

# Build to Tenants' Requirements: On-Demand Application-Driven vSD-EON Slicing [Invited]

Zuqing Zhu, *Senior Member, IEEE*, Bingxin Kong, Jie Yin, Sicheng Zhao, Shengru Li

**Abstract**—Application-driven networks (ADNs) aim to build logically separate virtual networks (VNs) to meet the distinct demands of different applications. In this paper, we study how to realize the on-demand slicing of application-driven virtual software-defined elastic optical networks (vSD-EONs) based on the concept of ADN. We design the network system for on-demand application-driven vSD-EON slicing, and demonstrate the building and operating of application-driven vSD-EONs with it experimentally. Specifically, our experimental demonstrations consider three scenarios: 1) the tenant's application requires high availability for the data plane, 2) the tenant's application is interactive and thus requires short end-to-end latency, and 3) the tenant's application needs enhanced physical-layer security guarantee. With an experimental testbed that consists of commercial optical transmission facilities (OTF), bandwidth-variable wavelength-selective switches (BV-WSS'), erbium-doped fiber amplifiers (EDFAs), and high-performance servers with both optical and electrical ports, we verify that the proposed vSD-EON slicing system can build vSD-EONs on-demand according to tenants' application demands and operate them correctly.

**Index Terms**—Elastic optical networks (EONs), Application-driven network (ADN), Network virtualization, Software-defined networking (SDN).

## I. INTRODUCTION

NOWADAYS, with the booming of emerging applications, the current Internet infrastructure is facing more and more challenges due to its ossification [1]. Specifically, emerging applications, such as multimedia cloud computing [2], virtual reality [3], and big data analytics [4], may require different quality-of-service (QoS) guarantees (*e.g.*, bandwidth, end-to-end latency, service availability and security level), while the Internet infrastructure that was designed for delivering best-effort services can hardly satisfy these requirements simultaneously in a cost-efficient manner. Fortunately, this dilemma can be resolved by introducing network virtualization [1], which slices various virtual networks (VNs) over a substrate network and isolates network resources for the VNs to ensure customized network environments. Therefore, with network virtualization, each tenant can ask the infrastructure provider (InP) to customize a VN for its application(s) and multiple VNs can share the substrate network cost-effectively [5, 6].

Meanwhile, when it comes to metro or core networks, one has to consider the physical infrastructure based on optical networking technologies to ensure high capacity. For such networks, even though network virtualization can also be realized in the packet layer [1], slicing VNs directly in

the optical layer has a few benefits, *e.g.*, better bandwidth isolation and securer data transmission [7]. More promisingly, the recent advances on flexible-grid elastic optical networks (EONs) [8] have indicated that agile bandwidth management can be realized directly in the optical layer to make the slicing of optical VNs more adaptive and not be restricted by the fixed spectrum grids [9]. Hence, the InP has more freedom to customize the optical VNs according to tenants' demands/applications. This is extremely important for turning the idea of application-driven networks (ADNs) [10] (*i.e.*, application-led VN construction) into reality.

Specifically, ADNs aim to build VNs for applications and provide logically separate networks to meet the distinct demands of different applications [10, 11], which provide service providers the network environments to grow beyond the current resource-service model that focuses on traffic and bandwidth. As for ADNs, InPs need to identify applications' service characteristics and allocate network resources accordingly, the virtualization of EONs helps to realize this concept from the bottom up. In other words, since an EON works in the physical layer (*i.e.*, Layer-1), the InP would need its help to satisfy certain application demands, such as end-to-end latency, service availability and security level. For instance, future 5G interactive services may require an end-to-end latency less than 5 ms [12], while both fiber transmission and repeated optical-electrical-optical (O/E/O) conversions can easily generate latencies in milliseconds. As these latencies are intrinsic to Layer-1, they can only be reduced with the network control and management (NC&M) efforts in EONs but cannot be compensated with the techniques in upper layers.

To build and operate to tenants' requirements, ADNs not only need to virtualize resources in substrate EONs but also require centralized NC&M to map application demands to network slices and control them accordingly, which makes software-defined networking (SDN) a perfect match [11]. Hence, it would be relevant to consider how to build virtual software-defined EONs (vSD-EONs) over a substrate EON based on the concept of ADN. Previously, people have demonstrated that SD-EONs can facilitate programmable optical networking in [13–19], while the slicing of vSD-EONs has also been realized to further enhance the adaptivity of SD-EONs [20–23]. However, to the best of our knowledge, both the system design and experimental demonstration of application-driven vSD-EON slicing have not been explored before.

In this paper, we extend our preliminary work in [23] to design and implement a novel network system that can realize on-demand application-driven vSD-EON slicing. Compared with the work in [23], the major improvements are as follows.

Z. Zhu, B. Kong, J. Yin, S. Zhao and S. Li are with the School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, P. R. China (email: zqzhu@ieee.org).

Manuscript received June 30, 2017.

Firstly, we redesign the schemes for the communications in the vSD-EON slicing system and optimize the operation procedure, such that on-demand application-driven vSD-EON slicing can be realized. Specifically, the functionality of the virtual network manager (VNMgr) is greatly improved to cover more application requirements. Secondly, we design and implement new modules in the network system's data plane (DP) to detect and report substrate status changes, and thus make the failure recovery procedure in the vSD-EON slicing system fully operational. Lastly and most importantly, we experimentally demonstrate the building and operating of three types of application-driven vSD-EONs. More specifically, our experimental demonstrations consider three scenarios: 1) the tenant's application requires high availability for the DP (High Availability Scenario) [23], 2) the tenant's application is interactive and thus requires short end-to-end latency (Low Latency Scenario), and 3) the tenant's application needs enhanced physical-layer security guarantee (Highly Secure Scenario).

Our experimental demonstrations are based on a network testbed that consists of commercial optical transmission facilities (OTF), bandwidth-variable wavelength-selective switches (BV-WSS'), erbium-doped fiber amplifiers (EDFAs), and high-performance servers with both optical and electrical ports. Hence, the testbed consists of both control plane (CP) and DP elements to facilitate "full-stack" investigations. Our experimental results indicate that the proposed vSD-EON slicing system can build vSD-EONs on-demand according to tenants' application demands, and operate them correctly to satisfy the demands, which, to the best of our knowledge, has not been studied or experimentally demonstrated before.

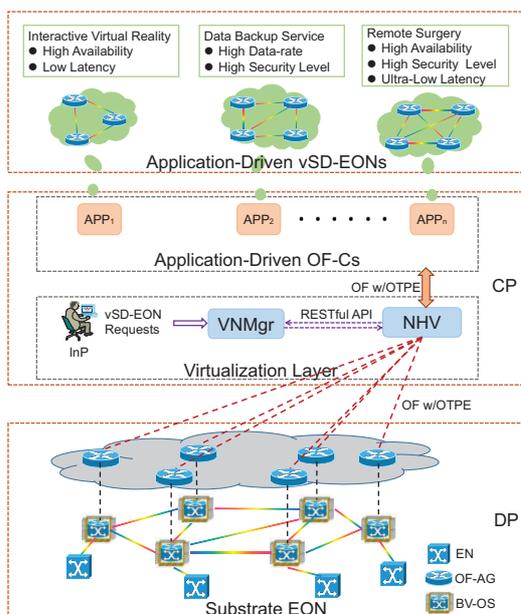


Fig. 1. System architecture to realize application-driven vSD-EON slicing.

