Leveraging master-slave OpenFlow controller arrangement to improve control plane resiliency in SD-EONs

Xiaoliang Chen,[1] **Bin Zhao,**[1] **Shoujiang Ma,**[1] **Cen Chen,**[1] **Daoyun Hu,**[1] **Wenshuang Zhou,**[1] **and Zuqing Zhu**[1,*]

[1]*University of Science and Technology of China, Hefei, Anhui, 230027, China*

[*]zqzhu@ieee.org

Abstract: In this paper, we study how to improve the control plane resiliency of software-defined elastic optical networks (SD-EONs) and design a master-slave OpenFlow (OF) controller arrangement. Specifically, we introduce two OF controllers (OF-Cs), *i.e.*, the master and slave OF-Cs, and make them work in a collaborative way to protect the SD-EON against controller failures. We develop a controller communication protocol (CCP) to facilitate the cooperation of the two OF-Cs. With the CCP, the master OF-C (M-OF-C) can synchronize network status to the slave OF-C (S-OF-C) in real time, while S-OF-C can quickly detect the failure of M-OF-C and take over the network control and management (NC&M) tasks timely to avoid service disruption. We implement the proposed framework in an SD-EON control plane testbed built with high-performance servers, and perform NC&M experiments with different network failure scenarios to demonstrate its effectiveness. Experimental results indicate that the proposed system can restore services in both the data and control planes of SD-EON jointly while maintaining relatively good scalability. To the best of our knowledge, this is the first demonstration that realizes control plane resiliency in SD-EONs.

© 2015 Optical Society of America

OCIS codes: (060.4250) Networks; (060.4510) Optical communications.

References and links

1. O. Gerstel, M. Jinno, A. Lord, and B. Yoo, "Elastic optical networking: a new dawn for the optical layer?" IEEE Commun. Mag. **50**, S12–S20 (2012).
2. Y. Sone, Y. Watanabe, W. Imajuku, Y. Tsukishima, B. Kozicki, H. Takara, and M. Jinno, "Bandwidth squeezed restoration in spectrum-sliced elastic optical path networks (SLICE)," J. Opt. Commun. Netw. **3**, 223–233 (2012).
3. X. Shao, Y. Yeo, Z. Xu, X. Cheng, and L. Zhou, "Shared-path protection in OFDM-based optical networks with elastic bandwidth allocation," in Proc. of OFC 2012, pp. 1–3 (2012).
4. A. Castro, L. Velasco, M. Ruiz, and J. Comellas, "Single-path provisioning with multi-path recovery in flexgrid optical networks," in Proc. of ICUMT 2012, pp. 745–751 (2012).
5. M. Liu, M. Tornatore, and B. Mukherjee, "Survivable traffic grooming in elastic optical networks - shared protection," J. Lightwave Technol. **31**, 903–909 (2013).
6. F. Ji, X. Chen, W. Lu, J. Rodrigues, and Z. Zhu, "Dynamic p-cycle protection in spectrum-sliced elastic optical networks," J. Lightwave Technol. **32**, 1190–1199 (2014).
7. 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. **5**, A57–A65 (2013).
8. 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 **21**, 4194–4204 (2013).

9. 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. **6**, 901–910, (2015).
10. A. Giorgetti, F. Paolucci, F. Cugini, and P. Castoldi, "Fast restoration in SDN-based flexible optical networks," in Proc. of OFC 2014, pp. 1–3 (2014).
11. L. Liu, W. Peng, R. Casellas, T. Tsuritani, I. Morita, R. Martinez, R. Munoz, and S. Yoo, "Experimental demonstration of OpenFlow-based dynamic restoration in elastic optical networks on GENI testbed," in Proc. of ECOC 2014, pp. 1–3 (2014).
12. A. Tootoonchian and Y. Ganjali, "HyperFlow: A distributed control plane for OpenFlow," in Proc. of INM/WREN 2010, pp. 1–6 (2010).
13. P. Fonseca, R. Bennesby, E. Mota and A. Passito, "A replication component for resilient OpenFlow-based networking," in Proc. of NOMS 2012, pp. 933–939 (2012).
14. D. Kreutz, F. Ramos and P. Verissimo, "Towards secure and dependable software-defined networks," in Proc. of HotSDN 2013, pp. 55–60 (2013).
15. H. Li, P. Li, S. Guo and A. Nayak, "Byzantine-resilient secure software-defined networks with multiple controllers in cloud," IEEE Trans. Cloud Comp. **in press**, 1–12 (2014).
16. OpenFlow Switch Specification 1.3.4. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.3.4.pdf
17. "Spectral grids for WDM applications: DWDM frequency grid," ITU-T Rec. G.694.1 (2012).
18. About POX [Online]. Available: http://www.noxrepo.org/pox/about-pox/.

