Software-Defined Optical Networks Technology and Infrastructure: Enabling Software-Defined Optical Network Operations [Invited]

Mayur Channegowda, Reza Nejabati, and Dimitra Simeonidou

***Abstract*—Software-defined networking (SDN) enables programmable SDN control and management functions at a number of layers, allowing applications to control network resources or information across different technology domains, e.g., Ethernet, wireless, and optical. Current cloud-based services are pushing networks to new boundaries by deploying cutting edge optical technologies to provide scalable and flexible services. SDN combined with the latest optical transport technologies, such as elastic optical networks, enables network operators and cloud service providers to customize their infrastructure dynamically to user/application requirements and therefore minimize the extra capital and operational costs required for hosting new services. In this paper a unified control plane architecture based on OpenFlow for optical SDN tailored to cloud services is introduced. Requirements for its implementation are discussed considering emerging optical transport technologies. Implementations of the architecture are proposed and demonstrated across heterogeneous state-of-the-art optical, packet, and IT resource integrated cloud infrastructure. Finally, its performance is evaluated using cloud use cases and its results are discussed.**

***Index Terms*—Flexible WDM grid; OpenFlow (OF); Optical networking; Software-defined networks (SDN).**

I. Introduction

Software-defined networking (SDN) [1] is defined as a control framework that supports programmability of network functions and protocols by decoupling the data plane and the control plane, which are currently integrated vertically in most network equipment. SDN technology allows the underlying infrastructure to be abstracted and used by applications and network services as a virtual entity. This allows network operators to define and manipulate logical maps of the network, creating multiple co-existing network slices (virtual networks) independent of underlying transport technology and network protocols. Furthermore, the separation of the control plane and data plane makes SDN a suitable candidate for an integrated control plane supporting multiple network domains and multiple transport technologies. OpenFlow (OF) [2] is an open standard vendor and technology agnostic protocol that allows separation of the data and control plane, and, therefore, it is a suitable candidate for the realization of SDN. It is based on flow switching with the capability to execute software/user-defined flow-based routing, control, and management in a controller (i.e., OF controller) outside the data path. Enabling SDN via OF extensions to support optical networks [3] can provide a new framework for evolving carrier grade and cloud networks. It can potentially facilitate application specific network slicing at the optical layer, coordination and orchestration of higher network layers, and applications with optical layers. It can also provide a unified control plane platform for integration of electronic packets and optical networks for access, metro, and core network segments, as well as in intra- and inter-data centers (DCs).

These features make SDN a suitable network control and management framework for cloud computing environments. Cloud computing services are characterized by the performance and availability of their services, which is highly dependent on cloud physical infrastructures. The cloud physical infrastructure comprises the DC infrastructure (i.e., computing, storage, and in general IT resources) and the network connectivity interconnecting DCs together and to the users. Network infrastructure is a key building block of cloud computing platforms both within DCs and between DCs for inter- and intra-DC connectivity. Furthermore, for delivering cloud services to end users and in order for users to utilize cloud-computing services, DC platforms need to be integrated with operator network infrastructures.

Extending SDN to support interconnectivity of IT resources, such as virtual computing [virtual machines (VMs)] and storage using emerging optical transport [4] and switching technologies (e.g., elastic optical networks), as well as existing packet networks, will enable application-aware/service-aware traffic flow handling and routing within DCs. SDN can facilitate implementation of programmable traffic engineering and load balancing schemes within a DC by taking into account the bandwidth and latency requirements of different traffic flows of different

Manuscript received May 10, 2013; revised September 6, 2013; accepted September 10, 2013; published September 30, 2013 (Doc. ID 190423).

The authors are with the High Performance Networks Group (HPNG), Department of Electrical and Electronic Engineering, University of Bristol, UK (e-mail: mayur.channegowda@bristol.ac.uk).

http://dx.doi.org/10.1364/JOCN.5.00A274

applications, enabling on-demand mobility and migration of services. With an abstraction mechanism like OF, SDN can also simplify the complexities of handling traffic among various networking technologies.

The SDN benefits for the cloud can be extended to service provider networks, as well. Enabling SDN at control and management of operator networks can facilitate coordination and orchestration of inter- and intra-DC networks involving the optical layer together with higher network layers. This can be achieved by providing a unified control plane platform [5,6] for integration of electronic packets and optical networks for DC, access, metro, and core network segments. In addition, SDN will enable creation of application/service specific network slices with guaranteed quality of service (QoS) between geographically distributed DCs and users. It also facilitates on-demand mobility and migration of services such as VMs and storage between geographically distributed DCs by unifying intra- and inter-DC network control and management.

In summary, deploying SDN in a multitechnology DC infrastructure will enable

- automated, efficient application-aware (including application level QoS, such as delay and jitter) mapping of traffic flows into optical and electronic packet transport layers within and between DCs regardless of transport technology and
- application specific and coordinated slicing of IT (computing and storage) and network resources (inter- and intra-DC) to create a virtual DC that supports multitenancy.

Deploying optical-technology-based SDN in cloud environments poses new challenges owing to the various traffic flow characteristics that are presented by diverse cloud services. An initial set of different service types along with their diverse service characteristics is described in Table I. The SDN-based control plane has to consider these characteristics (make it available to SDN applications in abstract form) in order to allocate suitable infrastructure resources for the user/application request. For instance, consider content delivery service in row 2 wherein different content types have different network requirements. For standard definition (SD) media traffic flows with low bandwidth (in megabits), short burst (known as mice flows) characteristics can be served by a packet flow, which can be set up in milliseconds, whereas HD media with medium capacity (~10 gigabits) is realized with a combination of packet–circuit flows. On the other hand, for a 4K media type with high-bandwidth (>10/40 gigabits) long-duration flows (elephant flows) can be served with flexible optical flows, thereby increasing overall network utilization and efficiency.

