SDN-based Parallel Link Discovery in Optical Transport Networks

Rafael Montero, Fernando Agraz, Albert Pagès, Jordi Perelló, Salvatore Spadaro
Universitat Politècnica de Catalunya (UPC)
Jordi Girona 1-3, 08034, Barcelona, Spain
rafael.montero@tsc.upc.edu

***Abstract*—The use of optical technologies in modern network scenarios has increased in the last decade, mostly due to their support in crucial networking topics (i.e., bandwidth, scalability). In parallel, these scenarios have also experienced the emergence of a new paradigm recognized as Software Defined Networking (SDN), which bases on the decoupling of forwarding and control functions, with aims to provide a more efficient way to manage network resources compared to legacy networking architectures. As both SDN and optical technologies are constantly being introduced in different networking scenarios (e.g., data centres, metro and access networks), their coexistence becomes a must. In this matter, it is important to notice that SDN was initially designed for electronic-based networks; hence its support for optical technologies is still at an early stage. Consequently, integration of both solutions still requires research efforts by the community. In this paper, we present a mechanism to address topology discovery in wavelength-switching Optical Transport Networks (OTNs). In particular, we discuss the importance of the topology discovery function and analyse the proposed mechanism, which bases itself on the use of wavelength-specific signalling tones as link-binding data to provide pre-service parallel link discovery in OTNs. Furthermore, we validate the method experimentally against an emulated OTN testbed with two different setups and compare the results to our previous work on this subject, achieving substantial reductions in the total topology discovery time.**

Keywords—Topology Discovery, Optical Transport Networks, Optical Subsystems, Software Defined Networking.

I. Introduction

The introduction of SDN in modern network scenarios has brought changes in how to utilize network resources more efficiently. In particular, the separation of control and infrastructure layers has represented a disruptive variation to what legacy network architectures exposed, especially when considering electronic-based networks. Likewise, early implementations of SDN were focused on this type of networks, building upon technology enablers such as the OpenFlow (OF) protocol [1]. As the interest on SDN technology by the community began to rise, deployments started to include not only electronic-based subsystems but also optical ones, mostly due to an increased use of optical networking solutions along the last decade. As a result, the control and management of optical resources became an essential matter for future SDN-based networks.

From the optics point of view, the separation of control and forwarding functions may not consider to be as disruptive as in electronic-based networks. Still, other principles of the SDN architecture [2] such as its logically centralized control and the exposure of abstract network resources and state towards an application layer, allow for the programmability of optical networks. This entails, not only reducing complexity in network operation and management but also accommodating these type of networks to the ever-changing needs of nowadays applications. Furthermore, the possibility to integrate both optical and electronic subsystems in a single SDN domain gives network managers granular control over the entire network.

Regarding the adoption of SDN technology, the discovery of underlying resources by the SDN controller plays an essential role, as it serves of topological information to applications. By these means, data consumers can use it to operate according to their particular behaviour. Consequently, the process of topology discovery with optical devices is required in order to provide a way of reaching correct topology mapping at layers above the physical one. In fact, static configurations of topology data, still performed in practice, arise as tedious tasks, error prone and no longer efficient. This is especially true in modern network scenarios that push forward to dynamic environments to accommodate to constantly growing and more complex network configurations. So, a method capable of helping optical networks in dynamically adapting to topology discovery events becomes crucial. Besides, such dynamicity would not only provide applications with up-to-date topological information, but also could improve their efficiency in front of network events, such as dimensioning or network failures, among others. To highlight the importance of automated and dynamic topology discovery in wavelength-switching OTNs, the following sub-sections review the main related work found in the literature, putting special focus on SDN-based discovery mechanisms, while also emphasising the contributions of the presented work before proceeding with the detailed explanation of the proposed mechanism.

A. **Related Work**

As SDN continues its adoption in both hybrid and optical-only scenarios, its level of compatibility with these types of networks starts to appear more and more significant. However, support for optical technologies is still limited to specific use cases. In this matter, efforts to standardize the southbound communication of optical related data started to appear in OF version 1.4.0 [3], where the protocol specification included a new set of port properties to support optical ports. Subsequently, the Open Networking Foundation (ONF) presented in [4] an analysis of the requirements for optical transport networks to work along with OF/SDN, addressing architectural and protocol requirements and considerations. Among them, the analysis identified a necessity for the OF protocol to support the controller validation of the topology and resources under its control in these types of networks. This topology validation requirement focuses on the discovery of interconnections between adjacent neighbours in the underlying data plane. Such interconnection discovery, more commonly known as link discovery, presents difficulties, as optical circuit-switching subsystems do not handle the insertion and extraction of packets from the data flow, which is a key part of typical discovery methods in electronic-based networks.

Regarding link discovery in OTNs, efforts have been present way before SDN rose as a promising option in the control of these types of networks. That said, the ITU presented a recommendation in [5] based in previously set requirements [6], where two methods for automatic link discovery were described. The first method proposed the exchange of discovery related data using the client layer payload while the second considered the in-band exchange of a discovery message embedded in the trail trace identifier (TTI) header. Both approaches provide an automatic link discovery but also come with some considerations. The first method sends discovery messages inside client payload, so it may impact on the client traffic at the link connection. In some cases, it could also make use of a dedicated control channel to exchange discovery information, as seen more typically in GMPLS-based implementations [7]. The second method requires of an in-service trail termination of the transport overhead so the discovery message data could be retrieved or added by the switch. This process does not affect the client signal, but it requires terminating the overhead at each link endpoint.

Following the ITU approach of exchanging discovery data using in-band overhead. The ONF presented in [8] OF protocol extensions to allow switch-controller-switch communication of data related to the discovery process. In particular, the document proposes the use of these extensions to configure and retrieve adjacency information at OF-agents, considering such process to be coordinated by the controller at both ends of a single link thus allowing confirmation of reachability between neighbouring nodes. By these means, the TTI-based discovery of optical links can be achieved. Nevertheless, these extensions are still in an experimental stage and waiting for their addition in future protocol specification releases. Another important aspect is that this work has been built based on version 1.4 of the OF protocol, which is not fully supported by most of the SDN controllers in the market at the time of writing this paper. In spite of this, if a wide adoption of these extensions is achieved at some point in the future, these could even be considered to manage the method presented in this document.