The rest of the paper is organized as follows. Section II describes the design of the application-driven vSD-EON slicing system. The operation procedure of the system is discussed in Section III, and we present the experimental results in Section IV. Finally, Section V summarizes the paper.

## II. SYSTEM DESIGN

It is known that in an SD-EON without virtualization, the bandwidth-variable optical switches (BV-OS') built with BV-WSS' are managed by a centralized OpenFlow controller (OF-C) to establish, reconfigure and remove lightpaths [13–15]. Note that, since the BV-OS' usually cannot take OF-based instructions from the OF-C directly, we need to develop an OF agent (OF-AG) as the proxy to communicate with the OF-C for NC&M operations and then configure a BV-OS accordingly. This can be done by expanding the well-known OpenvSwitch [24] to parse the OF messages (*i.e.*, OF v1.3.4 [25] with optical transport protocol extensions [26] (OF w/OTPE)) from the OF-C for spectrum management. An equipment controller should also be programmed to configure the BV-OS through a serial port according to the instructions in the OF messages [23]. Hence, in the DP, we have a shim layer of OF-AGs, each of which connects to a BV-OS locally. There are also a few edge nodes (ENs) in the substrate EON, which terminate the application traffics in the SD-EON. We implement the ENs with the high-performance servers, each of which has both optical and electrical Ethernet ports.

This architecture can be extended for realizing multiple vSD-EONs over a substrate EON. As shown in Fig. 1, since the InP makes the vSD-EONs to share the same substrate elements (*i.e.*, BV-OS' and fiber links), we need a network hypervisor (NHV) sitting in between the OF-Cs of the vSD-EONs and the substrate elements. Here, the NHV is leveraged from the SDN-based network virtualization framework [27] and we design our vSD-EON slicing system based on the framework. Note that, there are other frameworks that can be utilized to realize application-driven network virtualization, *e.g.*, the abstraction and control of traffic engineered networks (ACTN) framework [28, 29]. The NHV works as a translator, which realizes OF message interpretation between the virtual and substrate BV-OS' and enables the OF-C of each vSD-EON to realize application-driven NC&M. Hence, for an OF-C in the CP, it only sees the virtual BV-OS' in its own vSD-EON and controls them transparently to satisfy its application demands. Meanwhile, the OF-AGs of the substrate BV-OS' do not need to know the existence of multiple OF-Cs in the CP, since the OF messages from them are unified by the NHV just like that they were from a single OF-C.

We realize the NHV by modifying OpenVirteX [30] to support OF w/OTPE. However, we need to point out that the NHV itself cannot accomplish on-demand application-driven vSD-EON slicing. This is because to realize application-driven vSD-EON slicing, one needs to calculate the virtual network embedding (VNE) scheme [5, 6] for each vSD-EON carefully according to its application demands, which is out of the NHV's scope. Therefore, we introduce the virtual network manager (VNMgr) [23] in Fig. 1 to handle this task. Basically, the VNMgr and NHV form the virtualization layer, and they communicate with each other by using the RESTful API. The InP uses an interface, which is not included in Fig. 1, to listen for incoming vSD-EON requests from tenant applications. After a vSD-EON request has been admitted and documented by the interface, the InP will forward it to the VNMgr for

provisioning. The VNMgr examines the application demands (*e.g.*, high availability, low latency, large bandwidth, and high security level) of the vSD-EON, calculates the VNE scheme that includes node and link mapping results, and forwards the scheme to the NHV through the RESTful API. Then, the NHV implements the VNE scheme accordingly. Note that, the role of our VNMgr is different from that of the application-based network operation (ABNO) controller discussed in [31–33]. Specifically, once the vSD-EON has been created, the NC&M tasks will be handled by its OF-C while the VNMgr would not get involved. This is different from the ABNO controller, which gets directly involved in the NC&M operations.

Fig. 2 shows the detailed design of the VNMgr, which is a stand-alone software module. The sub-modules in it communicate with each other through internal software interfaces. The input/output module (I/O) handles its communications with both the NHV and the InP's NC&M system, through the RESTful API. When the I/O receives a vSD-EON request, it forwards the request to the request handler module (RHM), which first examines the application demands of the vSD-EON and then obtains a proper VNE algorithm from the VNE algorithm pool (VAP) to calculate the VNE scheme. Note that, the problem of VNE has already been investigated intensively, and various algorithms have been proposed to realize different optimization objectives [6, 7, 9]. Nevertheless, to realize application-driven vSD-EON slicing, there is no universal VNE algorithm that can handle all the possible application demands properly.

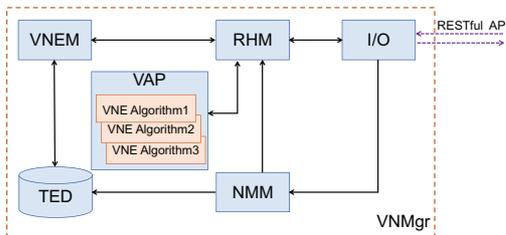


Fig. 2. System design of VNMgr.

Therefore, the VAP is introduced to return the proper VNE algorithms according to the vSD-EON's application demands. For instance, if high availability is required, a survivable VNE (S-VNE) as the one designed in [7] should be provided by the VAP. Also, if high security level is demanded, the VAP can return the location-constrained VNE (LC-VNE) algorithm in [6] to help the vSD-EON avoid certain insecure substrate elements. Moreover, to satisfy multiple application demands simultaneously, the VAP can point to the integrated version of two or more VNE algorithms. For example, a vSD-EON for remote surgery may require high availability and high security level at the same time. For it, the VAP can integrate LC-VNE and S-VNE together to return a VNE algorithm that provides an S-VNE scheme based on the location-constrained principle.

The obtained VNE algorithm is then forwarded to the VNE module (VNM), which calculates the VNE scheme for the vSD-EON with it based on the most updated status of the substrate EON stored in the traffic engineering database

(TED). Finally, if a feasible VNE scheme is obtained, the VNM updates the network status in the TED, and returns the VNE scheme to the RHM, which in turn instructs the I/O to forward the scheme to the NHV. In the meantime, the VNMgr monitors the status of the substrate EON proactively with the network monitoring module (NMM) to update the TED. Also, the NMM shares the changes of status with the RHM and lets it invoke vSD-EON remapping when necessary. For example, in case of a substrate link (SL) failure, the RHM needs to recover the services of the affected vSD-EONs, which demand for high availability, with remapping [23]. We implement the VNMgr with in-house developed software based on Python.

The most significant advantage of our proposed system in Figs. 1 and 2 is that the application-related Layer-1 assistance is all taken care of by the virtualization layer, which is made transparent to the OF-Cs of the vSD-EONs. This saves a lot of hassles for the tenants, and enables them to concentrate more on their applications when developing the OF-Cs. In this work, we realize the OF-C of each vSD-EON based on the ONOS platform [34], since it is compatible with OF w/OTPE and provides enriched supports for application-driven OF-C development. In our testbed, we run the VNMgr, NHV, and OF-AGs on high-performance Linux servers, and use Finisar 1×9 BV-WSS' to realize the BV-OS'. On each input port of a BV-OS, we implement a link monitoring module (LMM) to monitor the optical power level, which will generate an alert message to its OF-AG to report an SL failure, if it finds that the power level is below a certain threshold (*e.g.*, -35 dBm).