This paper introduces, in Section II, a control plane architecture based on OF for software-defined optical networks suitable for cloud computing services that takes into account the aforementioned requirements and features. The proposed architecture allows implementation of agile, elastic cloud networks that can adapt to application requirements on demand. Subsequently, the architecture subsection discusses technological considerations and requirements for OF protocol extensions to support optical networks. In Section III the paper describes two technical implementations of the proposed SDN architecture and demonstrates in Section IV these approaches over a heterogeneous testbed using SDN applications. Finally, using cloud use cases, the performance of the proposed architecture is evaluated over the integrated network plus IT resources testbed.

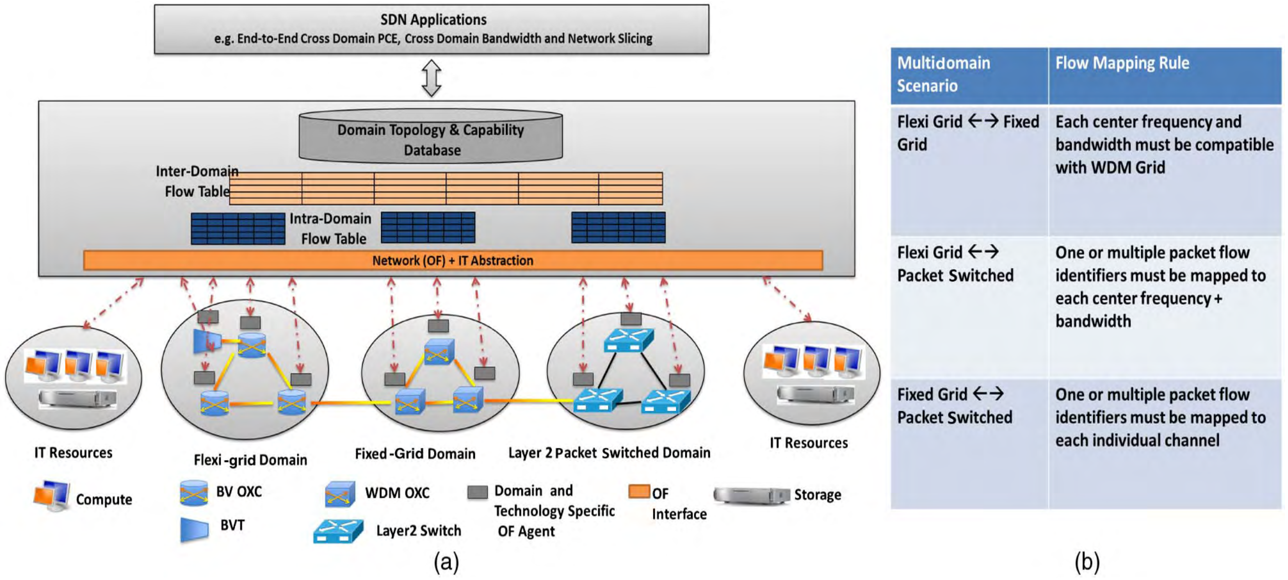
II. Architecture

In order to enable SDN-based unified control and management of an optical network the following challenges need to be addressed:

- Definition of a unified optical transport and switching granularity (i.e., optical flow) that can be generalized for different optical transport technologies (fixed DWDM, flexi DWDM, etc.) and be compatible with electronic packet switching technology. References [5–7] describe

TABLE I
DC Service Characteristics

Service Type		Service Characteristics	Service Composer			
			Switching	Rate	Configuration Time	Connection
Web		Low BW, mice flow	Ethernet/mpls flows	Mb/s	ms	Packet
Media	SD	Low BW, low priority, mice flow	Ethernet/mpls flows	Mb/s	ms	Packet
	HD	High BW, low latency, mice flow	Ethernet/mpls-circuit flows	Gb/s	ms to s	Packet, optical
	4K	Low latency, high BW, low loss, elephant flow	Ethernet, circuit-fiber, λ-sub λ flows	10–100 Gb/s	s	Packet, optical (fixed & flexi)
Migration	VM	Medium BW, mice/elephant flow	Ethernet/mpls-circuit flows	10–100 Gb/s	ms to s	Packet, optical
	Storage	High BW, elephant flow	Circuit-fiber, λ-sub λ flows	40–400 Gb/s	ms to s	Packet, optical (fixed & flexi)
Distribution File System (DFS)		Low latency, low BW, low loss, mice flow	Ethernet/mpls flows	Mb/s	ms	Packet


SDN Applications
e.g. End-to-End Cross Domain PCE, Cross Domain Bandwidth and Network Slicing
Domain Topology & Capability Database
Inter-Domain Flow Table
Intra-Domain Flow Table
Network (OF) + IT Abstraction
IT Resources
Flexi-grid Domain
Fixed-Grid Domain
Layer 2 Packet Switched Domain
IT Resources
Compute
BV OXC
BVT
WDM OXC
Layer2 Switch
Domain and Technology Specific OF Agent
OF Interface
Storage
(a)

Multidomain Scenario	Flow Mapping Rule
Flexi Grid <-> Fixed Grid	Each center frequency and bandwidth must be compatible with WDM Grid
Flexi Grid <-> Packet Switched	One or multiple packet flow identifiers must be mapped to each center frequency + bandwidth
Fixed Grid <-> Packet Switched	One or multiple packet flow identifiers must be mapped to each individual channel

Fig. 1. (a) Architecture of multilayer multitechnology control plane. (b) Flow mappings between technologies.

such a unification over a SDN architecture, the benefits of which are discussed in [8].