1. Introduction

Recently, elastic optical networks (EONs) have been proposed to achieve flexible bandwidth allocation in the optical layer [1]. Different from the fixed-grid wavelength division multiplexing (WDM) networks, EONs utilize bandwidth-variable transponders (BV-Ts) and wavelength-selective switches (BV-WSS') to provision lightpaths with blocks of spectrally-contiguous frequency slots (FS'), which have much narrower bandwidth than WDM channels.

As maintaining network survivability is always important and necessary for optical networks, people have proposed several protection and restoration schemes for EONs to ensure service continuity during single-link failures [2–6]. Meanwhile, by leveraging the idea of software defined networking (SDN), researchers have realized software-defined EONs (SD-EONs) with OpenFlow (OF) [7–9] and demonstrated dynamic link-failure recovery experimentally [10,11]. However, these studies only addressed the protection and restoration schemes for the data plane and assumed that the control plane, especially the OF controller (OF-C), is intact during network failures. Note that OF-C itself is not failure-proof, especially when there are severe and unpredictable incidents, *e.g.*, natural disasters, which can cause regional network failures. Due to the centralized nature of network control and management (NC&M) in SD-EONs, a failure on OF-C can bring the whole control plane down and cause unimaginable losses. Therefore, how to improve the control plane resilience of SD-EONs should be studied.

Previously, people have considered to use multiple controllers for improving the scalability and resilience of the SDN control plane for packet networks [12–15]. The authors of [12] proposed a logically centralized but physically distributed control architecture for SDN, which leverages event distribution to facilitate the collaboration among controllers. In [13], Fonseca *et al.* designed a replication component to synchronize the network status among controllers, and their protocol counted on the switches to detect controller failure and invoke NC&M task handover. The benefits and challenges of deploying multiple controllers in an SDN were discussed in [14] from the perspectives of resilience and security. More recently, the authors of [15] studied the problem of survivable controller placement in cloud networks, in which they assumed that each switch can communicate with multiple controllers for achieving enhanced resilience.

In this work, inspired by the aforementioned multi-controller proposals, we design a master-slave OF-C arrangement to improve the control plane resiliency in SD-EONs. Specifically, we introduce two OF-Cs, *i.e.*, the master and slave OF-Cs, and make them operate collaboratively to protect the SD-EON control plane against single controller failures. We design the functional

modules for the survivable control plane framework and develop a controller communication protocol (CCP) to facilitate the collaboration between the master and slave OF-Cs. With the CCP, the master OF-C (M-OF-C) can synchronize network status to the slave OF-C (S-OF-C) in real time, while S-OF-C can quickly detect the outage of M-OF-C and take over the NC&M tasks to avoid service disruption. We implement the proposed framework in an SD-EON control plane testbed built with high-performance servers, and perform NC&M experiments with different scenarios of network failures to demonstrate its effectiveness. Experimental results indicate that the proposed system can restore the services in both the data and control planes of SD-EON jointly and achieve relatively good scalability for the control plane. To the best of our knowledge, this is the first demonstration of realizing control plane resiliency in SD-EONs.

The rest of this paper is organized as follows. Section 2 describes the network architecture and the system and protocol designs for realizing control plane resiliency in SD-EONs. The operation principle of the proposed framework is presented in Section 3, and we discuss the experimental demonstrations in Section 4. Finally, Section 5 summarizes the paper.

