JOURNAL OF
Optical Communications
and Networking

Open and disaggregated optical transport networks for data center interconnects [Invited]

Chongjin Xie,[1,*] Lei Wang,[2] Liang Dou,[2] Ming Xia,[1] Sai Chen,[3] Huan Zhang,[3] Zhao Sun,[2] and Jingchi Cheng[3]

[1]Alibaba Cloud, Alibaba Group, Sunnyvale, California 94085, USA
[2]Alibaba Cloud, Alibaba Group, Beijing, China
[3]Alibaba Cloud, Alibaba Group, Hangzhou, China
*Corresponding author: Chongjin.xie@alibaba-inc.com

Received 17 October 2019; revised 28 January 2020; accepted 1 February 2020; published 2 March 2020 (Doc. ID 380721)

In this paper, we present Alibaba's practices and views on open and disaggregated optical transport networks, with a focus on metro data center interconnects (DCIs). We first discuss technology developments that have enabled open optical transport networks, followed by a presentation of our open optical network architecture for DCI networks. We then describe an overall data center network architecture and the application scenario of open and disaggregated DCI technology. Details of the point-to-point DCI network architecture and the need for reconfigurable optical add–drop multiplexers (ROADMs) in DCI networks are then discussed. We present the data models of DCI equipment, including optical terminal transponders, point-to-point optical line systems, and ROADMs, developed based on OpenConfig YANG models. The detailed design and implementation of a home-grown control and management software platform for DCI networks are also presented. We conclude with a discussion of the real deployment of the technology in Alibaba's networks.

https://doi.org/10.1364/JOCN.380721

1. INTRODUCTION

Due to the analog nature of optical signals, optical transport systems and networks have historically been closed and proprietary systems. In addition to the hardware [including optical transponders, optical amplifiers, optical switches, wavelength multiplexers/demultiplexers, wavelength selective switches (WSSs), and gain equalizers] being tightly coupled together, the control and management software has been tightly bound with the hardware as well. Although there have been numerous efforts to open up optical transport networks and move away from closed proprietary systems (e.g., the black link model supporting alien wavelengths [1]), all of the optical subsystems still needed to be co-designed and optimized to get the best system performance, resulting in little progress toward open optical systems in the past.

Recently, however, open and disaggregated optical networks have regained the interest of the industry. Numerous activities have been undertaken by academia, standard bodies, consortia, and network operators [2–9]. Wide deployment of open and disaggregated optical networks in data center interconnect (DCI) networks has begun by some hyperscale data center operators; further, it is expected that the adoption of the technology by other network operators will soon follow. The shift of the optical transport network paradigm to openness and disaggregation is mainly due to the following reasons: 1) The advent of digital coherent technology, which not only increases the spectral efficiency and receiver sensitivity but also significantly simplifies the design of optical communication systems. This simplification enables the decoupling of optical line systems from optical terminal transponders. 2) Current fiber deployments are nearing their nonlinear-Shannon-limit capacity [10]. Before we implement new technology to make the optical fiber "pipes" much larger, an alternative approach to effectively increase the capacity of an optical transport network is to increase network efficiency. This can be better achieved with open and disaggregated networks, as network operators can effectively control and reconfigure their networks and have end-to-end optimization of their networks according to their needs. 3) The development of software defined network (SDN) technology [11]. The practices of SDN in the past decade and its successful deployment in real networks have given network operators experience with, and confidence in, open network technologies. 4) The large demand of DCI networks. DCI networks are different from telecom networks. They grow much faster, require much larger bandwidths, and have a much shorter technology cycle. Thus, they can benefit more from an open and disaggregated architecture, which enables network operators to adopt new technologies much faster and more easily. Furthermore, DCI network architecture is simpler, and

there is little legacy equipment in the networks, which makes it easier to deploy open and disaggregated technology.

Due to the rapid growth of businesses such as e-commerce, cloud computing, and e-entertainment, Alibaba data center networks grow very fast, with DCI traffic doubling almost every year. To cope with fast-growing traffic and fast-changing businesses, Alibaba has been embracing the idea of open and disaggregated optical transport networks since 2016. We have developed open and disaggregated optical transport systems for metro DCI networks with an internally developed control and management software platform, which has been widely deployed in our networks. In this paper, we present our practices on open and disaggregated optical transport technology for metro DCI networks.

The paper is organized as follows. Section 2 describes the architecture of open optical transport networks. In Section 3, we discuss DCI network architectures, including point-to-point systems and mesh topologies that use reconfigurable optical add–drop multiplexers (ROADMs). Data models are presented in Section 4, including models for optical terminal transponders and optical line systems. The design and implementation of our internal control and management software platform are presented in Section 5. Section 6 discusses the real deployment of the technology in our networks. Section 7 summarizes the paper.

2. OPEN OPTICAL TRANSPORT NETWORK ARCHITECTURE

Figure 1 shows different disaggregation levels of an optical transport system. At one extreme is a conventional closed single-vendor system, where the whole system, including optical line equipment, terminal equipment, and associated control and management software, is from a single vendor. At the other extreme is an open module system, where each individual building block, including optical amplifiers, WSSs, and optical transponders, can come from different vendors, and network operators build their own systems based on these modules. An open module system is also called a white box system [6].

Between these two extremes is the open line system architecture. This is currently the most studied and widely deployed open and disaggregated optical transport system. (It is sometimes referred to as a partially disaggregated system.) In an open line system, the optical line system is from one vendor and accepts "foreign" or "alien" wavelengths; the control software, whether it is home-grown or from a third party, can manage equipment from different vendors. Different wavelengths on the same line system can be from different vendors, but an open line system does not require transponder interoperability (i.e., it does not require support for transponders from different vendors/generations at the two endpoints of a connection). Furthermore, it does not require ROADM node interoperability.

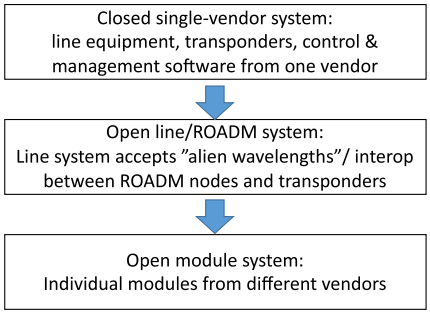

Fig. 1. Different disaggregation levels of optical transport systems.