In our previous work [9], we presented a topology discovery method for optical networks considering a sequential mechanism. Such mechanism is based on the exchange of Test-signals to discover optical links on a one-by-one basis, providing a correct mapping of links at the control plane although limiting scalability and time efficiency, especially when addressing the discovery of large network topologies. The work presented in this paper enhances it by allowing the discovery of links in a parallel manner by means of using specific link-binding data, thus helping to overcome the sequential method limitations. In particular, the conducted experiments show how this new mechanism allows to reduce the total topology discovery time when compared with the sequential one. It is very important to mention that these proposals fit in the pre-service category of layer 1 adjacency discovery use cases as defined in [6], where the mechanism acts before client link connections have been allocated, thus not affecting any traffic, in contrast to the previously discussed in-service method that could cause behavioural problems in the network.

B. **Contributions**

In this paper, we present a novel mechanism to address the necessity of link discovery in OTNs considering all previously discussed facts, providing a new variant in adjacency discovery techniques for transport networks. We propose the use of the optical extensions defined in [10], where additions to the OF protocol are built on version 1.0, widely supported by SDN controllers. Such extensions allow for the exchange of optical related information between the OF agents and the controller to support the discovery of optical subsystems. On this basis, we define a parallel link discovery method where the controller coordinates a pre-service exchange of wavelength specific *Test-signals* through all active fibre ports in the network, so that connected neighbours can advise the controller of a successful exchange. Then, the controller can use this information to create optical links at the topology database. The parallelism of the method regards to the ability of the controller in handling multiple discovery processes in the same period of time. Besides implementing the extensions, the proposed mechanism demands extra configurations at both control and data layers. At the control level, an SDN application is required to trigger the discovery process, coordinate the OF message exchange, correlate the received information and finally create the links. At the data layer, more specifically at the OF agent of the optical node, configurations are also required so it can understand orders coming from the controller and reply with the discovery related data. In order to highlight the added value of this work in respect of the sequential mechanism, Table 1 summarizes a high level feature comparison between methods.

Table 1. Feature comparison of Parallel vs Sequential Methods

	Parallel Discovery	**Sequential Discovery**
Pre-service mechanism behavior	X	X
Use of OF v1.0 optical extensions	X	X
Awareness of Port supported λ's	X	
Per-process use of available λ	X	
λ Tag embedding in OF-message	X	
Frequency grid based λ mapping	X	
Simultaneous discovery processes handling	X	

Under such premises, the remainder of this paper continues as follows. An architectural overview analyses the importance of the topology discovery function in SDN in Section II, putting emphasis on network scenarios including OTNs. Section III characterizes the proposed link discovery mechanism, together with an explanation of the required extensions at the OF agents and the SDN controller for its proper operation. In Section IV, we validate the mechanism at the control level by performing tests against an emulated OTN test-bed, showing results besides a comparison with our previous work in this subject [9]. Finally, Section V summarizes the main achievements and conclusions of the presented work.

II. Link Discovery in SDN-based Optical Networks

The topology discovery function has always been an important matter in SDN scenarios, as it comprises the representation of the underlying network resources in terms of host, node and link elements. In this respect, the study presented in [12] discussed the significance of such a function for a network environment to behave as expected. In general, this work identifies a precise topology discovery to be essential, so that tasks such as monitoring, resource management, scheduling, diagnosing, among many others, can operate in a correct manner. Actually, most services at both application and control layers depend on using this topological information. From the services point of view, the importance is set in their access to a correct abstraction of the network, so that the methods that the controller employs to construct such a topology representation are not particularly relevant to consumers.

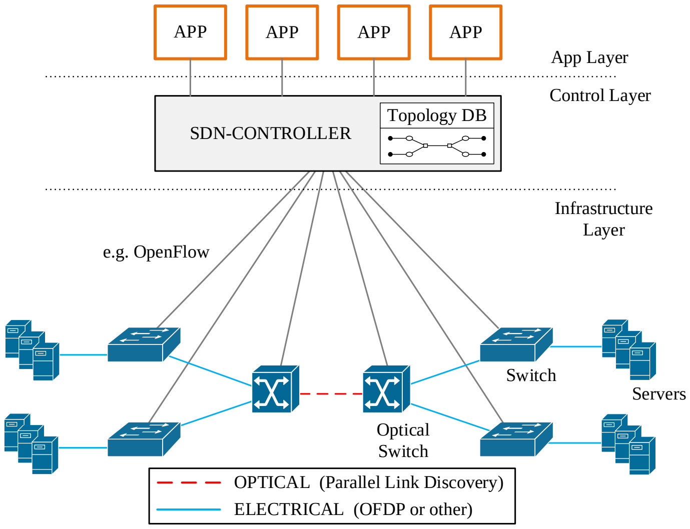


Fig. 1. SDN-enabled Hybrid Opto-Electrical Network.

Fig. 1 depicts a representative hybrid opto-electrical network scenario where the controller needs to discover both electrical and optical links, besides hosts and nodes, so that SDN applications can consume the abstracted network information. The complete topology data in this case is usually stored at the topology database in the controller. Applications then use the northbound communication to access and retrieve the data. Link discovery, more specifically, can use a combination of the OpenFlow Discovery Protocol (OFDP) [11] (or other method) to map the electrical interconnections and the proposed Parallel Link Discovery method for the optical ones. Even assuming that scenarios in practice may vary, this example serves to highlight the importance of having dynamic link discovery for both optical and electronic-based devices in modern SDN-controlled networks.

As exposed before, the controller is required to map network elements from the underlying data plane, where such elements can have different capabilities, either electrical or optical in the case of OF-enabled switches. Therefore, the controller must allow handling different procedures for host, node and link recognition according to the type of network resources under its control.

As introduced in the previous section, link discovery particularly presents a higher level of complexity in comparison to other topological elements (i.e., hosts and nodes). Therefore, it usually requires the controller to use the help of a discovery protocol, or mechanism, to fulfil the correct mapping of links. In packet-based networks for instance, OFDP is the common option for discovering adjacencies between network nodes. Such protocol is based on the exchange of Link Layer Discovery Protocol (LLDP) packets containing Chassis-ID and Port-ID identifiers in their payload, so that the controller can use this information, along with other metadata, to map the underlying links at the control level. In this regard, various analysis (e.g., [12] - [15]) have remarked the vulnerabilities of using OFDP, proposing both enhancements to the current protocol and other methods for accomplishing link discovery in these types of networks. Specifically, the work in [13] presents an enhanced version of the protocol with less LLDP message exchanges (OFDPv2). Then, the work in [14] presents the sOFTD (Secure OF Topology Discovery) which adds security to the protocol by introducing minimal changes to the OF switch design. Finally, the one work reported in [15] proposes to enhance the authentication of LLDP packets by using HMAC (Hash-based Message Authentication Code).