2. Survivable SD-EON with control plane resiliency

2.1. Network architecture

Figure 1 shows the network architecture of the survivable SD-EON that uses the master-slave OF-C arrangement to realize control plane resiliency. We introduce two OF-Cs, *i.e.*, M-OF-C and S-OF-C, in the control plane and connect each of them to all of the OF agents (OF-AGs). Each OF-AG is locally attached to an optical network element to control the service provisioning in the data plane [8, 9]. When the SD-EON is in its normal state, M-OF-C processes the OF messages from the OF-AGs, calculates the service provisioning schemes for lightpath requests, and instructs the OF-AGs to manage lightpaths in the data plane. However, when there is a network outage and M-OF-C is down, S-OF-C detects the failure immediately and takes over the NC&M tasks to avoid service disruption. In the SD-EON, OF-Cs communicate with OF-AGs with an extended OF protocol [9], while OF-Cs interact with each other using a controller communication protocol (CCP) that we design in this work. Note that the latest OF specification, *i.e.*, OF v1.3.4 [16], has included the support for a switch to connect with multiple OF-Cs, but the coordination among OF-Cs and the hand-over mechanism upon failures are not specified.

M-OF-C and S-OF-C have the same structure, as shown in Fig. 2(a). Here, the OF-Cs inherit the structure that we designed in [9], except that the controller communication module (CCM) is added to synchronize the most-updated network status between OF-Cs. Specifically, when
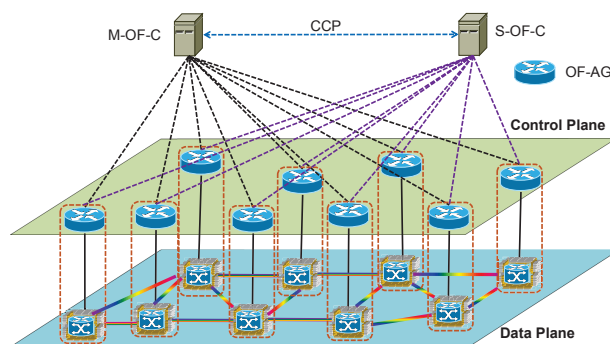

Fig. 1. Architecture of survivable SD-EON using master-slave OF controller arrangement.

a lightpath is provisioned, the resource provisioning module (RPM) in M-OF-C synchronizes the information to S-OF-C using its CCM. To guarantee reasonably good security, the communication between CCMs is built over a virtual private network (VPN) channel, over which the actual control messages are sent through TCP connections. Figure 2(b) shows the structure of an optical network element, which carries the lightpaths in the SD-EONs.

2.2. Protocol design

For the communication between OF-Cs and OF-AGs, we use an extended OF protocol, similar to our previous work in [9]. Basically, we modify the *Packet_In* message to include the information of a lightpath request, and extend the flow-matching fields to ensure that a lightpath can be correctly identified by each network element. Figure 3(a) shows the details of the OF extensions. Here, the *Priority* field indicates whether the flow-table is used for a working (*Priority* = 2) or a backup (*Priority* = 1) lightpath. In this work, we consider the resiliency in both the control and data planes. For the control plane, we leverage the master-slave OF-C arrangement to protect the service against single controller failures. While in the data plane, we incorporate shared path protection (SPP) to improve the availability of data transmission service. In order to support M-OF-C and S-OF-C, we extend the *Vendor* message in OF such that an OF-C can change its role dynamically with it. Specifically, the *Subtype* field indicates whether the message is a role request (from OF-C to OF-AG) or a role reply (from OF-AG to OF-C), while the *Role* field conveys the OF-C's role, *i.e.*, M-OF-C (*Role* = 1) and S-OF-C (*Role* = 2).

