Survivable Control Plane Establishment With Live Control Service Backup and Migration in SD-EONs

Bin Zhao, Xiaoliang Chen, Jing Zhu, and Zuqing Zhu

***Abstract*—It is known that software-defined elastic optical networks (SD-EONs) are programmable and application-aware. However, due to the centralized network control and management, SD-EONs are vulnerable to the network failures that can affect control plane operations. In this paper, we study the problem of survivable control plane establishment (SCPE), i.e., the controller placement and related communication planning for control service backup and migration to protect the control plane of an SD-EON against single node failures. We first propose a novel mutual backup model to improve the survivability of the control plane with high protection efficiency. An integer linear programming (ILP) model is then formulated to solve the SCPE problem with the objective to minimize both the number of deployed OpenFlow controllers (OF-Cs) and the average control channel latency. We also propose a time-efficient heuristic and use simulations to verify that it can obtain similar solutions to those of the ILP. On top of the theoretical investigation, we design and implement the system to facilitate live control service backup and migration with SCPE in an SD-EON control plane testbed. Experimental results demonstrate that the proposed scheme works efficiently, and compared with that using a single OF-C, our scheme achieves much shorter average provisioning latency in dynamic provisioning.**

***Index Terms*—Controller placement; Control plane resiliency; Mutual backup; OpenFlow; Software-defined elastic optical networks (SD-EONs).**

I. Introduction

Since its inception, the flexible-grid elastic optical network (EON) has been considered as a promising candidate for next-generation optical networks [1]. Specifically, compared with the existing fixed-grid wavelength division multiplexing (WDM) networks, EONs achieve higher spectrum efficiency and more adaptive resource allocation in the optical layer [2]. These advantages are realized with the bandwidth variable transponders (BV-Ts) and switches (BV-WSSs), which manipulate a set of spectrally contiguous frequency slots (FSs) to establish and manage each lightpath. Meanwhile, since the FSs have much narrower bandwidth than the conventional wavelength channels and the resource allocation based on them is more flexible, the problem scale of network control and management (NC&M) becomes significantly larger in EONs. Hence, we need to incorporate more intelligence in NC&M to operate EONs cost-effectively [3,4].

It is known that by decoupling the control and data planes, software-defined networking (SDN) can make the NC&M in optical networks programmable, adaptive, and application-aware [5]. Specifically, software-defined optical networks (SDONs) can typically be realized by leveraging protocols such as the OpenFlow (OF) [6], the network configuration protocol (NETCONF) with YANG as a data modeling language [7], and the path computation element communication protocol (PCEP) [8]. Here, OF is an open standard protocol that incorporates flow-based switching and centralized controller(s) to facilitate software-defined routing, switching, and network management. Moreover, to support software-defined EONs (SD-EONs), the latest OF specification (version 1.5) [6] has included the extensions for identifying flexible-grid optical flows. Hence, by implementing the OF protocol in the control plane, the network operator can manage the data plane elements (i.e., BV-Ts and BV-WSSs) in an EON intelligently with one or more controllers and realize a SD-EON [5,9–12].

In the meantime, as an optical fiber can carry over Tb/s traffic, ensuring network survivability is vital in optical networks. Hence, researchers have considered the data plane failures in SD-EONs and proposed a few schemes to improve the data plane resiliency [13–15]. Note that in a practical SD-EON, the control plane is definitely not failure-proof without specific considerations on the control plane resiliency. For instance, as it counts on the centralized controller to operate correctly, a controller failure can make the network unable to respond to any requests from the clients, and thus cause unimaginable losses to the network operator.

Moreover, to make the controller respond in a timely fashion to the clients' requests, the operator has to consider the round-trip delay between the controller and its data plane elements carefully; otherwise, the quality-of-service (QoS) of the SD-EON can be affected. This is especially true for the case in which the SD-EON is built for a backbone network with geographically distributed (geo-distributed) nodes. For such a backbone SD-EON, building fiber infrastructure among the nodes is very expensive, and thus the

Manuscript received January 21, 2016; revised April 8, 2016; accepted April 8, 2016; published May 13, 2016 (Doc. ID 257770).

The authors are with the School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China (e-mail: zqzhu@ieee.org).

http://dx.doi.org/10.1364/JOCN.8.000371

communications between the control and data planes (i.e., the control channels) have to share the fiber links with the data plane communications. Therefore, a component failure in the data plane can bring down the control channels too, even though the controller is intact. Although the control channels can be recovered by leveraging either the IP rerouting in upper-layer packet networks or the restoration mechanism implemented for the data plane, the round-trip delays may be increased to a level such that the SD-EON's QoS is impacted badly. Hence, we can see that in SD-EONs, the control plane resiliency is also very important and thus needs immediate attention. However, to the best of our knowledge, methods to improve the control plane resiliency in SD-EONs are still under-explored.

In this paper, we focus on improving the control plane resiliency in SD-EONs with survivable control plane establishment (SCPE) and try to protect the NC&M operations against single node failures. First, to address the controller failures caused by node failures, we propose a novel mutual backup model to improve the controller's survivability with relatively high protection efficiency. Specifically, our mutual backup model uses multiple controllers to manage an SD-EON, where they serve as the backups of one another, and for different data plane elements, a controller's role can be either working (i.e., a master controller) or backup (i.e., a slave one). Second, with the consideration of the latencies experienced by the control channels, we formulate an integer linear programming (ILP) model to solve the SCPE problem with the objective of minimizing both the number of deployed controllers and the average path length of the control channels. We also propose a heuristic to solve the SCPE problem quickly and verify that it can obtain near-optimal solutions. Finally, we implement an SD-EON control plane system with SCPE to demonstrate our proposal experimentally and realize live control service backup and migration in it.[1]

The rest of the paper is organized as follows. Section II briefly reviews the related work. In Section III, we discuss the network model and formulate the SCPE problem for SD-EONs. The ILP model and heuristic that solve the SCPE problem are presented in Section IV. Section V introduces the system design for live control service backup and migration, and the experimental demonstrations are shown in Section VI. Finally, Section VII summarizes the paper.

II. Related Work

Previously, in [5,9,11,16], based on the idea of SDONs, researchers have designed and demonstrated the network architectures and mechanisms for realizing dynamic lightpath provisioning. Also, with the architecture of SDONs, several applications, such as data-center service migration [17], spectrum defragmentation [18,19], and advance reservation request provisioning [20], have been studied too. However, none of these studies have addressed the problem of network survivability. The work in [13–15,21] considered how to improve the data plane resiliency in SDONs and demonstrated several fast and efficient failure recovery mechanisms. Nevertheless, they considered neither the controller failures nor the data plane failures that might affect control channels.

Previously, for packet networks using SDN, researchers have made various efforts toward improving the control plane resiliency. The studies in [22,23] considered controller failures and proposed to leverage a logically centralized but physically distributed control plane to improve the control plane resiliency. However, they did not explain the detailed procedure for control service backup and migration. Moreover, for a large-scale network, a fully distributed control plane will limit the network programmability and cause synchronization and scalability issues. Meanwhile, in this work, we provide detailed system design for realizing live control service backup and migration and also formulate an optimization to reduce the number of deployed controllers for balancing the tradeoff between the control plane resiliency and the system scalability. In [24], the authors designed the mechanisms and relevant protocol for control plane service recovery, but they did not address the problem of network planning with controller placement. Note that several existing SDN systems, such as ONOS [25] and OpenDaylight [26], also offered control plane resiliency with controller backup and migration, and their performance was documented in [27]. However, the controller protection schemes defined in [25,26] do not consider the mutual backup model discussed in this work, and thus their protection efficiency could still be improved.