### III. OPERATION PROCEDURE

The major functionalities of our vSD-EON slicing system are: 1) creating application-driven vSD-EONs, 2) managing lightpaths in the vSD-EONs to support upper-layer applications, and 3) remapping vSD-EON(s) automatically after a substrate status change. The detailed operation procedure for these functionalities is designed as in Fig. 3.

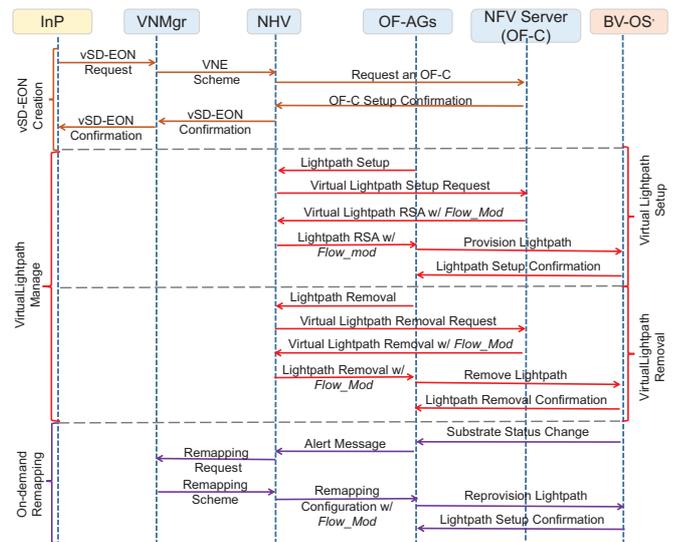


Fig. 3. Operation procedure in application-driven vSD-EON slicing system.

When the InP has admitted a vSD-EON request from the tenant, it forwards the request to the VNMgr, which will in turn

examine the application demands of the vSD-EON and then calculate a proper VNE scheme based on the most updated status of the substrate EON. Note that, the vSD-EON request should contain the information such as the virtual topology, bandwidth requirements on virtual links (VLs), application demands, and EN locations. Upon receiving the VNE scheme from the VNMgr, the NHV implements it by mapping the virtual BV-OS' and VLs to substrate BV-OS' and paths as suggested. Note that, since there is no actual virtual lightpath operating in the vSD-EON at this moment, the NHV only needs to record the VNE scheme internally and prepare the data structure for flow entry translation of the vSD-EON. Then, the NHV requests an OF-C for the vSD-EON, which is instantiated on a network function virtualization (NFV) server and opens to the tenant for installing application-related modules. Finally, the vSD-EON is created, and when the InP receives the VNMgr's confirmation, it hands the OF-C to the tenant, which will then operate the vSD-EON and manage lightpaths in it to support upper-layer applications for the ENs.

Next, when the vSD-EON is operational, its ENs can launch applications by requesting for virtual lightpaths. Specifically, an EN can forward a lightpath setup request to the OF-AG that is on its local BV-OS. The OF-AG encodes the tenant's ID in the request and sends it to the NHV. In the NHV, the tenant ID helps it determine which vSD-EON the lightpath request belongs to, and it then translates the request to a virtual lightpath request accordingly. Next, the virtual lightpath request is forwarded to the corresponding OF-C in the NFV server. The OF-C determines a routing and spectrum assignment (RSA) scheme for the virtual lightpath based on the virtual topology information of the vSD-EON, uses the flow entries in *Flow\_Mod* messages to encode the RSA scheme, and sends the messages to its virtual BV-OS' through the NHV. The NHV parses the *Flow\_Mod* messages, translates the flow entries' contents into substrate EON information (*i.e.*, what can be used by the OF-AGs to configure the BV-OS'), and sends the translated messages to the related OF-AGs. For instance, the virtual lightpath's spectrum assignment in the *Flow\_Mod* messages from the OF-C should be translated into physical spectrum assignment in the substrate EON. The OF-AGs provision the lightpath in the substrate EON by configuring their BV-OS' according to the received *Flow\_Mod* messages. Then, the virtual lightpath has been established for the vSD-EON, and the OF-C then can route application traffics over it. On the other hand, Fig. 3 also shows the procedure for virtual lightpath removal, which follows the similar principle.

Finally, we should notice that the network status of the substrate EON can change when the vSD-EONs are operational, and some of the changes can actually affect their operations. For instance, an SL failure can interrupt the services of a vSD-EON that demands for high availability, and a change on the security level of a substrate BV-OS' can make a vSD-EON that requires high security level not secure anymore. Since these substrate changes are first sensed by the OF-AG and then forwarded to NHV, taking care of them automatically in the virtualization layer can make the response timely and avoid the hassles in the OF-Cs. Therefore, in our vSD-EON slicing system, these changes are handled by remapping the

affected vSD-EONs automatically with the NHV and VNMgr and made transparent to the OF-Cs. As shown in Fig. 3, when a BV-OS detects an SL failure, its LMM reports the exception to the OF-AG on the BV-OS, which will in turn report the substrate status change to the NHV with a *Port\_Status* message. Then, the NHV determines whether a vSD-EON remapping is necessary. If yes, the NHV updates the substrate status in the VNMgr and communicates with it to get the remapping schemes for the affected vSD-EONs. Then, the remapping is implemented by the NHV, which updates the vSD-EONs' embedding schemes and sends corresponding *Flow\_Mod* messages to the related OF-AGs to adjust the RSA schemes of the affected virtual lightpaths.

#### IV. EXPERIMENTAL DEMONSTRATIONS

In this section, we discuss the experimental setup and results to show the proof-of-concept demonstrations of our application-driven vSD-EON slicing system.

##### A. Experimental Setup

We build a network testbed based on the system architecture in Fig. 1, and Fig. 4 shows the setup of the substrate EON, which consists of six BV-OS', fiber links with EDFAs, and a few ENs. The application traffics from/to the ENs can either be aggregated/de-aggregated by the commercial OTF (Huawei Optix OSN3500), which realizes both O/E/O and data-rate conversions, or be directly handled by the 10GbE optical ports on the ENs. The substrate BV-OS' are realized based on Finisar 1×9 BV-WSS', which operate within [1528.43, 1566.88] nm in the C-Band and have a minimum bandwidth allocation granularity of 12.5 GHz, *i.e.*, the bandwidth of a frequency slot (FS) in the EON is 12.5 GHz. The other network elements in the testbed, including the ENs, OF-AGs, NHV, VNMgr, and OF-Cs are implemented on ThinkServer RD540 servers. We use the testbed to conduct three experiments: 1) the tenant's application requires high availability for its DP (High Availability Scenario) [23], 2) the tenant's application is interactive and thus requires short end-to-end latency (Low Latency Scenario), and 3) the tenant's application needs enhanced physical-layer security guarantee (Highly Secure Scenario). Note that, our experimental demonstrations include both CP and DP operations, and thus are "full-stack".