Taking disaggregation one step further, an open ROADM system requires both ROADM node and transponder interoperability between vendors [5]. Strictly speaking, an open ROADM system has to follow the specifications defined by a standard such as the OpenROADM Multi-Source Agreement (MSA) [5]. However, more generally, when one upgrades an optical line system to enable the interoperability of ROADM nodes from different vendors and uses interoperable transponders (such as 400G ZR transponders that are compliant with OIF specifications [12]), the open line system can also be called an open ROADM system.

This discussion of open line versus open ROADM systems is further illustrated by the open and disaggregated optical transport system shown in Fig. 2. The system is divided into two decoupled parts. One part is composed of open, standardized, and modular hardware; the other part is a unified cloud-based software platform. The hardware includes the optical line system [optical WDM multiplexers/demultiplexers (OMDs), optical amplifiers (OAs), optical switches, ROADMs, etc.], and the terminal transponders. If all of the hardware equipment of the line system is provided by a single vendor, it is simply termed an "open line system." In contrast, the OpenROADM MSA specifies interfaces for some of the hardware so that equipment from different vendors is allowed and can interoperate in one optical line system. There is also some effort to allow the interoperability of terminal transponders from different vendors, such as the aforementioned 400G ZR OIF initiative.

The disaggregation model to use will partially depend on the operator's deployment scenario. For operators who care more about optical reach and performance, e.g., in long-haul transport systems, they may accept a model with proprietary terminal transponders (i.e., no interoperability) but an

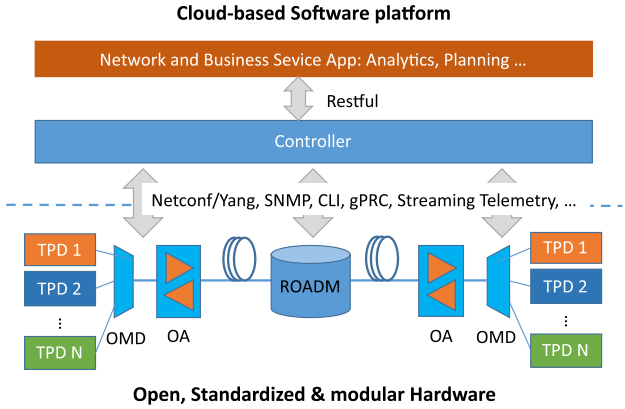

Fig. 2. Architecture of an open and disaggregated optical transport system. TPD, transponder; OMD, optical multiplexer and demultiplexer, OA, optical amplifier.

open line system, where operators have the freedom to use transponders from different vendors at different wavelengths.

In our open metro DCI networks, which currently are not ROADM-based, we utilize the open line system model. With the new development of businesses and data center/network architecture, we are considering the deployment of ROADMs in our networks. This involves the migration of an open line system to an open ROADM system, which is a subject for a separate paper and is not discussed here.

Hardware equipment is typically implemented as standalone compact boxes with open northbound interfaces that include the data model and the protocols. Industry groups and standard bodies have made significant efforts to standardize the data models and protocols. The YANG models [13] and NETCONF protocol [14] have become a widely accepted northbound interface standard on open compact optical transport boxes. Most equipment is also required to support legacy simple network management protocol (SNMP) and command line interface (CLI) protocols. Most performance monitoring (PM) of hardware equipment is based on SNMP [15], which is designed for legacy implementations; it demonstrates poor scaling for today's high-density platforms and limited extensibility. Streaming telemetry is a new approach for network monitoring in which data are streamed from devices continuously with efficient, incremental updates [16].

In modern systems, the controller provides control and management capability of hardware equipment and the overall network, including configuration, performance monitoring, and alarm management. Equipment abstraction is provided to the controller through YANG models so that the controller can discover equipment capabilities and functions. The configuration of the equipment is done through NETCONF or other protocols. Performance monitoring can be done through NETCONF or streaming telemetry, and alarm notification is done either through SNMP or streaming telemetry. The controller can also construct the network topology model through the discovery of equipment functions and their connections. The controller provides its capabilities to the upper-layer business and service applications such as planning and data analysis, through restful application programming interfaces (APIs).

3. DATA CENTER INTERCONNECT NETWORKS

A typical data center network is depicted in Fig. 3, which includes intra-data-center networks with link distances typically less than 2 km; metro networks whose distances are generally less than 80 km; wide-area networks, which can span up to approximately thousands of kilometers for national and global networks; and access networks that are used to connect data centers to points of presence, and service points (SPs), so that end users can be connected to data center networks with low latency. DCI in general refers to inter-data-center interconnects. In this paper, we focus on DCI in metro networks, which have much shorter distances (~ 80 km) and much higher capacity than telecom metro networks, due to the latency and bandwidth requirements of synchronous duplication of data among data centers in metro areas.

Due to the latency and bandwidth requirements, most data centers in a metro area are connected with point-to-point links. Figure 4 is a typical point-to-point metro DCI architecture. In most cases, it is a one-span system, with optical booster amplifiers (BAs) at the transmitter and optical preamplifiers (PAs) at the receiver. For a system with a lossy link, an inline optical amplifier may be used. To increase the reliability of the system, $1+1$ optical multiplex section protection (OMSP) is usually implemented, where the signal from the multiplexer at the transmitter is split into two parts by a 3 dB coupler, and a switch at the receiver side selects the signal from the better path. [This is illustrated by the automatic protection switch (APS) shown in the figure.] OMSP can protect against both fiber cuts and optical amplifier failures.

The number of fibers required for a point-to-point architecture is proportional to the square of the number of data centers (i.e., for a fully meshed network with direct fiber link connectivity between N data centers, $N(N-1)/2$ protected pairs of fibers are needed). Studies show that, when there are more than four data centers in a metro area, the total cost of ownership of a ROADM-based mesh DCI architecture is lower than that of a point-to-point architecture [17]. This is typically the case in metropolitan areas where a metro network consists of many small data centers (e.g., due to some applications requiring data centers to be close to users and limited spaces for large data centers in metropolitan areas as well). With a small number of ROADMs in the network, the number of fibers

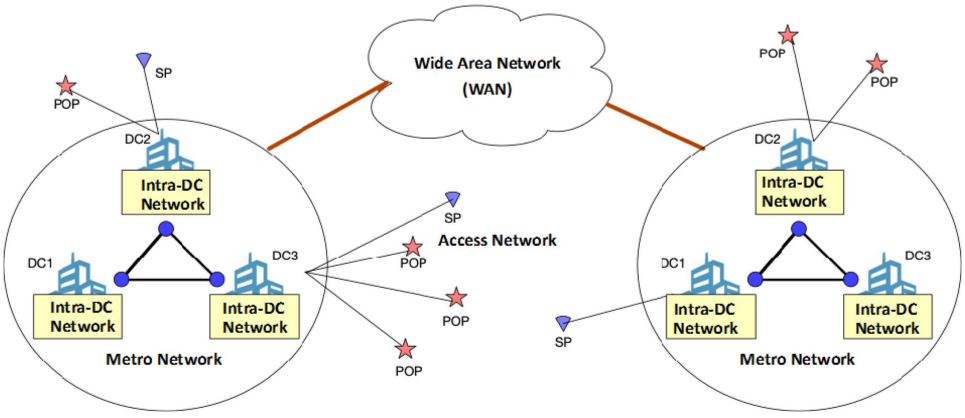

Fig. 3. Architecture of a data center network. DC, data center; POP, point of presence; SP, service point.