Figure 3(b) shows the structure of the *Synch_Request* message that the CPP uses for the communication between M-OF-C and S-OF-C. Basically, *Synch_Request* can be used for two purposes, **1)** M-OF-C and S-OF-C use it to synchronize the network status (*Flag = 1*), and **2)** the OF-Cs use it as a keep-alive signal when there is no need to synchronize network status (*Flag = 0*). For the former purpose, the *Request_ID*, *Service_Duration*, *Working_Path* and *Backup_Path* fields carry the information of a provisioned lightpath, and the last three fields are only used when there are data plane failures and certain lightpaths need to be restored. When *Synch_Request* is used as a keep-alive signal, S-OF-C sends it to M-OF-C every T_1, and upon receiving it, M-OF-C should reply with a *Synch_Reply* message immediately. If S-OF-C cannot
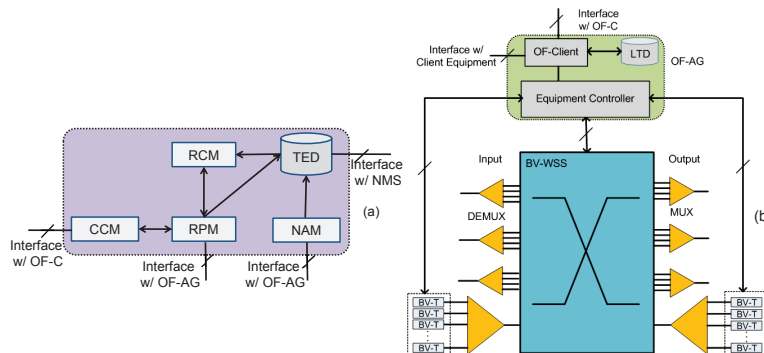

Fig. 2. Detailed structures of (a) OF-C and (b) OF-AG and optical network element in SD-EON. NMS: Network management system, RCM: Resource computation module, TED: Traffic engineering database, RPM: Resource provisioning module, NAM: Network abstraction module, CCM: Controller communication module, LTD: Local traffic database, OF-Client: OpenFlow client, BV-WSS: Bandwidth-variable wavelength-selective switch, BV-T: Bandwidth-variable transponder.

receive the *Synch_Reply* within T_2, it records a keep-alive timeout. And after N consecutive timeouts, S-OF-C determines that M-OF-C is down and starts to take over the NC&M tasks in the SD-EON. The actual values of T_1, T_2 and N should be decided based on the network environment. In our implementation, we use $T_1 = 5$ seconds, $T_2 = 100$ milliseconds and $N = 1$ for the simplicity. When S-OF-C has done the switchover and become the new M-OF-C, it reports the incident to the network operator. Then, the network operator will check the network and find out the proper procedure to take for recovering the failure.

Note that in practical network operations, CCP may need to synchronize more information between M-OF-C and S-OF-C, such as topology changes *etc*. Since this work is just a proof-of-concept demonstration of realizing control plane resilience in SD-EONs, we will incorporate more functionalities on synchronization in our future work.

3. Operation principle

This section discusses the operation principle of the survivable SD-EON. When the SD-EON is in the normal state, each OF-AG connects to both M-OF-C and S-OF-C simultaneously but only interacts with M-OF-C for sending and receiving OF messages. Upon receiving a lightpath request for client traffic, the OF-AG on the source node encodes the request's information in a *Packet-In* message and sends it to M-OF-C. M-OF-C receives the *Packet-In* and calculates the service provisioning scheme for the working and backup lightpaths. If the request is provisioned successfully, M-OF-C encodes its information in a *Synch_Request* message and sends it to S-OF-C. S-OF-C parses the *Synch_Request* for the information of the newly provisioned request, updates its traffic engineering database (TED) accordingly and replies with a *Synch_Reply*. Meanwhile, S-OF-C can query the status of M-OF-C by sending *Synch_Request* periodically.