In transparent optical networks, the scenario is entirely different as no insertion or retrieval of packets is possible between neighboring nodes. Thus, other type of information exchange or technique to identify optical links should be employed, allowing the controller to have complete visibility of the SDN domain. To this end, this paper proposes the use of a mechanism capable of providing an efficient discovery of optical links so upper layer services can use valid topological information. Moreover, the proposed link mapping is fully dynamic, so that changes in the topology do not affect the performance of the network services.

Next section presents the basics of the Parallel Link Discovery method along with the required configurations for its implementation and the message exchange involved in the discovery process.

III. Parallel link discovery

The Parallel Link Discovery mechanism bases on exchanging signalling tones between neighbouring nodes, using a different wavelength for each exchange performed at the same period in time. This wavelength information allows correlating OF status messages coming from the OF agents at the SDN controller in order to construct the topology representation at the control layer, thus avoiding the need for discovery message insertion/retrieval from optical channels.

As introduced, the proposed method requires a set of techniques, extensions and additions to operate. The following subsections present an in-depth analysis of each of them, addressing topics such as how the controller coordinates the triggering of the discovery process, which OF-messages are involved in the message exchange flow between layers, which information the controller uses to correlate the creation of optical links, among many other specifics.

A. **Wavelength-specific Test signal mechanism**

The first consideration made when designing the mechanism, was to overcome the limitation of not being able to exchange packet-based link information between neighbouring nodes, which discards the possibility of using LLDP. Instead, we considered the exchange of test signals between optical ports as a valid way of confirming reachability at both ends of a link. A test signal entails the transmission of light pulse by an active emitter fibre port and its detection at the receiver port, and only represents a signalling tone sent port-to-port without carrying any control related message or information. By these means, instructions sent to adjacent nodes would order them to execute such process and let the controller know about successful test signal exchanges by sending back confirmations as in [9]. However, this procedure presents some drawbacks, as it is only possible for the controller to recognize one optical link at a time, because confirmations sent by the OF-agents do not carry data to make the discovery of a link independent of any other discovery related confirmations. Thus, the procedure bounds to the use of a time window for the discovery of each optical link. Such limitation could represent penalties not only in the total time of topology discovery but also in scalability when discovering larger topologies with high number of nodes/links.

In light of the above, in this work we enhance the test signal-based mechanism to allow mapping links in parallel and without the constrains of a sequential per fibre port discovery. To achieve this, it is necessary to have some sort of link-binding data to allow the correlation of a confirmation sent by a peer node to the one sent by the node where the discovery process was initiated. Moreover, the controller needs to be able to trigger the process as fast as possible after detecting newly connected optical ports, while also being capable of creating detected links in the topology database by using the confirmations sent by OF-agents, despite of their arrival time or order of arrival. To achieve this, the mechanism uses a pool of wavelengths/frequencies, so that test signals involved in the discovery of a specific fibre link are sent over the same wavelength. The controller in this regard, must maintain knowledge of which wavelengths are being used by other active discovery processes, so it can select an available one to avoid link discovery inconsistencies. A previously selected wavelength from the control-level database is made again available after its related discovery process is finished. This wavelength information, mapped as a code for its exchange between controller and agent, serves as the link-binding data for the controller to use in order to recognize the correlation between arrival confirmations. More specifically, the controller is now able to handle simultaneous link discovery processes.

The use of code mapping for wavelengths is also present at the node in the form of a common frequency grid, which, although being common for all nodes and containing all set of frequencies, it will only be used to map the ones supported by each specific node. As a result, it is also important that the controller assigns a wavelength supported by the optical port, otherwise the test signal exchange could not be performed. Therefore, it must gather this information first at time of retrieving each node's features and consider it at the time of triggering a discovery process. Since the link discovery is performed at the network provisioning phase, prior to the network operation, that is prior to service allocations, to send/receive the test signals related to different discovery processes at the specified ports and through the specified wavelengths the optical transceivers equipping the optical nodes can be used.

Fig. 2 shows an example of the test signal exchange mechanism enhanced by the use of an assigned wavelength. In this case, the details of the procedure are explained as follows.

•Step 1 - Transmit Initial Tone: Once the controller selects an available wavelength from the database, the transceiver receives detailed instructions from the controller to begin the discovery process; it starts by sending the first test signal at the specified fibre port and over the specified wavelength.

•Step 2 - Process Received Tone: When the receiver senses the test signal, it processes the signal digitally to identify the wavelength.

•Step 3 - Code Mapping: In addition, it uses a common frequency grid to map the wavelength to a corresponding code.

•Step 4 - Notify Controller (Receiver): The receiver constructs a specific OF-message containing the code (in this example W=1) and flags to identify it as a receiver. It sends the message to the controller as a test signal confirmation.

•Step 5 - Transmit Response Tone: In parallel, the receiver sends another test signal back using the port where it sensed the original signal and through the same wavelength.

•Step 6 - Process Response Tone & Code Mapping: Back at the transmitter, it receives the response test signal and repeats the procedure to identify the wavelength and get the code.

•Step 7 - Notify Controller (Transmitter): Finally, it also constructs the confirmation message with the code and transmitter identifier flag to send it to the SDN controller.

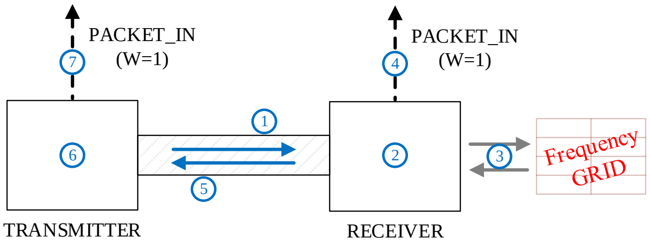


Fig. 2. Wavelength-specific Test signal mechanism.

In general, the controller executes this process before the link starts transferring data (i.e., at the pre-service stage). Another important matter is the deletion of links, where the controller should be able to delete links from the topology database upon the reception of a Port-Down status from the OF-agent. This would allow guaranteeing a correct mapping of the network resources, so that the controller can always maintain an up-to-date topology.

