# Dynamic Topology Discovery in SDN-enabled Transparent Optical 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*—Optical technologies play nowadays a crucial role in different network scenarios (e.g., data centres, metro and access networks), mainly due to their support for higher bandwidth and scalability in comparison to electronic-based opaque solutions. In this context, the deployment of the Software Defined Networking (SDN) paradigm over optical networks has gained interest from both industry and research communities, spawning several implementations over multiple well documented use cases. However, in Transparent Optical Networks (TONs), where nodes offer optical switching capabilities, neighbouring nodes adjacencies have to be manually configured at both data and control plane levels, which is a lengthy task that can potentially lead to misconfiguration. Alternative solutions rely on the use of supervisory networks or dedicated channels. Nevertheless, implementing these methods requires both effort and resources to provide adjacencies discovery, making the awareness of the correct underlying topology by the SDN controller a tedious and occasionally very complex process. In this paper, we present a novel SDN-based cost-effective topology discovery method, allowing TONs to automatically learn physical adjacencies between optical devices. In particular, this is achieved by means of a *test-signal mechanism* and the OpenFlow protocol. The SDN control plane and optical agent implementations, as well as the message exchange flow between the subsystems and the controller are described. Then, the proposed discovery mechanism is experimentally assessed over an emulated TON test-bed, analysing the average time required for the optical topology discovery in different network scenarios.**

***Keywords—Topology Discovery, Transparent Optical Networks, Optical Subsystems, SDN Controller.***

## I. INTRODUCTION

The constant growth of data traffic in current telecom networks fostered by the high bandwidth requirements of emerging applications and paradigms (e.g., cloud-based services, Internet of Things, Big Data, etc.) has driven network implementations to incorporate Transparent Optical Network (TON) technologies towards high bandwidth capacity and scalability. These technological deployments usually require efforts not only in the planning stage, but also in operation, control and support. To this end, networking solutions already deployed in electronic-based networks are also being applied to optical networks in aims of achieving an efficient use of the overall network resources.

In this regard, the current network software-ization trend has promoted deployments of network architectural solutions decoupling control plane from data forwarding functions, by means of a logically centralized control plane capable of controlling such underlying forwarding resources. This approach, commonly identified as Software Defined Networking (SDN), gives network managers the opportunity to manage, configure, automate and optimize network resources via software applications, thus avoiding inefficient per-subsystem configuration methods.

From the SDN architecture point of view, a correct mapping of the underlying topology at the control plane level is paramount to guarantee the functionality of control layer applications and the exposure of data plane resources to upper layers (e.g., application layer, orchestrator). In fact, a miscorrelation between the topology known by the controller and the actual physical topology can lead to an incorrect global visualization of how the resources are allocated and connected to each other. Additionally, it can become the root source of errors in SDN applications when deploying services over the network.

In this context, different methods have been implemented to guarantee the correct exposure of the physical topology in TONs. These methods range from fully static approaches, manually configuring neighbour information directly at the controller, to dynamic but more complex ones, such as establishing supervisory networks on top of optical subsystems to allow an automatic topology discovery. The former method completely lacks dynamicity. Hence, although it may work well when configuring small network topologies, it can become a tedious task in large-scale networks, potentially increasing the misconfiguration probability as well. The second method provides dynamic discovery, but it requires a control network to interconnect devices agents, which breaks the concept of the logically centralized control plane, a key principle of the SDN architecture [1]. Moreover, it requires the use of additional resources, which could not only result in operational and capital expenses but also become difficult to implement and control in large scale networks. It is also important to consider that enabling automatic topology discovery by using dedicated channels would require electrical link termination at incoming fibre ports of optical subsystems, thus imposing the need for additional hardware, which increases the overall complexity and cost of the network.

In the current SDN paradigm, the OpenFlow (OF) protocol [2] is enhanced with the proposed optical extensions in [3]. In this regard, the OF protocol has been recognized by the industry as the true enabler of the SDN paradigm, considering it the first standard interface specifically designed to provide communication between control and data planes [4]. Furthermore, a subset of recently documented implementations related to TONs and SDN-based software solutions have shown deployments using OF as the main communication interface between the optical subsystems and the controller [5].