##### B. High Availability Scenario

In this scenario, since the tenant's application requires high availability for its DP, the VNMgr needs to leverage the S-VNE algorithm in [7] to determine both the working and backup mapping schemes of a vSD-EON. This means that when the vSD-EON is created, the backup resources are reserved in the substrate EON for it such that its service would be recoverable during any single SL failure, *i.e.*, the vSD-EON always has a remapping scheme with sufficient backup resources to satisfy the high availability requirement. Then, when an SL failure affects the operation of its VL(s), the VNMgr can invoke on-demand remapping to recover the services on the VL(s). In this demonstration, we conduct experiments for vSD-EON creation, virtual lightpath activation, and on-demand remapping

during an SL failure [23]. Fig. 5 illustrates the experimental scheme for the High Availability Scenario. Here, for simplicity, we only show the scheme in the DP. The tenant requests for a vSD-EON that consists of three virtual BV-OS'. As shown in Fig. 5(a), the original link mapping result for the vSD-EON is  $\{(a, b) \rightarrow \{2-3\}, (b, c) \rightarrow \{3-5\}, (a, c) \rightarrow \{2-5\}\}$ . Then, we disconnect the fiber between substrate nodes (SNs) 3 and 5 to emulate an SL failure that impacts VL  $(b, c)$ . To ensure the availability of the vSD-EON, the NHV and the VNMgr perform link remapping to recover VL  $(b, c)$ , and the remapping result is  $\{(b, c) \rightarrow \{3-4-5\}\}$  as in Fig. 5(b).

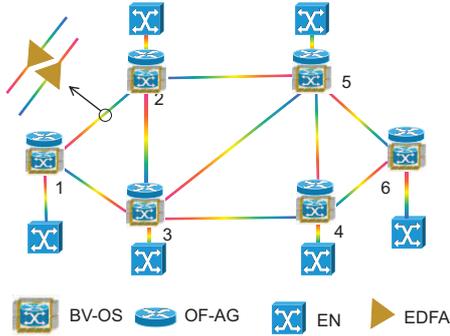


Fig. 4. Experimental setup and topology of the substrate EON.

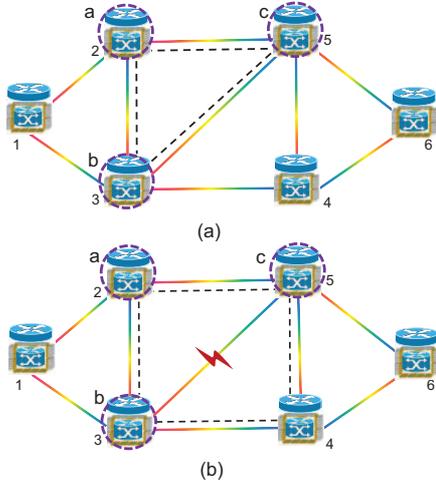


Fig. 5. Experimental scheme for the High Availability Scenario, (a) original mapping result, and (b) remapping result after an SL failure.

Fig. 6(a) shows the Wireshark capture of the control messages used for creating the vSD-EON. In **Step 1**, the InP forwards the vSD-EON request from the tenant to the VNMgr. The VNMgr calculates the VNE scheme for the vSD-EON and sends the result to the NHV in **Step 2**. Next, when the NHV has successfully created the vSD-EON, it notifies the VNMgr in **Step 3**, for confirmation. Finally, the VNMgr confirms the InP regarding the creation of the vSD-EON in **Step 4**. Here, we assume that each VL of the vSD-EON requires 100 GHz bandwidth (*i.e.*, 8 FS'), and thus the NHV reserves the FS-block [344, 351], which covers the FS' whose center wavelengths range within [1531.85, 1532.53] nm on the SLs for the vSD-EON. It takes our system 389 milliseconds to

create the vSD-EON. After the vSD-EON has been created, the InP hands it over to the tenant, which will then set up virtual lightpaths through its OF-C.

The experimental result of activating a virtual lightpath in the vSD-EON is illustrated in Fig. 6(b). Here, we assume that the tenant needs to set up a virtual lightpath on VL  $(b, c)$  for a capacity of 10 Gbps. In **Step 1**, the OF-AG on SN 3 encodes the lightpath request in a *Packet\_In* message and sends it to the NHV. The NHV determines the ownership of the virtual lightpath based on the tenant ID encoded in the message, translates the message, and forwards the translated message to the OF-C of the tenant. Then, in **Step 2**, the OF-C determines that the virtual lightpath should occupy an FS, generates two *Flow\_Mod* messages to install on its virtual BV-OS'  $b$  and  $c$ . The two *Flow\_Mod* messages are sent to the NHV, which translates them to two *Flow\_Mod* messages for SNs 3 and 5 to set up the lightpath in the substrate EON. Upon receiving the *Flow\_Mod* messages, the OF-AGs on the SNs configure the corresponding substrate BV-OS' according to the instructions in them. Hence, the virtual lightpath has been established and the ENs can use it to carry application traffics in the vSD-EON. **Step 3** covers the communications between the NHV and SNs for actually setting up the lightpath.

| Time      | Source | Destination | Protocol | Length | Info   |   |
|-----------|--------|-------------|----------|--------|--|---|
| 15.538829 | InP    | VNMgr       | HTTP     | 1351   | POST /vnManager HTTP/1.1 (application/json)  | ① |
| 15.536266 | VNMgr  | NHV         | HTTP     | 424    | POST /tenant HTTP/1.1 (application/json-rpc) | ② |
| 15.814090 | NHV    | VNMgr       | HTTP     | 66     | HTTP/1.1 200 OK (application/json)           | ③ |
| 15.919544 | VNMgr  | InP         | HTTP     | 66     | Continuation or non-HTTP traffic             | ④ |

(a)

| Time      | Source | Destination | Protocol  | Length | Info                         |   |
|-----------|--------|-------------|-----------|--------|------------------------------|---|
| 78.370439 | Node-3 | NHV         | OF-w-OTPE | 168    | 33209 > 6633 [Type:PacketIn] | ① |
| 78.371524 | NHV    | OF-C        | OF-w-OTPE | 168    | 56742 > 6633 [Type:PacketIn] | ② |
| 78.387396 | OF-C   | NHV         | OF-w-OTPE | 170    | 6633 > 56742 [Type:FlowMod]  | ③ |
| 78.387432 | OF-C   | NHV         | OF-w-OTPE | 170    | 6633 > 56743 [Type:FlowMod]  | ④ |
| 78.407317 | NHV    | Node-5      | OF-w-OTPE | 170    | 6633 > 55102 [Type:FlowMod]  | ⑤ |
| 78.413908 | NHV    | Node-3      | OF-w-OTPE | 170    | 6633 > 33209 [Type:FlowMod]  | ⑥ |