The studies in [28,29] investigated the controller placement problem [30] for survivable SDN network planning and proposed a few algorithms to improve the control plane resiliency. Nevertheless, they still relied on the one-to-one backup model that only resulted in relatively low protection efficiency and also did not design a detailed procedure for control service backup and migration. More importantly, as the background of their problems was the packet networks that only covered a relatively small geographical area, IP rerouting can recover the control channels promptly without introducing intolerable round-trip delay, upon component/link failures. This, however, might not be the case in our problem, which considers a backbone SD-EON with geo-distributed nodes.

Inspired by the aforementioned studies for packet SDN networks, we have designed and demonstrated a master–slave controller arrangement for SD-EONs in [31]. However, the problem of SCPE, i.e., the controller placement and related control channel planning for control service backup and migration, still has not been addressed and should be investigated carefully to make the solution more practical.

[1]Note that, with minor or even no modifications, the SCPE approaches proposed in this work can also be applied to the SDONs based on other physical infrastructures, e.g., fixed-grid single- and mixed-line-rate WDM networks. However, to be scientific, we would not generalize the background of the SCPE approaches, since our system implementation consists of special considerations on the service provisioning in EONs, i.e., the OF extensions for flexible grids and the calculations on routing and spectrum assignments.

III. Survivable Control Plane Establishment in SD-EONs

We consider a backbone SD-EON with geo-distributed nodes, each of which is treated as an equipment site that can carry either only the data plane elements (i.e., BV-Ts and BV-WSSs) or both the data plane elements and an OpenFlow controller (OF-C). Hence, if a node carries an OF-C, a failure on it will bring down both its data plane elements and the OF-C. In the SD-EON, the nodes are interconnected by fiber links. Note that, if we consider the large geographical coverage of the backbone SD-EON and the very low bandwidth requirement of the control channels (i.e., normally less than 100 Mb/s), building independent fiber infrastructure just for the control channels would be both prohibitively expensive and unnecessary. Therefore, we assume that the control channels share the fiber links with the data plane communications.

Due to the low data rate of a control channel for control information exchange, it may not fully occupy a lightpath between its two end nodes. Instead, its traffic should be statistically multiplexed with other data transmissions between the same node pair and share the corresponding lightpath with them. Then, at its destination node, the control traffic will be electrically terminated and demultiplexed so that it can be processed by the control plane element (i.e., OF-C or OF-AG) on the node. In this work, we determine the routing of the lightpaths that carry the control channels. This is because the propagation delay between an OF-C and its data plane elements contributes to the round-trip delay of the control messages, and hence the control channels' path lengths should be maintained within a reasonable range for QoS guarantee.[2]

Figure 1 shows the network architecture to realize SCPE in an SD-EON. The data plane consists of optical switches, each of which is placed in a node, and includes network elements such as BV-Ts and BV-WSSs to set up lightpaths for the actual data transmission. The network elements are managed by the survivable control plane that is built with OF-Cs and OF agents (OF-AGs). Here, each OF-AG attaches to an optical switch locally and controls its operation for lightpath configuration, while each OF-C is placed in the equipment site of a node. Hence, in the SD-EON, a node includes at least an optical switch and an OF-AG, while it may also contain an OF-C. For example, in Fig. 1, *Node* 1 only includes an optical switch and an OF-AG, while OF-C-1 is placed in *Node* 4 (i.e., OF-C-1 is connected to the OF-AG in *Node* 4 with a solid line) to make it contain an optical switch, an OF-AG, and an OF-C.

Based on the aforementioned network model, we propose a novel mutual backup model to improve the control plane resiliency with high protection efficiency. Specifically, the mutual backup model makes OF-Cs serve as the backups of one another, and for different switches, an OF-C's role can be either working (i.e., a master OF-C) or backup

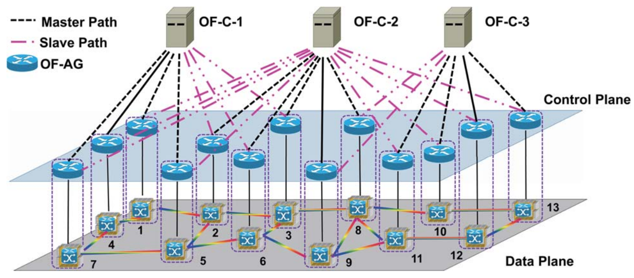


Fig. 1. Network architecture of an SD-EON with SCPE.

(i.e., a slave OF-C). We consider protecting the control plane against single node failures, and hence, for each optical switch, we allocate two OF-Cs to manage it; i.e., one is the master OF-C in the working state (i.e., connected with a black dashed line in Fig. 1) and the other is the slave OF-C for backup (i.e., connected with a pink dashed-dotted line in Fig. 1). For instance, the switch in *Node* 1 has OF-C-1 as its master OF-C, while its slave OF-C is OF-C-2. Meanwhile, for a particular OF-C, its role can be either master or slave, depending on which switch we talk about. For example, OF-C-1 is the master OF-C for the switches in *Nodes* 1, 4, 5, and 7, while for those in *Nodes* 2, 3, and 6, it becomes their slave OF-C.

During network operation, each master OF-C processes the OF messages from its OF-AGs, calculates the service provisioning schemes for lightpath requests, and instructs the OF-AGs to manage their optical switches accordingly for lightpath configuration. For the situation in which a lightpath needs to traverse the optical switches that are managed by multiple master OF-Cs, e.g., a lightpath from *Node* 1 to *Node* 12 in Fig. 1, the master OF-Cs work collaboratively to set it up, with the similar provisioning scheme demonstrated in [19] for multi-domain SD-EONs. A slave OF-C will remain silent if its master OF-C is working. But when there is a node failure that brings the master OF-C down, it detects the failure and takes over the NC&M tasks quickly to avoid service disruption.

More specifically, during a single node failure, the recovery mechanism is designed to work as follows. First of all, we make sure that there are keep-alive message exchanges between the OF-Cs and their switches and between each pair of master and slave OF-Cs, with which the OF-Cs can recognize and locate the failure. Then, based on the network model, we find that the failure can affect the control plane in three scenarios: 1) the failure brings down a master OF-C, 2) the failure brings down the control channel between a pair of master and slave OF-Cs, and 3) the failure brings down the control channel between a master OF-C and its switch.[3]

For the first scenario, the slave OF-C(s) of the failed master OF-C will detect the failure and take over its NC&M tasks. For the second scenario, we assign two node-disjoint paths between each pair of master and slave

[2]Note that, in addition to the propagation delay, the round-trip delay may also consist of other latencies, which are fixed and cannot be optimized with network planning.

[3]Note that, as the control channels may share nodes and fiber links, the failure can affect the control plane with mixed scenarios, but each of them can still be addressed independently.

OF-Cs, and when the failure happens, the OF-Cs will detect it and switch their control channel to use the backup path. For the third scenario, we ensure that the path between a switch and its master OF-C and the one between it and its slave OF-C are node-disjoint. Hence, when the master OF-C detects the failure, it will inform its slave OF-C(s) about it and the slave OF-C(s) will then inform the switches whose control channels are impacted to use them as the master OF-C(s).