The OF protocol defines specific attributes and messages to allow the exposure of neighbour information from optical devices up to the controller level. However, these extensions lack effective ways to automatically acquire such information between optical devices in TONs. All this being said, the use of the OF protocol appears as an appropriate way of exposing optical devices connectivity to a logically centralized controller. However, the use of this protocol still requires a way to avoid statically configuring neighbour data at the physical devices in order to step up an automatic discovery process.

In this paper, we present a novel cost-effective method to make the discovery process fully automatic. To achieve this goal, certain implementations and enhancements are still required at both data and control layers in order to allow the correct exposure of the topological information. From the data plane perspective, the OF agent running on top of the optical sub-system must gather its neighbours' information (i.e., data-path and port identifiers), without relying on a static configuration. Similarly as in [6], the solution that we propose to this goal is a Test Signal Mechanism (TSM), which consists in emitting light from each fibre port of a particular device and letting neighbours detect it, employing the same technique in the other way around. The detection and emission of these light signals triggers the mapping of fibre links between neighbours by exchanging specific OF messages between the agents and the controller, thus making the support by the OF agents of this technique a mandatory requirement to achieve the automatic discovery of the topology. In addition, at the controller level, the intelligence to control the exchange of the previously mentioned OF messages, the ability to retrieve peer information from them and the creation of the optical links at the controller level topology is required. Hence, a new application capable of providing such intelligence is also defined in this proposal.

The remainder of this paper continues as follows. An in-depth characterization of the proposed TSM enabling automatic topology discovery is elaborated in Section II, together with an explanation of the required implementations at the OF agents and SDN controller. Next, the mechanism is validated at the control level by performing tests over an emulated data plane test-bed, which are reported in Section III. Finally, Section IV summarizes the key achievements and conclusions of the paper.

## II. SDN-CONTROLLED TOPOLOGY DISCOVERY IN TONS

Fig. 1 depicts the general scenario of the use case under study. Therein, the SDN controller, which implements most of the intelligence required to operate the network, lies on top of the optical data plane. The optical devices are equipped with OF agents that enable their configurability via SDN. To implement the proposed discovery mechanism, the optical devices need to be provided with the capability to measure optical power in their fibre ports in order to detect the test signals arriving from neighbouring devices. In turn, the OF agents have to implement the management of the test signal emitted and received by the optical devices and send the appropriate OF messages to the controller. Finally, the controller has to be extended with a new module to handle these messages and coordinate the dynamic discovery of the optical links.

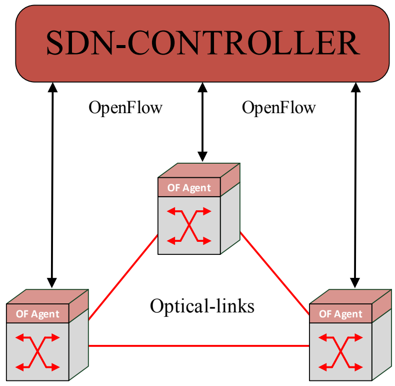


Fig. 1. SDN-enabled Transparent Optical Network.

The basics of the TSM along with the message exchange involved in the discovery process are presented in the following subsection. Furthermore, the newly implemented internal modules and the proposed extensions at the SDN control level are also introduced.

### *A. Test signal mechanism*

Discovery methods to be used in TONs vary from the ones used in electrical packet-based networks. This is mainly because optical devices lack the capability to insert or extract packets from the established optical connections, which is the basic functionality that allows discovery protocols to share peer/port information between adjacent subsystems and to expose such data towards the controller via protocols like OF. Considering such limitations, a different approach must be implemented taking advantage of the capabilities provided by TONs. In this regard, ITU Recommendations G7714 [7] and G7714.1 [8] define discovery procedures by exchanging identifier information in-fibre at the in-band trail overhead channel and using an out-of-fibre message exchange between discovery agents to verify the correlation of such information. Such in-band exchange of information is commonly supported by separate associated channels, one in each direction, between optical subsystems. The out-of-fibre exchange, on the other hand, is usually supported by a Data Communications Network (DCN) at a higher level. By applying such discovery methodology recommendations to an SDN-enabled TON scenario, it is possible to define an enhanced discovery procedure that takes advantage of the centralized control architecture. Specifically, the logically centralized control can be used to trigger the discovery process, to coordinate the data layer adjacency discovery, and to correlate link information between connected fibre ports. This eliminates the need for the aforementioned DCN between agents. Moreover, in order to avoid the use of separate associated channels, the use of orderly exchanged test signals between optical ports is proposed.