B. OF-based Discovery Message Flow

The next step in the implementation of the link discovery process was to address the communication between control and infrastructure layers, more specifically between the SDN controller and the OF-agents. As commonly supported OF versions (i.e., OpenFlow v1.0, OpenFlow v1.3), do not handle optical node discovery, extensions in [10] were considered to provide the OF version 1.0 of this functionality, so that optical nodes and all their ports could be identified along with their specific characteristics. With this feature, the controller would not only be able to gather information regarding which fibre ports are in a Port-Up state but also trigger the discovery process on them.

The discovery mechanism takes advantage of a set of OF messages that are exchanged using independent control TCP/IP connections between OF-agents and the controller, where each single connection is established by means of a management network. In fact, all OF discovery related messages are sent at this level and through these control channels. Fig. 3 depicts the complete message sequence. First, when a new optical node (i.e., OF-Agent 2 in this case) connects to the controller, both exchange HELLO messages to establish the connection and negotiate the OF version to be used. Next, the SDN controller sends a FEATURES_REQUEST message asking for capabilities that, the agent itself responds with a FEATURES_REPLY message. At this point, the controller is able to identify which fibre ports are active and begin sending discovery related OF messages.

In order to trigger the discovery process, the controller sends PACKET_OUT messages (1) towards OF-Agent 2, instructing it to begin testing reachability through the specified ports and using specific wavelengths. These particular messages use SET_DL_SRC action flag to distinguish themselves as discovery triggering messages. So, after instructed, the agent sends test signals (2) to both neighbouring nodes (i.e., OF-Agents 1 & 3). Once peers detect and process these signals, both send back a confirmation test signal (3) and construct PACKET_IN messages to let the controller know about the recent events (4). The initiating agent receives both confirmation signals from peers and sends PACKET_IN messages to the controller (5), one for each successful test signal exchange. The difference between all PACKET_IN messages arriving to the controller is that OF-Agent 2 uses an ACTION reason flag to identify itself as the node which generated the initial reachability tests, while OF-Agent 1 & 3 use a NO_MATCH reason flag to identify themselves as peers. To be able to correlate all the received information, the controller uses these reason flags along with other link-discovery related data (i.e., port numbers and wavelength codes). As introduced in the previous subsection, by using this data, the number of discovery processes executed at a specific point in time is not relevant for the controller. In fact, the only constraint now becomes the number of supported wavelengths by the identified fibre ports, as only one wavelength can be assigned per port for testing at the same time. After discovering a link, the controller can release the wavelength so it becomes available for triggering the discovery process in another fibre port.

Once the controller pairs the information received from nodes at both ends of a link, it notifies these agents about their directly connected peers by means of PACKET_OUT messages (6) (7). In contrast to the ones in (1), these messages use a SET_DL_DST action flag, so that the agents can identify them as the ones confirming peer data. Finally, after adjacent nodes learn peer information, they expose it again to the controller in the form of CPORT_STATUS messages, which allows topology manager modules at the controller level to gather directly this data and construct new optical links at the topological database.

In summary, the depicted message flow allows to correctly map optical links in a parallel manner using standard OF messages related to electrical networks such as PACKET_IN, PACKET_OUT messages, while also employing extended, specific to optics messages like the CPORT_STATUS. Consequently, the employment of the extended OF protocol version in the SDN environment is considered crucial for the whole message flow to work.

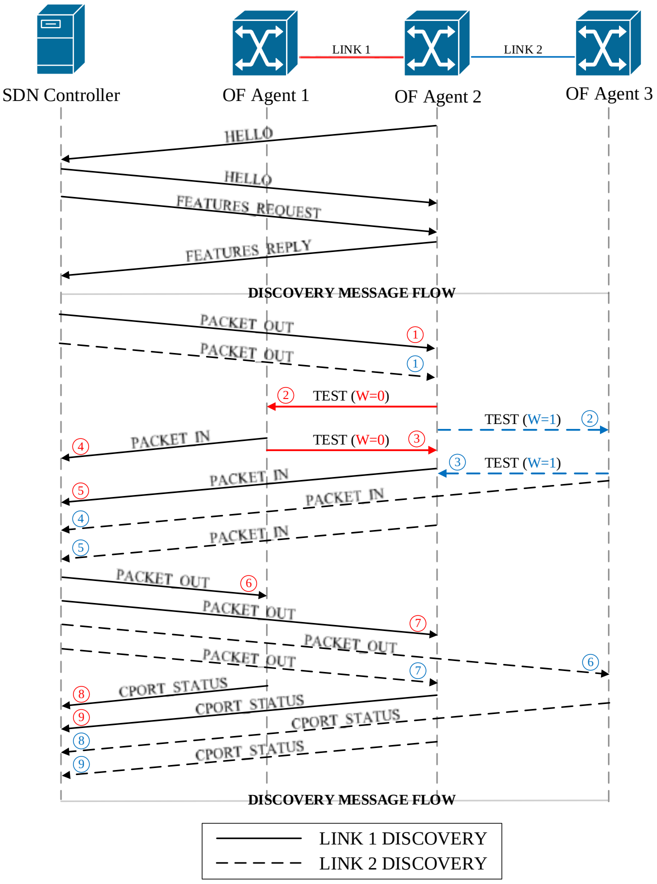


Fig. 3. Discovery Message Flow.

C. Mechanism Software Requirements

A classification of the requirements for the discovery mechanism is possible by considering layers exposed in the general SDN architecture [2]. Following this, we divide requirements between the ones needed at: 1) the Infrastructure Layer, or 2) the Control Layer. In both cases, configurations could consider implementing new software, extending an existent one or simply using available resources.

1) Infrastructure Layer: At this level, extensions to OF-agents are required, as they need the intelligence to understand instructions coming from their designated SDN controller. More specifically, agents must handle incoming OF-messages related to the discovery process and use the information contained in them to instruct the use of the wavelength-specific test signal mechanism at the optical nodes. Additionally, they must also allow creating OF specific messages to send back confirmations towards the controller while considering the previously analysed usage of flags.

Another major implementation required at the agent level is the ability to map wavelengths to their correspondent code. For this, it is necessary, first of all, to digitally process the received test signals to detect the used wavelength. Then, the agent must use a common Frequency Grid to get the code, where such a grid must only contain the supported frequencies by the receiver fibre port. Finally, the agent should add this information into the OF-message used to advertise the controller of discovery events.