(b)

| Time      | Source | Destination | Protocol  | Length | Info   |   |
|-----------|--------|-------------|-----------|--------|--|---|
| 16.885014 | LMM    | Node-5      | HTTP      | 314    | POST /Proxv HTTP/1.1 (application/json)      | ① |
| 4.134623  | Node-5 | NHV         | OF-w-OTPE | 146    | 54072 > 6633 [Type:PortStatus]               | ② |
| 4.136342  | NHV    | VNMgr       | HTTP      | 188    | POST /vnManager HTTP/1.1 (application/json)  | ③ |
| 4.142609  | VNMgr  | NHV         | HTTP      | 368    | POST /tenant HTTP/1.1 (application/json-rpc) | ④ |
| 4.152990  | NHV    | VNMgr       | HTTP      | 66     | HTTP/1.1 200 OK (application/json)           | ⑤ |
| 4.200687  | VNMgr  | NHV         | HTTP      | 455    | POST /status HTTP/1.1 (application/json-rpc) | ⑥ |
| 4.201488  | NHV    | Node-3      | OF-w-OTPE | 170    | 6633 > 35007 [Type:FlowMod]                  | ⑦ |
| 4.201576  | NHV    | Node-4      | OF-w-OTPE | 170    | 6633 > 38350 [Type:FlowMod]                  | ⑧ |
| 4.201586  | NHV    | Node-5      | OF-w-OTPE | 170    | 6633 > 54072 [Type:FlowMod]                  | ⑨ |
| 4.201789  | NHV    | VNMgr       | HTTP      | 66     | HTTP/1.1 200 OK (application/json)           | ⑩ |
| 4.204786  | VNMgr  | NHV         | HTTP      | 194    | HTTP/1.0 200 OK (application/json)           | ⑪ |
| 16.261718 | Node-5 | LMM         | HTTP      | 66     | HTTP/1.0 200 OK (application/json)           | ⑫ |

(c)

Fig. 6. Wireshark capture of control messages collected in the High Availability Scenario for (a) vSD-EON creation, (b) virtual lightpath activation, and (c) on-demand remapping during an SL failure.

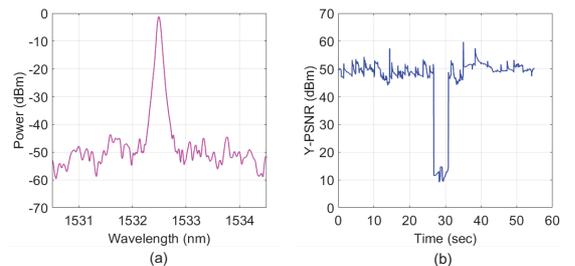


Fig. 7. Experimental results of the High Availability Scenario, (a) optical spectrum of the virtual lightpath and (b) PSNR of video playback.

On the ENs that locally connects to SNs 3 and 5, we run software to realize real-time high-definition (HD) video

streaming over the virtual lightpath. Specifically, the software multiplexes/demultiplexes 50 video streams, each of which has a resolution of  $1920 \times 1080$ , and the total data-rate of the application traffic is around 1 Gbps. The application traffic is sent/received with the 1GbE ports on the ENs, and the data transfer between SNs 3 and 5 is realized with the virtual lightpath generated by the OTF. Specifically, the ENs' 1GbE ports are connected to the 1GbE ports on the OTF, where the application traffic is groomed onto or de-groomed from an optical signal that uses a center wavelength of 1532.47 nm and has 10 Gbps data-rate based on the synchronous transport module level 64 (STM-64). Fig. 7(a) shows the optical spectrum of the virtual lightpath.

When the application of real-time HD video streaming is running, we disconnect the fiber link between SNs 3 and 5 to emulate an SL failure. Fig. 6(c) shows the Wireshark capture of control messages that are used for the on-demand remapping to recover the service of the vSD-EON from the SL failure. In **Step 1**, an LMM on the related BV-OS' (*i.e.*, SNs 3 and 5) detects the SL failure and reports the exception to its OF-AG. Then, in **Step 2**, the OF-AG notifies the NHV about the SL failure with a *Port\_Status* message. After that, the NHV communicates with the VNMgr to get the remapping scheme of the vSD-EON, as shown in **Step 3**. Specifically, VL ( $b, c$ ) will be mapped to SL 3-4-5 instead of SL 3-5, as shown in Fig. 5(b). Next, in **Step 4**, the NHV sends *Flow\_Mod* messages to SNs 3, 4, and 5 to reroute the lightpath that is affected by the SL failure. Note that, the NHV and VNMgr conduct the remapping automatically to recover the service of the vSD-EON, which is made transparent to its OF-C. Finally, in **Step 5**, the NHV talks with the VNMgr to notify that the restoration has been accomplished, and the OF-AG on SN 5 informs its LMM about the same thing.

To confirm that the video streaming does get recovered after the lightpath rerouting, we randomly select a video stream and plot the luminance components peak signal-to-noise-ratio (Y-PSNR) of its playback in Fig. 7(b), which indicates that the video's playback quality is recovered after 4.8 seconds.

### C. Low Latency Scenario

It is known that the VNE schemes of vSD-EONs can be categorized into: 1) transparent and 2) opaque ones [9]. Specifically, in the transparent VNE, we have to ensure that all the VLs in a vSD-EON use the same spectra on SLs, while this constraint does not need to be satisfied in the opaque VNE. Therefore, in a vSD-EON based on the transparent VNE, the virtual lightpaths that go through multiple VLs can always be set up all-optically. On the other hand, the opaque VNE may force the same type of virtual lightpaths in a vSD-EON to use repeated O/E/O conversions and electrical processing in the packet layer, which can generate additional latencies. Meanwhile, in terms of the spectrum utilization, the opaque VNE significantly outperforms the transparent VNE [9]. Hence, between the transparent and opaque VNE schemes, there is a tradeoff between the end-to-end latency and the spectrum utilization, which should be carefully considered by the VNMgr when slicing application-driven vSD-EONs.

In the experiment for the Low Latency Scenario, we consider two vSD-EONs, which are for interactive applications and bulk-data transfers, respectively. Therefore, for the vSD-EON to carry interactive applications, the VNMgr should apply the transparent VNE to avoid the additional latency due to the repeated O/E/O conversions and electrical processing in the packet layer, while for the one to carry bulk-data transfers, the VNMgr can leverage the opaque VNE to reduce spectrum fragmentation [35] and thus save spectrum utilization [36, 37].

Fig. 8 shows the experimental scheme for the Low Latency Scenario. Note that, the workflow to create the vSD-EONs in the Low Latency Scenario is the same as that in the High Availability Scenario, because from the system point of view, the operation procedure of the vSD-EON slicing system should not be affected by application demands. Meanwhile, although the workflows for vSD-EON creation are the same, the actual VNE schemes provided by the VNMgr are different in the two scenarios. In Fig. 8(a), the vSD-EON for interactive applications consists of three virtual BV-OS', and its link mapping result is  $\{(a, b) \rightarrow \{1-2\}, (b, c) \rightarrow \{2-3\}, (a, c) \rightarrow \{1-3\}\}$ . We assume that each of its VL requires a bandwidth of 100 GHz (*i.e.*, 8 FS'), and thus the NHV reserves the FS-block [158, 165], which covers the FS' whose center wavelengths range within [1550.27, 1550.97] nm, on the corresponding SLs. The VNE result of the vSD-EON for bulk-data transfers is shown in Fig. 8(b), in which the link mapping result is  $\{(a', b') \rightarrow \{2-3\}, (b', c') \rightarrow \{3-4\}, (c', d') \rightarrow \{4-5\}, (a', d') \rightarrow \{2-5\}\}$ . This vSD-EON also requires 100 GHz bandwidth on each of its VLs. Since the VNMgr applies the opaque VNE for it, the NHV reserves the FS-block [158, 165] on SLs 3-4, 4-5, and 2-5, while for SL 2-3, the FS-block [344, 351] (*i.e.*, [1531.85, 1532.53] nm) is assigned since the FS-block [158, 165] has already been occupied.

