Can Open Flow make transport networks smarter and dynamic? An overview on Transport SDN

Rodolfo Alvizu, Guido Maier
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano
Milan, Italy
rodolfoenrique.alvizu@polimi.it, guido.maier@polimi.it

***Abstract*—The growth of intra data center communications, cloud computing and multimedia content applications force transport network providers to allocate resources faster, smarter and dynamically. Software-defined Networking (SDN), has been proposed to create a unified control plane for transport networks (Transport SDN). This article presents an overview on Transport SDN proposals based on OpenFlow, the de facto SDN protocol. OpenFlow is at the forefront of the Transport SDN models and several testbeds have proved the implementation of a unified control plane for multi-domain and multi-technology optical transport networks. We show how OpenFlow can be enabled in current and future network devices through agents and new hardware respectively. Transport SDN can boost the programmability and scalability of the network, increase the network intelligence and allow for dynamic resource allocation and restoration. The review highlights a rapid development of Transport SDN, which seems to tackle the problems that GMPLS encountered for commercial deployment. Finally a comparison between the main research efforts towards a multi-domain transport SDN is given.**

Keywords—Software Defined Networks; Optical Networks; Transport SDN; Open Flow; GMPLS; IP and Optical convergence; Unified Control Plane.

I. INTRODUCTION

The fast evolution of mobile devices, multimedia content and cloud computing applications, together with the rise of intra data center communications, exacerbate the need for an intelligent and dynamic transport network, and for multi-domain schemes to increase the resource allocation efficiency of the backbone segment.

In general, the transport network is composed by layer 0 (photonics) and layer 1 (Synchronous Optical Network/ Synchronous Digital Hierarchy and Optical Transport Network) circuit switched connections. It includes heterogeneous vendor-specific solutions and has to deal with analog domain constraints related to photonics. As a consequence, to implement an efficient unified control plane for smart and dynamic multi-domain optical networks is a challenging task.

General Multi-Protocol Label Switching (GMPLS) was conceived as the first unified control plane for optical transport networks, for dynamic resource provision and network survivability assurance [1]. GMPLS extends MPLS to cover circuit switching technologies (e.g., time slots, wavelengths and fibers). Despite a long-lasting standardization process and a number of GMPLS-compliant transport equipment implementations, today there are no large commercial deployments of GMPLS as unified control plane because of its high level of complexity [2, 3].

A new promising solution to create a unified control plane for multi-domain optical transport networks is based on Software Defined Networks (Transport SDN). The main difference between GMPLS and transport SDN is the centralized nature of SDN, instead of a distributed control of GMPLS.

The Open Networking Foundation (ONF) [4], an organization dedicated to the promotion and adoption of SDN, defines SDN as: "*an emerging network architecture where network control is decoupled from forwarding and is directly programmable*". SDN has created a revolution into the networking world. Data centers and big companies like Google were the first to deploy SDN based solutions [5]. OpenFlow, the de facto SDN standard for communications between the data plane and the control plane, was designed for packet switching networks [6]. OpenFlow provides a data plane abstraction based on flow tables. A flow is an n-tuple of any combination of layer 2 to layer 4 headers.

There are two works that elaborate on the comparison between GMPLS and OpenFlow based control plane for transport networks [3, 7]. The discussion presented in [3] arguments that the SDN model is superior in terms of complexity, programmability, extensibility, and adoption path. The second work presents and experimental assessment of performance by means of two testbeds of 1000 nodes, which demonstrated that the SDN model is superior in terms of blocking probability, wavelength utilization and lightpath setup time [7].

It is interesting to notice that the main ongoing efforts on Transport SDN are OpenFlow-based. In fact, the data plane abstraction of OpenFlow can be easily extended to support circuit switching technologies. Hence, this article elaborates an overview on transport SDN focusing on OpenFlow-based models to answer the question: Can OpenFlow make transport networks smarter and dynamic?