- Design and implementation of an abstraction mechanism that can hide the heterogeneous optical transport layer technology details and realize the aforementioned generalized switching entity definition. Abstraction models similar to mobile phone operating systems like Android [9] and embedded systems, namely, tiny OS [10], provide insights on how separating concerns can be advantageous. Similar attempts for networks are currently being realized via approaches like Forces [11] and OF.
- Taking into account physical layer specific features of different optical transport technologies, such as power, impairments, and switching constraints. [12].
- Cross technology constraints for bandwidth allocation and traffic mapping in networks comprising heterogeneous technological domains, e.g., packet over single or hybrid optical transport technologies. This plays an important role in provider networks where multiple operational units are consumed to maintain different technology domains. An SDN-based solution where the separation of a data plane with a common control plane can lead to lower operating expenditures and more efficient networks [13].

Figure 1(a) shows an architectural block diagram of the proposed OF-based optical SDN control plane that addresses the aforementioned challenges. Central to the proposed architecture is an abstraction mechanism, realized by an extended OF controller and the OF protocol. This mechanism enables generalization of the flow switching concept for the underlying heterogeneous optical transport technologies, as well as its integration with packet switched domains. The architecture encompasses three critical components, which are described in detail in the following subsections.

A. *Hardware Abstractions*

The goal of the resource or hardware abstraction is to hide the technological details of underlying heterogeneous transport network resources and enable a programmable interface for hardware state configuration. We present here a complimentary hardware abstraction layer based on tinyOS, as shown on the left side of Fig. 2, which includes a hardware presentation layer (HPL), a hardware interface layer (HIL), and an OF application programming interface (API). The HPL provides all the capabilities of the device. It hides the hardware intricacies and exports the device features and capabilities based on a unified information model (represented in a uniform way) to the upper HIL. The HIL utilizes the raw interfaces provided by the HPL components to build useful abstractions, hiding the complexity naturally associated with the use of hardware resources.

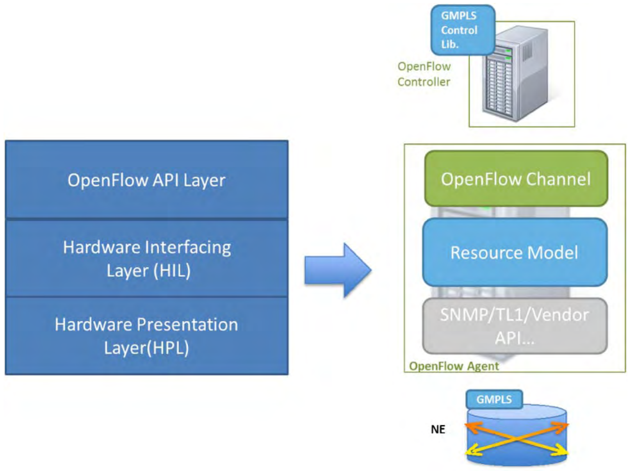


Fig. 2. OpenFlow agent abstractions.

The HIL exposes only the required features and information that can be used in an OF-based network. The HIL is also capable of maintaining a state that can be used for performing arbitration and resource control. HILs are tailored to the concrete device class represented in OF circuit addendum v0.3 [14], which provides the necessary specifications to represent an optical device class. The difference between HPLs and HILs is that the former exposes all available capabilities of a device and the latter exposes only those necessary for flow-based general abstraction, thereby keeping the API simple and light.

The OF API maps abstracted information provided by the HIL into the OF protocol and its extensions. An example to explain the interworking between the layers can be a lightpath setup in the optical domain. An end-to-end lightpath establishment consists of wavelength-based cross-connections on nodes and also requires equalizing power across the lightpath. The HPL exposes both cross-connect and equalization configuration features of the node as an optical device class, but the HIL uses only the cross-connect feature from the class and implicitly performs equalization when required. However, in a case where the application requires all features from the device, it can directly use the HPL interface.

Furthermore, these abstractions can be supported on vendor devices in two ways: 1) softpath, which is a software-based implementation of abstraction layers wherein the flow matches are software based, and 2) hardpath, where implementation of abstraction layers is done using fast hardware, e.g., ternary content-addressable memory (TCAM)-based flow matches. Since current optical devices have no embedded hardware for implementing hardware abstraction, a software-based approach is used. We use this model to build our modular OF hardware abstraction layer, as shown on the right in Fig. 2, henceforth called the OF agent. The agent provides a novel optical switch abstraction that supports an extended OF protocol (beyond v0.3 as explained in the next section). This agent can utilize the network element (NE) management interface [simple network management protocol (SNMP), vendor API, etc.] to communicate with the data plane, in a case where an OF implementation is not supported, and provide the HPL functionalities. To implement an HIL, a generic and novel resource model is designed and implemented to maintain the NE's configuration (wavelengths, port capabilities, and switching constraints). The OF agent also includes the OF channel, which is responsible for communication with the extended OF controller (Fig. 2) and provides an API for programming flows.

The IT resource abstraction is already well exploited with many commercial hypervisors from VMware (vSphere) [15], Citrix (Xen) [16], etc., and also open source ones, like KVM. They can be managed and configured with the help of the various APIs and tools that are inbuilt with the virtualization technology. For example, the Xen technology virtualized server has an in-built Xen API (XAPI) [17] for VM management. The network + IT abstraction layer uses the IT abstraction provided by hypervisors and network abstraction provided by OF (described earlier) and exposes these programmable interfaces to the upper application or components. Thus the architecture provides a common abstraction layer that includes network resource exposed by OF and IT resources, enabling a pluggable environment.

