Integration of Legacy Non-SDN Optical ROADMs in a Software Defined Network

Imad Alawe, Bernard Cousin, Olivier Thorey, Rodolphe Legouable

Integration of Legacy Non-SDN Optical ROADMs in a Software Defined Network

Imad Alawe
IRT b<>com
Rennes, France
Imad.Alawe@b-com.com

Bernard Cousin
IRT b<>com
Université de Rennes 1/IRISA
Rennes, France
Bernard.Cousin@irisa.fr

Olivier Thorey
IRT b<>com
Ekinops
Rennes, France
Olivier.Thorey@b-com.com

Rodolphe Legouable
IRT b<>com
Orange
Rennes, France
Rodolphe.Legouable@b-com.com

***Abstract*—Optical networks are well known to offer high rate bandwidth by sending data over light wavelengths. Those optical networks are composed of fiber optics, amplifiers, ROADMs, etc. Nowadays, we wish to manage and monitor that type of networks, especially the ROADMs, flexibly and intelligently by integrating SDN in optical networks. Unfortunately, legacy ROADMs aren't SDN compatible yet. They have all the basic optical functionalities we wish to deploy (offered via the SNMP protocol in our case). However, we aren't always able to update hardware or software in the ROADMs to adapt legacy ROADMs to SDN architecture. The proposed architecture in this article, a proxy that translates OpenFlow messages sent by ONOS into SNMP messages to configure the ROADMs and conversely, is a solution. This solution allows not only to flexibly monitor and manage an optical network via SDN architecture, but also to recover and reroute wavelengths when a link is down. It is also the most adapted solution to legacy networks because it doesn't require any upgrade on the optical network elements.**

***Keywords*—SDN, OpenFlow, SNMP, Optical Networks.**

I. Introduction

Currently, any core network must handle enormous amounts of data, mainly due to the popularity of new services as: video, cloud computing or other new bandwidth-intensive networked applications. As yet, optical networks are appropriated for handling increased bandwidth demands. However, they are not very flexible, difficult to provision and challenging to manage. As new services arise and become significantly popular, network operators need to rapidly adapt and balance network capacity and service requests. Network operators also need to decrease the time of deployment and therefore reduce their cost of deploying, operating and scaling optical core networks.

In order to provide faster packet forwarding to answer the digital society needs, Multiprotocol Label Switching (MPLS) was introduced in the late 1990s. Few years later, Generalized Multi-Protocol Label Switching (GMPLS) [1], a protocol suite extending MPLS, emerged to introduce this new switching technology for other interfaces than packet interfaces (i.e. time-division multiplexing (TDM), wavelength switching and fiber-switching).

Following MPLS/GMPLS, Software Defined Networking (SDN) has emerged recently, to allow networks to be flexibly deployable, elastic, cost-efficient and programmable. SDN proposes to use a unified control plane suitable for managing both Internet Protocol (IP) packet networks and optical networks offering significant advantages over GMPLS as mentioned in [2].

In optical networks, legacy equipment may typically be configured via the Simple Network Management Protocol (SNMP) [3]. Its a simple, light protocol allowing to control the required management interactions, including imperative actions [4]. Optical networks are typically managed by a Network Management System (NMS), which uses SNMP. So, the combination of SNMP protocol and the NMS offers programmability and monitoring in optical networks. However, it still lacks flexibility and elasticity.

SDN is a solution to add logically centralized control and flexibility to optical networks. In this case, legacy SNMP optical equipment and SDN controller must be modified in order to form a homogenous working system.

Many attempts are conducted to integrate SDN in legacy networks and adapting the network equipment to insure full communication between the SDN controller and the network elements.

One of the proposed solutions is accommodating legacy switches in an OpenFlow (OF) network [5]. It consists of developing a proxy that interprets OF commands received from the controller and applies the actions in a real switch. Each OF rule is translated in a legacy switch as a Virtual Local Area Network (VLAN). This solution is only applicable to Packet IP networks, and may not work in large scale networks due to the limitation of number of VLANs supported by a legacy switch.

A second solution was proposed by [6]. It consists of integrating an OF Management Gateway in order to manage OF networks with legacy NMSs. This solution may be useful to enable communication between an OF network and a legacy Packet IP network. However, it is not adapted to optical networks.

This paper proposes SDN architecture to control, manage and monitor an optical network based on SNMP optical equipment using an SDN controller and an OF to SNMP proxy. Proxy is also the most most adapted solution to legacy networks because it doesn't require any upgrade on neither the optical network elements nor the SDN controller.