With this recovery mechanism, we can realize a survivable SD-EON in which the control services can be restored during a node failure. Hence, the SCPE problem becomes how to place the OF-Cs and plan the control channels' routing paths such that both the number of the deployed OF-Cs and the average path length of the control channels are minimized, and during a single node failure, the control plane operation is intact.

IV. SD-EON Control Plane Planning With SCPE

A. ILP Formulation

We first formulate the following ILP model to solve the SCPE problem exactly for SD-EON control plane planning.

Parameters:

- $G(V,E)$: physical topology of the SD-EON, where V and E represent the sets of nodes and fiber links in G.
- $l_{(u,v)}$: length of the fiber link $(u,v) \in E$, where $u,v \in V$.
- L_m: maximum transmission distance between a master OF-C and its optical switches.
- L_s: maximum transmission distance between a slave OF-C and its switches, and we normally have $L_s \geq L_m$.
- L_c: maximum transmission distance between a pair of master and slave OF-Cs.

Variables:

- c_v: Boolean variable that equals 1 if an OF-C is placed in node v, and 0 otherwise.
- $w_{u,v}$: Boolean variable that equals 1 if the OF-C placed in node u is the master OF-C of the switch in node v, and 0 otherwise.
- $b_{u,v}$: Boolean variable that equals 1 if the OF-C placed in node u is the slave OF-C of the switch in node v, and 0 otherwise.
- $x^{s,d}_{(u,v)}$: Boolean variable that equals 1 if we decide to use link $(u,v) \in E$ to set up the control channel from node s to node d $(s,d \in V)$, and 0 otherwise.

Objective:

To ensure the QoS of the control services in the SD-EON and improve its cost-effectiveness and scalability, we should minimize both the number of deployed OF-Cs and the average path length of the control channels, which are among the OF-Cs and between the OF-Cs and their switches. Basically, since during dynamic network operation, the OF-Cs need to communicate with their switches constantly, reducing the average path length helps to shorten the round-trip delays and thus can expedite the service provisioning. Hence, the optimization objective is

$$\text{Minimize}\left(\sum_{v\in V} c_v \cdot \sum_{(u,v)\in E} l_{(u,v)}\right) + \left(\frac{1}{2\cdot|V|}\sum_{s,d\in V}\sum_{(u,v)\in E} l_{(u,v)}\cdot x^{s,d}_{(u,v)}\right). \quad (1)$$

The optimization objective in Eq. (1) contains two terms. In the first term, $\sum_{v\in V} c_v$ is the number of deployed OF-Cs, and we multiply it by the total link length in the topology (i.e., $\sum_{(u,v)\in E} l_{(u,v)}$) to make its value much larger than that of the second term. Hence, minimizing the number of deployed OF-Cs becomes the major objective. The second term in Eq. (1) is for the average path length of the control channels.[4]

Constraints:

$$\sum_{u\in V} w_{u,v} = 1, \qquad \sum_{u\in V} b_{u,v} = 1, \quad \forall\, v \in V. \quad (2)$$

Equation (2) ensures that each switch in the SD-EON has one and only one master OF-C, and the same condition applies to its slave OF-C.

$$w_{u,v} + b_{u,v} \leq c_u, \qquad \forall\, u,v \in V. \quad (3)$$

Equation (3) ensures that the placement of a master or slave OF-C is valid, i.e., in a node that has a local OF-C:

$$c_s + c_d + w_{s,d} + b_{s,d} \leq 2\sum_{v\in V} x^{s,d}_{(s,v)} + 1, \qquad \{s,d \in V{:}s \neq d\}, \quad (4)$$

$$c_s + c_d + w_{s,d} + b_{s,d} \leq 2\sum_{v\in V} x^{s,d}_{(v,d)} + 1, \qquad \{s,d \in V{:}s \neq d\}, \quad (5)$$

$$\sum_{v\in V} x^{s,d}_{(v,s)} = \sum_{u\in V} x^{s,d}_{(d,u)} = 0, \qquad \{s,d \in V{:}s \neq d\}, \quad (6)$$

$$\sum_{v\in V} x^{s,d}_{(u,v)} = \sum_{v\in V} x^{s,d}_{(v,u)}, \qquad \{u,s,d \in V{:}s \neq d, u \neq s,d\}, \quad (7)$$

$$x^{s,d}_{(u,v)} + x^{s,d}_{(v,u)} \leq 1, \qquad \{u,v,s,d \in V{:}s \neq d, u \neq v\}, \quad (8)$$

[4]Note that the control channels include the communications between all the master–slave OF-C pairs and between OF-Cs and their switches. Therefore, the total number of control channels is upper-bounded by $\frac{1}{2}\sum_{v\in V} c_v \cdot [(\sum_{v\in V} c_v) - 1] + 2|V|$, which can be approximated as $2|V|$ as we usually have $\sum_{v\in V} c_v \ll |V|$.

$$\sum_{v\in V}(x^{s,d}_{(u,v)} + x^{d,s}_{(u,v)}) \leq 1, \qquad \{u,s,d \in V: s \neq d\}, \tag{9}$$

$$\sum_{v\in V}(x^{s,d}_{(u,v)} + x^{z,d}_{(u,v)}) \leq 3 - w_{s,d} - b_{s,d} - w_{z,d} - b_{z,d}, \\ \{u,z,s,d \in V: s \neq d, z \neq s\}. \tag{10}$$

Equations (4)–(10) are the link-based flow constraints to determine the control channels' routing paths. To provide resiliency against single node failures, we ensure that the path between a switch and its master OF-C and the one between it and its slave OF-C are node-disjoint and also assign two node-disjoint paths between each pair of master and slave OF-Cs:

$$\sum_{u\in V}\sum_{(u,v)\in E} x^{s,d}_{(u,v)} \cdot l_{(u,v)} \leq w_{s,d} \cdot L_m + b_{s,d} \cdot L_s + c_d \cdot L_c, \\ \{s,d \in V: s \neq d\}, \tag{11}$$

$$\sum_{u\in V}\sum_{(u,v)\in E} x^{s,d}_{(u,v)} \cdot l_{(u,v)} \leq (3 - w_{s,d} - c_d) \cdot \min(L_m, L_c) \\ + b_{s,d} \cdot L_s, \{s,d \in V: s \neq d\}, \tag{12}$$

$$\sum_{u\in V}\sum_{(u,v)\in E} x^{s,d}_{(u,v)} \cdot l_{(u,v)} \leq (3 - b_{s,d} - c_d) \cdot \min(L_s, L_c) \\ + w_{s,d} \cdot L_m, \{s,d \in V: s \neq d\}. \tag{13}$$

Equations (11)–(13) ensure that the path of each control channel satisfies the distance constraints. The distance constraints are included because we want to ensure that the round-trip delay for exchanging control messages is tolerable and reasonably good QoS can be achieved for the control services.

B. Heuristic Algorithm

In order to reduce the computational complexity, we propose a heuristic to solve the SCPE problem with relevancy sets. Basically, the SCPE problem can be solved in three steps: 1) determining the number and locations of OF-Cs, 2) assigning master and slave OF-Cs to each switch in the SD-EON, and 3) planning the control channels between each pair of master and slave OF-Cs and between OF-Cs and their switches.