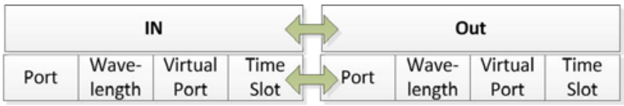


Fig. 2 OpenFlow circuit switch cross connection table.

The article is organized as follows. In section II the main OpenFlow-based models for transport SDN are described. In section III a comparison table and a discussion of the described models are presented. Finally in section IV the conclusions of the overview are exposed.

II. OPENFLOW- BASED SOLUTIONS FOR TRANSPORT SDN

In this section an overview of proposed OpenFlow-based solutions for transport SDN is presented. The classification made in [8] is an interesting criterion for sorting the OpenFlow-based transport SDN solutions. Adopting a similar classification criterion, we sort the main research efforts in four working groups.

A. *Packet and Circuit Network Convergence (PAC.C)*

The Packet and Circuit Network Convergence is the first result of Stanford studies on extensions to OpenFlow in support of circuit switching [9, 10]. The proposal of PAC.C was motivated by the OpenFlow innovation capabilities and the fact that the data plane abstraction of OpenFlow can be easily extended to support circuit switching. The authors showed how the cross-connect tables of transport switches can be adapted to the OpenFlow data plane abstraction, and become a circuit flow table. The flows in the cross-connect tables are defined as layer 1 and layer 0 circuit flows.

It is important to notice that in PAC.C the circuit flow table is not used to perform circuits' lookup as in the case of flow tables from vanilla OpenFlow (packet switching). In the OpenFlow extensions, the Circuit flow table corresponds with established circuits in the switching matrix.

As OpenFlow is mainly focused on packet domains, the OpenFlow Circuit Switched Addendum v.03 [11] presents the required OpenFlow protocol extensions (version 1.0) to support circuit switching technologies, and the description of OpenFlow circuit switches. In the addendum, the optical cross connect keeps a circuit switching flow table (separated from the packet switching flow table) defined as layer 1 or layer 0 circuits. Fig. 1 presents the specific circuit switched cross-connection table proposed in the Addendum. The circuit flows are defined by four fields per input and output ports, specifically, port number, wavelength, virtual port associated with the Virtual Concatenation Group (VCG) and starting time slot of the SONET/SDH allowing Ethernet/TDM convergence. The VCGs are used for mapping packet flows to circuit flows, allowing the interconnection between packet and circuit domains. Fig. 2 depicts the hybrid packet-circuit switch proposed in the specification [11]. An OpenFlow circuit switch will be composed by the right part of the hybrid switch.

The extensions proposed in [9-11] allows for wider and flexible definition of flows, which are defined as combination of headers from Layer 2 to 4 and circuits from Layer 1 and 0. Moreover, the circuit switching flow tables in the cross-connects can be dynamically updated in order to adapt the transport network to traffic pattern variations or failures. PAC.C extensions lead to a unified control plane to manage OpenFlow enabled packet switches, circuit switches and hybrid switches.

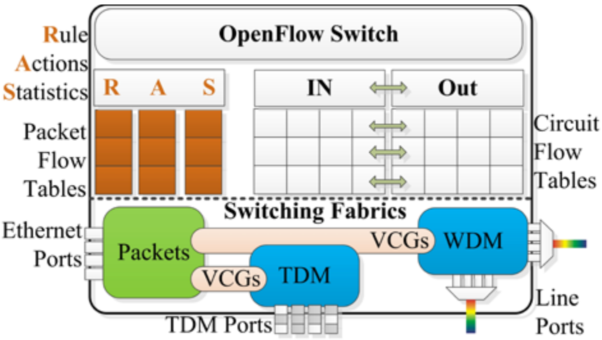


Fig. 1 OpenFlow hybrid switch with packet and circuit flow tables.

The authors of PAC.C demonstrated a converged Ethernet/TDM network using extended OpenFlow and the NOX controller [12]. They implemented an application for dynamic circuit Switching which provides application-aware aggregation and traffic engineering [13].

The hybrid packet optical circuit switch was proposed to replace backbone routers in order to achieve a fully meshed IP core [14]. Potential cost savings were evaluated by a detailed analysis of the capital expenditure. A saving of 60% was assessed for a typical backbone operator.