The aforementioned test signals are involved in a method that entails the transmission of light by an active emitter fibre port, its detection at the receiver port, and a final confirmation of this reception by another test signal sent through the same fibre link back to the emitter, thus requiring the presence of a transmitter in each node of the topology. The use of this method allows the exposure, via specific OF messages, of the peer/port information at both ends between which the TSM has been successfully performed. As no link correlation method is available at the data plane level due to its transparent nature, the link mapping is performed at the controller level by means of correlating such received peer/port information. In order to guarantee the correct mapping of links, the controller implements a queue where newly recognized active ports are stored. In this way, the TSM is triggered for each port of the queue sequentially. By these means, the controller is able to configure each link based on the confirmation of the successfully tested TSM by the two connected devices.

Another important consideration regarding the use of this method is that the exchange of test signals should be used while links are not carrying any traffic, which in principle will not represent an issue since the discovery process is triggered at the moment the controller recognizes a new active port. In fact, the time required to discover an adjacency through an active port should be considered negligible compared to the time the port takes in being set to begin transferring data. Moreover, the performance of the current implementation of the mechanism could be affected if used in a scenario where an optical device containing ports with active transmission of data was connected to the controller. In such situation, the TSM could not be properly initiated due to the already operational optical links. Nonetheless, this limitation could be overcome by extending the mechanism to avoid the discovery of already active ports. As a final consideration, the deletion of links at the control level topology is handled by consuming the notifications sent by the OF agents at the optical subsystems regarding the change of the fibre ports to a down state, thus allowing the process of discovery and deletion of links to be fully dynamic.

### B. *Discovery message flow*

This section describes the sequence of OF messages involved in the discovery process, which is illustrated in Fig. 2. The OF agents of the optical devices connect, first of all, to the SDN controller through a HELLO message exchange. Once the controller detects an optical device, it requests for its capabilities by sending a FEATURES_REQUEST message. Upon reception of this message, the OF agent of the optical device exposes the information of its physical ports in a FEATURES_REPLY message. At that point, the controller identifies active optical fibre ports and orderly triggers the discovery process for each one of them. To this end, the controller firstly sends a PACKET_OUT message (1) to the OF agent, specifying the active port identifier and a SET_DL_SRC output action, so the agent knows where the TSM should be used to test adjacencies. Then, the agent activates the specified port to send a test signal (2). When the neighbour detects the reception of light at one of its ports, it notifies its agent, which activates the transmission of a test signal back through the same port to confirm such reception (3). Then, it sends a PACKET_IN message with a NO_MATCH value in the reason field, and the information of the port where the signal was detected towards the controller (5). At the same time, when the first agent receives the confirmation signal at the port where the test started, it also sends a PACKET_IN message to the controller (4), but this time with an ACTION reason and the port where the initiating test was transmitted, and the test reply was detected.

These PACKET_IN messages are processed by the controller that, using the discovery module, associates the port information exposed in both messages to determine which node corresponds to the emitter or the receiver side. When such an association is accomplished, the controller realizes that there exists a link between both nodes at the data plane level. It is worth noting here that, at this point, the controller could create the link logical instance and store it in the topology. However, OF standards [3], [9] suggest that data plane devices are the ones that send the peer information to the controller. In light of this, the mechanism is finalized as follows. The controller sends empty PACKET_OUT messages (6) (7) towards both agents related to the newly discovered link with SET_DL_DST action and the link specific peer information (i.e., the peer data path and fibre port identifiers). When the agents receive these messages, they become aware of the correct adjacencies and expose the peer information via CPORT_STATUS messages to the controller (8) (9). These messages are finally consumed by the extended topology module that completes the discovery process adding the new link to the topology data store.