First of all, it is easy to verify that there are at least two OF-Cs in the network, while the actual number and locations of the OF-Cs strongly depend on the distance constraints. Hence, for each node pair in $G(V,E)$, we calculate K shortest paths. Then, we put all the paths whose length is not longer than L_s in the slave path set P^s, and select those whose length is not longer than L_m to be included in the master path set P^m.

Definition. In a path set, if we can find at least one path between s and d, we say that s and d are relevant based on the path set. We define the master and slave relevancy sets of $v \in V$ [i.e., $R^m(v)$ and $R^s(v)$] to include all the relevant nodes of v based on the master path set P^m and slave path set P^s, respectively. Note that the relevancy sets of v, $R^m(v)$, and $R^s(v)$ at least include v itself.

Algorithm 1 shows the detailed procedure for the preprocessing to obtain the path sets and relevancy sets (i.e., in *Lines* 1–11). Moreover, Algorithm 1 also finds the lower bound on the number of required OF-Cs. We can easily verify that if there are at least two different nodes u and v in V such that $|R^s(u)| = |R^s(v)| = |V|$, i.e., their slave relevancy sets cover all the nodes in the topology, the lower bound on the number of required OF-Cs is $n_c = 2$; otherwise $n_c = 3$. The procedure for determining n_c is in *Lines* 12–15. The time complexity of Algorithm 1 can be analyzed as follows. In a connected graph, we have $|V|$ as the number of nodes in it, and the total number of node pairs is $|V| \cdot (|V| - 1)$, which is upper-bounded by $|V|^2$. In *Line* 2, the complexity of calculating K shortest paths for an arbitrary node pair is $O(K \cdot |V|)$. Then, the complexity of the first for-loop is $O(K \cdot |V|^3)$, while the second for-loop that determines the value of n_c has a complexity of $O(|V|)$. Therefore, the time complexity of Algorithm 1 is $O(K \cdot |V|^3)$.

Algorithm 1 Preprocessing

1 **for** each node pair s–d in V **for**
2 calculate K shortest paths from s to d;
3 **if** at least one path has a length $\leq L_s$ **then**
4 insert the path(s) into P^s;
5 $R^s(d) \leftarrow s$, $R^s(s) \leftarrow d$;
6 **end**
7 **if** at least one path has a length $\leq L_m$ **then**
8 insert the path(s) into P^m;
9 $R^m(d) \leftarrow s$, $R^m(s) \leftarrow d$;
10 **end**
11 **end**
12 $n_c = 3$;
13 **if** there are at least two different nodes u and v such that $|R^s(u)| = |R^s(v)| = |V|$ **then**
14 $n_c = 2$;
15 **end**

Algorithm 2 illustrates the proposed heuristic for solving the SCPE problem with relevancy sets. Basically, it tries to solve the SCPE problem in a greedy manner. The while-loop covering *Lines* 1–38 increases the value of n_c by 1 each time and checks whether n_c OF-Cs can cover all the nodes in V while satisfying all the design constraints. As shown in *Lines* 2–3, if we can find at least one group of n_c nodes whose master relevancy sets' union covers all the nodes in V, we check each of such feasible node groups with *Lines* 4–34 to solve the SCPE problem with n_c OF-Cs. Here, *Lines* 4–10 place OF-Cs in the selected nodes and set up the master control channels (i.e., the communications between the switches in the nodes and their master OF-Cs). Note that if the switch in *Node* v can select more than one OF-C as its master OF-C, i.e., v is included in multiple OF-Cs' master relevancy sets, the switch will take the OF-C that has the shortest path to it as the master

OF-C. We call this master OF-C selection scheme the **shortest-fit scheme**, as shown in *Line* 7. In *Line* 9, if an OF-C in *Node* u is selected as the master OF-C of the switch in *Node* v, we remove v from the slave relevancy set of u [i.e., $R^s(u)$] to ensure that a switch cannot use the same OF-C as its master and slave OF-Cs.

After setting up the master control channels, *Line* 12 checks whether the selected OF-Cs' slave relevancy sets can still cover V. If yes, we use *Lines* 13–17 to determine the slave OF-C of each node in V and set up the corresponding slave control channels. In this procedure, if there is a failure, we mark $flag = 0$ to record it, as shown in *Lines* 16 and 19. Then, the for-loop that covers *Lines* 21–27 tries to establish two node-disjoint control channels between each pair of master and slave OF-Cs to enable the mutual backup scenario. Finally, if all of the aforementioned steps can be finished successfully, *Lines* 32–33 record the SCPE solution and return it; otherwise, *Lines* 29–30 make the algorithm continue to search.

For Algorithm 2, its time complexity can be analyzed as follows. We can see that the while-loop in Algorithm 2 would be executed for $|V| - 2$ times at most. In each iteration, the search for candidate controller union (i.e., *Lines* 3–35) would run $\binom{|V|}{n_c}$ times at most. The for-loop covering *Lines* 6–10 would run $|V|$ times. The complexity of *Lines* 12–20 is $O(K \cdot |V|)$, and the for-loop that covers *Lines* 21–27 can iterate for $\binom{n_c}{2}$ times. Hence, the overall time complexity of Algorithm 2 is $O(|V|^2 \cdot 2^{|\widehat{n_c}|})$, where we define $\widehat{n_c}$ as the upper bound of n_c. This suggests that Algorithm 2 is a pseudo-polynomial algorithm since the maximum value of $\widehat{n_c}$ is $|V|$, even though we normally have $\widehat{n_c} \ll |V|$.

Algorithm 2 Solve SCPE With Relevancy Sets

input: $G(V,E)$, P^m, P^s, $\{R^m(v), \forall v \in V\}$, $\{R^s(v), \forall v \in V\}$, n_c.
output: SCPE solution

1 **while** $n_c \le |V|$ **do**
2   **if** we can find n_c nodes such that their master relevancy sets' union covers V **then**
3   **for** each feasible group of n_c nodes **do**
4   select these nodes as OF-C locations;
5   backup states of all the slave relevancy sets;
6   **for** each node v in V **do**
7   select u as its master OF-C with the shortest-fit scheme;
8   get master control channel of v as the shortest path from u to v;
9   remove v from $R^s(u)$;
10   **end**
11       $flag = 1$;
12   **if** the union of the n_c OF-Cs' slave relevancy sets still covers V **then**
13   select slave OF-C for each node in V;
14   get slave control channel of each node in V and ensure that it is node-disjoint with the node's master control channel;
15   **if** one or more slave control channels cannot be found **then**
16           $flag = 0$;
17   **end**
18   **else**
19         $flag = 0$;
20   **end**
21   **for** each pair of master and slave OF-Cs in the n_c nodes **do**
22   set up two node-disjoint control channels between them;
23   **if** the control channel(s) cannot be found **then**
24           $flag = 0$;
25   **break**;
26   **end**
27   **end**
28   **if** $flag = 0$ **then**
29   restore all the slave relevancy sets to their original states;
30   **continue**;
31   **else**
32   store the SCPE solution;
33   **return**;
34   **end**
35   **end**
36   **end**
37   $n_c = n_c + 1$;
38 **end**