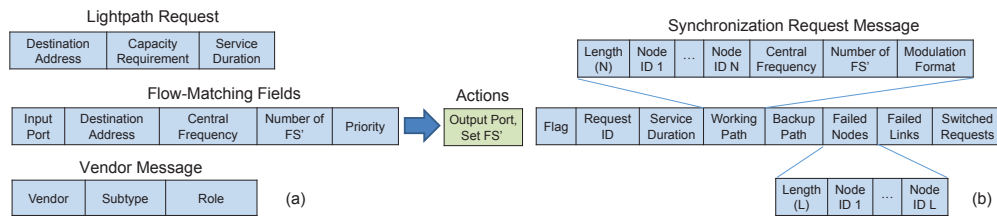In this work, we try to protect the SD-EON against the failures in both the control and data

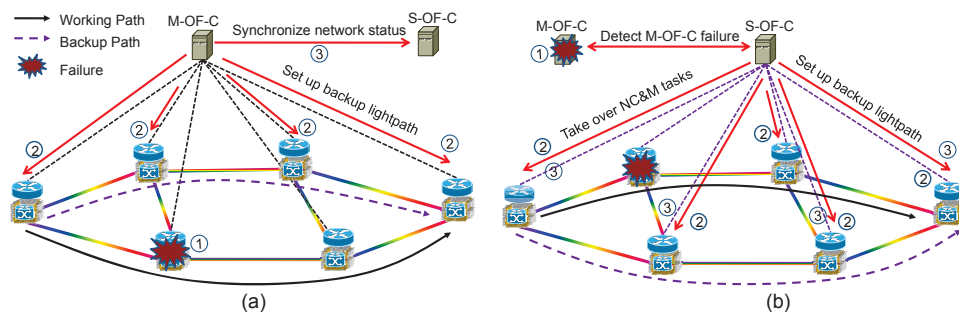Fig. 3. Protocol design to realize survivable SD-EON, (a) Extended OF and (b) CPP.

Fig. 4. Network restoration procedure for the cases: (a) Failure only impacts the data plane, and (b) Failure impacts both the data and control planes.

planes, and hence consider two failure scenarios, **1)** the failure only affects the data plane but M-OF-C is intact, and **2)** M-OF-C fails simultaneously with a data plane element. The corresponding restoration procedures for them are described as follows.

- **Scenario 1**: As shown in Fig. 4(a), M-OF-C detects the failure in the data plane by monitoring the corresponding OF connection or examining the *Port_Status* messages from OF-AGs. Then, it finds out the affected lightpaths and recovers them by setting up the backup ones. When the lightpaths are restored, M-OF-C includes the information of both the failed nodes/links and the restored lightpaths in a *Synch_Request* message and sends it to S-OF-C for network status synchronization.

- **Scenario 2**: S-OF-C detects the outage of M-OF-C after a timeout has happened when it expects a *Synch_Reply* message. Then, it sends *Vendor* messages to all the OF-AGs, informs them to change its role to M-OF-C (*i.e., Role = 1*), and takes over the NC&M tasks in the SD-EON. Next, S-OF-C detects the failure in the data plane and restores the affected lightpaths with the method depicted in **Scenario 1**. An illustrative example for the aforementioned operations is in Fig. 4(b).