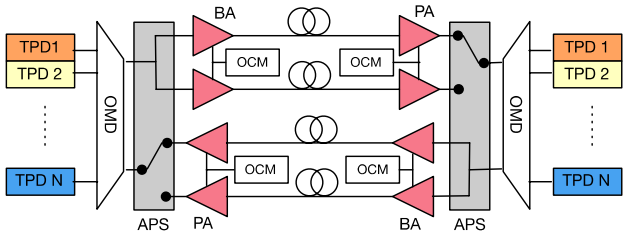

Fig. 4. Point-to-point metro DCI architecture. APS, automatic protection switch; BA, booster amplifier; PA, preamplifier; OCM, optical channel monitor; TPD, transponder; OMD, optical multiplexer and demultiplexer.

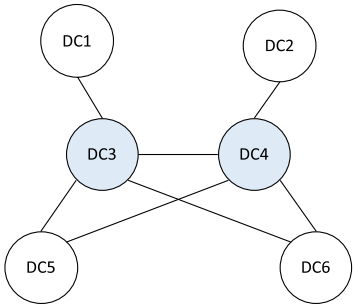

Fig. 5. ROADM architecture DCI network. ROADMs are located in DC3 and DC4. DC: data center.

can be significantly reduced, with little increase in latency, as shown in Fig. 5. With ROADMs located in data centers DC3 and DC4, seven fiber links are used to provide full-mesh connectivity between all six data centers, whereas 15 fiber links are required when a point-to-point architecture is used.

We currently use point-to-point architecture in our metro networks. However, we are considering deploying ROADMs in our networks in the coming years. Lots of effort has been taken to make ROADMs compatible with open line systems [18].

Although studies on direct-detection technologies for metro DCI applications are numerous, today's DCI predominantly implements coherent detection technology due to its high spectral efficiency, high sensitivity, less stringent requirements on optical line systems, and simplicity of operation. The last two characteristics are especially important for open and disaggregated DCI networks. In addition, coherent detection provides programmable capability to transponders, so that the bit rates of coherent transponders can be programmed according to network requirements, which provide additional flexibility to networks.

4. DATA MODELS

In multivendor environments, it is a big challenge to configure a network programmatically if data models vary from device to device. Vendors may argue that they need proprietary data models to address their own special capabilities. However, from an operator's perspective, it is preferable to manage and control network devices in a unified manner, regardless of the vendor, in order to simplify device adaptation, system integration, network operation, and automation.

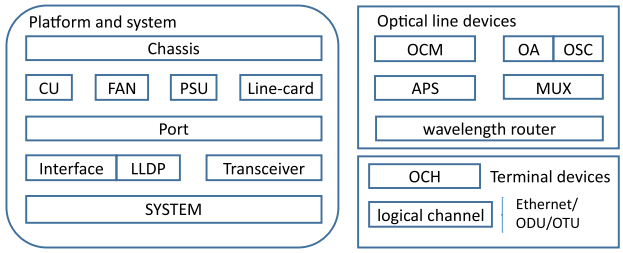

Fig. 6. Optical transport system device data model architecture. CU, control unit; PSU, power supply unit; LLDP, link layer discovery protocol; OSC, optical supervisory channel; OCH, optical channel; ODU, optical data unit; OUT, optical transport unit.

At least two organizations are working on standardization of data models for optical networks, OpenROADM and OpenConfig. OpenROADM is an MSA that defines interoperability specifications for ROADMs as well as transponders and pluggable optics. Specifications consist of both optical interfaces as well as data models [5]. OpenConfig is an informal working group of network operators with the goal of developing dynamic and programmable network infrastructure based on software-defined networking principles [4]. OpenConfig allows vendors and operators to add extensions to the existing data models to promote innovation, as long as they do not make incompatible changes or add extensions to the models that violate the core model aspects.

As OpenConfig models are more mature than those of OpenROADM (OpenConfig models have already been deployed in real networks) and simpler, we believe they are more suitable for today's DCI networks. Therefore, we have adopted OpenConfig data models as southbound device interfaces on our own open and disaggregated optical transport systems for metro DCI networks. In our systems, OpenConfig models are modularized into three main groups: platform/system module, terminal device module, and optical line device module; further, the three groups can be combined into different types of optical transport devices, as shown in Fig. 6. Some device specification data, such as device size, typical/maximum power, line-side pre-FEC (forward error correction), bit error ratio (BER) threshold, and amplifier noise figure, are not included in OpenConfig, although they are important for optical transport network planning and management. Thus, we introduced device/module profiles into our system to supplement and improve upon OpenConfig models, to better support device abstraction functions. Note that the specification data structures in the profiles are vendor agnostic, with only specification data values, which are provided in a vendor's equipment datasheets. The values are provided for different vendors/platforms/card types. The overheads of the profiles are acceptable.

A. Platform and System Module

As shown in Fig. 6, the OpenConfig platform module defines a flat data model for device component inventory management. In our system, it mainly contains the chassis, line-cards, fans, power supply units, controller units, ports and transceivers; each of them is modeled as a component, with corresponding

configuration, operational state, and PM data. In addition, the hierarchical relationship between components is represented by the component references in the subcomponent and parent fields, so that one can select proper platform components and build an arbitrary device structure as needed.

The OpenConfig system module is used for managing system-wide services and functions in network devices, such as network time protocol server configuration, log service, alarm management, and so on. Most importantly, the platform module and the system module employ common data models, which can be used for abstracting and modeling any network device. This enables various types of devices to have a common structure with respect to component inventory and system services/functions, which significantly reduce network automation development and operation cost.

B. Optical Transport Terminal Device Module

As shown in Fig. 6, the OpenConfig transport terminal device module defines the optical channel (OCH) component and logical channel unit for managing terminal transponders in an optical transport network.