PAC.C is aimed to an efficient unified control plane thus, the proposed model directly integrates with OpenFlow's packet switch model. However it incurs in substantial architectural changes that need to be implemented to support OpenFlow in the transport network elements (NE). Therefore, PAC.C wager for a disruptive model from current network elements of transport technologies.

B. *OpenFlow in Europe Linking Infrastructure and Applications (OFELIA)*

The European project OFELIA proposed OpenFlow-based transport SDN which includes for the first time both fixed and flexible grid optical networks [15-18]. The support of flexible grid DWDM technology (Flexi-Grid) in OFELIA's proposal follows the progressive advances and promises of this technology. In flexi-grid spectral and modulation format flexibility is introduced, increasing the adaptability of lightpaths to meet the variable requirements of services and applications of users [18].

The main feature of OFELIA's model is the abstraction of the optical switch created by introducing a piece of software called OpenFlow agent. The agent bridges the lack of OpenFlow support at hardware level in commercial optical transport equipment (e.g., the flow table presented in Fig. 2). Thus, OpenFlow can be enabled in current optical nodes just by adding a software agent, providing a smooth transition path towards transport SDN.

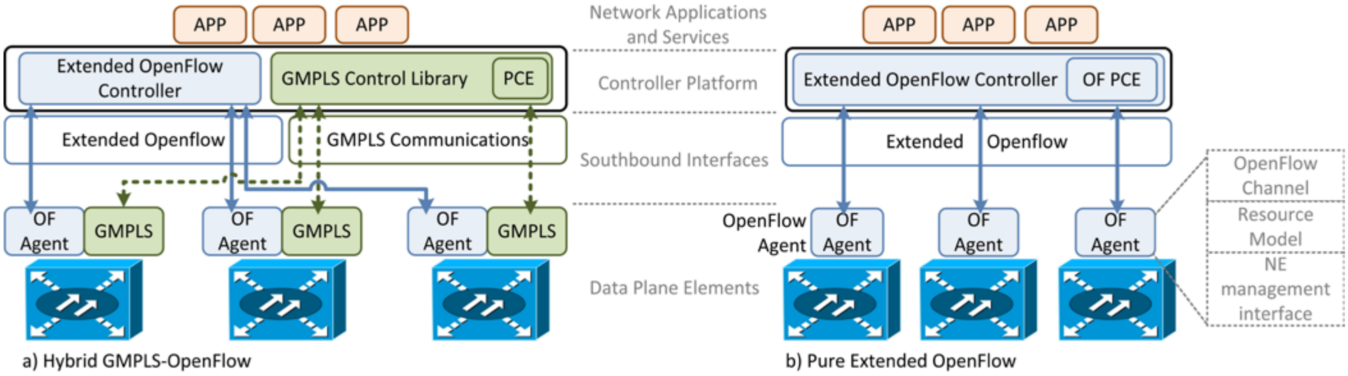


Fig. 3 OFELIAs' OpenFlow-based SDN optical network for packet and circuit switching. a) Hybrid GMPLS-OpenFlow. b) Pure Extended OpenFlow.

The OpenFlow agent is composed by three vertical modules as depicted at the right side of Fig. 3b. From top to down, the first module is the OpenFlow Channel, responsible of the establishment of a secure channel with the controller. The Second module is the Resource Model, which creates a generic abstraction of the optical data plane. Such abstraction is composed by the circuit flow tables, the multi-domain mapping information (e.g., mapping packet to circuit) and the vendor specific NE parameters: switching and power constraints, recovery mechanisms and optical layer impairments. The third module interfaces the agent with the data plane exploiting the NE's management interface, e.g., the Simple Network Management Protocol (SNMP) or vendor-specific API.