C. Performance Comparison

We use the three topologies in Fig. 2 to evaluate the performance of the ILP and heuristic. For the simulations with the six-node and NSFNET topologies, we set $L_m = 1500$ km, $L_s = 5000$ km, and $L_c = 5000$ km, while the one with the US-Backbone topology uses $L_m = 900$ km, $L_s = 2000$ km, and $L_c = 3800$ km. For each topology, we use both the ILP and heuristic to solve the SCPE problem and compare their solutions in terms of the number of deployed OF-Cs, the average control latency, and the

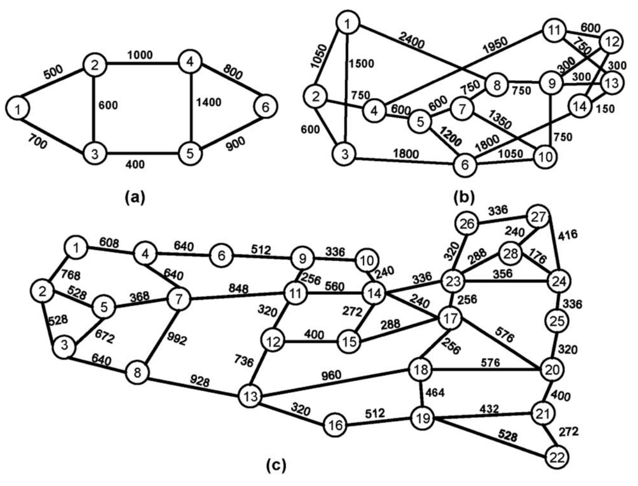


Fig. 2. Network topologies in simulations (link lengths in kilometers): (a) six-node topology, (b) NSFNET topology, and (c) US-Backbone topology.

TABLE I
PERFORMANCE COMPARISON BETWEEN ILP AND HEURISTIC ON SCPE

		# of OF-Cs	Average Path Length (km)	Running Time (s)
Six-Node	ILP	2	1216.67	1
	Heuristic	2	1341.67	0.61
NSFNET	ILP	3	1687.50	2354
	Heuristic	3	1901.61	0.88
US-Backbone	ILP	–	–	–
	Heuristic	5	1265.12	1024.11

computation time. We use Lingo v11.0 [32] to solve the ILP and simulate the heuristic with MATLAB R2011b, and all the simulations run on a computer with 3.10 GHz Intel Core i3-2100 CPU and 4.00 GB RAM. Table I summarizes the simulation results. It can be seen that the heuristic deploys the same number of controllers for SCPE as the ILP model, while the average path length provided by it is slightly longer than that from the ILP. The results on the running time verify that the heuristic has significantly lower time complexity than the ILP. Due to its high complexity, the ILP cannot obtain an optimal solution for the US-Backbone topology.

In order to compare the performance of the ILP and heuristic further, we show the whole SCPE solutions from them for the NSFNET topology in Tables II and III, respectively. Here, “M-N Path” refers to the control channel from a master OF-C to its switch, “S-N Path” is for the control channel from a slave OF-C to its switch, and “M-S Path” is for the control channel between a pair of master and slave OF-Cs.

TABLE II
ILP'S SOLUTION TO SCPE FOR NSFNET TOPOLOGY

	OF-C-1	OF-C-2	OF-C-3
Location	6	2	9
Master OF-C for	5,6,10	1,2,3,4	7,8,9,11,12,13,14
Slave OF-C for	3,4,7,8,11,12,13,14	5,9	1,2,6,10
M-N Paths	6-5; 6-10	2-1; 2-3; 2-4	9-8-7; 9-8; 9-12-11; 9-12; 9-13; 9-13-14
S-N Paths	6-3; 6-5-4; 6-5-7; 6-5-7-8; 6-14-13-11; 6-14-12; 6-14-13; 6-14	2-4-5; 2-4-5-7-8-9	9-8-1; 9-10-6-3-2; 9-13-14-6; 9-10
M-S Paths	6-3-2; 6-10-9	2-4-5-6; 2-4-5-7-8-9	9-13-14-6; 9-10-6-3-2

TABLE III
HEURISTIC'S SOLUTION TO SCPE FOR NSFNET TOPOLOGY

	OF-C-1	OF-C-2	OF-C-3
Location	5	1	9
Master OF-C for	4,5,6,7	1,2,3	8,9,10,11,12,13,14
Slave OF-C for	1,2,3,8,9,10,11,12,13,14	4	5,6,7
M-N Paths	5-4; 5-6; 5-7	1-2; 1-3	9-8; 9-10; 9-12-11; 9-12; 9-13; 9-12-14
S-N Paths	5-4-2-1; 5-4-2; 5-4-2-3; 5-7-8; 5-7-8-9; 5-6-10; 5-4-11; 5-6-14-12; 5-6-14-13; 5-6-14	1-2-4	9-8-7; 9-10-6; 9-8-7-5;
M-S Paths	5-4-2-1; 5-7-8-9	1-3-6-5; 1-8-9	9-12-11-4-2-1; 9-10-6-5

V. SYSTEM DESIGN FOR LIVE CONTROL SERVICE BACKUP AND MIGRATION IN SD-EONS

We utilize the SCPE solution from the ILP to build a control plane testbed for experimental demonstration. The NSFNET topology in Fig. 3 is used, and we place three OF-Cs in it. For each optical switch, its master and slave OF-Cs are assigned according to the ILP's solution in Table II, and the control channels are also set up accordingly. The OF-Cs are programmed based on the POX platform [33].

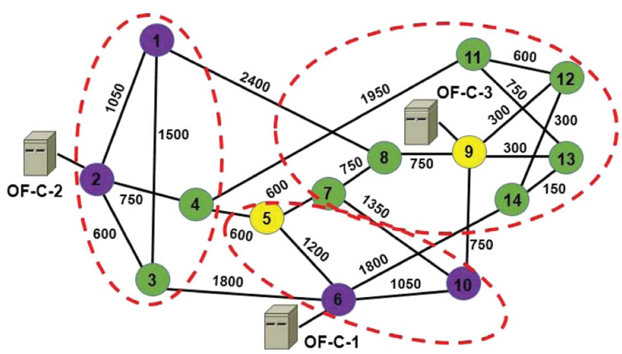


Fig. 3. Experimental testbed with NSFNET topology (emulated data plane fiber lengths in kilometers).

Note that POX only complies with OF v1.0, which does not support the extensions for identifying flexible-grid optical flows, and hence we implement the extensions by ourselves. Meanwhile, we notice that the latest OF v1.5 [6] has already standardized the extensions mentioned above and also included a few other modifications related to OF v1.0. Hence, our implementation with POX does not fully comply with OF v1.5 and can only be viewed as a partial implementation of the standardized OF-C for SD-EONs. Currently, our POX implementation is good for proof-of-concept demonstrations, but in the future, we should consider other platforms (e.g., OpenDaylight [26]) for better OF support.

In each node in the testbed, the OF-AG is realized by programming Open-vSwitch and running it on an independent Linux server [19], while the optical switch is software-emulated due to the budget constraint. To realize live control service backup and migration efficiently, we separate the traffic engineering database (TED) module from the POX platform for OF-C, and realize it with a MySQL database. Specifically, each OF-C still has a TED, but it operates independently to record the control service information, i.e., flow-entries for lightpaths, topology abstraction, etc. Then, during the dynamic network operation, a master OF-C synchronizes the control service information to its slave OF-C with the background database replication provided by MySQL. Hence, the status synchronization is automatic and becomes independent of other operations in the OF-Cs.