2) Control Layer: The configurations at this level require more efforts in terms of software extensions and the creation of new modules inside the SDN controller. Fig. 4 helps to appreciate the overall architecture, making emphasis on the control layer. In principle, the intelligence required for the link discovery process could be implemented in any controller compliant with the v1.0 OF protocol. In this case and as exposed in the figure, we have selected the OpenDaylight (ODL) platform [16] as the base controller for this implementation; the controller has been modified to support the extended OF protocol used. Moreover, a new link discovery module has been introduced while other existent service modules have also been extended. Note that the use of the discovery mechanism is not bounded to ODL, but the implementations and extensions required at this level can be accommodated to any other particular SDN controller, given the aforementioned specifications.

Starting from the southbound interfaces, the existent OF Plugin & OF Java modules were modified to accommodate the extensions proposed in [10] to the support for the OF v1.0 by these modules. By implementing these extensions, the protocol allows the use of wavelength-switching OTNs related data, so that the controller can identify these capabilities when a new node connects via the extended OF protocol. After an optical node exposes its features, these are usually stored at databases in the Model-Driven Abstraction Layer (MD-SAL), which allocates specific model-defined data structures that provide a source of self-describing data to components inside the controller. Furthermore, these data structures also allow consumers (i.e., internal/external apps, orchestrator, etc.) to retrieve required data for their operation. In the case of components at the application layer, this is typically done via northbound Application Programmable Interfaces (APIs).

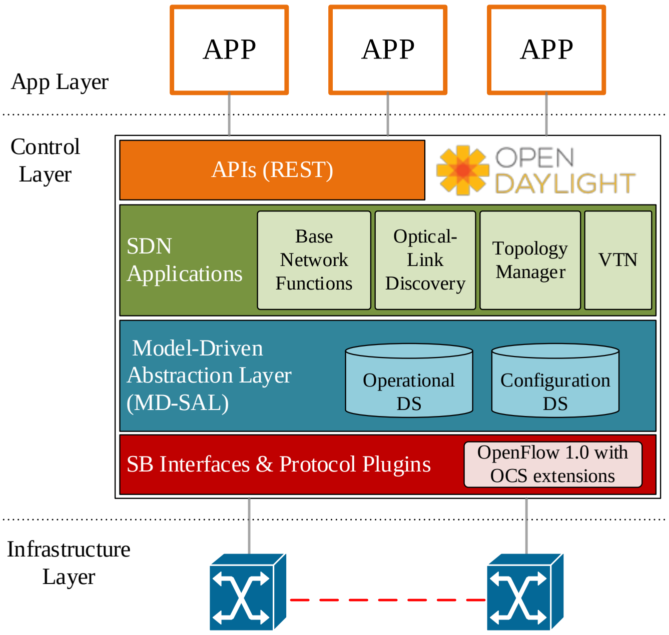


Fig. 4. SDN architecture for OTNs.

In order to manage the OF-messages related to the link discovery, we designed a new Optical-Link Discovery (OLD) module. Such module is responsible of creating a list of all the newly recognized active fibre ports and triggering the discovery process over them, considering the maximum number of simultaneous processes according to the number of supported wavelengths. Besides this, the OLD is capable of correlating all the confirmation messages arriving back from OF-agents and of advising them of their peer information. Hence, the OLD is the true enabler of the parallel link discovery process.

Another module involved is the Topology Manager (TM) that was adapted to use data received in the port status updates, such as the CPORT_STATUS message, allowing to create or delete optical links. This module in particular is the one responsible of dynamically maintaining the topological database up-to-date. Finally, the Virtual Tenant Manager (VTN) module presents also extensions. This particular module is the one in charge of providing L2 functionalities, requiring the ability of identifying optical discovery related (PACKET_IN and PACKET_OUT) packets for forwarding them to the OLD module for their further processing.

It is worth noting here, that, by design, the illustrated mechanism is sensitive to network changes (by addition or removal of equipment). In particular, the controller dynamically reacts to such changes either by deleting the removed hardware from the topology or by triggering the discovery over the added equipment. Thanks to that, an up-to-date topological representation at the control layer is made possible, reacting to the dynamics of the underlying physical topology, essential for applications consuming the connectivity services provided through the SDN controller.

IV. Experimental validation

To validate the proposed discovery mechanism experimentally, we used an SDN testbed running two emulated OTN scenarios with different topology configurations. To do this, we used software-based emulators of the optical switching nodes (multi-degree Reconfigurable Optical Add&Drop Multiplexers [17]), which also emulate the test signal transmission and receiption, and deployed them according to the selected topological lay-outs. We also deployed an OF-agent over each optical node emulator. These OF-agents where connected to the extended SDN controller. Table 2 presents the key parameters of these scenarios where we consider bidirectional links, meaning two per interconnection. We used a 9-node and a 14-node scenario with 10 supported wavelengths by link (i.e., optical fibre port). Fig. 5 shows a graphical representation of both topologies.