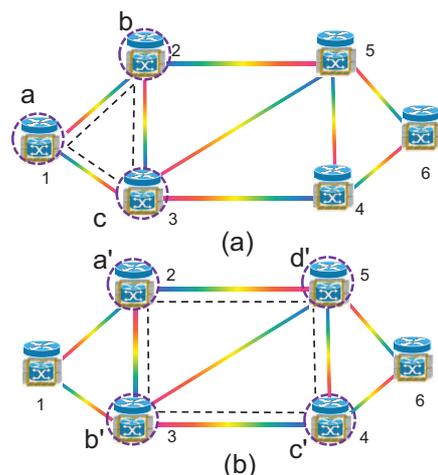


Fig. 8. Experimental scheme for the Low Latency Scenario, (a) vSD-EON with transparent VNE, and (b) vSD-EON with opaque VNE.

Then, two virtual lightpaths are established in the vSD-EONs, for  $a-b-c$  and  $a'-b'-c'$ , respectively. We can see that the virtual lightpath  $a-b-c$  can be set up all-optically, while the application traffics on the virtual lightpath  $a'-b'-c'$  have to experience O/E/O conversions and electrical processing in the packet layer on virtual BV-OS  $b'$  (*i.e.*, SN 3). Since

the creation of the vSD-EONs and the activation of virtual lightpaths follow the same procedures as those in the previous subsection, we omit the experimental results for them here. Figs. 9(a) and 9(b) illustrate the optical spectra on the SLs for the virtual lightpaths in the vSD-EONs based on the transparent and opaque VNE schemes, respectively. The optical spectra confirm our analysis above. Then, we send 100 packets through the two virtual lightpaths and measure the average end-to-end latencies, respectively. The results suggest that for the vSD-EON to carry interactive applications, the average latency is 0.1 millisecond, while for the one to carry bulk-data transfers, the average latency is 0.45 millisecond. Note that, in the experiment, we ensure that the fiber lengths of SLs 1-2 and 3-4 are the same, and thus the additional latency on the virtual lightpath in Fig. 9(b) is due to the O/E/O conversions and electrical processing in the packet layer on SN 3. Meanwhile, we would like to point out that in a practical substrate EON, the SLs may have different fiber lengths, and thus in addition to choosing the transparent VNE scheme, mapping VLs to relatively short substrate paths can also help to reduce the end-to-end latencies of virtual lightpaths.

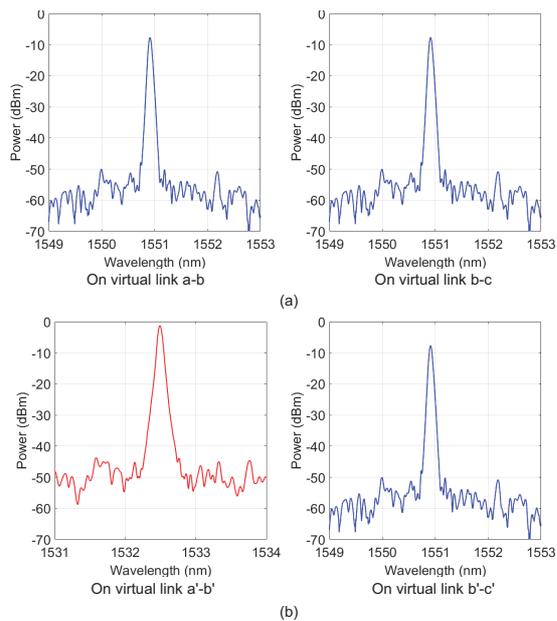


Fig. 9. Optical spectra on SLs for virtual lightpaths in the vSD-EONs with (a) transparent VNE and (b) opaque VNE.

#### D. Highly Secure Scenario

When performing vSD-EON slicing, the InP makes different vSD-EONs share the same substrate elements to improve the cost-effectiveness of its service provisioning. Nevertheless, it is known that in EONs, there would be physical-layer vulnerabilities if different clients share the same physical components [38–40]. For instance, a malicious client can launch a high power jamming attack in the physical-layer to interrupt the data transmissions in other vSD-EONs that share the same fiber link(s) and/or substrate BV-OS' with it. Apparently, these physical-layer vulnerabilities can seriously affect the QoS of the applications that require relatively high

security-levels, *e.g.*, the backup service of sensitive data and the remote surgery. Hence, our application-driven vSD-EON slicing system tries to address this issue by leveraging the LC-VNE algorithm in [6] to help the vSD-EONs avoid insecure substrate elements (*i.e.*, SLs and substrate BV-OS').

Fig. 10 shows the experimental scheme for the Highly Secure Scenario. Here, we consider the situation in which SN 4 and SL 1-2 are insecure. Specifically, a malicious client can connect a broadband amplified spontaneous emission (ASE) noise source to SN 4 to launch the deny-of-service (DoS) attack in the physical-layer, while a high power jamming attack can be launched on SL 1-2 to seize most of its EDFA's gain. Then, there are three vSD-EONs to be provisioned. Among them, vSD-EONs 1 and 2 do not require a high security-level, and thus their VNE schemes in Figs. 10(a) and 10(b) use the insecure substrate elements, while vSD-EON 3 is the one for highly secure applications and its VNE scheme in Fig. 10(c) avoids both SN 4 and SL 1-2.

Next, we first turn on the ASE noise source on SN 4 to emulate a DoS attack. Fig. 11(a) shows the optical spectrum of the ASE noise source. If we consider a virtual lightpath whose substrate path is 3-4-6 in vSD-EON 1, its optical spectra with and without the DoS attack are plotted in Fig. 11(b). It can be seen that the ASE noise degrades the lightpath's optical signal-to-noise ratio (OSNR) from 46.69 dB to 10.48 dB, which is lower than the sensitivity of the optical receiver. Hence, when the DoS attack is present, the application traffics that use the lightpath would be lost. To verify this, we send a 2 Mbps traffic flow over the lightpath in vSD-EON 1 and measure its receiving bandwidth over time. Fig. 11(c) illustrates the result, and we can see that during  $t \in [14, 16]$  seconds, the traffic flow gets disrupted by the DoS attack. Meanwhile, for comparison, we consider a virtual lightpath whose substrate path is 2-3-5 in vSD-EON 3, and send a 2.6 Mbps traffic flow over it. As indicated in Fig. 11(c), the receiving bandwidth of the application using the lightpath in the highly secure vSD-EON 3 has not been affected by the DoS attack.

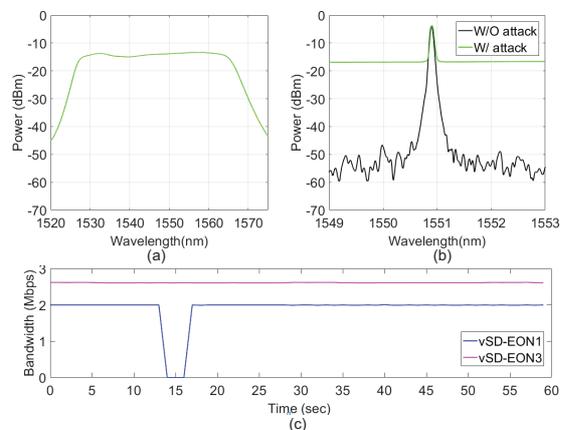


Fig. 11. Experimental results of DoS attack, (a) optical spectrum of ASE noise source on SN 4, (b) optical spectra of lightpath 3-4-6 in vSD-EON 1 with and without DoS attack, and (c) receiving bandwidths of applications using lightpaths in vSD-EONs 1 and 3.