B. OpenFlow Extensions

An OF-enabled switch is represented in the controller by one or more flow tables (see Fig. 3), and each table entry consists of match fields, counters, and a set of associated actions. The current OF version concentrates mainly on packet domains, and an addendum was added to address the optical domain considering synchronous optical network/synchronous digital hierarchy, optical cross-connects (OXCs), and Ethernet/time division multiplexing convergence as circuit switched technologies. We use OF version 1.0 with extensions supporting circuit switching, which is documented as addendum v0.3. This current specification does not support optical network features like switching constraints and optical impairments, which are key functions required by an optical control plane. Furthermore, it does not support advanced and emerging optical transport technologies, such as a flexible DWDM grid. To address the shortcomings of the current OF extension on supporting optical network technologies, we have proposed a generic and extended optical flow specification [18], as shown in Fig. 3. In the proposed definition, an optical flow can be identified by a flow identifier comprising port, wavelength or center frequency (CF) of the optical carrier, bandwidth associated with the wavelength or CF, signal type (e.g., optical transport format: subwavelength switching header information, time slot, bitrate, protocol, modulation format) associated with a specific optical transport and switching technology, and constraints specific to the physical layer (e.g., sensitivity to impairments and power range). This definition is generic enough to allow applying the concept of optical flow [Fig. 1(b) top] to both existing and emerging optical transport technologies. Moreover, it is in line with the packet domain OF flow matching.

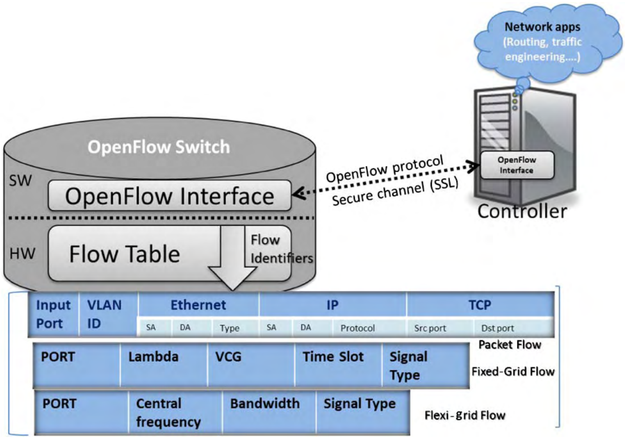


Fig. 3. Flow definitions for different technology domains.

These flow generalizations are used to extend the OF protocol, which includes Switch_Feature and CFlow_Mod messages. A Switch_Feature message advertises the device capabilities and a CFlow_Mod message is used to configure the node state. The Switch_Feature (i.e., reply message) extension supports optical NE capabilities, including central frequency, spectrum range, and bandwidth granularity of transponders and switches; number of ports and the wavelength channels of the switches; peering connectivity inside and across multiple domains; signal types; and NE optical constraints, e.g., attenuation. We use the extended CFlow_Mod messages for configuring NEs, i.e., transponders and switching/cross-connect nodes for both fixed- and flexible-grid DWDM compatible NEs based on the International Telecommunication Union Telecommunication Standardization Sector G.694.1 recommendation [19]. Notably for the flexible WDM grid the equation $193.1 + n \times 0.00625$ (THz) is used to calculate the central frequency of a frequency slot, while $12.5\ \text{GHz} \times m$ yields the slot width. Here n is an integer and m is a positive integer. So for flexi domains, the exchange of m and n values between controller and optical elements (or OF agent) determines the spectrum for the node.

Apart from the core messages, two other vendor-based OF messages are also included, extending the specification to include switching constraints and power equalization functions. Switching constraints describe how physical ports are connected with each other. This relationship between ports results from internal NE configuration and tells what optical signal (wavelength) can flow between the ports. Some devices require power equalization to be triggered after a cross-connection, so OF equalization messages are used to trigger power equalization along the internal signal path between ports.

The network control plane using the extended OF protocol is able to abstract the switching entity and transport format of each technological domain in the form of generic flows (Fig. 3) and to configure NEs using technology specific flow tables. For multitechnology domain aspects, the controller is made aware of each domain constraint by utilizing intradomain and interdomain flow tables. An intradomain flow table holds flow identifiers and associated actions for each NE within a particular domain. In addition, the architecture utilizes an interdomain flow table for enforcing cross technology constraints for bandwidth allocation when traffic traverses from one technology domain to another [Fig. 1(b), e.g., flexi DWDM to fixed WDM, or packet to DWDM]. The domain flow tables stored in the controller map the technology domain abstractions, whereas the flow tables in the device provide individual network node abstraction.

In this architecture, the details pertaining to the topology and technology/domain constraints are stored in the domain capability database, whose updated information is utilized by SDN applications over a well-defined northbound API. Depending upon the desired service, a DC application can then utilize the full infrastructure abstraction from the API to orchestrate resource allocation based on user/application requirements.

C. SDN Application

Applications are critical components of SDN architecture. SDN applications can provide isolated network functionalities that provide a modular way to add or remove new functionalities. It also opens doors to create new functionalities; for example, it can be used to create tenant virtual topologies based on a cloud user request, provide traffic access management, or policy-based service management like FlowVisor [20]. In our proposed architecture we foresee that different algorithms, such as routing wavelength/spectrum algorithms, can be used as apps. They are responsible for tasks such as path computation, routing, wavelength assignment, loop avoidance, and many more that are critical in an integrated packet–optical network. The OF controller exposes a well-defined API wherein multiple algorithms, i.e., SDN applications, can be used in conjunction to provide a multitude of functionalities.