Table 2. OTN Topologies Key Parameters

	# Nodes	# Links	# Supported Wavelengths
Topology 1	9	22	10
Topology 2	14	46	10

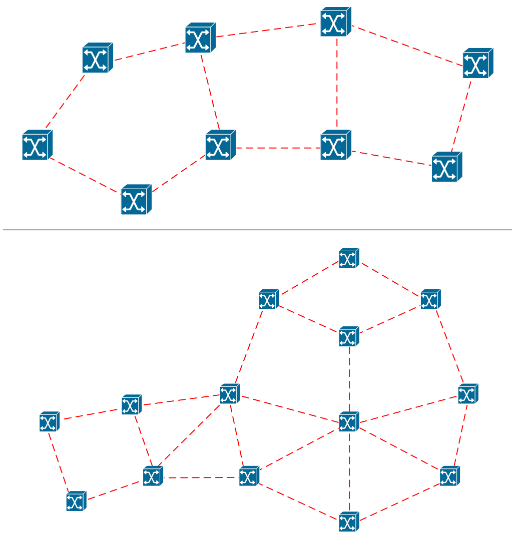
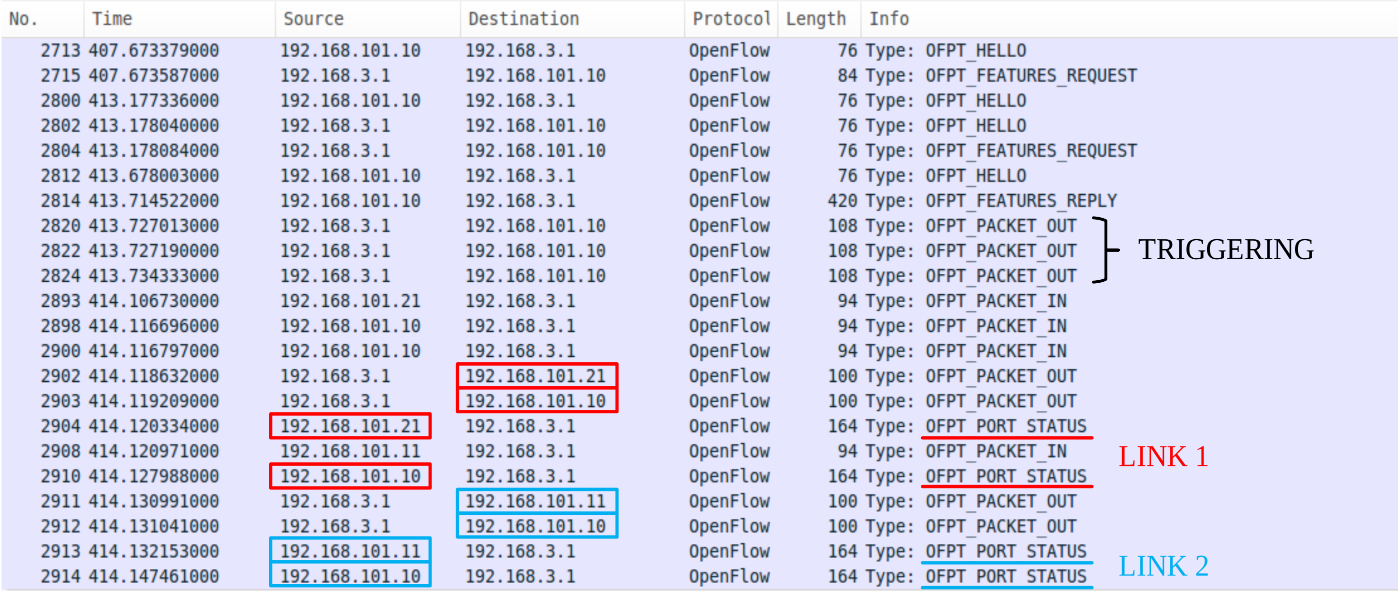


Fig. 5. OTN Topology configurations: 9-node (top), 14-node (bottom).

In order to illustrate the mechanism operation, Fig. 6 shows a capture of the discovery messages exchange between a newly connected optical node (IP: 192.168.101.10) and two of its adjacent nodes. More specifically, the mechanism is required to discovery two links, LINK 1 towards the first adjacent node (IP: 192.168.101.21) and LINK 2 towards the second one (IP: 192.168.101.11).

The message flow starts with the node exchanging HELLO messages with the controller and exposing its features through the FEATURES_REPLY message. As the figure shows, at this point the controller triggers the discovery mechanism through all the recently recognized active fibre ports, so the OF-agent of the node can instruct the sending of the test signals at the specified ports/wavelengths. When the other agents sense the signals and send responses back, they confirm the events to the controller by means of PACKET_IN messages.



```
No.   Time            Source           Destination      Protocol Length Info
 2713 407.673379000   192.168.101.10   192.168.3.1      OpenFlow     76 Type: OFPT_HELLO
 2715 407.673587000   192.168.3.1      192.168.101.10   OpenFlow     84 Type: OFPT_FEATURES_REQUEST
 2800 413.177336000   192.168.101.10   192.168.3.1      OpenFlow     76 Type: OFPT_HELLO
 2802 413.178040000   192.168.3.1      192.168.101.10   OpenFlow     76 Type: OFPT_HELLO
 2804 413.178084000   192.168.3.1      192.168.101.10   OpenFlow     76 Type: OFPT_FEATURES_REQUEST
 2812 413.678003000   192.168.101.10   192.168.3.1      OpenFlow     76 Type: OFPT_HELLO
 2814 413.714522000   192.168.101.10   192.168.3.1      OpenFlow    420 Type: OFPT_FEATURES_REPLY
 2820 413.727013000   192.168.3.1      192.168.101.10   OpenFlow    108 Type: OFPT_PACKET_OUT
 2822 413.727190000   192.168.3.1      192.168.101.10   OpenFlow    108 Type: OFPT_PACKET_OUT
 2824 413.734333000   192.168.3.1      192.168.101.10   OpenFlow    108 Type: OFPT_PACKET_OUT
 2893 414.106730000   192.168.101.21   192.168.3.1      OpenFlow     94 Type: OFPT_PACKET_IN
 2898 414.116696000   192.168.101.10   192.168.3.1      OpenFlow     94 Type: OFPT_PACKET_IN
 2900 414.116797000   192.168.101.10   192.168.3.1      OpenFlow     94 Type: OFPT_PACKET_IN
 2902 414.118632000   192.168.3.1      192.168.101.21   OpenFlow    100 Type: OFPT_PACKET_OUT
 2903 414.119209000   192.168.3.1      192.168.101.10   OpenFlow    100 Type: OFPT_PACKET_OUT
 2904 414.120334000   192.168.101.21   192.168.3.1      OpenFlow    164 Type: OFPT_PORT_STATUS
 2908 414.120971000   192.168.101.11   192.168.3.1      OpenFlow     94 Type: OFPT_PACKET_IN
 2910 414.127988000   192.168.101.10   192.168.3.1      OpenFlow    164 Type: OFPT_PORT_STATUS
 2911 414.130991000   192.168.3.1      192.168.101.11   OpenFlow    100 Type: OFPT_PACKET_OUT
 2912 414.131041000   192.168.3.1      192.168.101.10   OpenFlow    100 Type: OFPT_PACKET_OUT
 2913 414.132153000   192.168.101.11   192.168.3.1      OpenFlow    164 Type: OFPT_PORT_STATUS
 2914 414.147461000   192.168.101.10   192.168.3.1      OpenFlow    164 Type: OFPT_PORT_STATUS
```

Fig. 6. Wireshark capture depicting the message exchange for the adjacency discovery between three network nodes.