Figure 4 shows the functional design of the OF-C for SCPE. The controller communication module (CCM) deals with the communication between OF-Cs using the controller communication protocol (CCP) developed in [19,31]. The purpose of the CCP is twofold. First, it realizes the keep-alive signaling between master and slave OF-Cs. Second, for a lightpath that needs to traverse the optical switches managed by multiple master OF-Cs, the CCP helps the master OF-Cs to work collaboratively to set up the lightpath. The controller-to-database module (CDM) works as the interface between an OF-C and its external TED. The rest of the modules in Fig. 4 work similarly to those designed in [19,31].

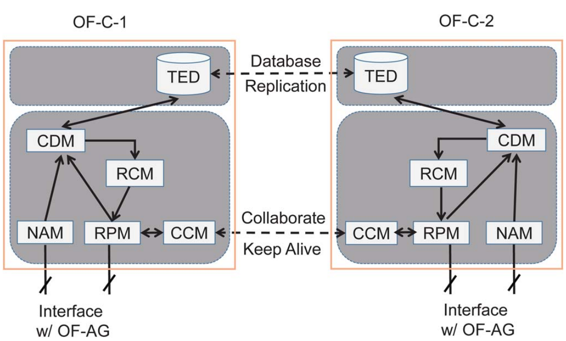


Fig. 4. Functional design of OF-Cs. RCM, resource computation module; RPM, resource provision module; TED, traffic engineering database; NAM, network abstraction module; CCM, controller communication module; CDM, controller-to-database module.

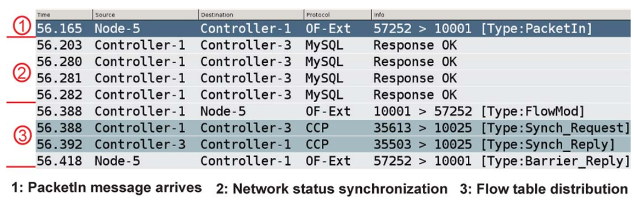
VI. Experimental Demonstrations

A. Live Control Service Backup

Figures 5 and 6 show the experimental results on cooperative service provisioning with control service backup. Here, the lightpath request is from *Node* 5 to *Node* 9. Since the lightpath will traverse the optical switches that are controlled by multiple OF-Cs, they have to work collaboratively to provision it. In Fig. 5, the first line is for the *PacketIn* message from *Node* 5 to report the request. Upon receiving the *PacketIn*, OF-C-1 (i.e., the master OF-C of the OF-AG on *Node* 5) calculates the provisioning scheme, distributes a partial provisioning scheme to the network elements that it controls with *FlowMod* messages, and sends *Synch_Request* messages to OF-C-3 to ask for their cooperation for setting up the rest of the lightpath. Note that, even though the OF-Cs work distributedly, they still reside in a single domain and hence each OF-C has the global view of the network. Therefore, OF-C-1 can obtain the end-to-end path for the request (i.e., 5-7-8-9 as shown in Fig. 6) and include the information in *Synch_Request* messages. When OF-C-3 has set up the rest of the lightpath, it replies with a *Synch_Reply* message to OF-C-1 and the lightpath is provisioned. Figure 5 also shows the MySQL messages used for background database replication, which verify that the mutual backup functions well.

B. Live Control Service Migration

We then perform an experiment to demonstrate the control service migration when a master OF-C is down due to network failure. Figure 7 shows the messages captured on



	Time	Source	Destination	Protocol	Info
①	56.165	Node-5	Controller-1	OF-Ext	57252 > 10001 [Type:PacketIn]
	56.203	Controller-1	Controller-3	MySQL	Response OK
	56.280	Controller-1	Controller-3	MySQL	Response OK
②	56.281	Controller-1	Controller-3	MySQL	Response OK
	56.282	Controller-1	Controller-3	MySQL	Response OK
	56.388	Controller-1	Node-5	OF-Ext	10001 > 57252 [Type:FlowMod]
③	56.388	Controller-1	Controller-3	CCP	35613 > 10025 [Type:Synch_Request]
	56.392	Controller-3	Controller-1	CCP	35503 > 10025 [Type:Synch_Reply]
	56.418	Node-5	Controller-1	OF-Ext	57252 > 10001 [Type:Barrier_Reply]

1: PacketIn message arrives 2: Network status synchronization 3: Flow table distribution

Fig. 5. Messages captured on OF-C-1 for cooperative provisioning.

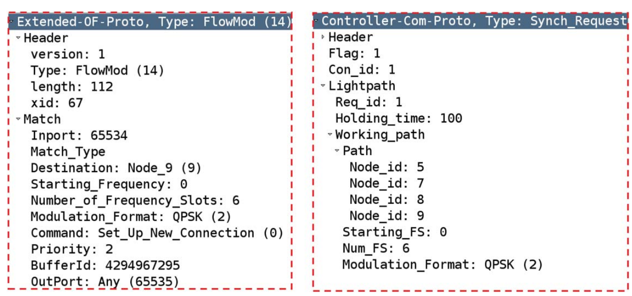


Fig. 6. Wireshark captures for the FlowMod and *Synch_Request* messages from OF-C-1.

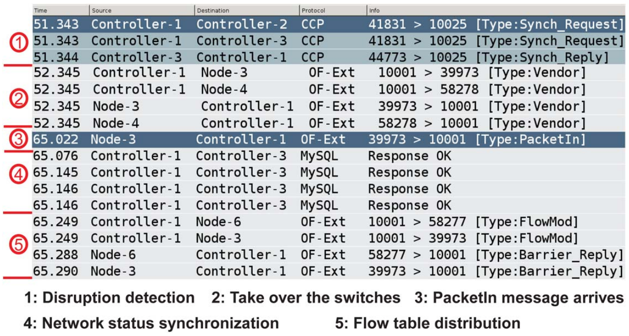


Fig. 7. Messages captured on OF-C-1 for control service migration.

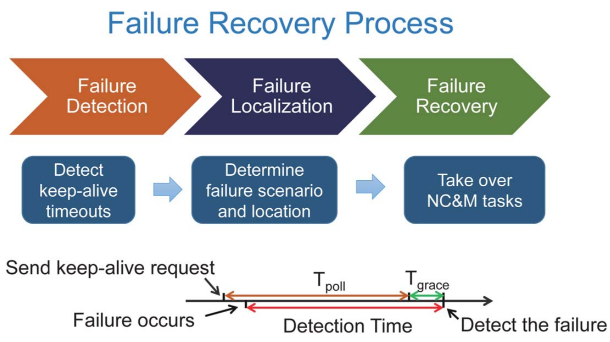


Fig. 8. Procedure of failure recovery.