4. Experimental demonstration

We implement the proposed survivable SD-EON framework in an OF-based control plane testbed that consists of 14 stand-alone OF-AGs and two OF-Cs. Each OF-AG is programmed based on Open-vSwitch running on high performance Linux servers (ThinkServer RD540), while the OF-Cs are realized with the POX platform and they also run on independent servers. Note that the servers for OF-Cs also host the network management system (NMS), which provides a web-based graphical user interface (GUI). Figure 5(a) shows the GUI on M-OF-C, which illustrates the topology of the SD-EON and the connections among OF-AGs and OF-Cs. Similar to our previous work in [9], we focus on the control plane of the survivable SD-EON and the data plane is emulated. In other words, each OF-AG configures a virtual software-emulated network element but not a real BV-T or BV-WSS. The emulated network topology is shown in Fig. 5(b), which is the 14-node NSFNET topology and in the SD-EON, M-OF-C and S-OF-C locate close to *Nodes* 8 and 4, respectively.
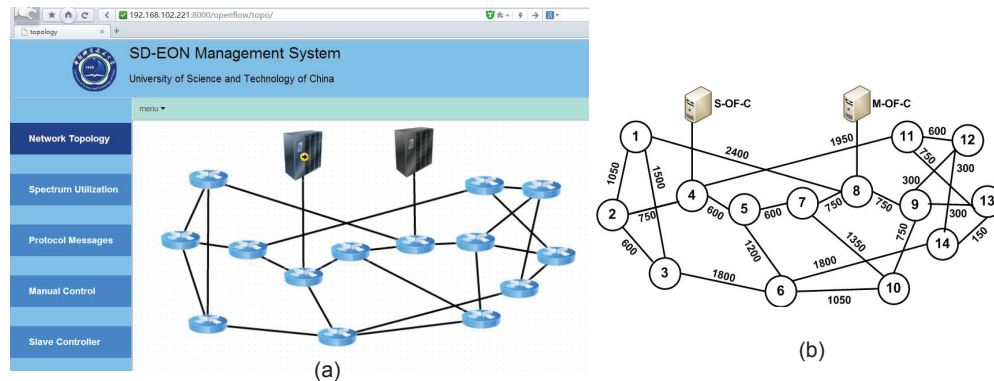

Fig. 5. (a) Interface of the network management system (NMS) for the SD-EON (b) Emulated NSFNET topology to in the experimental testbed (fiber lengths in kilometers).

4.1. Normal service provisioning

We first conduct experiments on normal service provisioning in the survivable SD-EON. Figure 6(a) presents the messages captured in M-OF-C for setting up a lightpath from *Node* 7 to *Node* 9. It can be seen that the control plane latency is around 44 msec. During the service provisioning, M-OF-C also sends a *Synch_Request* message to S-OF-C to synchronize the network status and the details of the message is shown in Fig. 6(b), where we can see that M-OF-C determines the working lightpath as 7→8→9, assigns the FS-block [10, 14] to it, and selects QPSK as its modulation format. In the meantime, S-OF-C sends a *Synch_Request* message to M-OF-C every 5 seconds as the control plane keep-alive, which can also be seen in Fig. 6(a).

We then conduct dynamic lightpath provisioning experiments to stress-test the proposed survivable SD-EON framework. We assume that the EON works in C-Band and each fiber link can accommodate 358 FS', each of which has a bandwidth of 12.5 GHz. The dynamic lightpath requests are generated by each OF-AG according to the Poisson process. The bandwidth requirement of each request is uniformly distributed within [25, 250] Gb/s. Note that the lightpath requests in optical networks usually do not come in as rapidly as the traffic demands in packet networks. Therefore, in practice, the operator usually serves lightpath requests with a period of tens of minutes or even hours. However, in the experiments, since we want to stress-test the proposed system, we use the scenario in which one or more than one requests can arrive in each second in the SD-EON and run each experiment for 500 seconds. The holding time of each request follows a negative exponential distribution with an average of 50 seconds.

We record the maximum number of flow entries (*i.e.*, the number of in-service lightpaths) stored in M-OF-C, and the average numbers of CCP and OF messages per second in Table 1. We can see that with a relatively high traffic load such that the request blocking probability is around 12.3%, the maximum number of flow entries stored in M-OF-C is 151 and the numbers of CCP and OF messages per second reach 5.66 and 25.92, respectively. Meanwhile, it has been shown that POX platform can handle over 30,000 flows per second [18]. Therefore, we verify that the proposed system has relatively good scalability.