For our proposed architecture an issue with the packet–optical integration in a dynamic cloud environment is the optimal resource utilization. Flows have to be carefully traffic engineered so as not to have underutilization, especially in the high-capacity optical domain. For example a high-capacity low-latency traffic flow might be attractive for the optical domain, but if it is in short bursts then it leads to inefficient resource mapping. Therefore, we developed an application-aware load balancer that balances the traffic flow based on the application requirement, taking into consideration the technology domain constraints and bandwidth. This is based on the service characteristics depicted in Table I and the application carefully maps the elephant flows to the appropriate packet, fixed, or flexi domain. For example, a critical cloud service like storage migration might require very high bandwidth, which is appropriate for flexible WDM grid nodes, whereas a short burst voice-over-IP (VOIP) call is suitable over the packet domain.

III. Implementation

We build a prototype of this OF agent for the ADVA fixed WDM [21] and for an in-house built flexible-grid WDM node, which will be used to compose the architecture. The agent uses the NE management interface (SNMP) to communicate with the data plane to provide the HPL functionalities. The available OF library v1.0 provided the base OF library, which was first extended according to the circuit addendum and then the proposed OF extensions described in Subsection II.B were included.

NOX [22] controller version 1.0 was extended to incorporate the circuit specification. Again the same extensions proposed in Subsection II.B were incorporated. The controller also exposed an API to the SDN applications. The packet domain interworking required two main functions, network discovery and an L2 learning switch, which were included as part of the NOX applications. As part of the optical domain application, a bundle of algorithms was included for each technology domain (fixed grid, flexi grid) and their corresponding cross domain networking.

As an SDN application, we have developed an algorithm bundle, including several algorithms designed for different scenarios (i.e., single/multiple fixed grid, single/multiple flexi grid, mixed fixed and flexi grid), which is running on top of the OF controller to compose virtual network (VN) slices over flexi- and fixed-grid domains. The algorithm supports two main functionalities: one is to calculate the best path from source to destination and the other is to find the optimum spectrum across domains to fulfill user requests. The algorithm bundle reads the information of physical networks and the user requests from the OF controller. The physical network information obtained from the topology database of the OF controller not only involves the nodes and their connectivities but also the domain constraints and impairments. Utilizing the flow mapping description in Fig. 1(b), the application can serve requests, taking into consideration the domain constraints, such as wavelength-supported impairments.

Based on the aforementioned flow definition and OF protocol extensions, we introduce two methods for implementation of the proposed control plane architecture: 1) integrated generalized multiprotocol label switching (GMPLS) [23,24] and OF and 2) standalone OF [25]. Management extensions were introduced to support control plane (CP)-assisted optical OF, which assumes cooperation with ADVA's GMPLS CP. In CP-assisted OF, an OF controller uses the GMPLS Control Library module, which sets up or tears down lightpaths using ADVA's management interface, namely, the SNMP protocol. In the integrated GMPLS-OF approach, the OF controller receives information regarding the topology and resources using extended OF protocol and can expose them to applications. SDN applications based on this information can request for a path or compute explicitly the path. However, a detailed path computation, lightpath establishment, and teardown are performed utilizing the GMPLS CP. An extended OF controller and associated SDN applications are developed that consider loose and explicit lightpath establishment. In the former case, only ingress and egress NEs and ports are specified and the GMPLS controller handles the path computation and establishment. In other words, the OF controller exploits GMPLS available functionalities in order to compute flow tables and, consequently, to establish and verify the lightpaths. In the explicit lightpath establishment case, the controller is able to specify the full details of the lightpath (i.e., address all the switches and ports along the lightpath), to verify the feasibility of the lightpath and perform its establishment. The controller utilizes the Switch_Feature messages to construct the network topology and CFlow_Mod messages to control optical transponders and switches. The extended OF controller, unlike the loose lightpath establishment approach, relies on the SDN application for computing flow tables in the controller and, consequently, for establishing and verifying end-to-end lightpaths.

IV. Demonstration

A. Testbed Setup

The experimental setup in the University of Bristol laboratory is depicted in Fig. 4(a) and consists of heterogeneous resources. We configured the testbed to emulate