OF-C-1. Here, we assume that OF-C-2 is down. OF-C-1 detects the outage of OF-C-2 when it records a keep-alive timeout because OF-C-2 has not responded with a *Synch_Reply* message in a timely fashion. Then, OF-C-1 changes its role from slave to master for *Nodes* 3 and 4 by sending *Vendor* messages to them. Meanwhile, OF-C-3 conducts similar operations to *Nodes* 1 and 2. When the control service migration is done, OF-C-1 and OF-C-3 cover the whole network and they can work collaboratively to set up lightpaths. In order to verify this, we make *Node* 3 generate a lightpath request destined to *Node* 6. Here, because the master OF-Cs of *Nodes* 3 and 6 are both OF-C-1, OF-C-1 can provision the lightpath just by itself. Note that, later on, when OF-C-2 is repaired and returns online, we do not switch back its role of master OF-C for *Nodes* 1–4. Instead, we treat it as the slave OF-C of these nodes; i.e., OF-C-2 becomes the slave OF-C for *Nodes* 1–5 and 9. We only switch its role back when there is a failure on other OF-Cs or it is the time for network maintenance. The rationale behind this is to minimize the role-switches on OF-Cs during network operation and avoid unnecessary service interruptions.

C. *Failure Recovery Process*

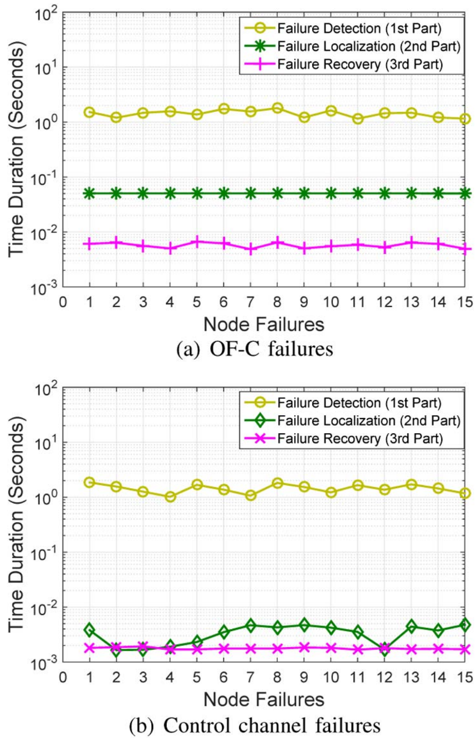
As we have explained in Section III, our system with SCPE relies on the keep-alive polling between the OF-Cs and their switches and between each pair of master and slave OF-Cs to detect and locate the network failures. Specifically, when an OF-C detects a keep-alive timeout or receives a report on keep-alive timeout from its switches, it can identify the failure scenario and invoke the corresponding recovery procedure. When the recovery is finished, the OF-C will distribute the changes to other OF-Cs with status synchronization. Therefore, the recovery latency generally consists of three parts, i.e., time for failure detection, time for locating the failure, and time for recovering the impacted control service(s), as shown in Fig. 8. Basically, the time for failure detection is the period between when the failure happens and when it is detected by an OF-C, and its value depends on the polling interval of keep-alive messages T_{poll} and the grace period for timeout T_{grace}. This is because only after waiting for a keep-alive message for more than $T_{\text{poll}} + T_{\text{grace}}$ does an OF-C claim the detection of a keep-alive timeout.

According to the typical values used in the existing OpenFlow systems [27,34,35], we set $T_{\text{poll}} = 1$ s and $T_{\text{grace}} = 0.05$ s and conduct experiments to measure the three parts that contribute to the recovery latency. Here, we consider two of the three failure scenarios mentioned in Section III, i.e., 1) the failure brings down a master OF-C and 3) the failure brings down the control channel between a master OF-C and its switch. We omit the second failure scenario in which the control channel between a pair of master and slave OF-Cs is brought down. Basically, in terms of failure recovery, the procedure for the second scenario is almost the same as that for the third one, with the only exception that the OF-Cs switch their control channel to the backup path for recovering the impacted control service. Hence, in the recovery latency, the third time part of the recovery latency would be much shorter.

Figure 9 shows the experimental results on the time parts of recovery latency. Basically, we randomly invoke a failure in the testbed and record the time parts during the recovery process. It can be seen that for the two failure scenarios, the time durations used for failure detection are similar since the detection scheme is the same as we have explained above. For failure localization, the time used in OF-C failures is longer because when a slave OF-C detects a keep-alive timeout on its master OF-C, it also needs to check the reports from its switches to determine whether the failure is an OF-C failure or just a failure on the control channel between it and its master OF-C. On the other hand, if the failures are on the control channels between a master OF-C and its switches, the master OF-C can easily detect them and inform its slave OF-Cs about the failure locations. Hence, the time used for failure localization in Fig. 9(b) is shorter. Finally, when an OF-C failure happens, the slave OF-Cs normally need to take over more switches than those in the failure scenario such that only the control channels between master OF-Cs and their switches are impacted. This explains why the third time part in Fig. 9(a) is longer than that in Fig. 9(b).

D. *Dynamic Network Operation*

Finally, we try to stress-test the proposed system with dynamic network operation and compare its performance with an SD-EON that is controlled by a single OF-C. Basically, we assume that the SD-EON is deployed in



(a) OF-C failures

(b) Control channel failures

Fig. 9. Experimental results on time parts of recovery latency.

the C-band and each fiber link can accommodate 358 FSs, each of which has a bandwidth of 12.5 GHz. The dynamic lightpath requests are generated by each OF-AG according to the Poisson traffic model. For the requests, we set the average holding time as 500 s, set the bandwidth requirements as within [25,250] Gb/s, and select their source and destination nodes randomly. Each experiment serves around 2000 dynamic lightpath requests, and the benchmark scheme is the SD-EON that only has a single OF-C located on *Node* 8. We collect the experimental results on average provisioning latency per request, which are shown in Fig. 10. It is interesting to notice that the SD-EON with SCPE can achieve shorter average provisioning latency than the one that uses a single OF-C, especially when the traffic load is relatively high. This is because the multi-controller scheme in SCPE can reduce the processing load on each OF-C by parallelizing the request handling procedure.

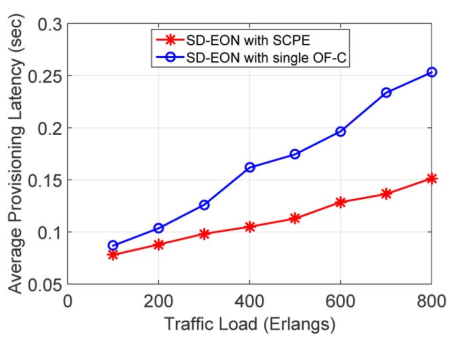


Fig. 10. Average provisioning latency in dynamic network operation.

VII. Conclusion

This paper studied the problem of SCPE in SD-EONs and proposed a novel mutual backup model to improve the survivability of the control plane. We first formulated an ILP model to solve the SCPE problem with the objective to minimize both the number of deployed OF-Cs and the average control channel latency. Then, we proposed a time-efficient heuristic and use simulations to verify that it can obtain similar solutions to the ILP. In addition to the theoretical investigation, we also designed and implemented the system to facilitate live control service backup and migration in an SD-EON with SCPE. We came up with a protocol for the control service backup and migration and ensured that the network status synchronization among OF-Cs is independent of service provisioning. Our experiments demonstrated that the proposed scheme works efficiently for live control service backup and migration. We also tested the recovery operation in the system, and verified that the recovery worked fine as designed and the recovery latency is relatively short. Moreover, compared with the SD-EON using a single OF-C, our scheme achieved much shorter average provisioning latency in dynamic provisioning.

Acknowledgment

This work was supported in part by the NSFC Project 61371117, the Fundamental Research Funds for the Central Universities (WK2100060010), and the Strategic Priority Research Program of the CAS (XDA06011202). It was also supported by the State Key Laboratory of Advanced Optical Communication Systems and Networks in China.