Two approaches have been presented by OFELIA, both based on the OpenFlow agent: Hybrid GMPLS-OpenFlow (Fig. 3a) [17] and Pure Extended OpenFlow (Fig. 3b) [16]. In the Hybrid approach, the authors reuse the standardized GMPLS control plane to offload the OpenFlow controller from part of the circuit switching complexity. As presented in Fig. 3a there are two separated control platforms and communication protocols. Extended OpenFlow is used for topology and resource information control that can be delivered to applications. GMPLS control plane is responsible for lightpath computation, establishment and verification reusing the GMPLS Path Computation Entity (PCE). GMPLS control plane, through the device management interface protocol, compute vendor-specific physical layer constraints.

Two methods for lightpath establishment were considered, called loose and explicit [15]. In the loose lightpath establishment only the edge nodes and ports are specified by OpenFlow and the GMPLS control plane manage the path computation and establishment. In the explicit method the OpenFlow controller exploits the centralized information on topology and resources to compute the lightpaths, while the GMPLS control plane manages verification and establishment of the computed lightpaths.

The pure Extended OpenFlow approach (Fig. 3b) is similar to the PAC.C proposal. However, the OpenFlow agent exchanges information with the network elements and the controller (extended NOX) through the extended OpenFlow protocol and the network element management interface.

The extended NOX is responsible for creating the topology, using the switch request/reply features). By exchange of CFLOW_MOD messages with the agent, the controller can manage the cross-connection flow tables, it is thus in charge of controlling lightpath establishment and teardown. Through generic vendor extension messages, the controller gathers resource and switching-constraint information, to be used by the OpenFlow PCE module (OF PCE). The OF PCE module is nested in the extended NOX, and it is responsible for constraint-aware lightpath computation.

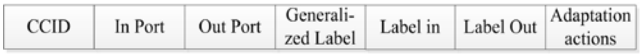
For identification and control purposes of several optical transport and switching technologies, including elastic optical networks, the OFELIAS's proposal defines the following fields: optical flow identifier, wavelength or center frequency (CF) for fixed and flexi-grid technologies, bandwidth, signal type and physical layer constraints.

For achieving multi domain capabilities, OFELIAS's model defined two kinds of flow tables: intra domain (same domain) and inter domains (interface of different domains). By defining both tables, the constraints related to actions involving different domain flows (e.g., packets, fixed grid, flexi grid) can be specified to the controller using multi domain mapping rules.

Despite the relieve of some optical switching related issues that the Hybrid model obtains through GMPLS, the evaluation results presented in [18] exposes that the Pure Extended OpenFlow outperforms the Hybrid OpenFlow-GMPLS model in terms of path-setup times and control stability.

C. Multi-layer Multi-region (ML-MR)

The Multi-layer Multi-region proposal [19] is based on OpenFlow 1.1 protocol extensions that envision a unified control of ML-MR transport network switches. The name comes from GMPLS that refers to switching technologies as regions instead of domains, while the different granularities inside regions are called layers. The authors of [19] believe that GMPLS is a key to enable an easier and gradual migration towards Transport SDN. Thus ML-MR has been proposed as a GMPLS-based model; specifically it is based on the GMPLS way of provisioning new connections, reusing the standardized label encodings. However, in the context of SDN the



CCID	In Port	Out Port	Generalized Label	Label in	Label Out	Adaptation actions

Fig. 5 GMPLS-based OpenFlow circuit switch cross connection table.

distributed nature of GMPLS is dropped, in order to implement a centralized controller.

To describe the circuit characteristics, the packet and circuit port structure is also reused from the GMPLS specifications. ML-MR emulates the Label Switched Path (LSP) of GMPLS to create interconnections between packet switches, layer 2 packet switches (e.g., Ethernet bridges), TDM circuit switches (e.g., SONET/SDH), lambda and fiber switches (e.g., optical cross connects) [8].

ML-MR requires a set of extensions similar to those proposed in PAC.C [11]. Notwithstanding, the OpenFlow extensions proposed by ML-MR reuse GMPLS path establishment signaling. Specifically, ML-MR introduces the labels exchanged by the Resource Reservation Protocol - Traffic Engineering (RSVP-TE). TE and PCE capabilities of GMPLS are included in the OpenFlow controller. Information gathering from devices to the controller remains (to our knowledge) as an open issue for this proposal.