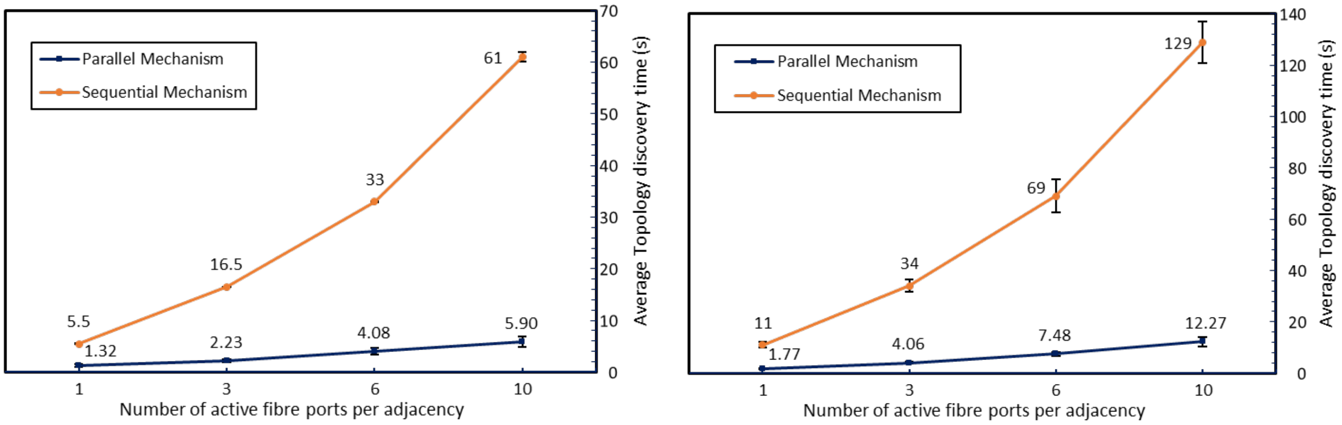


Fig. 7. Average topology discovery time comparison with Parallel / Sequential mechanisms in OTN emulated scenarios: 9-node (left) and 14-node (right).

Then, the node, that started the test signals, receives the responses and, in turn, sends the confirmations. With all this information at the controller level, the OLD application is able to correlate the link data and send peer information to all the OF-agents involved by using PACKET-OUT messages. Finally, the agents perform a final port update with the CPORT_STATUS message, so the TM itself can retrieve these messages and create the optical links at the topology database.

At the end of the discovery process, the TM refreshes the topology database with the new link information; consumers now can retrieve this data and show it to the user. For example, consumers could represent the applications in charge of the ODL graphical interface, but in general, these could be any application at the control or application layer. As it is discussed ahead in this paper, links would now be visible between each newly connected node and its adjacent nodes, thus allowing the logical construction of the complete topology at the control level.

The main objective for the tests was checking the total time of topology discovery for both scenarios and compare times with the ones achieved in [9]. In this regard, it is worth mentioning that these results used for comparison use a sequential link discovery mechanism in contrast to the parallelism of the presented one.

The work related to the sequential method considers an orderly triggered per fibre port discovery, where time windows are set between each discovery process. Note that the benchmarking between the two mechanisms does not include the per fibre port discovery time, since it is similar for both. On the other hand, when focusing on the total time of discovery for a complete topology, it is worth to be noted that the presented method triggers the discovery in fibre ports in parallel, without requiring to wait for the previous discovery process to finish. Therefore, we expect a significant reduction of the total time of topology discovery when compare the two different mechanisms.

To this purpose, Fig. 7 exposes the topology discovery time comparison between both mechanisms including the 95% confidence interval. These times are the resulting averages after executing the discovery process 10 times on each different configuration. To get these measures we have used the controller log to identify the moment where it triggers the discovery process upon recognizing active optical ports and compare it to the time it makes available link information in the topology database. It is worth noting here how both figures follow similar behaviours in despite of the tested scenario, which relates directly to the one-by-one link discovery basis of the sequential method versus the simultaneous link discovery of the Parallel one. Besides, for the two emulated OTN scenarios, we used a different number of fibre ports or links between adjacent nodes. More specifically, we deployed scenarios using 1, 3, 6 and 10 bidirectional fibre ports. As a result, we managed to increase the number of links to further test the scalability of the mechanism when there is a large number of optical links in the topology. Fig. 8 shows the relation between the number of fibre ports between adjacent nodes and the total number of links.

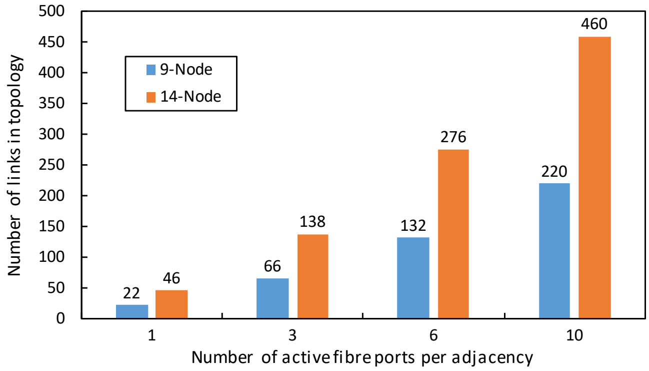


Fig. 8. Links to be discovered according to fibre ports per adjacency.

In order to let the controller discover all these links as fast as possible, we connect all the OF-agents belonging to the scenario at once towards the controller. In this way, the OLD application can identify both the total number of ports where the discovery process should be triggered and the supported wavelengths by these ports. In this case, and as Table 1 shows, each fibre port supports 10 different wavelengths, so it limits in this way the maximum number of concurrent discovery processes to 10.

Back in Fig. 7, it is possible to see how the execution time results obtained with the proposed parallel discovery mechanism, even with only 10 supported wavelengths, present important reductions in all types of configurations compared to the ones of the sequential discovery. For instance, in the 9-node scenario, starting with one port per adjacency (22 links) the topology discovery time reduces from 5.5 seconds in sequential mode to only 1.3 when using parallel discovery. Next, the gap starts to get bigger as the number of links for discovery continue to increase. For example, with three ports per adjacency (66 links) the time reduces from 16.5 to 2.23 seconds. Moreover, in the cases of six ports (132 links) and ten ports (220 links), the topology discovery time reduces from 33 to 4 and from 61 to 5.9 seconds respectively.