The OCH is augmented into the platform module as a special component, which corresponds to a line-side channel with an optical carrier of assigned wavelength/frequency. An OCH has several configuration parameters, including channel frequency, target output power, operational mode (channel baud rate, FEC mode, modulation format, data rate), and line-side physical port reference (mapping an OCH to a line-side physical port). In addition, the OCH model also provides operational states and performance monitors, such as input power, output power, chromatic dispersion (CD), polarization mode dispersion (PMD), and polarization dependent loss (PDL).

A logical channel is a group of logical grooming elements that may be assigned to subsequent stages for multiplexing/demultiplexing or to an optical channel for line transmission. There are three types of logical channels, Ethernet, optical data unit (ODU), and optical transport unit (OTU), and each type has its own PMs. For example, an Ethernet logical channel has Ethernet layer counters and physical coding sublayer (PCS) error counters, an ODU/OTU logical channel has frame/block error counters, and an OTU logical channel also has pre-FEC BER and FEC-related counters. Each logical channel has an assignment container, which contains the logical-channel/optical-channel reference in the next stage. In this way, a logical channel chain can be built to represent the multiplexing and mapping structure from client-side port to line-side port. Although data center traffic is predominantly Ethernet, some optical transport network (OTN) layer functions are still useful in data center transport networks; as such, OTN is widely used in DCI networks. For example, loopback functions can help isolate link failure points, overhead "DM" can provide L1 delay measurement for fiber links, "GCC" can be used for remote device management for sites without data control network (DCN) availability, and OTN can provide layer 1 data encryption.

Figure 7 is an example of an OpenConfig terminal transponder channel model. The leftmost component is a client-side transceiver. It contains several physical channels, and each channel corresponds to a physical lane in the transceiver, which provides lane-level configuration and PM capability. When an Ethernet logical channel is bound on a client-side transceiver and corresponding physical port, it is labeled as an Ethernet port on the client side and provides Ethernet-related configuration and PM. Ethernet packets may be encapsulated into an optical payload unit (OPU)/ODU frame (such as Ethernet to OPU4/ODU4), indicating the mapping/multiplexing relationship through the channel reference in the assignment container. The other mapping/multiplexing stages (such as ODU4 to ODUc2, ODUc2 to OTUc2, and OTUc2 to OCH) are similar and represented by a channel chain. The right-most component in the figure is a line-side transceiver, with a physical port and corresponding OCH. Note that Fig. 7 is only an example, and OpenConfig does not assume a particular mapping/multiplexing structure on a terminal transponder. The channel model details depend on the transponder itself and the requirements of the network operators.

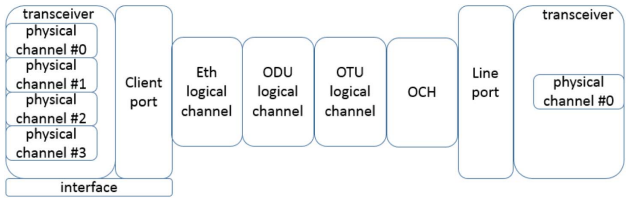

Fig. 7. Example of a terminal transponder channel model.

C. Point-to-Point Optical Line Device Module

As shown in Fig. 6, the optical line device module defines all of the devices in a point-to-point system, including the OA, optical supervisory channel (OSC), OCM, APS, and optical multiplexer (MUX). Each optical component has its own configuration and operational state data. OpenConfig disaggregates a whole optical transport line system into basic optical components and defines connections between them, conforming to the target of an open and disaggregated optical transport system.

As shown in Fig. 4, the OMD provides wavelength multiplexing/demultiplexing between terminal devices and the optical line system. In the forward direction, the multiplexed optical signal passes through an APS module for bidirectional $1+1$ protection and is split into two parts. In the reverse direction, one of the incoming signals from the two different paths is selected and passed to the OMD in the APS. The APS module has many configuration parameters to enable user-defined protection switch strategies; the parameters include switch mode, switch threshold and hysteresis, threshold offset, and switch hold-off time. The APS module also provides an operational state such as the active path to represent the switch working state and PM in each port. Each OA module exposes its basic capabilities to adapt to various application scenarios, including module enable/disable, amplifier mode, input power range, target gain, target gain tilt, target output power, and also operational state and PM. The OCM module is simple and

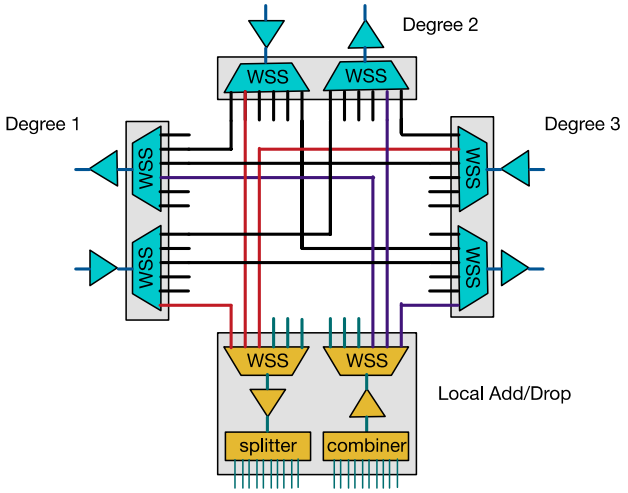

Fig. 8. Typical structure of a colorless and directionless ROADM.

contains channel power spectrum density at each monitored port.

D. ROADM

Figure 8 shows a typical structure of a colorless and directionless ROADM. With a directionless ROADM, any wavelength from any degree can be routed to any other degree, including local add–drop ports, by software control, which makes an optical network programmable and reconfigurable. OpenConfig defines a ROADM as an optical-wavelength-router; in its YANG model, a collection of media channels is used to specify a frequency band with lower and upper frequency, which is compatible with flexgrid operation. For each media channel, both source and destination ports are configurable. Thereby, it can describe the wavelength routing in a ROADM node in principle. However, there are limitations of this model for real implementation. As shown in Fig. 8, the WSS is an important component in a ROADM, which in general is deployed as a card in a ROADM. However, the OpenConfig optical-wavelength-router data model does not describe all attributes of a WSS card as a component, e.g., component name, port numbers, optical performance on each port, and control mode for each media channel. In addition, the OpenConfig model does not include all of the key properties of a ROADM in terms of a node. For our own network, we enhanced the OpenConfig optical-wavelength-router YANG model to support both WSS components and ROADM node controllers.