4.2. Data plane service restoration

We then perform experiments to verify that the survivable SD-EON can achieve data plane service restoration. Here, we consider the failure scenario in which the data plane has outage but M-OF-C is intact. Figure 7(a) shows the messages used for recovering the lightpath 7→8→9 when *Link* 7→8 is broken. It can be seen that M-OF-C first detects the link failure by receiving
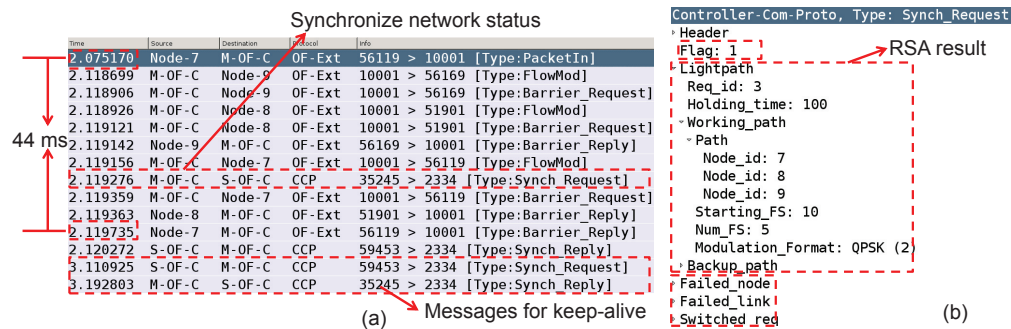

Fig. 6. (a) Messages captured in M-OF-C for setting up a working lightpath, and (b) Details of the *Synch_Request* message for synchronizing network status from M-OF-C to S-OF-C.

Table 1. Results from Stress-Tests with Dynamic Lightpath Provisioning

Traffic Load (requests per second)	1	1.5	2	2.5	3
Request Blocking Probability	0	0	0.0380	0.0584	0.1227
Maximum Number of Flow Entries	64	98	120	136	151
Average Number of CCP Messages per second	2.40	3.40	4.25	5.11	5.66
Average Number of OF Messages per second	12.41	17.62	19.32	23.79	25.92

a *Port_Status* message from *Node* 7, and then it sets up the backup lightpath 7→10→9 to restore the data transmission service. The whole restoration process takes around 24 msec. Figure 7(b) presents the details of the *Flow_Mod* message used for installing the backup lightpath, and M-OF-C uses *Priority = 1* to tell the OF-AG that the message is for service recovery. We also observe that M-OF-C informs S-OF-C about the link failure with a *Synch_Request* message.

4.3. Joint service restoration for the control and data planes

Finally, we conduct experiments to show that the survivable SD-EON can restore the services in both the control and data planes jointly. Here, we consider the situation that both *Node* 8 and M-OF-C becomes unavailable during network operation. Figure 8(a) shows the messages captured in S-OF-C for the joint restoration process. First of all, S-OF-C detects the failure of M-OF-C when it sees a timeout during expecting a *Synch_Reply* message from M-OF-C. Note that in a more practical case, the mechanism for detecting the failure of M-OF-C should be more sophisticated, *e.g.*, S-OF-C may try to retransmit the "keep-alive" *Synch_Request* message for several times before claiming that M-OF-C is down. Then, S-OF-C takes over the NC&M tasks in the SD-EON by sending the *Vendor* messages to all the OF-AGs and letting them switch its role from S-OF-C to M-OF-C. Figure 8(b) shows the *Vendor* message captured on an OF-AG. Note that, in Fig. 8(a), we do not include all the *Vendor* messages due to the space limitation. After taking over the NC&M tasks, S-OF-C finds the lightpaths that are impacted by the failure and restores them. The messages in Fig. 8(a) show that S-OF-C recovers the broken lightpath 7→8→9 with the backup lightpath 7→10→9.

To evaluate the scalability of the survivable SD-EON for handling network failures, we also conduct experiments to measure the control plane latencies for restoring different numbers of lightpaths in a batch. The results are plotted in Fig. 8(c). It can be seen that the restoration latencies increase steadily with the number of lightpaths to be restored, and it takes the SD-EON around 94 msec to restore 50 lightpaths. These results verify that the proposed survivable
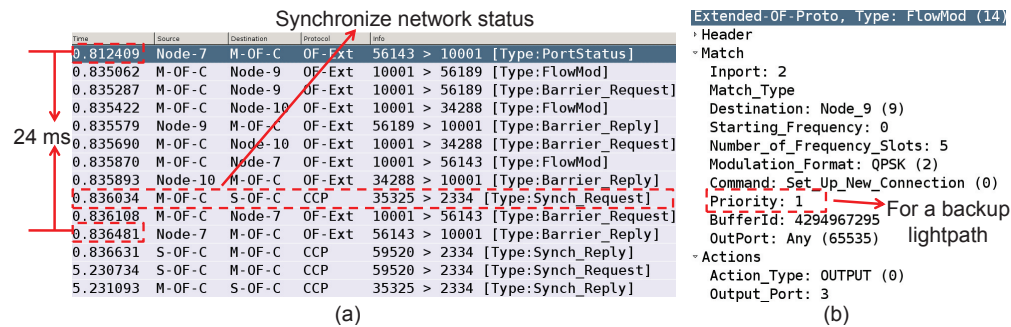