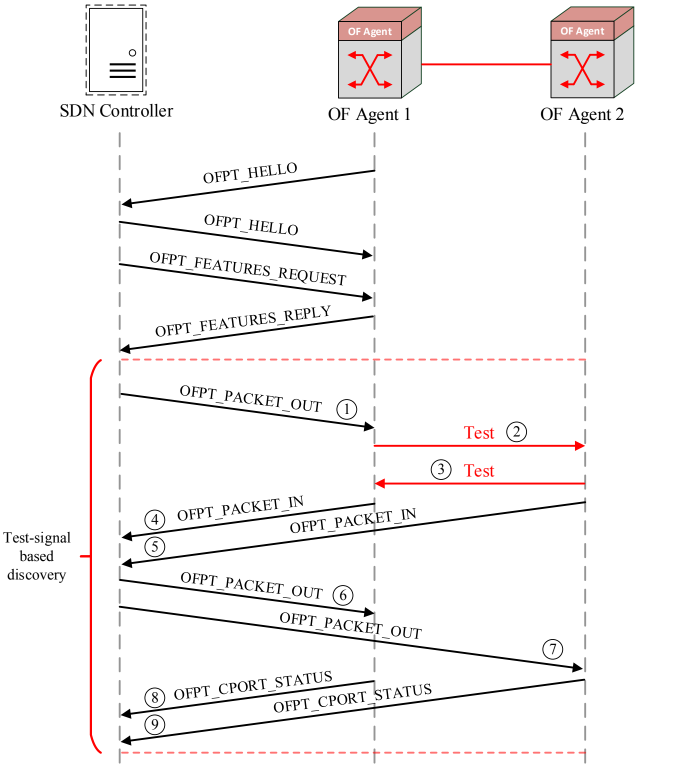


Fig. 2. OpenFlow 1.0 message flow for optical link discovery based on the proposed TSM.

### C. *Extended SDN control plane*

The overall architecture considered for the SDN-based optical topology discovery mechanism is depicted in Fig. 3, where focus is on the software-defined functionalities contained at the control layer, but considering also the optical devices/agents at the data plane and other external applications that could be running on top of the controller at the application layer. The figure also identifies how modules are logically placed inside the controller, based on their interaction with lower or higher layers. The OpenDaylight (ODL) platform [10] in its Lithium release has been selected as the base SDN controller for our implementation. This open source controller provides support for the OF protocol version 1.0 [9], which has been provided with the Optical Circuit Switching (OCS) extensions defined in [3] to configure the optical devices. In addition, ODL has been extended with a module that implements the described discovery mechanism.

At the lower layer of the controller, the OF protocol plugin implements the OF OCS extensions to provide a direct communication point with the optical devices via the so-called southbound interface. The MD-SAL represents the core communication element between modules inside the controller. This module allocates specific model-defined data structures that provide a source of self-describing data to components inside the controller, and also to external applications via northbound Application Programming Interfaces (APIs). SDN applications provide specific behaviour and intelligence defined at a software level. A set of APIs allows for the exposure of data and services to upper layers and external applications.

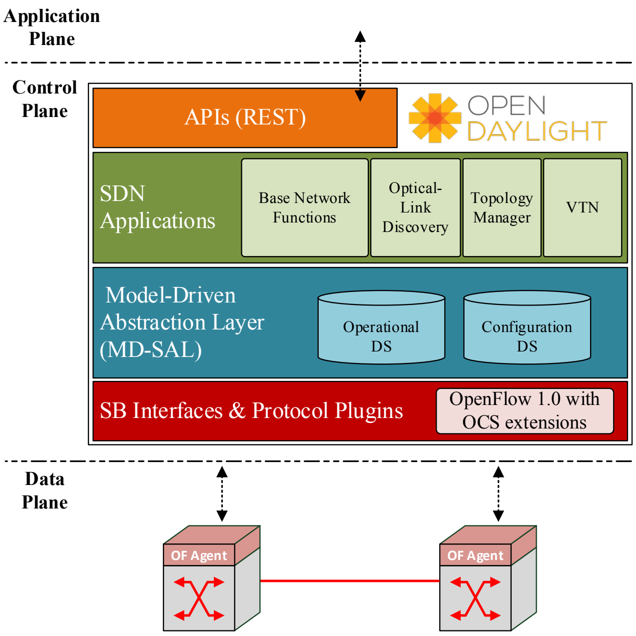


Fig. 3. SDN Architecture for TONs.

The SDN application layer has been extended to implement the proposed automatic topology discovery method and to expose this information to other applications by means of the abstraction layer (MD-SAL). In particular, the Optical Link Discovery (OLD) module has been introduced to accommodate the new requirements. This module implements the fibre ports queue to organize the sequential triggering of the discovery process by sending specific OF PACKET_OUT messages towards the OF agents of newly connected optical devices, instructing them to orderly use the proposed TSM over each of their active optical ports in order to test neighbour reachability. Moreover, the module provides the appropriate processing of the incoming OF messages associated to the discovery mechanism, being capable of mapping the optical links from the underlying physical topology and providing the OF agents with their particular peer and fibre port neighbour information. It is worth noting here, that the OF agents support the same extended version of the OF protocol and have been provided with the capability to support the proposed discovery mechanism.