The ML-MR authors' position about the OpenFlow agents is that it is a short term solution in enabling OpenFlow on transport layers and that it is not an efficient solution for packet-optical integration [8]. Hence, the interface between the OpenFlow communication channel and the data plane is presented as vendor specific implementation. ML-MR only specifies that the OpenFlow communications from controller to nodes will be based on GMPLS encodings. A difference with OFELIA hybrid approach is that the TE and PCE applications are centralized in the OpenFlow controller.

The ML-MR switch proposal is very similar to PAC.C (presented in Fig. 2), where the packet flow table is kept and a new circuit flow table is introduced. Fig. 4 shows that each entry of the circuit flow table is composed by seven fields. In this case the circuit identifier (CCTID) identifies the circuit flow and specifies the virtual port associated to it. The generalized label based on GMPLS standards specifies encoding, switch type and payload ID. Label/in – Label/out, a 32 bit unsigned integer, represents the incoming/outgoing label following GMPLS standardized technology-specific labels (TDM, WDM, etc.).

D. Multi-layer Multi-granularity (ML-MG)

The Multi-layer Multi-granularity is a collaborative work by KDDI R&D Laboratories (Japan) and the University of Posts and Telecommunications (BUPT, China), later joined by other groups.

The ML-MG is the first proposal with real experimental test of OpenFlow-based control plane for transparent optical networks [20, 21]. Fig. 5 shows the proposed architecture of the OpenFlow enabled switch. In ML-MG the abstraction layer is created by means of a Virtual OpenFlow switch (VOFS) that provides a virtualized view of the optical device. The VOFS is composed by *n* virtual Ethernet interfaces associated with the *n* physical ports, and the flow table. The VOFS establishes an OpenFlow secure channel to communicate with the SDN controller. Based on the circuit flow table, the VOFS sends standard Transaction Language 1 (TL1) commands through the TCP interface to configure the cross connection of the optical device.

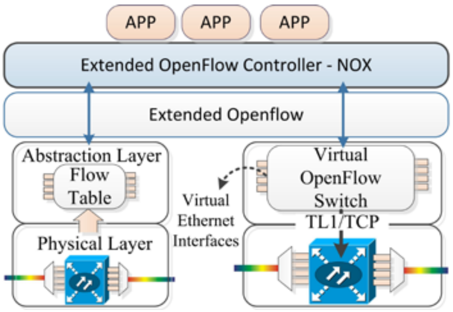


Fig. 4 OpenFlow Optical Cross Connect proposed in ML-MG.

Later the ML-MG group continues to extend OpenFlow and the VOFS virtualization for a unified control plane of multiple layers: packet switching, Optical Burst Switching (OBS), and Optical Circuit Switching (OCS). OBS extensions were considered due to its statistical multiplexing capability, and to show that the proposed model can be applied to packet, burst and circuit switching. The first field-trial connecting Japan, China and Spain, demonstrated the control capabilities of ML-MG to dynamically establish, tear down and restore end-to-end paths across multiple layers and granularities [22, 23]. The ML-MG also addressed transponder control including failure-alarm control, by translating the TL1 messages to OpenFlow; therefore upon a link failure, the SDN controller is able to compute and establish a restoration path.

The more recent publication [24] proposed a new protocol extension and stateless-PCE integrated control-plane to address several new technical and architectural issues related to the introduction of flexi grid technologies.

Due to SDN centralized nature, scalability represents a major concern for future deployments of SDN and OpenFlow-based architectures. The controller manages both path computation and the signaling. In consequence, the authors of ML-MG proposed to offload the path computation tasks from the SDN controller by means of a dedicated path computation element (PCE) [25]. An extended NOX controller was proposed, which communicates with a dedicated stateless PCE, via PCE communication protocol (PCEP) [26].

The ML-MG is similar to OFELIA model the OFVS is a different implementation of the OpenFlow agent used in OFELIA that enables OpenFlow in the network elements. In fact in [24] the OFVS was called OpenFlow agent.