1. Data Model of a WSS

The general WSS data model includes two parts, one is for the component itself, and the other is for the media-channel management. The description of the WSS component itself is similar to that of other component models, including inventory information, such as card name, chassis name, port name, the number of input ports and output ports, operation wavelength range, and basic state information of each port such as optical power.

```
+--rw media-channels
 +--rw media-channel* [index]
   +--rw index   -> ./config/index
   +--rw config
   | +--rw index?        uint32
   | +--rw name?         string
   | +--rw lower-frequency?  oc-opt-types:frequency-type
   | +--rw upper-frequency?  oc-opt-types:frequency-type
   | +--rw admin-status?    oc-opt-types:admin-state-type
   | +--rw source
   | | +--rw port-name?  -> /oc-platform:components/component/name
   | +--rw dest
   | | +--rw port-name?  -> /oc-platform:components/component/name
   | +--rw frequency-slots
   |  +--rw frequency-slot* [lower-frequency upper-frequency]
   |    +--rw lower-frequency     oc-opt-types:frequency-type
   |    +--rw upper-frequency     oc-opt-types:frequency-type
   |    +--rw control-mode?       identityref
   |    +--rw attenuation-value?  decimal64
   |    +--rw target-power-value?  decimal64
   +--ro state
    +--ro index?        uint32
    +--ro name?         string
    +--ro lower-frequency?  oc-opt-types:frequency-type
    +--ro upper-frequency?  oc-opt-types:frequency-type
    +--ro admin-status?    oc-opt-types:admin-state-type
    +--ro source
    | +--ro port-name?  -> /oc-platform:components/component/name
    +--ro dest
    | +--ro port-name?  -> /oc-platform:components/component/name
    +--ro oper-status?    identityref
    +--ro channel-power?   decimal64
    +--ro frequency-slots
     +--ro frequency-slot* [lower-frequency upper-frequency]
       +--ro lower-frequency     oc-opt-types:frequency-type
       +--ro upper-frequency     oc-opt-types:frequency-type
       +--ro control-mode?       identityref
       +--ro attenuation-value?  decimal64
       +--ro target-power-value?  decimal64
       +--ro actual-power-value?  decimal64
```

Fig. 9. YANG model of a WSS component (media channel related part).

Figure 9 shows one part of the YANG model we designed for the media channel in a WSS. There are two main modifications. One is to support a frequency slot with a finer granularity inside a media channel; this is useful for both network media channel operation and optical domain equalization of a media channel, such as in-band frequency equalization to reduce cascaded filtering effects. The other one is focused on the control mode of each frequency slot. "*ATTENUATION-CONTROL*" mode means directly adjusting the attenuation value of a WSS. "*POWER-CONTROL*" mode is realized with a feedback loop, and the expected output power of this frequency slot is configured by the target-power value. In the state container, the actual-power value shows the current power of this frequency slot, usually obtained from an optical channel monitor.

2. Data Model of a ROADM

A complete ROADM is composed of two types of blocks, as shown in Fig. 8. The first part is called "degree," which is connected to the degree of another ROADM on one side and the internal structure on the other side. The second is called "add/drop groups," allowing the adding and dropping of wavelengths between fiber and client ports. A complete ROADM model includes the description of the ROADM node, its degrees, media channels, and connections. In Fig. 10, we provide one portion of our ROADM YANG model, which models connections among degrees within a ROADM, including the connections between two different WSS tributary ports. In this

```
+--rw connections
  | +--rw connection* [source-name destination-name]
  |  +--rw source-name      -> ../config/source/degree-name
  |  +--rw destination-name  -> ../config/destination/degree-name
  |  +--rw config
  |  | +--rw source
  |  | | +--rw degree-name?  -> /wavelength-router/degrees/degree/config/name
  |  | | +--rw port-name?    -> /oc-platform:components/component/name
  |  | +--rw destination
  |  |  +--rw degree-name?  -> /wavelength-router/degrees/degree/config/name
  |  |  +--rw port-name?    -> /oc-platform:components/component/name
  |  +--rw state
  |   +--rw source
  |    | +--rw degree-name?  -> /wavelength-router/degrees/degree/config/name
  |    | +--rw port-name?    -> /oc-platform:components/component/name
  |    +--rw destination
  |      +--rw degree-name?  -> /wavelength-router/degrees/degree/config/name
  |      +--rw port-name?    -> /oc-platform:components/component/name
```

Fig. 10. YANG model of a ROADM node (connections).

model, we first map the degree with a chassis and a component; then, we use the degree and port to show the connections. Each connection in the model is indexed by the source and destination degrees. The add–drop groups can be represented as one or several local degrees, so the connection between the express degree and local degree is described in the same way as shown in Fig. 10.

5. CONTROL AND MANAGEMENT PLATFORM

A control and management system is an integral part of a network system. Prior to the emergence of open DCI, a DCI optical network could consist of equipment from multiple vendors with proprietary control and management systems. Furthermore, a network with equipment from different vendors typically had multiple vendor-specific control and management systems, and a network engineer had to log in to each vendor's system to manage different parts of their network. The nonunified control and management systems resulted in the inability to have global views and unified operation of the whole network. In addition, these control and management systems usually have different features and operating flows, which further reduce operational efficiency.

A unified control and management platform for an open DCI network provides an abstraction over DCI devices to manage their standard capability, regardless of vendor, implementation, southbound protocol, and vendor-specific features. For instance, an optical terminal device is abstracted as an entity backed up by an OpenConfig-compatible data model, regardless of how its internal data are structured or which protocol it utilizes. This abstraction builds up the foundation for unified management over open DCI equipment from different vendors. On top of this abstraction, network manageability is exposed through APIs to compose high-level, vendor-agnostic business logic and network applications.

Alibaba developed an in-house control and management suite called Transport Software Defined Networking (TSDN) to manage its metro open DCI optical networks. TSDN is built on top of several open-source frameworks. Our stack includes Spring, ExpressJS, Netty, Pouch (equivalent to Docker), Junit, etc. We initially used OpenDaylight (ODL) [19] for translation between device-northbound protocol to RESTCONF but later implemented the agent layer ourselves to replace ODL due to the simplicity of DCI networks and the uniqueness of our operational scenarios. Decoupled from the development of these middleware functionalities, our main effort is focused on implementation of DCI control and management business logic, such as construction, inventory, device/network control and monitoring, and data processing. We designed and engineered TSDN from day one with a number of considerations.

A. Simplicity

TSDN was not born with the role to serve as a conventional OTN network control and management platform. In contrast, it leverages the southbound abstraction to manage DCI equipment following OpenConfig models. Our simplified network architecture and deployment scenario further reduce the complexity of TSDN implementation. For example, as all of the cross-connect requirements in our networks are on the wavelength level, and there is no need for finer granularity cross-connects, we drop the feature to configure electronic cross-connects in terminal devices and start from a simple scenario with only point-to-point connections, which helps us to design a clean work flow for network management. This flow largely has remained the same until the present time, with potential to allow more complex scenarios in the future.