This article is organized as follows: Section II introduces

SDN, OF protocol, optical networks and SNMP protocol. Section III lays down our proposition. Section IV presents the proof of concept. Finally, conclusions are drawn in Section V.

II. Context

A. SDN

Based on [7], SDN main goal is to break the vertical integration by separating the control plane and the data plane. With this separation, network equipment become simple forwarding devices and the logic is controlled by a logically centralized controller, simplifying policy enforcement, network (re)configuration and network evolution.

On one side, SDN is a new network architecture with four pillars:

1) Control plane and data plane are decoupled. Network devices will become simple forwarding devices.
2) Forwarding decisions are flow based, instead of destination based.
3) Control logic is moved to an external entity (SDN controller).
4) The network is programmable through software applications running on top of the SDN controller.

On the other side, SDN can also be defined by three fundamental abstractions: Forwarding, Distribution, and Specification.

1) Forwarding abstraction should allow any forwarding behavior desired by the network application while hiding details of the underlying hardware.
2) Distribution abstraction should make the distributed control problem a logically centralized one.
3) Specification abstraction should allow network applications to express the desired network behavior without being responsible of implementing that behavior itself.

B. OF Protocol

The OF protocol [8] was defined by the Open Networking Foundation (ONF) as the first standard communication interface between a SDN controller and network devices of SDN architecture.

In OF protocol, the communication is established between a network device and its SDN controller through an OF channel initiated by the ROADM.

Mainly, OF uses flows as rules to be applied on the traffic. Those flows are defined by the controller and stored in flow tables on both sides (the network device and the controller).

An optical flow in OF can be defined as depicted in Figure 1.

- Match: the criteria to match against traffic.
- Actions: the actions to apply on the matched traffic. For instance, Switch, Add, Drop or Delete an optical flow.
- Statistics: the number of matched traffic, dropped traffic, etc.

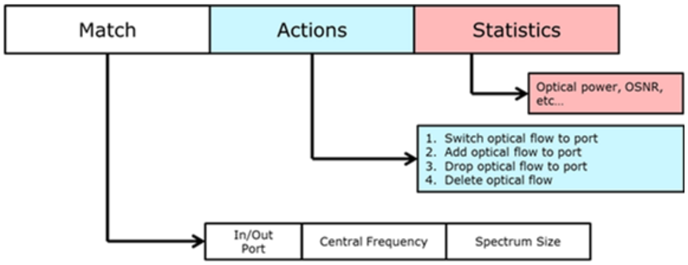
Match
Actions
Statistics
Optical power, OSNR, etc...
1. Switch optical flow to port
2. Add optical flow to port
3. Drop optical flow to port
4. Delete optical flow
In/Out Port
Central Frequency
Spectrum Size

Fig. 1. Flows for wavelength switching in optical networks

So when data is received by a ROADM, it tries to match the traffic to a flow. In optical network, this match is based for instance on the input port and/or on the central frequency of the wavelength. If no matching criteria were defined for the data, the NE (Network Equipment), i.e. the ROADM, will send the data to the SDN controller. However, if a matching criterion is found, the ROADM applies the specific action on the data and the statistics are updated.

C. Optical Networks

Optical networking transmits encoded data onto light through optical nodes of a network. In optical networks specific equipment are used. It includes fiber, laser, prism, multiplexer/demultiplexer, reconfigurable optical add-drop multiplexers (ROADM), optical amplifiers etc.

Wavelength division multiplexing (WDM) is a technology used in optical networking. By using different wavelengths from lasers, it allows to multiplex a number of optical carriers into a single optical fiber.

Figure 2 , a ROADM adds to optical networks the ability to remotely switch traffic from a WDM system. It allows individual or multiple wavelengths to be added and/or dropped from line fiber without affecting the transit traffic.