As for the 14-node emulated OTN scenario, the differences in time for both mechanisms are even bigger. In this case, we can analyse the reduction impact in the configuration with more links. Hence, with ten ports per adjacency (460 links) it goes from taking 129 seconds to discovery the full topology with the sequential approach to only 12.2 seconds with the parallel mechanism. While previous cases show, at least, an 80% of improvement in the discovery time, this last case shows a reduction of around 90% of the total time it used to take before, thus showing the scalability potential of this method when working with scenarios with an even larger number of links. In this matter, we also consider important the use of the mechanism only in front of a new port event (i.e., Port-Up, Port-Down), and not periodically as regular discovery protocols. This approach is required to avoid stressing the controller by constant message processing, which could affect not only the discovery process but other applications as well.

The substantial reduction of total discovery times achieved with the Parallel Link Discovery method is an expected effect of the enhanced controller ability to trigger multiple discovery processes in the same period of time. As now, in comparison to the required time window in the Sequential mechanism to confirm each link, the controller can have a number of on-going discovery processes equal to the number of wavelengths supported by the links (i.e., 10 in the present experiment).

The times and effects previously analysed demonstrate the significant improvement when changing from a per port basis to one capable of handling discovery processes in parallel. For this, we consider the introduced use of wavelengths in test signals crucial, so the controller can use this information as an identifier when correlating links at the control layer. Furthermore, we also consider the configurations realized at both the controller and OF-agents to be fundamental, where the most relevant additions are the introduction of the OLD application, extensions at TM, VTN, OF-JAVA and OF-PLUGIN projects, besides provisioning agents with the intelligence to interact with the controller and handle the use of test signals to test reachability between adjacent nodes. As a result of these additions, the ODL graphical interface will now feed from the topological database, containing optical link information such as Source-Port, Destination-Port and Link-Id, and be able show a representative view of the up-to-date OTN topology. This information will allow correlating particular links to their correspondent statistics data received from optical subsystems in order to analyse the status of the link, resource utilization, error recovery, etc. Moreover, besides the visual view, a newly recognized link will be identified as a modification event to the topological database, which could also trigger any application related task listening to the link-created event.

Finally, after completing the validation at a software level and as a last concern, further testing with this mechanism is still required, especially considering its implementation with physical optical equipment, which could bring other variables to the tests performed in this paper.

V. CONCLUSIONS

Following the current trend of employing optical solutions in different network scenarios, helped by the use of modern technologies such as SDN, this work analysed the importance of having a correctly mapped topology in scenarios with optical capabilities such as OTNs, so that consumers of this information can operate as expected. In this regard, we identified the need in OTNs for a process or method to allow the controller operating in these networks recognizing optical links. As a result, this work has also presented a mechanism to address this necessity by means of a parallel link discovery process coordinated between controller and OF-agents, which brings a new variant to the state of the art in adjacency discovery techniques for transport networks, due to its pre-service nature.

The paper furtherly presents tests involving the use of the proposed mechanism against emulated OTN scenarios with 9-node and 14-node configurations, where we compared the obtained results to our previous work in the subject. Regarding this, various setups in the number of fibre links between neighbours were used, so that tests could be performed over different topology alignments. In overall, results showed a significant improvement of above 80% in the average time of topology discovery between both compared methods. Moreover, results exposed how the simultaneous link discovery basis of the parallel method overcomes the limitations of a one-by-one link recognition process. Considering the obtained results, the mechanism proved to guarantee the correct mapping of optical links in OTNs while also demonstrating its scalability potential when discovering a high number of links in the topology.

ACKNOWLEDGMENT

This work has been partially funded by the Spanish National project SUNSET (TEC2014-59583-C2-1-R) with FEDER contribution.

REFERENCES

[1] Nick McKeown et al. “OpenFlow: Enabling innovation in campus networks”. ACM Communications Review, vol. 38, num. 2, pp. 69-74, April 2008.

[2] ONF, “SDN Architecture Overview”, Version 1.0, Open Networking Foundation, November 2014.

[3] ONF, “OpenFlow Switch Specification”, Version 1.4.0, Open Networking Foundation, October 2013.

[4] ONF, “Requirements Analysis for Transport OpenFlow/SDN”. Open Networking Foundation, August 2014.

[5] ITU-T Recommendation G.7714.1/Y.1705.1, “Protocol for automatic discovery in SDH and OTN networks”, 8/2017.

[6] ITU-T Recommendation G.7714/Y.1705, “Generalized automatic discovery for transport entities”, 8/2015.

[7] J. Oliveira et al., "Experimental testbed of reconfigurable flexgrid optical network with virtualized GMPLS control plane and autonomic controls towards SDN," 2013 SBMO/IEEE MTT-S International Microwave & Optoelectronics Conference (IMOC), Rio de Janeiro, 2013, pp. 1-5.

[8] ONF, “Optical Transport Protocol Extensions”. Open Networking Foundation, March 2015.

[9] R. Montero, F. Agraz, A. Pagès, J. Perelló and S. Spadaro, "Dynamic topology discovery in SDN-enabled Transparent Optical Networks," 2017 International Conference on Optical Network Design and Modeling (ONDM), Budapest, 2017, pp. 1-6.

[10] S. Das, “Extensions to the OF Protocol in support of Circuit Switching”, addendum v0.3, June 2010.

[11] GENI Wiki. [Online]. Available: http://groups.geni.net/geni/wiki/OpenFlowDiscoveryProtocol

[12] S. Khan, A. Gani, A. W. Abdul Wahab, M. Guizani and M. K. Khan, "Topology Discovery in Software Defined Networks: Threats, Taxonomy, and State-of-the-Art," in IEEE Communications Surveys & Tutorials, vol. 19, no. 1, pp. 303-324, Firstquarter 2017.

[13] F. Pakzad, M. Portmann, W. L. Tan and J. Indulska, "Efficient topology discovery in software defined networks," 2014 8th International Conference on Signal Processing and Communication Systems (ICSPCS), Gold Coast, QLD, 2014, pp. 1-8.

[14] A. Azzouni, N. T. Mai Trang, R. Boutaba and G. Pujolle, "Limitations of openflow topology discovery protocol," 2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), Budva, 2017, pp. 1-3.

[15] T. Alharbi, M. Portmann and F. Pakzad, "The (in)security of Topology Discovery in Software Defined Networks," 2015 IEEE 40th Conference on Local Computer Networks (LCN), Clearwater Beach, FL, 2015, pp. 502-505.

[16] OpenDaylight, https://www.opendaylight.org

[17] R. Egorov, "Next Generation ROADM architecture and design," 2013 Optical Fiber Communication Conference and Exposition (OFC), Anaheim, CA, 2013, pp. 1-3.