B. Modularization

TSDN models DCI equipment as a composition of multiple functionally decoupled components, where a component is defined by an interface to regulate its behavior and provides a set of closely related functionalities chainable by other components. A component can be a port, a line card, an optical module, or an amplifier. Components can also form a tree structure where the root component represents DCI equipment. This design shares similarity with the OpenConfig hierarchy but is more coarse-grained with northbound-friendly operations. The common components act as LEGO-like blocks, which assists us in building various types of open DCI equipment such as optical terminal transponders, optical amplifiers, and ROADMs.

C. Open API

We believe a complex network task can be decomposed into a number of fine-grained operations. Therefore, a major role of TSDN is to serve tasks by exposing a set of well-defined APIs. TSDN's API is categorized into several groups: resource—support CRUD operations (creating, reading, updating, deleting) for resources at different levels, such as city, data center, network element, device; config—support getting real-time state from, and setting configuration to, actual devices; topology—support getting, filtering, and transforming topology at the physical or OTN layers, as well as end-to-end path and wavelength-assignment computations; warning—support getting, confirming, and clearing warnings. TSDN APIs follow a RESTful flavor with each resource defined by a uniform resource identifier and employing JavaScript object notation for data exchange. For user friendliness, TSDN provides automatically generated online API documents that list all of the supported operations and parameter specifications.

D. High Availability

TSDN is designed to tolerate concurrent failures such as software crashes, server failures, virtual machine system failures, network disconnections, and even data-center-level disasters. A complete deployment of TSDN spreads across multiple geo-disjoint data centers to provide redundancy at multiple levels. Each data center hosts a slice of TSDN on a cluster of servers. This setup allows TSDN to survive under machine or network failures or when an entire data center is down. In addition, an agent service (to be described below) is clusterized and deployed in the proximity of DCI equipment. Since TSDN communicates with devices via an agent, this multi-agent deployment adds connectivity redundancy in the case of agent failures or management plane disconnections.

E. Fast Iteration

TSDN is deployable to multiple environments (test, integration, staging, production). We employ containerization technologies to achieve parity, such that an image runs with a definite environment regardless of the hosting server, except for the environment-specific variables. Furthermore, we leverage a continuous integration/continuous delivery (CI/CD) pipeline to expedite our engineering efficiency and release cycle. For small features and quick fixes, they can go online from coding as fast as within 30 min.

Figure 11 illustrates several major components that jointly compose TSDN. The first one is TSDN Service, which is the core entity that provides a majority of control and management functions through open APIs. Internally, it has a resource manager that manages resources at all levels, including fiber links, devices, layered metadata (city, data center, network element, plane, etc.). Besides managing resource entities, this resource manager maintains references and dependencies among peer resources or resources at different layers. Once a resource entity is updated, this update is immediately received in the associated reference and dependency and is propagated to the resources that are interested in the update.

To model and manage DCI equipment, TSDN Service implements a process to generate a device instance based on a statically configured template. A template describes the typical configuration and hardware specifications for devices of the same type. For example, a template for a terminal device may describe the number of line cards, the number of client ports per card, and the power range of its optical module for

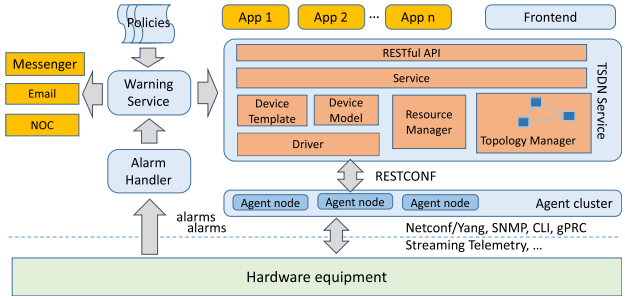

Fig. 11. Overview of the Alibaba control and management platform, TSDN, for open DCI networks. NOC, network operation center.

a line port. Based on the template, a device instance can be dynamically populated with all the components specified in the template. A device template bridges between vendor-specific hardware and the abstract device model in TSDN Service.

In addition to management of individual devices, a topology manager is integrated with TSDN Service to provide topology-related features. It supports generating and manipulating topology at physical fiber links, equivalent to an optical transmission section (OTS), optical multiplex section (OMS), or end-to-end OCH section [20]. The topology is built from a set of statically configured physical fiber links. Several graph algorithms are used to construct the topology, which is done automatically and does not require further human intervention. Transformation between different layers can be done automatically to facilitate path computation, wavelength assignment, signal quality evaluation, wavelength utilization computation, etc. On top of the functional blocks described above, a service layer sits between these blocks and the API layer to compose high-level services and eventually expose them through a set of open APIs.

In order to manage heterogenous hardware in an open DCI network, we introduce an agent service to achieve southbound abstraction and simplify TSDN Service by black-boxing hardware details. An agent service is provided by a cluster of agent nodes, and each agent node runs an ODL instance with light adaptations for Alibaba's production environment. Specifically, the agent provides translation between device-facing protocols (SNMP [15] and NETCONF [13]), and RESTCONF [21] facing TSDN Service. An agent is not part of the TSDN deployment; rather, it is a separate deployment on a different set of physical machines. For better performance such as latency, an agent is usually deployed in the proximity of the DCI equipment that is designated to the agent. One agent can handle many DCI devices, and typically one agent is used in one data center. Therefore, in contrast with TSDN deployment, which is located within a few data centers, an agent deployment may spread to a much larger number of data centers.

A device, when being added to our network, is designated to an agent node to bridge the device and TSDN. For devices that follow the OpenConfig data model, the agent forwards the device config and state data to TSDN in the same way defined by OpenConfig, and our OpenConfig-aware drivers in TSDN Service are able to parse the data. For devices that do not follow the OpenConfig data models, a proprietary data model is employed for data exchange between the agent and TSDN Service. Currently, TSDN uses the agent service to manage DCI devices with SNMP and NETCONF protocols. With such an agent service, TSDN can evolve independently of vendor-specific features and implementations. When a device with a new northbound protocol is introduced, the agent will adapt the device-facing changes, while leaving the RESTCONF between the agent and the TSDN untouched. We aim to support devices that implement gRPC [22] and streaming telemetry. We will deploy them in our networks as they become available.