III. COMPARISON OF OPENFLOW-BASED TRANSPORT SDN

A comparison between the OpenFlow-based unified control planes for the transport optical network is developed in this section. Table 1 summarizes the proposals based on six properties: 1) approach used to develop the extensions to the

TABLE I. COMPARISON OF SDN/OPENFLOW EXTENSIONS FOR TRANSPORT NETWORKS

Property	**SDN/OpenFlow Extension Proposals**			
	PAC.C [9-14]	***OFELIA [15-18]***	***ML-MR [8, 19]***	***ML-MG [20-24]***
Approach to develop the Extensions	Pure Extended OpenFlow	Pure Extended OpenFlow & Hybrid GMPLS	Hybrid GMPLS	Pure Extended OpenFlow
Data Plane & OpenFlow Interface	Hardware Vendor implementation	Software or Hardware OpenFlow Agent	Hardware Vendor implementation	Hardware OpenFlow Agent
Optical Tech. Supported	TDM, Fixed DWDM Grid	TDM, Fixed and Flexible DWDM Grid	GMPLS features	Fixed and Flexible DWDM Grid, OBS
Path Computation	Integrated into the Controller	Integrated into the Controller	GMPLS	Dedicated PCE
Controller	NOX	Extended NOX	Open Issue	Extended NOX
Migration Path	Disruptive	Smooth	Smooth	Smooth

OpenFlow and the SDN controller; 2) interface between the heterogeneous data plane and the OpenFlow protocol; 3) optical technologies supported; 4) path computation tasks; 5) controller used in the experimental deployment; 6) migration path towards Transport SDN.

Hybrid-GMPLS and Pure OpenFlow are the two main approaches to develop SDN/OpenFlow extensions. More attention had been given to pure OpenFlow extensions. Only the ML-MR [19] proposes full GMPLS-based extensions. The OFELIA group evaluated both Hybrid-GMPLS and pure OpenFlow extensions; however the later outperforms the former in terms of latency for setting up and tearing down the paths. The authors of ML-MG evaluated three variations of the Hybrid-GMPLS approach named parallel, overlay and integrated, however this is their only work using the Hybrid-GMPLS approach [27].

There is the need of an abstraction layer between the heterogeneous optical nodes and the OpenFlow in order to provide a standardized view to the controller. There are two main approaches, the one supported by PAC.C [14] and ML-MR [19], which is to leave this task to vendor deployment. PAC.C is OpenFlow-friendly, while ML-MR is GMPLS-based. The second approach is to insert a generic agent in order to enable OpenFlow in the optical nodes. ML-MG [24] only considers a hardware agent using virtualization of ports and TL1, while OFELIA [15] considers both the introduction of hardware and software agents using the NE management interface. The interfacing agent represents a short- to mid-term approach in order to enable deployed optical devices into the OpenFlow network and to allow a smoother transition towards SDN. The PAC.C and ML-MR approach represents a long-term solution for new commercial equipments with hardware OpenFlow interfaces.

Appropriate SDN/OpenFlow extensions can support all the optical transport technologies. The first extensions to the OpenFlow Protocol in support of circuit switching [11] only consider TDM circuits and fixed DWDM grid; however the evolution of transport SDN has lead to field trials with fixed and flexible DWDM grid technologies [15, 24] and even several switching paradigms like burst switching [22].

The path computation is a task with high complexity due to heterogeneity, multi-domain nature and geographical extension of transport networks. Therefore the PCE is an important issue for scalability of the transport SDN solutions. The ML-MG and ML-MR offload the controller of this task by using dedicated PCE or by letting GMPLS perform the path computation. On the other hand in PAC.C and OFELIA the path computation tasks are integrated into the controller.

There are several open source SDN controllers; despite that, NOX [12]; which was the first OpenFlow controller, was selected in all the trials and testbeds presented in section II.