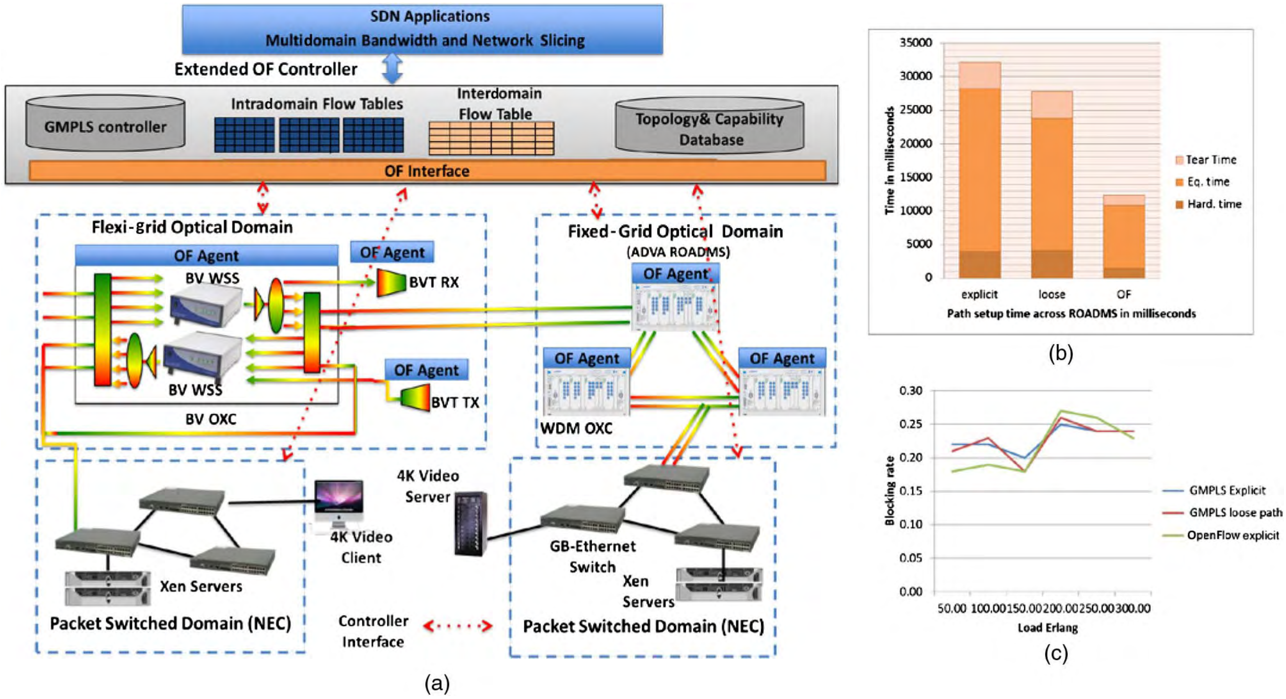


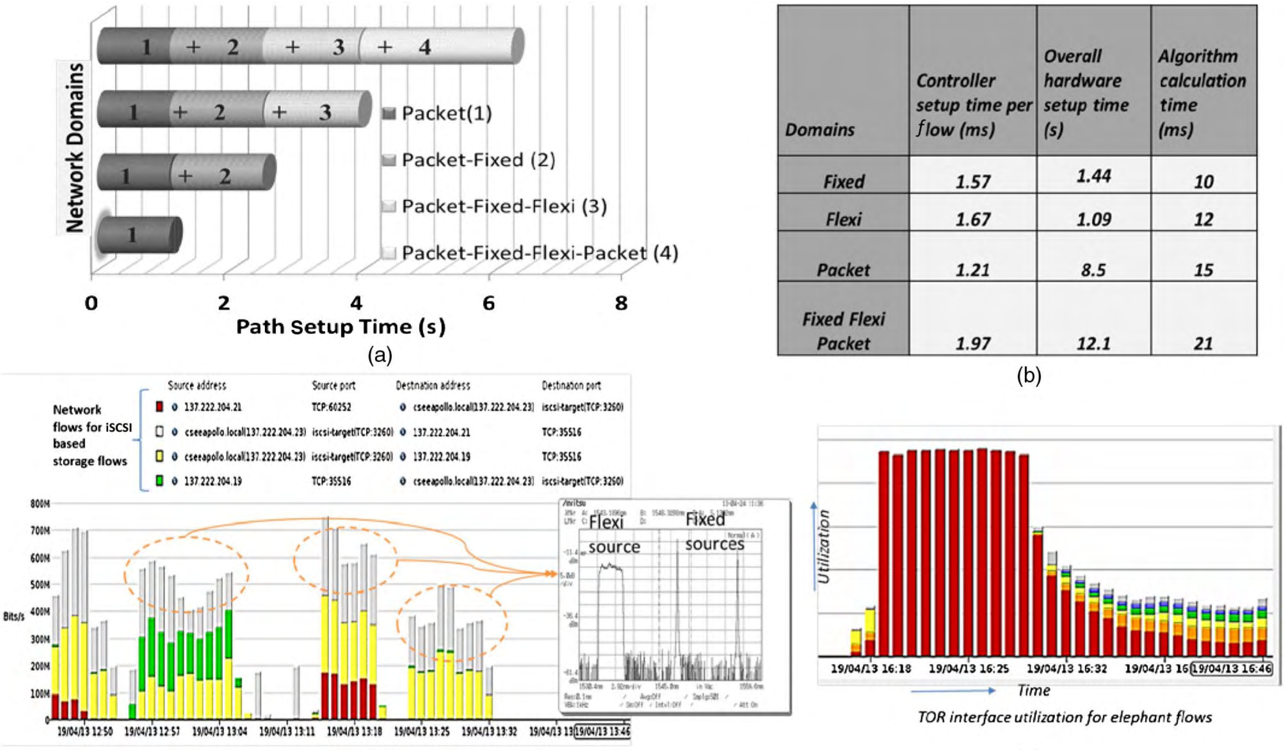
Fig. 4. (a) Demonstration setup: packet-fixed-flexible devices. (b) Path setup times for fixed WDM nodes. (c) Blocking probability versus load for GMPLS–OF and standalone OF approaches.

a heterogeneous cloud environment comprising packet and optical (fixed and flexi) network resources combined with high performance virtualized IT resources (servers and storage). The fixed–flexible-grid testbed is comprised of an in-house built 8 × 8 (4 × 4 bidirectional) bandwidth variable (BV) OXC utilizing two BV wavelength selective switches (WSSs) with internal recirculation fiber loops to emulate multiple nodes, a BV transponder [BV transmitter (TX) and BV receiver (RX)] supporting the C-band, and three ADVA FSP3000 reconfigurable optical add/drop multiplexers (ROADMs) with two active wavelength channels. The packet switched testbed comprises four NEC IPX, one Arista 7050s, and one Extreme Summit OF-enabled 1/10/40GE top of the rack (TOR) switches. The computer resources are represented by a commercial Xen server virtualization powered by Xen hypervisor on a dozen high-performance virtualization servers backed with a 10 Tbyte hard drive. This lets us create a large number of virtual machines, which are used to generate DC application traffic. Following Table I, different service types are generated on the experimental testbed and the performance of the service composition using the SDN control plane is measured.