References

[1] P. Lu, L. Zhang, X. Liu, J. Yao, and Z. Zhu, "Highly-efficient data migration and backup for big data applications in elastic optical inter-datacenter networks," *IEEE Netw.*, vol. 29, pp. 36–42, Sept./Oct. 2015.

[2] Z. Zhu, W. Lu, L. Zhang, and N. Ansari, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," *J. Lightwave Technol.*, vol. 31, pp. 15–22, Jan. 2013.

[3] L. Gong, W. Zhao, Y. Wen, and Z. Zhu, "Dynamic transparent virtual network embedding over elastic optical infrastructures," in *Proc. of Int. Conf. on Communications (ICC)*, June 2013, pp. 3466–3470.

[4] 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, July 2013.

[5] 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.

[6] "OpenFlow Switch Specification 1.5.0" [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf.

[7] J. Schonwalder, M. Bjorklund, and P. Shafer, "Network configuration management using NETCONF and YANG," *IEEE Commun. Mag.*, vol. 48, no. 9, pp. 166–173, Sept. 2010.

[8] S. Dasgupta, J. de Oliveira, and J. Vasseur, "Path-computation-element-based architecture for interdomain MPLS/GMPLS traffic engineering: Overview and performance," *IEEE Netw.*, vol. 21, pp. 38–45, July 2007.

[9] 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.

[10] S. Ma, C. Chen, S. Li, M. Zhang, S. Li, Y. Shao, Z. Zhu, L. Liu, and S. Yoo, "Demonstration of online spectrum defragmentation enabled by OpenFlow in software-defined elastic optical networks," in *Optical Fiber Communication Conf. (OFC)*, Mar. 2014, pp. 1–3.

[11] Z. Zhu, C. Chen, S. Ma, L. Liu, X. Feng, and S. Yoo, "Demonstration of cooperative resource allocation in an OpenFlow-controlled multi-domain and multinational SD-EON testbed," *J. Lightwave Technol.*, vol. 33, pp. 1508–1514, Apr. 2015.

[12] S. Li, W. Lu, X. Liu, and Z. Zhu, "Fragmentation-aware service provisioning for advance reservation multicast in SD-EONs," *Opt. Express*, vol. 23, pp. 25804–25813, Oct. 2015.

[13] A. Giorgetti, F. Paolucci, F. Cugini, and P. Castoldi, "Fast restoration in SDN-based flexible optical networks," in *Optical Fiber Communication Conf. (OFC)*, Mar. 2014, pp. 1–3.

[14] 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 European Conf. on Optical Communication*, Sept. 2014, pp. 1–3.

[15] X. Chen, M. Tornatore, S. Zhu, F. Ji, W. Zhou, C. Chen, D. Hu, L. Jiang, and Z. Zhu, "Flexible availability-aware differentiated protection in software-defined elastic optical networks," *J. Lightwave Technol.*, vol. 33, no. 18, pp. 3872–3882, Sept. 2015.

[16] M. Channegowda, R. Nejabati, and D. Simeonidou, "Software-defined optical networks technology and infrastructure: Enabling software-defined optical network operations," *J. Opt. Commun. Netw.*, vol. 5, pp. A274–A282, Oct. 2013.

[17] J. Zhang, Y. Zhao, H. Yang, Y. Ji, H. Li, Y. Lin, G. Li, J. Han, Y. Lee, and T. Ma, "First demonstration of enhanced software defined networking (eSDN) over elastic grid (eGrid) optical networks for data center service migration," in *Optical Fiber Communication Conf. (OFC)*, Mar. 2013, pp. 1–3.

[18] C. Chen, X. Chen, M. Zhang, S. Ma, Y. Shao, L. Li, M. Suleiman, and Z. Zhu, "Demonstrations of efficient online spectrum defragmentation in software-defined elastic optical networks," *J. Lightwave Technol.*, vol. 32, pp. 4701–4711, Dec. 2014.

[19] Z. Zhu, X. Chen, C. Chen, S. Ma, M. Zhang, L. Liu, and S. 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.

[20] W. Lu, S. Ma, C. Chen, C. Chen, and Z. Zhu, "Implementation and demonstration of revenue-driven provisioning for advance reservation requests in OpenFlow-controlled SD-EONs," *IEEE Commun. Lett.*, vol. 18, pp. 1727–1730, Oct. 2014.

[21] A. Sgambelluri, A. Giorgetti, F. Cugini, F. Paolucci, and P. Castoldi, "OpenFlow-based segment protection in Ethernet networks," *J. Opt. Commun. Netw.*, vol. 5, pp. 1066–1075, Sept. 2013.

[22] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, and S. Shenker, "Onix: A distributed control platform for large-scale production networks," in *Proc. of USENIX/OSDI*, Oct. 2010, pp. 1–6.

[23] A. Tootoonchian and Y. Ganjali, "HyperFlow: A distributed control plane for OpenFlow," in *Proc. of INM/WREN*, Apr. 2010, p. 3.

[24] M. Kuzniar, P. Peresini, N. Vasic, M. Canini, and D. Kostic, "Automatic failure recovery for software-defined networks," in *Proc. of HotSDN*, Aug. 2013, pp. 159–160.

[25] "Introducing ONOS—A SDN network operating system for service providers" [Online]. Available: http://onosproject.org/wp-content/uploads/2014/11/Whitepaper-ONOS-final.pdf.

[26] "OpenDaylight" [Online]. Available: https://www.opendaylight.org/.

[27] "Raising the bar on SDN control plane performance, scalability, and high availability" [Online]. Available: http://onosproject.org/wp-content/uploads/2014/11/PerformanceWhitepaperBlackbirdrelease-technical.pdf.

[28] D. Hock, S. Gebert, M. Hartmann, T. Zinner, and P. Tran-Gia, "POCO-framework for Pareto-optimal resilient controller placement in SDN-based core networks," in *Proc. of NOMS*, May 2014, pp. 1–2.

[29] H. Li, P. Li, S. Guo, and A. Nayak, "Byzantine-resilient secure software-defined networks with multiple controllers in cloud," *IEEE Trans. Cloud Comput.*, vol. 2, pp. 436–447, Oct. 2014.

[30] N. Feamster and J. Rexford, "The controller placement problem," in *Proc. of HotSDN*, Aug. 2012, pp. 7–12.

[31] X. Chen, B. Zhao, S. Ma, C. Chen, D. Hu, W. Zhou, and Z. Zhu, "Leveraging master-slave OpenFlow controller arrangement to improve control plane resiliency in SD-EONs," *Opt. Express*, vol. 23, pp. 7550–7558, Jan. 2015.

[32] "Lindo," v 11.0 [Online]. Available: http://www.lindo.com/.

[33] "POX" [Online]. Available: http://www.noxrepo.org/pox/about-pox/.

[34] "H3C S5830V2 OpenFlow Switch" [Online]. Available: http://www.h3c.com.cn/Service/Document_Center/Switches/Catalog/S5830V2/S5830V2/Command/Command_Manual/H3C_S5830V2[S5820V2]_CR-Release2307-6W100/16/201401/813770_30005_0.htm.

[35] "HP 12500 OpenFlow Switch" [Online]. Available: http://h10032.www1.hp.com/ctg/Manual/c04220314.