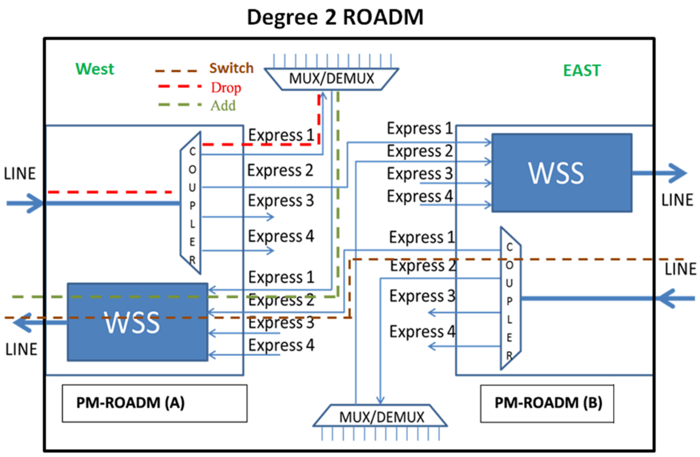

Fig. 2. Degree 2 ROADM Architecture (ADD/DROP/SWITCH)

For instance, one possible architecture of a ROADM is the following. On one side, when a wavelength gets in a ROADM on a line input port, an optical coupler reproduces the same signal on each express output port. On the opposite side, the Wavelength Selective Switching (WSS) selects one express input port and passes out its data on the line output port depending on its configuration.

In standard bi-directional application, all actions (Add/Drop/Switch/Delete) are applied on both directions. An “Add”action on the east direction line is coupled to a “Drop”action on the west direction line and conversely. For the “Switch/Delete”action, it is the same on both directions.

D. SNMP Protocol

Simple Network Management Protocol was defined by the Internet Engineering Task Force (IETF) and is a component of the IP Suite. It main purpose is to manage and monitor network elements.

The SNMP protocol consists of:

- SNMP Manager/NMS: monitors and controls managed network elements.
- Managed devices: the network elements that need to be monitored and managed.
- SNMP Agent: an application that collects, stores, retrieves management information and informs the manager about an event.
- Management Information Base (MIB): a database describing the managed device parameters.

The SNMP protocol operates in the Application Layer. Its basic commands are:

- GET Request: to request a value of a parameter in the MIB.
- GET NEXT Request: to request the value of the next parameter in the MIB.
- GET BULK Request: to request the values of many parameters in the MIB.
- SET Request: to attribute a value to a parameter in the MIB.
- TRAP: to inform the manager of an event occurrence.
- Response: The answer of a request.

III. Proposed Architecture

In optical networks, legacy ROADMs are deployed and support neither SDN architecture nor OF Protocol. However those ROADMs offer functionalities that are deployed manually today or just managed by a NMS using the SNMP protocol. Those ROADMs may be automatically managed and controlled by an SDN controller offering flexibility, elasticity. In order to achieve that and make optical networks SDN compatible, we propose the following solution, described in Figure 3 .The proposal consists of integrating a proxy between the SDN controller and each ROADM.

In our solution, we chose Open Network Operating System (ONOS) [9] as the SDN controller. ONOS is an open source SDN controller able to manage and monitor packet layer and optical layer. The proxy is an application coded in JAVA language. It communicates at the northbound interface with the SDN controller through OF, and in the southbound interface with the ROADM through SNMP.

To offer scalability, a proxy process is in charge of only one ROADM. Using this approach allows parallel network management operations. Also, in a case of a proxy failure, it restricts failure only to the ROADM associated to the proxy.

The proxy implementation is flexible. It can be implemented either on the SDN controller side by running the proxy process on the server running the SDN controller; or on the ROADM side by running the proxy process in the chassis management card; or on any station interconnected to internet and having access to the management network.

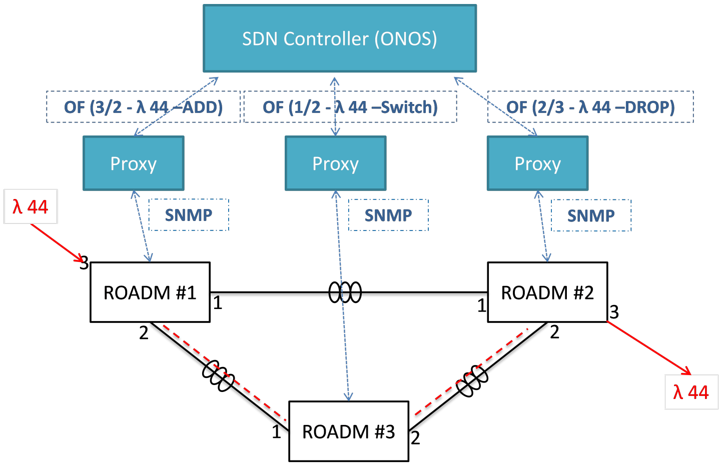

Fig. 3. Proposed Architecture

The benefit of this solution is that the SDN controller has the entire control of the network, the topology and the resources. It is also the most adapted solution to legacy networks because it doesnt require any upgrade on the optical network or on the ROADMs.