The deployed testbed includes the GMPLS–OF integrated controller, as well as the developed extended standalone OF controller. The developed OF agent abstraction is deployed on the ADVA fixed ROADMs and flexible nodes. The SDN applications described in Section II are used for path computation and traffic grooming over the heterogeneous testbed.

B. Results

We have evaluated the performance of both approaches, i.e., GMPLS–OF integrated and standalone OF, in terms of path setup times using an SDN application to create network slices. Figure 4(b) shows path setup times for a packet over the ADVA ROADMs domain (packet over the fixed WDM domain only) using the integrated GMPLS–OF (includes both loose and explicit modes) and standalone OF approaches for different path request and load values. The individual network element setup times were categorized based on hardware, power equalization, and teardown times. The OF approach was better, owing to its ability to cross-connect and equalize power concurrently on involved NEs. The results indicate faster path setup times for the standalone OF. Figure 4(c) shows that the blocking rate versus the load result of hybrid (explicit, loose path) and pure OF approaches were 23%, 23%, and 22%, respectively. Lightpath requests are generated according to a Poisson process and uniformly distributed among all node pairs. Both interarrival of requests and their holding times are exponentially distributed. Imposed load to the extended controller in terms of lightpath requests (100 requests) are varied from 50 to 300 Erlangs.



(a)

Domains	Controller setup time per flow (ms)	Overall hardware setup time (s)	Algorithm calculation time (ms)
Fixed	1.57	1.44	10
Flexi	1.67	1.09	12
Packet	1.21	8.5	15
Fixed Flexi Packet	1.97	12.1	21

(b)

(c)

(d)

Fig. 5. (a) Configuration times for different domains. (b) Total path setup times. (c) VM migration traffic grooming. (d) Application aware utilization reduction.

The high blocking rate is mainly due to the limited number of client ports per NE.

In order to evaluate the OF-based control plane performance we collated the various control plane timings along with the cross-connect setup times. Figure 5(a) shows the timings of the various operational parts of the OF controller. The controller setup time indicates the time required for creating and processing OF messages in both the OF controller and agents. The hardware setup time includes the controller setup time and the time taken for each agent to configure its corresponding NE upon receiving a CFlow_Mod message. The algorithm time is the time for the SDN application to compute the network path and slices. Figure 5(b) illustrates the performance of the standalone OF for end-to-end path setup times for different technology domains. Path setup times are compared for three different cases, i.e., the fixed DWDM domain only, the flexi–fixed DWDM domains, and packet over fixed–flexi DWDM domains. In addition, comparing results from the three test scenarios shows that the OF controller performance is stable for different network scenarios irrespective of the transport technology and the complexity of the network topology.

We further expanded our demonstration to include a typical cloud scenario to run migration use cases utilizing the standalone OF approach. Typical DC computing resource migration consists of two types: VM and storage migration [26]. Though both migrations are performed live, the distinction is that in the storage the actual VM disk moves, which requires huge bandwidth. This kind of storage migration for inter-DC flows can be aggregated and configured with flexible super-channel flows, as shown in Fig 5(c). The high-capacity Internet Small Computer System Interface (iSCSI) storage flows between the Xen servers (137.222.204.21/19) and storage are groomed to flexible-grid flows by the load balancer application running on the controller, whereas the low bandwidth VM migration is over fixed-grid flows. Figure 5(d) shows the utilization of packet switches for high BW media flows with other traffic captured on a popular industry sFlow monitoring application [27]. Upon receiving the first media packet, the SDN controller pushes the path flows using OF flow_mod messages for the packet domain to setup the service. During the course, if the monitor application detects a high-bandwidth long-duration flow (multiple media server clients), a suitable optical path is constructed in conjunction with a path computation application. Then the SDN controller programs the optical devices with wavelength flows, directing the media client flows to the optical layer, thereby drastically reducing the overall utilization in the packet domain, as is seen in the result at Fig. 5(d). The results show two major features: service deployment and automated reconfiguration based on the load.

V. Conclusion

We have proposed a control plane architecture based on OF for enabling SDN operations in integrated packet–optical networks. A novel abstraction mechanism for enabling OF on optical devices was developed and implemented on commercial hardware. We discuss requirements and describe implementations of OF protocol extensions for transport optical networks incorporating commercial optical equipment, as well as research prototypes of emerging optical transport technologies. Experimental validations and performance analysis of the proposed architecture demonstrate improved path setup times and control stability when OF is applied directly to optical transport technologies. Furthermore, the cloud migration use case results suggest improved network utilization with a unified SDN/OF control plane that is application aware. Our experiments demonstrate that SDN/OF provides an extensible control framework for packet over optical transport embracing existing and emerging wavelength switching technologies. The work pioneers new features to the OF circuit specifications and aims to enable dynamic, flexible networking in data centers.

Acknowledgments

This work is partially supported by the EU funded projects FIBRE and ALIEN and the UK funded EPSRC PATRON and Hyper Highway. This work is part of a joint collaboration with the ADVA Optical Networking for the OFELIA project.

References

[1] ONF, "Software-defined networking: the new norm for networks," Mar. 13, 2012 [Online]. Available: https://www.opennetworking.org/images/stories/downloads/white-papers/wp-sdn-newnorm.pdf.

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[3] S. Gringeri, N. Bitar, and T. J. Xia, "Extending software defined network principles to include optical transport," *IEEE Commun. Mag.*, vol. 51, no. 3, pp. 32–40, Mar. 2013.

[4] C. Kachris and I. Tomkos, "A survey on optical interconnects for data centers," *IEEE Commun. Surv. Tutorials*, vol. 14, no. 4, pp. 1021–1036, Fourth Quarter 2012.