PAC.C is the solution with the most disruptive migration path. In order to simplify the adoption of transport SDN, authors of PAC.C proposed network slicing to allow the network operators creating small trials in their networks without affecting the entire network. OFELIA and ML-MG employ OpenFlow agents to enable the current optical devices to OpenFlow. ML-MR is based on GMPLS; thus, it makes use of the port specifications and communication protocols of GMPLS, already standardized and implemented in commercial equipments.

IV. CONCLUSIONS

The overview presented in this paper provides evidence in supporting the conclusion that a smarter and dynamic future transport network can be achieved by enabling OpenFlow into the network elements to create a unified control plane for packet and circuit domains.

Even though OpenFlow was defined for packet networks, the flow abstraction can be adopted into the circuit switching tables in transport network devices. Once the network elements are OpenFlow enabled, an SDN controller can configure them through a unified control plane. The OpenFlow-based transport SDN models discussed in the review have the potential to improve the transport network efficiency with cross-domain and dynamic end-to-end path establishment, lightpath restoration and traffic-engineering capabilities.

Our review spans only 5 years since the first work on OpenFlow-based transport SDN (PAC.C) was published as a pioneer work by Stanford University. Notwithstanding, in such a short period of time several transport SDN testbeds and large-scale world field-trials were reported, with multi-domain dynamic path computation capabilities across different switching paradigms (e.g., packets, bursts and circuits) and technologies (e.g., TDM, fixed and flexible DWDM grids). This evidence is supporting that OpenFlow-based Transport SDN can accelerate the innovation at the transport network.

A major issue to tackle in the transport SDN, due to its centralized nature, is the scalability. PCE arise as an interesting option to offload the traffic engineering and path computation tasks from the SDN controller. However, it is an open issue waiting for more contributions to encourage the commercial adoption of this network architecture.

There is a hot debate about SDN vs. GMPLS in the academy and industry. Nonetheless, it seems that there is no real conflict between the two technologies. In our overview three out of four working groups proposed an integrated solution in which GMPLS is used to release part of the work load from the SDN controller.

We expect a continuous growth on the number of transport SDN proposals, impulsed by the current momentum of SDN, and the support by academy, vendors, service providers and other bodies like IETF.

REFERENCES

[1] E. Mannie, "Generalized multi-protocol label switching (GMPLS) architecture," RFC 3945, Oct. 2004.

[2] F. Hu, Network Innovation through OpenFlow and SDN: Principles and Design: CRC Press, 2014.

[3] S. Das, G. Parulkar, and N. McKeown, "Why OpenFlow/SDN Can Succeed Where GMPLS Failed," in European Conference and Exhibition on Optical Communication, Amsterdam, 2012, p. Tu.1.D.1.

[4] Open Networking Foundation. Available: https://www.opennetworking.org/index.php

[5] S. Jain, et al., "B4: experience with a globally-deployed software defined wan," presented at the Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM, Hong Kong, China, 2013.

[6] N. McKeown, et al., "OpenFlow, Enabling Innovation in Campus Networks," SIGCOMM Comput. Commun. Rev., vol. 38, pp. 69-74, Mar. 2008.

[7] Y. Zhao, J. Zhang, H. Yang, and Y. Yu, "Which Is More Suitable for the Control over Large Scale Optical Networks, GMPLS or OpenFlow?," in Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2013, Anaheim, California, 2013, p. NTu3F.2.

[8] M. Shirazipour, Z. Ying, N. Beheshti, G. Lefebvre, and M. Tatipamula, "OpenFlow and Multi-layer Extensions: Overview and Next Steps," in 2012 European Workshop on Software Defined Networking (EWSDN), 2012, pp. 13-17.

[9] S. Das, G. Parulkar, and N. McKeown, "Unifying Packet and Circuit Switched Networks," in GLOBECOM Workshops, 2009 IEEE, 2009, pp. 1-6.

[10] S. Das, G. Parulkar, and N. McKeown, "Simple Unified Control for Packet and Circuit Networks," in Summer Topical Meeting, 2009. LEOSST '09. IEEE/LEOS, 2009, pp. 147-148.