Then, we inject a high power optical signal in SL 1-2 to emulate a jamming attack. Fig. 12(a) shows the optical

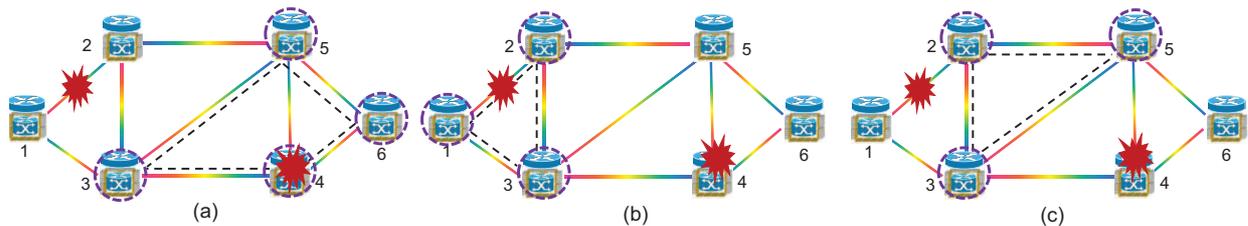


Fig. 10. Experimental scheme for the Highly Secure Scenario, (a) vSD-EON 1 using insecure SN 4, (b) vSD-EON 2 using insecure SL 1-2, and (c) vSD-EON 3 with a high security level.

spectrum of the jamming signal at 1559.78 nm. This time, we consider a virtual lightpath whose substrate path is 1-2 in vSD-EON 2, and its optical spectra for the cases with and without the jamming attack are plotted in Fig. 12(b). We observe that since the jamming signal seizes most of the EDFA's gain on SL 1-2, the power of the lightpath's output at SN 2 decreases from 7.34 dBm to -4.96 dBm, which is lower than the sensitivity of the optical receiver. Therefore, the application traffics that use the lightpath would be lost too. This can be confirmed with the result on the receiving bandwidth in Fig. 12(c), which shows that the receiving bandwidth of the 2 Mbps traffic flow using the lightpath in vSD-EON 2 decreases to 0 during  $t \in [27, 29]$  seconds when the jamming signal is on. On the other hand, Fig. 12(c) also suggests that the receiving bandwidth of the application using the lightpath in the highly secure vSD-EON 3 has not been affected by the jamming attack.

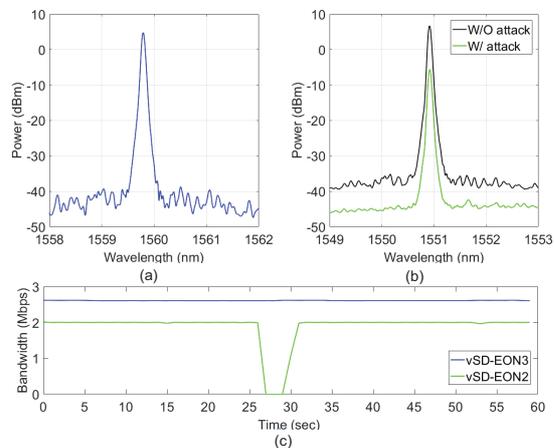


Fig. 12. Experimental results of high power jamming attack, (a) optical spectrum of jamming signal on SL 1-2, (b) optical spectra of lightpath 1-2 in vSD-EON 2 with and without jamming attack, and (c) receiving bandwidths of applications using lightpaths in vSD-EONs 2 and 3.

## V. CONCLUSION

In this paper, we studied how to realize the on-demand slicing of application-driven vSD-EONs based on the concept of ADN. We designed the network system and demonstrated the building and operating of application-driven vSD-EONs with it experimentally. Our experimental demonstrations considered three scenarios, *i.e.*, the High Availability, Low Latency and Highly Secure Scenarios. With an experimental testbed that consists of commercial OTF, BV-WSS, EDFAs, and high-performance servers with both optical and electrical ports,

we verified that the proposed vSD-EON slicing system can build vSD-EONs on-demand according to tenants' application demands and operate them correctly to satisfy the demands.

## ACKNOWLEDGMENTS

This work was supported in part by the NSFC Project 61371117, NSR Project for Universities in Anhui (KJ2014ZD38), the Key Research Project of the CAS (QYZDY-SSW-JSC003), and the NGBWMCN Key Project under Grant No. 2017ZX03001019-004.

## REFERENCES

- [1] M. Chowdhury and R. Boutaba, "A survey of network virtualization," *Comput. Netw.*, vol. 54, pp. 862–876, Apr. 2010.
- [2] P. Lu, Q. Sun, K. Wu, and Z. Zhu, "Distributed online hybrid cloud management for profit-driven multimedia cloud computing," *IEEE Trans. Multimedia*, vol. 17, pp. 1297–1308, Aug. 2015.
- [3] P. Rosedale, "Virtual reality: The next disruptor: A new kind of worldwide communication," *IEEE Consum. Electron. Mag.*, vol. 6, pp. 48–50, Jan. 2017.
- [4] P. Lu, L. Zhang, X. Liu, J. Yao, and Z. Zhu, "Highly-efficient data migration and backup for big data applications in elastic optical interdatacenter networks," *IEEE Netw.*, vol. 29, pp. 36–42, Sept./Oct. 2015.
- [5] M. Chowdhury, M. Rahman, and R. Boutaba, "VineYard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 20, pp. 206–219, Feb. 2012.
- [6] L. Gong, H. Jiang, Y. Wang, and Z. Zhu, "Novel location-constrained virtual network embedding (LC-VNE) in optical datacenter networks," *IEEE/ACM Trans. Netw.*, vol. 24, pp. 3648–3661, Dec. 2016.
- [7] H. Jiang, Y. Wang, L. Gong, and Z. Zhu, "Availability-aware survivable virtual network embedding (A-SVNE) in optical datacenter networks," *J. Opt. Commun. Netw.*, vol. 7, pp. 1160–1171, Dec. 2015.
- [8] Z. Zhu, W. Lu, L. Zhang, and N. Ansari, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," *J. Lightw. Technol.*, vol. 31, pp. 15–22, Jan. 2013.
- [9] L. Gong and Z. Zhu, "Virtual optical network embedding (VONE) over elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 450–460, Feb. 2014.
- [10] G. Zhang, D. Huo, and Y. Li, "All about the user with ADN," *Huawei Commun.*, no. 79, pp. 62–68, Jun. 2016. [Online]. Available: <http://www-file.huawei.com/-/media/CORPORATE/PDF/publications/communicate/79/14-Cutting-Edge-All-about-the-user-with-ADN.pdf>
- [11] Y. Wang, D. Lin, C. Li, J. Zhang, P. Liu, C. Hu, and G. Zhang, "Application driven network: providing on-demand services for applications," in *Proc. of ACM SIGCOMM 2016*, pp. 617–618, Aug. 2016.
- [12] 5G vision. [Online]. Available: <https://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf>
- [13] L. Liu, R. Munoz, R. Casellas, T. Tsuritani, R. Martinez, and I. Morita, "Openslice: an OpenFlow-based control plane for spectrum sliced elastic optical path networks," *Opt. Express*, vol. 21, pp. 4194–4204, Feb. 2013.
- [14] R. Casellas, R. Martinez, R. Munoz, R. Vilalta, L. Liu, T. Tsuritani, and I. Morita, "Control and management of flexi-grid optical networks with an integrated stateful path computation element and OpenFlow controller," *J. Opt. Commun. Netw.*, vol. 5, pp. A57–A65, Oct. 2013.