[5] S. Das, G. Parulkar, N. McKeown, P. Singh, D. Getachew, and L. Ong, "Packet and circuit network convergence with OpenFlow," in *Optical Fiber Communication Conf. and Expo. and the Nat. Fiber Optic Engineers Conf. (OFC/NFOEC)*, 2010, paper OTuG1.

[6] L. Liu, D. Zhang, T. Tsuritani, R. Vilalta, R. Casellas, L. Hong, I. Morita, H. Guo, J. Wu, R. Martínez, and R. Muñoz, "First field trial of an OpenFlow-based unified control plane for multi-layer multi-granularity optical networks," in *Optical Fiber Communication Conf. and Expo. and the Nat. Fiber Optic Engineers Conf. (OFC/NFOEC)*, 2012, paper PDP5D.2.

[7] L. Liu, R. Muñoz, R. Casellas, T. Tsuritani, R. Martínez, and I. Morita, "OpenSlice: An OpenFlow-based control plane for spectrum sliced elastic optical path networks," *Opt. Express*, vol. 21, no. 4, pp. 4194–4204, 2013.

[8] S. Das, Y. Yiakoumis, G. Parulkar, N. McKeown, P. Singh, D. Getachew, and P. D. Desai, "Application-aware aggregation and traffic engineering in a converged packet-circuit network,"

in *Optical Fiber Communication Conf. and Expo. and the Nat. Fiber Optic Engineers Conf. (OFC/NFOEC)*, Mar. 6–10, 2011.

[9] Android Hardware Abstraction Layer [Online]. Available: https://source.android.com/devices/reference/files.html.

[10] V. Handziski, J. Polastre, J. Hauer, C. Sharp, A. Wolisz, and D. Culler, "Flexible hardware abstraction for wireless sensor networks," in *Proc. 2nd European Workshop on Wireless Sensor Networks*, Jan. 31–Feb. 2, 2005, pp. 145–157.

[11] A. Doria, J. Hadi Salim, R. Haas, H. Khosravi, W. Wang, L. Dong, R. Gopal, and J. Halpern, "Forwarding and control element separation (ForCES) protocol specification," IETF RFC 5810, Mar. 2010 [Online]. Available: http://tools.ietf.org/html/rfc5810.

[12] C. V. Saradhi and S. Subramaniam, "Physical layer impairment aware routing (PLIAR) in WDM optical networks: Issues and challenges," *IEEE Commun. Surv. Tutorials*, vol. 11, no. 4, pp. 109–130, 2009.

[13] H. Yang, J. Zhang, Y. Zhao, S. Huang, Y. Ji, J. Han, Y. Lin, and Y. Lee, "First demonstration of cross stratum resilience for data center services in OpenFlow-based flexi-grid optical networks," in *Asia Communications and Photonics Conf.*, 2012, paper PAF4C.5.

[14] S. Das, "Extensions to the OF protocol in support of circuit switching," addendum v0.3, June 2010, http://archive.openflow.org/wk/images/8/81/OpenFlow_Circuit_Switch_Specification_v0.3.pdf.

[15] VMware ESX [Online]. Available: http://www.vmware.com/products/vsphere-hypervisor/overview.html.

[16] Citrix XenServer [Online]. Available: http://www.citrix.com/products/xenserver/overview.html.

[17] XEN API [Online]. Available: http://wiki.xenproject.org/wiki/Archived/Xen_API_Project.

[18] M. Channegowda, R. Nejabati, M. R. Fard, S. Peng, N. Amaya, G. Zervas, D. Simeonidou, R. Vilalta, R. Casellas, R. Martínez, R. Muñoz, L. Liu, T. Tsuritani, I. Morita, A. Autenrieth, J. P. Elbers, P. Kostecki, and P. Kaczmarek, "Experimental demonstration of an OpenFlow based software-defined optical network employing packet, fixed and flexible DWDM grid technologies on an international multi-domain testbed," *Opt. Express*, vol. 21, no. 5, pp. 5487–5498, 2013.

[19] "Spectral grids for WDM applications: DWDM frequency grid," ITU-T Recommendation G.694.1, June 2002.

[20] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "FlowVisor: A network virtualization layer," OPENFLOW-TR-2009-01, 2009.

[21] ADVA ROADMs [Online]. Available: http://www.advaoptical.com/en/products/scalable-optical-transport/fsp-3000.aspx.

[22] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: Towards an operating system for networks," *Comput. Commun. Rev.*, vol. 38, no. 3, pp. 105–110, 2008.

[23] S. Das, G. Parulkar, and N. McKeown, "Why OpenFlow/SDN can succeed where GMPLS failed," in *European Conf. and Exhibition on Optical Communication*, 2012, paper Tu.1.D.1.

[24] S. Azodolmolky, R. Nejabati, E. Escalona, R. Jayakumar, N. Efstathiou, and D. Simeonidou, "Integrated OpenFlow–GMPLS control plane: An overlay model for software defined packet over optical networks," *Opt. Express*, vol. 19, pp. B421–B428, 2011.

[25] M. Channegowda, P. Kostecki, N. Efstathiou, S. Azodolmolky, R. Nejabati, P. Kaczmarek, A. Autenrieth, J. P. Elbers, and D. Simeonidou, "Experimental Evaluation of Extended OpenFlow deployment for high-performance optical networks," in *European Conf. and Exhibition on Optical Communication*, 2012, paper Tu.1.D.2.

[26] Xenserver Storage Migraton [Online]. Available: http://www.citrix.com/content/dam/citrix/en_us/documents/products/live-storage-migration-with-xenserver.pdf.

[27] InMon sFlow Monitoring [Online]. Available: http://www.inmon.com/products/sFlowTrend.php.