On first run, the ROADM is declared to the proxy (for instance using a discovery protocol or by the network operator through command line tool). The proxy connects to the ROADM and verifies the information of the ROADM (Valid IP address, Valid ROADM parameters, etc.) by using SNMP protocol basic command GET NEXT Request. Once the information is verified, the proxy declares the ROADM to ONOS via OF by establishing an OF Channel. The OF channel establishment starts by a Hello OF message initiated by the proxy and sent to the SDN controller (ONOS). Once received, ONOS responds also by a Hello OF message followed by a number of initialization messages (Feature request, Port description request, Experimenter Request, Role Request). Actually the OF protocol version 1.4 supports optical ports, contrarily to the 1.3 version. However the 1.4 version is not supported by ONOS yet. So to support optical layer, ONOS uses Experimenter OF messages to declare the optical ports and all its metadata. Finally once the proxy responds to ONOS, the master role reply OF message, the OF Channel establishment is done and the ROADM is completely declared to ONOS and ready to be managed.

Before creating routes, optical links should be declared to ONOS because it doesn’t support automatic detection yet. In our case we choose to push optical links to ONOS using a JavaScript Object Notation (JSON) file [10]. In the JSON file, the links are described by indicating the endpoints of each link.

At this point, ONOS may receive a route demand from the user. In that case, it computes the best possible route on the optical network (e.g. the wavelength $\lambda 44$ on the route passing respectively by the ROADM #1, ROADM #3 and ROADM #2, cf. Figure 3). Then, it sends an OF flow rule to the proxy of each ROADM concerned in the route. For example, the flow rule sent to the proxy of the ROADM #1 (cf. Figure 3) could be illustrated as follow:

- OF: OpenFlow protocol.
- 3/2 - $\lambda 44$: The wavelength number 44 from source port number 3 and in destination to port number 2.
- ADD: Apply an ADD action.

Once the concerned proxy receives the flow rule, it analyses it, translates it into SNMP configuration messages compatible with legacy ROADMs.

The OF to SNMP translation consists on analyzing the OF flow rule by the proxy in order to determine which wavelength and which action (Add/Drop/Switch/Delete) to configure on the ROADM. Once those data are collected, the proxy prepares the corresponding SNMP Get commands (i.e. GET "WaveLength plan table"), and SNMP Set commands (i.e. SET "Switch on $\lambda 44$"), based on the ROADM SNMP Management Information Base (MIB), in order to apply the desired configuration.

Once the SNMP command is ready, the proxy will transfer it to the ROADM.

IV. Proof Of Concept (POC)

In Ekinops laboratory, three chassis were prepared with a two degree ROADM in each. They were interconnected as depicted in Figure 4. Two computers were used in this demo; one on which the SDN controller ONOS is running and the second one where the Java proxy applications are launched. We have not modified the software of neither the SDN controller nor the ROADMs. We have just introduced the proxys.

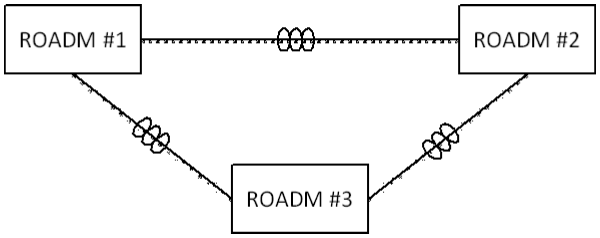

Fig. 4. POC Architecture

As illustrated in Figure 5, the POC starts by an equipment discovery. The proxies verify the existence of the equipment using a SNMP GET NEXT message in order to verify the given chassis components and then starts an OF Channel establishment with ONOS. Once all the network elements are declared to ONOS, the optical links are described to ONOS by pushing a JSON links description file.

A user application sends a request to ONOS to create an optical route between two ROADMs. ONOS chooses the best route to use, and sends the corresponding flow rule to each proxy in question. Each proxy analyses the flow rule received, translates it into SNMP configuration command as mentioned in section III and pushes it to the ROADM. Once the network configuration is done, the user application is able to use the optical channel (for instance it may ping the ending point from the starting point and conversely).