Extensions to other existent modules such as the Virtual Tenant Network Manager (VTN) and the Topology Manager (TM) have also been implemented. The VTN is the module responsible for providing L2 functionalities to the network, that is, it is the responsible for processing the OF PACKET_IN messages. Hence, the VTN has been extended to forward the incoming OF messages related to the discovery mechanism to the OLD module for their processing. Besides, the TM has been adapted to handle neighbour information exposed by the OF agents via OF CPORT_STATUS update messages, so it can use these data to create the mapped links logical instances at the common topology data store, thus making available optical link information to consumers via such data store.

## III. Experimental Validation

The experimental validation of the proposed mechanism has been conducted over two emulated 9-node and 14-node TON data plane scenarios, which are illustrated in Fig. 4. In particular, the figure shows the topologies under study as depicted by the OpenDaylight controller GUI after the discovery process has successfully finalized.

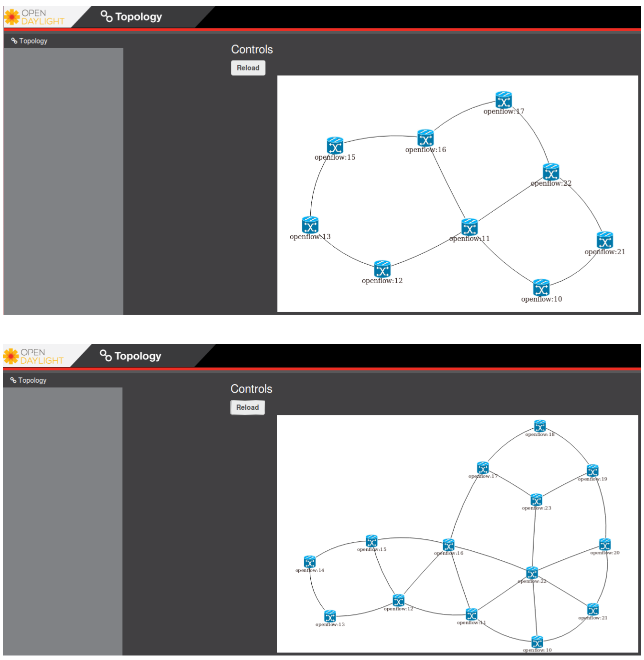


Fig. 4. Employed TON topologies as shown in the ODL graphical interface: 9-node (top) and 14-node (bottom).

| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| 124 | 13.787155000 | 192.168.101.21 | 192.168.3.1 | OpenFlow | 76 | Type: OFPT_HELLO |
| 126 | 13.787737000 | 192.168.3.1 | 192.168.101.21 | OpenFlow | 84 | Type: OFPT_FEATURES_REQUEST |
| 127 | 13.788223000 | 192.168.101.21 | 192.168.3.1 | OpenFlow | 76 | Type: OFPT_HELLO |
| 129 | 13.824290000 | 192.168.101.21 | 192.168.3.1 | OpenFlow | 340 | Type: OFPT_FEATURES_REPLY |
| 131 | 13.827185000 | 192.168.3.1 | 192.168.101.21 | OpenFlow | 80 | Type: OFPT_STATS_REQUEST |
| 132 | 13.836056000 | 192.168.101.21 | 192.168.3.1 | OpenFlow | 1136 | Type: OFPT_STATS_REPLY |
| 134 | 13.905168000 | 192.168.3.1 | 192.168.101.21 | OpenFlow | 176 | Type: OFPT_PACKET_OUT |
| 174 | 14.111269000 | 192.168.3.1 | 192.168.101.10 | OpenFlow | 88 | Type: OFPT_STATS_REQUEST |
| 175 | 14.111732000 | 192.168.101.10 | 192.168.3.1 | OpenFlow | 80 | Type: OFPT_STATS_REPLY |
| 177 | 14.111813000 | 192.168.101.10 | 192.168.3.1 | OpenFlow | 87 | Type: OFPT_PACKET_IN |
| 179 | 14.113726000 | 192.168.3.1 | 192.168.101.21 | OpenFlow | 88 | Type: OFPT_STATS_REQUEST |
| 181 | 14.114671000 | 192.168.101.21 | 192.168.3.1 | OpenFlow | 80 | Type: OFPT_STATS_REPLY |
| 183 | 14.114789000 | 192.168.101.21 | 192.168.3.1 | OpenFlow | 87 | Type: OFPT_PACKET_IN |
| 185 | 14.124050000 | 192.168.3.1 | 192.168.101.10 | OpenFlow | 100 | Type: OFPT_PACKET_OUT |
| 186 | 14.124126000 | 192.168.3.1 | 192.168.101.21 | OpenFlow | 100 | Type: OFPT_PACKET_OUT |
| 187 | 14.126141000 | 192.168.101.10 | 192.168.3.1 | OpenFlow | 164 | Type: OFPT_PORT_STATUS |
| 188 | 14.126141000 | 192.168.101.21 | 192.168.3.1 | OpenFlow | 164 | Type: OFPT_PORT_STATUS |

