchdog periodically obtaining the telemetry and sending it to the PLASE. This allows the engine to be able to exploit a telemetry history instead of a snapshot by perfecting the use of different quality of transmission estimator (QoT-E) strategies.

The last step consists in the request by the OLC to the PLASE to obtain the optimal working points for the amplifiers. Within the body of this request it is possible to specify the type of optimization requested; if not specified, the PLASE uses its own default strategy. For this experiment GNPy is used as QoT-E [5]. Once obtained the working point in terms of target gain and tilt parameters, the OLC accordingly configures each ILA.

V. CONCLUSIONS

In this work, a modular architecture for implementing an optical line controller following a partially disaggregated optical network model is proposed. An experiment is also conducted, demonstrating the feasibility in controlling optical amplifiers and paving the way for the creation of an open-source line controller. Its potential has been shown in a scenario where vendor support is absent. The experiment demonstrated how it is possible to send a telemetry to the PLASE in order to optimize the working point of the amplifiers.

VI. ACKNOWLEDGEMENTS

This work has been supported by the TIP and by the EU Horizon Europe research and innovation program, ALLEGRO Project, GA No. 101092766.

REFERENCES

[1] Jean-Luc Auge, Vittorio Curri, and Esther Le Rouzic. “Open design for multi-vendor optical networks”. In: *OFC*. Optica Publishing Group. 2019.

[2] Kamal Benzekki, Abdeslam El Fergougui, and Abdelbaki Elbelrhiti Elalaoui. “Software-defined networking (SDN): a survey”. In: *Security and communication networks* 9.18 (2016), pp. 5803–5833.

[3] Giacomo Borraccini et al. “Disaggregated Optical Network Orchestration based on the Physical Layer Digital Twin”. In: *OFC*. Optica Publishing Group. 2023.

[4] Giacomo Borraccini et al. “QoT-driven optical control and data plane in multi-vendor disaggregated networks”. In: *OFC*. Optica Publishing Group. 2022.

[5] Vittorio Curri. “GNPy model of the physical layer for open and disaggregated optical networking”. In: *Journal of optical communications and networking* 14.6 (2022), pp. C92–C104.

[6] Vittorio Curri. “Software-defined WDM optical transport in disaggregated open optical networks”. In: *2020 ICTON*. IEEE. 2020, pp. 1–4.

[7] *http://www.openroadm.org*.

[8] *https://www.openconfig.net*.

[9] Esther Le Rouzic et al. “Operationalizing partially disaggregated optical networks: an open standards-driven multi-vendor demonstration”. In: *OFC*. OSA. 2021.

[10] DB Mongo. *Mongodb*. 2015.

[11] Francesco Paolucci et al. “Network telemetry streaming services in SDN-based disaggregated optical networks”. In: *Journal of Lightwave Technology* 36.15 (2018), pp. 3142–3149.

[12] N Sambo et al. “Probe-based schemes to guarantee lightpath quality of transmission (QoT) in transparent optical networks”. In: *2008 34th European Conference on Optical Communication*. IEEE. 2008, pp. 1–2.

[13] Andrea Sgambelluri et al. “OpenConfig and OpenROADM automation of operational modes in disaggregated optical networks”. In: *IEEE Access* 8 (2020), pp. 190094–190107.

[14] Akhilesh S Thyagaturu et al. “Software defined optical networks (SDONs): A comprehensive survey”. In: *IEEE Communications Surveys & Tutorials* 18.4 (2016), pp. 2738–2786.

[15] Keyao Zhu and Biswanath Mukherjee. “A review of traffic grooming in WDM optical networks: Architectures and challenges”. In: *Optical Networks Magazine* 4.2 (2003), pp. 55–64.