The reactivity of ONOS is demonstrated by forcing a Link down on the route. ONOS detects the Link failure. Following the computation of a new path between the two points, ONOS asks the proxies to remove the old flow rules and sends to them the new flow rules to apply. Each proxy analyzes the received messages from ONOS, removes the old applied configurations on the ROADMS and then applies the new ones. At this point, all traffic is rerouted on the new path.

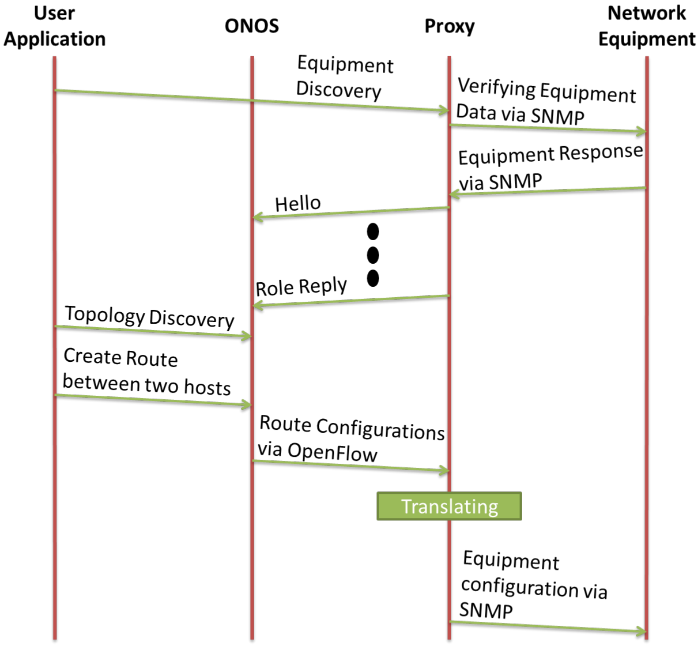
User Application
ONOS
Proxy
Network Equipment
Equipment Discovery
Verifying Equipment Data via SNMP
Equipment Response via SNMP
Hello
Role Reply
Topology Discovery
Create Route between two hosts
Route Configurations via OpenFlow
Translating
Equipment configuration via SNMP

Fig. 5. Proof of Concept Global Process

V. Conclusion

Optical networks are known for handling increased bandwidth demands. They are very well managed using SNMP and a NMS but it still lacks flexibility and elasticity. SDN appeared recently allowing networks to resolve the situation and many testbeds demonstrated its capabilities. By Integrating SDN to optical networks, the control plane is centralized in an SDN controller allowing flexibility, elasticity and also many other benefits depending on use cases.

In our paper, we proposed an SDN architecture adapted to include legacy non-SDN ROADMS. The key is to use a proxy that translates the OF messages sent by the controller into SNMP commands to apply the desired configurations on the ROADM and vice-versa. Moreover, our proposal does not require any software modification of the SDN controller or ROADM SNMP agent.

Our Proof of Concept, at Ekinops Labs, demonstrates the possibility of flexibly monitoring and managing an optical network via SDN architecture, and demonstrates the reactivity of ONOS when optical link fails.

References

[1] E. Mannie, "Generalized multi-protocol label switching (gmpls) architecture," *Interface*, vol. 501, p. 19, 2004.

[2] S. Das, G. Parulkar, and N. McKeown, "Why openflow/sdn can succeed where gmpls failed," in *European Conference and Exhibition on Optical Communication*. Optical Society of America, 2012, pp. Tu–1.

[3] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "Simple network management protocol (snmp)," Rfc 1157, 1990.

[4] A. Carvalho, F. De Carvalho, and G. Pavlou, "Management of optical networks: Snmp agents enabling tmn," in *Towards a Pan-European Telecommunication Service Infrastructure (IS&N'94)*. Springer, 1994, pp. 181–191.

[5] F. N. Farias, J. J. Salvatti, E. C. Cerqueira, and A. J. Abelém, "A proposal management of the legacy network environment using openflow control plane," in *Network Operations and Management Symposium (NOMS), 2012 IEEE*. IEEE, 2012, pp. 1143–1150.

[6] S. Waldbusser, "Monitoring gateway systems and methods for openflow type networks," Feb. 19 2015, US Patent App. 14/626,861.

[7] D. Kreutz, F. M. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[8] Open Networking Foundation, "Openflow switch specification."

[9] "Onos project," http://onosproject.org/.

[10] T. Bray, "The javascript object notation (json) data interchange format," RFC 7159, 2014.