Fig. 5. Wireshark capture depicting the message exchange for the adjacency discovery between two network nodes.

To start, Fig. 5 illustrates the message sequence exchanged between two network nodes for adjacency discovery. More specifically, the figure shows how after a node connects to the controller via the exchange of HELLO messages, its features are exposed by means of a FEATURES_REPLY message, so the controller is able to trigger the beginning of the discovery mechanism by sending a PACKET_OUT message to the node.

Next, after the TSM is executed between the initiating node and its neighbour, PACKET_IN messages are sent to the controller from both adjacent nodes to advertise the fibre ports where the signals have been respectively emitted and detected. Finally, after the controller has successfully correlated the adjacency, it sends PACKET_OUT messages to both nodes so they can expose their neighbour information via CPORT_STATUS messages back towards the controller to configure the topology database.

Fig. 6 and 7 depict the average topology discovery time (left Y-axis) with our proposed method in the emulated 9-node and 14-node TONs. Each data point in the graph has been obtained after averaging 10 executions. In both cases, tests have been performed for a different number of fibre ports per adjacency, thus accounting for multi-fibre TON scenarios. Particularly, 1, 3, 6 and 10 bidirectional fibre ports have been configured between neighbouring nodes. In this way, it is possible to analyse the impact of the number of network nodes and links on the topology discovery time. Besides, the average time of discovery per fibre port has also been considered in order to evaluate how a higher number of links to be discovered and the weight of handling more nodes at the controller affects the discovery time of a single link depending on the scenario.

Looking at the results it is possible to perceive that the number of network nodes and links has a proportional impact on the topology discovery time. For instance, in the 9-node TON case shown in Fig. 6, the average link discovery time while connecting the nodes to the controller was around 10 seconds when a number of 22 links had to be discovered. Next, the time starts to increase gradually as the scenario introduces more links. For example, with 66 links the time is around 16 seconds and reaches up to 61 seconds in the largest scenario (with 220 links). This proportional increase in the topology discovery time, even though it depends on the number of links to be discovered, shows a quite low total time for the overall discovery process, taking also in consideration that, in this case, all the nodes have been connected to the controller at once which is not necessarily the common case in practice. As for the 14-node TON case depicted in Fig. 7, similar results are also observed. In this case, the whole process takes around 11 seconds for 46 links, 34 seconds for 138 links, 70 seconds for 276 links and 129 seconds for 460 links in the topology.

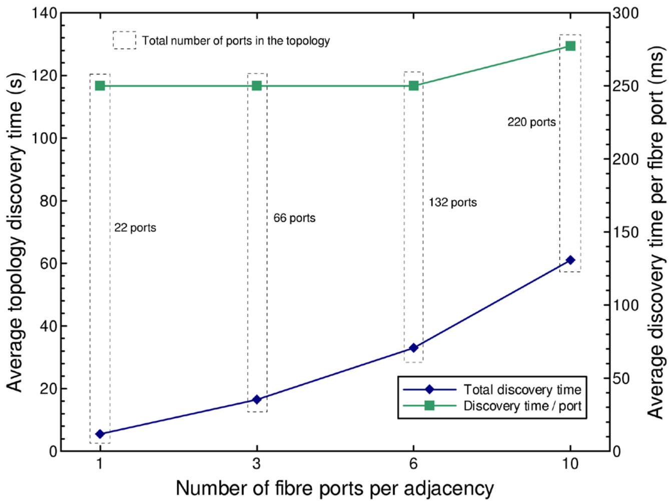


Fig. 6. Average topology discovery time (left Y-axis) and per-fibre port discovery time (right Y-axis) in 9-node TON scenario.