Fig. 7. (a) Messages captured in M-OF-C for installing a backup lightpath, and (b) Details of the *Flow_Mod* for setting up the backup lightpath.
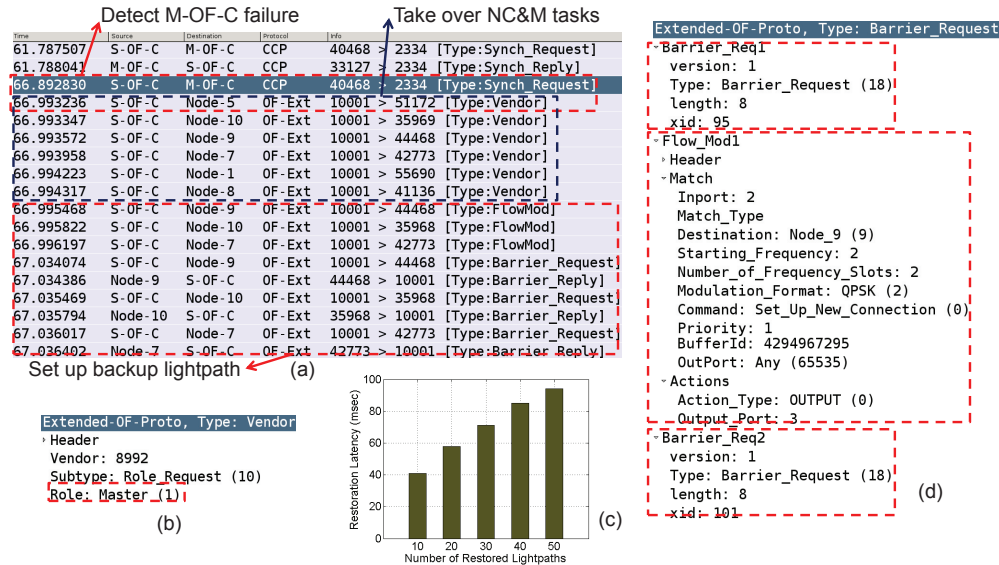
Fig. 8. (a) Messages captured in S-OF-C for restoring a simultaneous failure of M-OF-C and a node, (b) Details of a *Vendor* message, (c) Control plane latency for restoring different number of lightpaths, and (d) Details of a *Bundle* message.

SD-EON has a relatively good scalability. We achieve this by leveraging the message bundling mechanism in OF [16], which can merge several OF messages that need to be sent to the same OF-AG into a single *Bundle* message. This mechanism reduces the control plane signaling latency dramatically. An example of the *Bundle* message is illustrated in Fig. 8(d), where we merge two *Barrier_Request* messages and one *Flow_Mod* message into a *Bundle* message.

5. Conclusion

This paper designed a master-slave OF-C arrangement to improve the control plane resiliency in SD-EONs. We introduced two OF-Cs, *i.e.*, M-OF-C and S-OF-C, and made them operate collaboratively to protect the SD-EON control plane against single controller failures. The proposed framework was implemented in an SD-EON control plane testbed built with high-performance servers, and we performed NC&M experiments with different scenarios of network failures to demonstrate its effectiveness. Experimental results indicated that the proposed system could restore services in both the data and control planes of SD-EON jointly while maintaining relatively good scalability.