[11] S. Das, "Extensions to the OpenFlow Protocol in support of Circuit Switching," in Addendum to OpenFlow Protocol Specification (v1.0) – Circuit Switch Addendum v0.3, ed, 2010.

[12] N. Gude, et al., "NOX, towards an operating system for networks," SIGCOMM Comput. Commun. Rev., vol. 38, pp. 105-110, Jul. 2008.

[13] S. Das, et al., "Packet and circuit network convergence with OpenFlow," in 2010 Conference on Optical Fiber Communication (OFC), collocated National Fiber Optic Engineers (OFC/NFOEC), 2010, pp. 1-3.

[14] S. Das, G. Parulkar, and N. McKeown, "Rethinking IP core networks," J Opt Commun Netw, vol. 5, pp. 1431-1442, Dec. 2013.

[15] M. Channegowda, et al., "Experimental demonstration of an OpenFlow based software-defined optical network employing packet, fixed and flexible DWDM grid technologies on an international multi-domain testbed," Optics Express, vol. 21, pp. 5487-5498, Mar. 2013.

[16] M. Channegowda, et al., "Experimental evaluation of Extended OpenFlow deployment for high-performance optical networks," in 2012 38th European Conference and Exhibition on Optical Communications (ECOC), 2012, pp. 1-3.

[17] S. Azodolmolky, et al., "Integrated OpenFlow - GMPLS control plane: An overlay model for software defined packet over optical networks," in 2011 37th European Conference and Exhibition on Optical Communication (ECOC), 2011, pp. 1-3.

[18] M. Channegowda, R. Nejabati, and D. Simeonidou, "Software-Defined Optical Networks Technology and Infrastructure: Enabling Software-Defined Optical Network Operations [Invited]," J Opt Commun Netw, vol. 5, pp. A274-A282, Oct. 2013.

[19] M. Shirazipour, W. John, J. Kempf, H. Green, and M. Tatipamula, "Realizing packet-optical integration with SDN and OpenFlow 1.1 extensions," in 2012 IEEE International Conference on Communications (ICC), 2012, pp. 6633-6637.

[20] L. Liu, T. Tsuritani, I. Morita, H. Guo, and J. Wu, "Experimental validation and performance evaluation of OpenFlow-based wavelength path control in transparent optical networks," Optics Express, vol. 19, pp. 26578-26593, Dec. 2011.

[21] L. Liu, T. Tsuritani, I. Morita, G. Hongxiang, and W. Jian, "OpenFlow-based wavelength path control in transparent optical networks: A proof-of-concept demonstration," in Optical Communication (ECOC), 2011 37th European Conference and Exhibition on, 2011, pp. 1-3.

[22] L. Liu, et al., "First field trial of an OpenFlow-based unified control plane for multi-layer multi-granularity optical networks," in Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2012 and the National Fiber Optic Engineers Conference, 2012, pp. 1-3.

[23] L. Liu, et al., "Field Trial of an OpenFlow-Based Unified Control Plane for Multilayer Multigranularity Optical Switching Networks," Journal of Lightwave Technology, vol. 31, pp. 506-514, Feb. 2013.

[24] L. Liu, et al., "Demonstration of a Dynamic Transparent Optical Network Employing Flexible Transmitters/Receivers Controlled by an OpenFlow-Stateless PCE Integrated Control Plane [Invited]," J Opt Commun Netw, vol. 5, pp. A66-A75, Oct. 2013.

[25] A. Farrel, J.-P. Vasseur, and J. Ash, "A path computation element (PCE)-based architecture," RFC 4655, August 2006.

[26] J.-P. Vasseur and J.-L. Le Roux, " Path Computation Element (PCE) Communication Protocol (PCEP)," RFC 5440, March 2009.

[27] L. Liu, T. Tsuritani, and I. Morita, "Experimental demonstration of OpenFlow/GMPLS interworking control plane for IP/DWDM multi-layer optical networks," in Transparent Optical Networks (ICTON), 2012 14th International Conference on, 2012, pp. 1-4.