The TSDN Service has its own web-based frontend/UI that serves as the main portal for daily operations by the network engineers. This is a one-stop shop that offers core functions

such as device configuration, service provisioning, topology view, signal quality analysis, and inventory management. In addition to this UI, we support other teams to develop their network applications by calling TSDN APIs. This mechanism allows quick response to various dynamic business needs that emerge from daily network operations, while keeping TSDN core designs and features in a focused and isolated manner. Our practice has shown proven success for a wide scope of network-operation scenarios.

F. Warning Service and Alarm Handler

These two components are stand-alone, deployable services for handling alarms reported by equipment spontaneously. Alarms can be reported via different mechanisms such as SNMP traps, NETCONF notifications, streaming telemetry, etc. The alarm handler is a thin layer service adapting to a variety of alarm protocols and performs basic preprocessing without understanding or interpreting the actual alarm content. In contrast, the warning service implements the main logic for handling alarms. Each alarm goes through a configurable pipeline, which is composed of a number of policies. A policy is a customized statement defined by network engineers to specify the desired logic of processing an alarm. A typical policy may include checking impacted services, selectively pushing to subscribers, or silently dropping the alarm.

6. DEPLOYMENT

We started the development of open and disaggregated DCI technology in early 2016, and the first pilot test was conducted in early 2017. Massive deployment of open and disaggregated DCI systems in our metro networks began in 2018; today, almost all our new metro optical network deployments use open and disaggregated DCI technology. Over the past few years, we have seen large shifts in the industry's position toward adopting such technology, from reluctance of big system vendors to provide open and disaggregated DCI products at the beginning to the embracing of the technology by the whole industry today, including not only by system vendors but also by component and subsystem suppliers as well. Two years ago, most DCI products used vendor-specific private data models, whereas today, almost all DCI products support or will support OpenConfig models as demanded by hyperscale data center operators.

DCI technology evolves very fast, and the technology cycle is accelerating, as illustrated in Fig. 12. The deployment of coherent technology in DCI networks started with 100G in 2010, and the third-generation 400G coherent technology has been widely deployed in real networks recently; further, 800G and 1.2T per channel will be implemented in the next two to three years. Although there is not much increase in the fiber capacity above a 400G per channel rate (i.e., the spectral efficiency will remain about the same, with the higher channel rates accompanied by coarser channel spacing), the increase of channel speeds will result in the reduction of per bit bandwidth cost [10]. At the same time, real implementation of 400G ZR will start in 2020, which may have a major impact on DCI

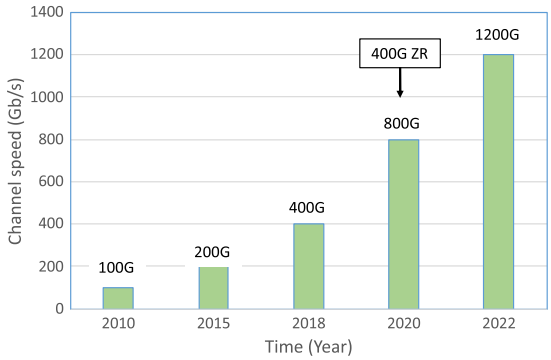

Fig. 12. Channel speed evolution of DCI networks.

networks. Open and disaggregated DCI transport technology allows us to adopt new technology quickly, build DCI networks better suited for our business with a more digestible innovation cycle, and have a better vendor ecosystem.

The open and disaggregated architecture gives us a wide choice of hardware and technologies from different vendors. Selection of hardware from a particular vendor is mainly determined by the advanced technology features of the hardware, such as bit rates per channel, required optical signal-to-noise ratio of terminal transponders, and the support of OpenConfig data models, along with vendor capability for field technical support and the alignment of the technology with our network architecture and roadmap. We have deployed 400G in our networks and are planning to deploy higher bit rate products such as 600G and 800G.

Our control and management platform, TSDN, is used to manage all of the deployed open and disaggregated DCI networks. All of the operations on the DCI networks, including building, service provisioning, configuration, performance monitoring, alarm management, and resource and inventory management, are performed on this platform. Figure 13 provides a quick view of the TSDN dashboard. Due to the sensitivity of business data, the real values on the figure are purposely hidden. The dashboard can be personalized by adding/removing panels according to the roles of the various users. For example, network-planning personnel may wish to see the distribution of OCHs by different locations (as shown in Fig. 13, bottom right), the number of total devices, or a snapshot of high-level geo-coverage statistics. On the other hand, operation personnel may be interested in the current state of the DCI network such as the count of active alarms (Fig. 13, upper left), the distribution of historical alarms by time windows (Fig. 13, bottom left), and the number of devices online (Fig. 13, upper right). On the left, the dashboard provides portals to key operations on topology, resources, alarms, and performance. For example, on the topology page, TSDN supports display of the real-time link state at the OTS/OMS with protection/OMS layers. On the resource page, a user can build a DCI network by defining resources at different levels from data center sites to fiber/devices. Based on these physical resources, OTN resources can then be configured before services are fully activated and the network is being monitored.

Figure 14 provides an example of one of our deployed DCI systems, which illustrates the spectrum that can be viewed

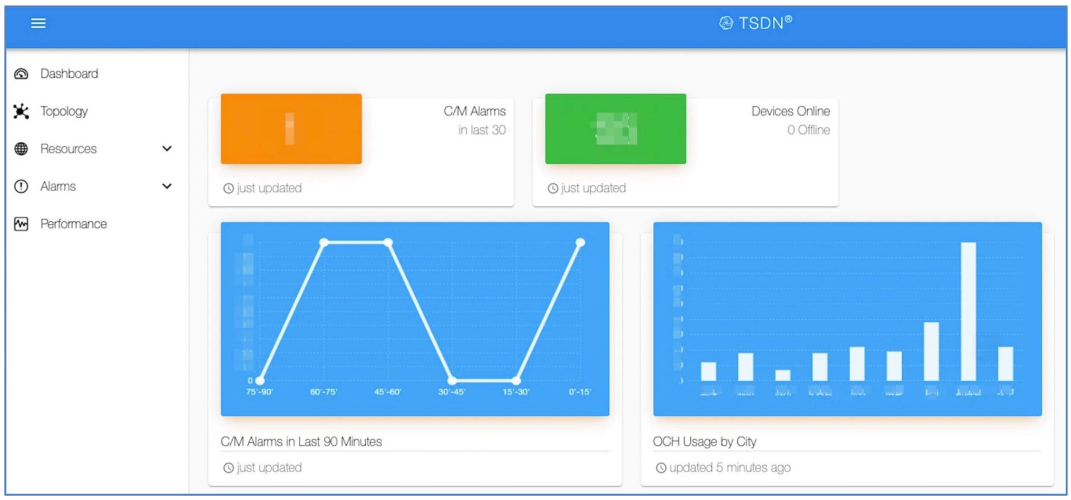

Fig. 13. Dashboard of Alibaba's DCI control and management platform, TSDN.