- [15] C. Chen, X. Chen, M. Zhang, S. Ma, Y. Shao, S. Li, M. Suleiman, and Z. Zhu, "Demonstrations of efficient online spectrum defragmentation in software-defined elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 4701–4711, Dec. 2014.
- [16] Z. Zhu, X. Chen, C. Chen, S. Ma, M. Zhang, L. Liu, and B. Yoo, "OpenFlow-assisted online defragmentation in single-/multi-domain software-defined elastic optical networks," *J. Opt. Commun. Netw.*, vol. 7, pp. A7–A15, Jan. 2015.
- [17] Z. Zhu, C. Chen, S. Ma, L. Liu, X. Feng, and B. Yoo, "Demonstration of cooperative resource allocation in an OpenFlow-controlled multidomain and multinational SD-EON testbed," *J. Lightw. Technol.*, vol. 33, pp. 1508–1514, Apr. 2015.
- [18] X. Chen, M. Tornatore, F. Ji, W. Zhou, C. Chen, D. Hu, S. Zhu, L. Jiang, and Z. Zhu, "Flexible availability-aware differentiated protection in software-defined elastic optical networks," *J. Lightw. Technol.*, vol. 33, pp. 3872–3882, Sept. 2015.
- [19] M. Zeng, Y. Li, W. Fang, W. Lu, X. Liu, H. Yu, and Z. Zhu, "Control plane innovations to realize dynamic formulation of multicast sessions in inter-DC software-defined elastic optical networks," *Opt. Switch. Netw.*, vol. 23, pp. 259–269, Jan. 2017.
- [20] R. Munoz, R. Vilalta, R. Casellas, R. Martinez, T. Szyrkowicz, A. Autenrieth, V. Lopez, and D. Lopez, "Integrated SDN/NFV management and orchestration architecture for dynamic deployment of virtual SDN control instances for virtual tenant networks," *J. Opt. Commun. Netw.*, vol. 7, pp. B62–B70, Nov. 2015.
- [21] J. Yin, J. Guo, B. Kong, H. Yin, and Z. Zhu, "Experimental demonstration of building and operating QoS-aware survivable vSD-EONs with transparent resiliency," *Opt. Express*, vol. 25, pp. 15 468–15 480, 2017.
- [22] J. Yin, H. Yin, S. Liu, Z. Zhou, X. Chen, and Z. Zhu, "On-demand and reliable vSD-EON provisioning with correlated data and control plane embedding," in *Proc. of GLOBECOM*, pp. 1–6, Dec. 2016.
- [23] J. Yin, J. Guo, B. Kong, and Z. Zhu, "Demonstration of survivable vSD-EON slicing with automatic data plane restoration to support reliable video streaming," in *Proc. of OFC*, pp. 1–3, Mar. 2017.
- [24] Open vSwitch. [Online]. Available: <http://www.openvswitch.org/>
- [25] OpenFlow Switch Specification version 1.3.4. [Online]. Available: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.3.4.pdf>
- [26] Optical Transport Protocol Extensions. [Online]. Available: [https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/Optical\\_Transport\\_Protocol\\_Extensions\\_V1.0.pdf](https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/Optical_Transport_Protocol_Extensions_V1.0.pdf)
- [27] A. Blenk, A. Basta, M. Reisslein, and W. Kellerer, "Survey on network virtualization hypervisors for software defined networking," *IEEE Commun. Surveys Tuts.*, vol. 18, pp. 655–685, First Quarter 2016.
- [28] D. Ceccarelli and Y. Lee. Framework for abstraction and control of transport networks, IETF Draft. [Online]. Available: <https://tools.ietf.org/html/draft-ceccarelli-actn-framework-07>
- [29] R. Vilalta, R. Munoz, R. Casellas, R. Martinez, F. Francois, S. Peng, R. Nejabati, D. Simeonidou, N. Yoshikane, T. Tsuritani, I. Morita, V. Lopez, T. Szyrkowicz, and A. Autenrieth, "Network virtualization controller for abstraction and control of OpenFlow-enabled multi-tenant multi-technology transport networks," in *Proc. of OFC 2015*, pp. 1–3, Mar. 2015.
- [30] OpenVirteX. [Online]. Available: <http://ovx.onlab.us/>
- [31] D. King and A. Farrel. A PCE-based architecture for application-based network operations, IETF Draft. [Online]. Available: <https://tools.ietf.org/pdf/draft-farrkingel-pce-abno-architecture-05.pdf>
- [32] R. Munoz, R. Vilalta, R. Casellas, R. Martinez, F. Francois, M. Channegowda, A. Hammad, S. Peng, R. Nejabati, D. Simeonidou, N. Yoshikane, T. Tsuritani, V. Lopez, and A. Autenrieth, "Experimental assessment of ABNO-based network orchestration of end-to-end multi-layer (OPS/OCS) provisioning across SDN/OpenFlow and GMPLS/PCE control domains," in *Proc. of ECOC 2014*, pp. 1–3, Sept. 2014.
- [33] A. Aguado, V. Lopez, J. Marhuenda, O. Dios, and J. Fernandez-Palacios, "ABNO: a feasible SDN approach for multivendor IP and optical networks," *J. Opt. Commun. Netw.*, vol. 7, pp. A356–A362, Feb. 2015.
- [34] Open Network Operating System (ONOS). [Online]. Available: <http://onosproject.org/>
- [35] W. Shi, Z. Zhu, M. Zhang, and N. Ansari, "On the effect of bandwidth fragmentation on blocking probability in elastic optical networks," *IEEE Trans. Commun.*, vol. 61, pp. 2970–2978, Jul. 2013.
- [36] M. Zhang, C. You, H. Jiang, and Z. Zhu, "Dynamic and adaptive bandwidth defragmentation in spectrum-sliced elastic optical networks with time-varying traffic," *J. Lightw. Technol.*, vol. 32, pp. 1014–1023, Mar. 2014.
- [37] M. Zhang, C. You, and Z. Zhu, "On the parallelization of spectrum defragmentation reconfigurations in elastic optical networks," *IEEE/ACM Trans. Netw.*, vol. 24, pp. 2819–2833, Oct. 2016.
- [38] J. Zhu, B. Zhao, and Z. Zhu, "Attack-aware service provisioning to enhance physical-layer security in multi-domain EONs," *J. Lightw. Technol.*, vol. 34, pp. 2645–2655, Jun. 2016.
- [39] J. Zhu and Z. Zhu, "Physical-layer security in MCF-based SDM-EONs: Would crosstalk-aware service provisioning be good enough?" *J. Lightw. Technol.*, in Press, 2017.
- [40] J. Zhu, B. Zhao, and Z. Zhu, "Leveraging game theory to achieve efficient attack-aware service provisioning in EONs," *J. Lightw. Technol.*, vol. 35, pp. 1785–1796, May 2017.