On the other hand, when analysing the average discovery time per fibre port (right Y-axis of the graphs), the results show that regardless of the number of nodes and links in the topology the impact on the per-port discovery time is minimal. For instance, when comparing both scenarios between cases allocating less number of links to the ones with higher amount of links, the resultant variances are around 30 to 40 ms. Considering these results and the fact that the registered time for this variable is around 250 ms in the cases with fewer links, such variance can be considered negligible and not directly related to the discovery mechanism. In fact, it can be explained as mostly associated to the higher data processing required by the controller when handling topologies with greater number of nodes and links.

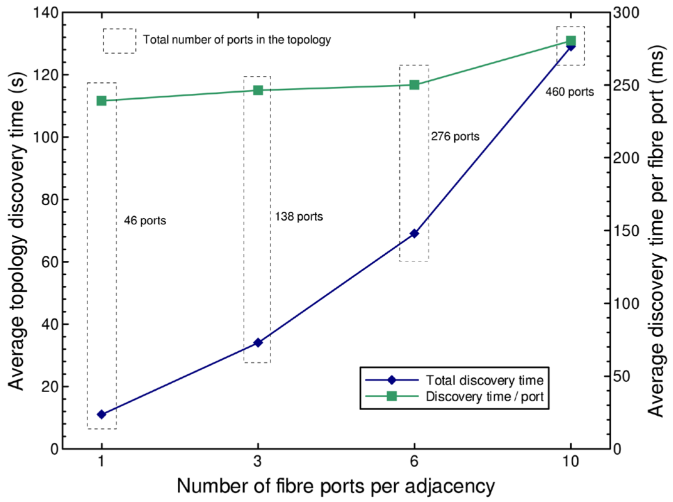


Fig. 7. Average topology discovery time (left Y-axis) and per-fibre port discovery time (right Y-axis) in 14-node TON scenario.

Consequently, it can be noted that the introduced TSM-based OF-enabled discovery procedure allows for a low-variant per fibre port adjacency discovery when increasing the number links to be discovered between two neighbouring nodes while at the same time providing of a relatively low total time of topology discovery in different scenarios, thus remarking the scalability of the overall proposed discovery mechanism.

## IV. Conclusions

With the more common application of optical technologies in modern network implementations, more particularly in the case of TONs, the discovery of the optical topology when using SDN to control network resources, has been identified as a crucial matter for the correct function of SDN applications that require a precise visualization of the topology to be able to provide the provisioning of services over the network.

In this regard, an optical topology discovery procedure is presented in this paper to address these necessities, by using optical test signals and the OpenFlow (OF) protocol. Such a method has been experimentally validated in a 9-node and 14-node emulated transparent optical networks with different numbers of parallel fibres between neighbouring nodes. The obtained results not only demonstrate the correct mapping of the topologies at the controller level, but also show low total times of network topology discovery and a low variance in per-fibre port discovery time when the method is applied to scenarios with different amount of configured links and nodes.

## Acknowledgment


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


## References


[1] ONF, "SDN Architecture Overview", Version 1.0, Open Networking Foundation, December 2013.

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

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

[4] "Software-Defined Networking: The New Norm for Networks". Open Networking Foundation, April 2012.

[5] A. Pagès et al., "Experimental Assessment of VDC Provisioning in SDN/OpenStack-based DC Infrastructures with Optical DCN," 42nd European Conference on Optical Communication (ECOC 2016), Dusseldorf, Germany, September 2016, pp. 1-3.

[6] J. Perelló, E. Escalona, S. Spadaro, J. Comellas and G. Junyent, "Resource discovery in ASON/GMPLS Transport Networks," in IEEE Communications Magazine, vol. 45, no. 10, pp. 86-92, October 2007.

[7] ITU-T Rec. G.7714/Y.1705, "Generalized Automatic Discovery for Transport Entities," August 2005.

[8] ITU-T Rec. G.7714.1/Y1705.1, "Protocol for Automatic Discovery in SDH and OTN Networks," April 2003.

[9] ONF, "OpenFlow Switch Specification", Version 1.0.0, , December 2009.

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