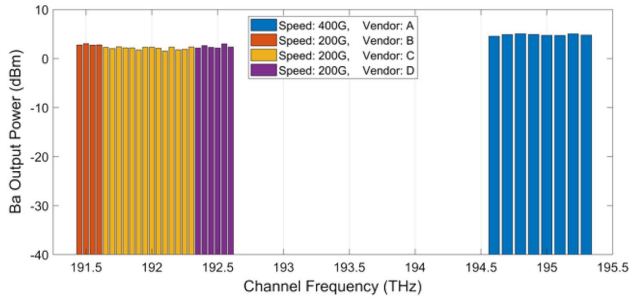

Fig. 14. Spectrum read from the BA in one of Alibaba's deployed DCI networks.

in real time in our TSDN platform with the data read from the OCM in the BA in Fig. 4. In this example, the open and disaggregated point-to-point DCI system with the architecture of Fig. 4 is composed of equipment from five vendors, one for the optical line system and four for the terminal transponders. In this system, we have deployed some 400G channels from vendor A and some 200G channels from vendors B, C, and D. The system is ready to accept 800G and 400G ZR when they become commercially available.

7. SUMMARY

We started to work on open and disaggregated DCI transport networks in 2016 and developed our own open optical transport systems for DCI applications with a home-grown control and management software platform, which have been widely deployed in our networks. This paper presented our views and practices on this technology, including system architecture, data model, control and management platform, and discussion on deployment of the technology in real networks. Open and disaggregated DCI transport networks are still in the early stages, mostly focused on simple point-to-point metro network applications today. This is the first step toward open, programmable, and flexible optical networks. Eventually, hyperscale data center operators and Internet service providers will expand this technology to their mesh networks and wide-area networks, in which ROADMs and more complicated control mechanisms have to be adopted to make optical transport networks more flexible and efficient.

REFERENCES

1. "Amplified multichannel dense wavelength division multiplexing applications with single channel optical interfaces," ITU-T Recommendation G.698.2, Version 1 (2007).
2. V. Kamalov, V. Dangui, T. Hofmeister, B. Koley, C. Mitchell, M. Newland, J. O'Shea, C. Tomblin, V. Vusirikala, and X. Zhao, "Lessons learned from open line system deployments," in *Optical Fiber Communication Conference* (2017), paper M2E.2.
3. M. De Leenheer, Y. Higuchi, and G. Parulkar, "An open controller for the disaggregated optical network," in *International Conference on Network Design and Modelling*, Dublin, Ireland, 2018.
4. http://www.openconfig.net/.
5. http://www.openroadm.org/.
6. N. Sambo, K. Christodoulopoulos, N. Argyris, P. Giardina, C. Delezoide, A. Sgambelluri, A. Kretsis, G. Kanakis, F. Fresi, G. Bernini, H. Avramopoulos, E. Varvarigos, and P. Castoldi, "Experimental demonstration of fully disaggregated white box including different types of transponders and monitors, controlled by NETCONF and YANG," in *Optical Fiber Communication Conference* (2018), paper M4A.3.
7. Y. Yin, T. Wang, L. Dou, S. Zhang, M. Xia, and C. Xie, "Standardized northbound interface testing automation on the open and disaggregated optical transport equipment," in *Optical Fiber Communication Conference* (2019), paper M3Z.3.
8. A. Campanella, B. Yan, R. Casellas, A. Giorgetti, V. Lopez, and A. Mayoral, "Reliable optical networks with ODTN, resiliency and failover in data plane and control plane," in *European Conference on Optical Communications* (2019), demo session.
9. M. Filer, H. Chaouch, and X. Wu, "Toward transport ecosystem interoperability enabled by vendor-diverse coherent optical sources over an open line system," J. Opt. Commun. Netw. **10**, A216–A224 (2018).
10. E. Agrell, M. Karlsson, A. R. Chraplyvy, D. J. Richardson, P. M. Krummrich, P. Winzer, K. Roberts, J. K. Fischer, S. J. Savory, B. J. Eggleton, M. Secondini, F. R. Kschischang, A. Lord, J. Prat, I. Tomkos, J. E. Bowers, S. Srinivasan, M. Brandt-Pearce, and N. Gisin, "Roadmap of optical communications," J. Opt. **18**, 063002 (2016).
11. S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Holzle, S. Stuart,

and A. Vahdat, "B4: experience with a globally-deployed software defined WAN," ACM SIGCOMM Comput. Commun. Rev. **43**, 3–14 (2013).
12. Optical Internetworking Forum, "Implementation agreement 400ZR," oif2017.245.11 (2019).
13. M. Bjorklund, ed., "YANG—a data modeling language for the network configuration protocol (NETCONF)," IETF RFC 6020 (2010).
14. R. Enns, ed., "Network configuration protocol (NETCONF)," IETF RFC 6241 (2011).
15. "An architecture for describing simple network management protocol (SNMP) management frameworks," IETF RFC 3411, https://tools.ietf.org/html/rfc3411.
16. A. Sadasivarao, S. Jain, S. Syed, K. Pithewan, P. Kantak, B. Lu, and L. Paraschis, "High performance streaming telemetry in optical transport networks," in *Optical Fiber Communication Conference* (2018), paper Tu3D.3.
17. J. Babbitt, "Optical DCI architecture: point-to-point versus ROADM," Lightwave, 2017, https://www.lightwaveonline.com/data-center/data-center-interconnectivity/article/16673421/optical-dci-architecture-pointtopoint-versus-roadm.
18. L. Dou, L. Wang, S. Chen, J. Cheng, S. Zhao, M. Xia, H. Zhang, L. Xiao, J. Xu, J. Yu, and C. Xie, "Demonstration of open and disaggregated ROADM networks based on augmented OpenConfig data model and node controller," in *Optical Fiber Communication Conference* (2020), paper M3Z.12.
19. "Open Daylight," https://www.opendaylight.org/.
20. "Interfaces for the optical transport network (OTN)," ITU-T Recommendation G.709, https://www.itu.int/rec/T-REC-G.709/.
21. "RESTCONF protocol," IETF RFC 8040, https://tools.ietf.org/html/rfc8040.
22. "gRPC," https://grpc.io/.