# Demonstrations of Efficient Online Spectrum Defragmentation in Software-Defined Elastic Optical Networks

Cen Chen, Xiaoliang Chen, Mingyang Zhang, Shoujiang Ma, Yan Shao, Suoheng Li, Munir Said Suleiman, and Zuqing Zhu, *Senior Member, IEEE*

***Abstract*—Elastic optical networks (EONs) facilitate agile spectrum management in the optical layer. When coupling with software-defined networking, they function as software-defined EONs (SD-EONs) and provide service providers more freedom to customize their infrastructure dynamically. In this paper, we investigate how to overcome spectrum fragmentation in SD-EONs with OpenFlow-controlled online spectrum defragmentation (DF), and conduct system implementations to facilitate highly-efficient online DF. We first consider sequential DF, i.e., the scenario that involves a sequence of lightpath reconfigurations to progressively consolidate the spectrum utilization. We modify our previous DF algorithm to make sure that the reconfigurations can be performed in batches and the "make-before-break" scheme can be applied to all of them. The modified algorithm is implemented in an OpenFlow (OF) controller, and we design OF extensions to facilitate synchronous batch reconfiguration. Then, we further simplify the DF operations by designing and implementing parallel DF that can accomplish all the DF-related lightpath reconfigurations simultaneously. All these DF implementations are experimentally demonstrated in an SD-EON control plane testbed that consists of 14 stand-alone OF agents and one OF controller, which are all implemented based on high-performance Linux servers. The experimental results indicate that our OF-controlled online DF implementations perform well and can improve network performance in an efficient way.**

***Index Terms*—Adaptive defragmentation, elastic optical network (EON), openflow, parallel defragmentation, software-defined networking (SDN), synchronous batch reconfiguration (SBR).**

## I. Introduction

WITH agile spectrum management, the elastic optical networks (EONs) improve the spectral efficiency of lightpaths and bring intelligence into the optical layer [1], [2]. Specifically, different from the fixed-grid wavelength-division multiplexing (WDM) systems that operate on wavelength grids at 50 or 100 GHz, bandwidth-variable (BV) transponders and switches in EONs can allocate spectrum with a granularity at 12.5 GHz or less [1]. Then, the EON equipments groom a few narrow-band frequency slots (FS') that are spectrally contiguous to set up lightpaths with various bandwidths.

Meanwhile, in order to fully exploit the aforementioned advantages in the data plane, EONs need a more sophisticated control plane, as it needs to manage blocks of contiguous FS' instead of independent wavelength channels [3], [4]. More importantly, the control plane needs to address the spectrum fragmentation, i.e., the existing of non-aligned, isolated, and small-sized unused FS blocks in the optical spectrum [5]. Spectrum fragmentation leads to low spectrum utilization and increases the blocking probability of future requests [6]. The situation becomes more severe when EONs have to support highly dynamic traffic with frequent lightpath assembling/disassembling (e.g., in an inter-datacenter network [7]). To relieve spectrum fragmentation, people have proposed various defragmentation (DF) algorithms that involve reconfigurations of in-service lightpaths [5], [6], [8]–[13]. As it uses network-wide information to re-optimize the spectrum utilization, DF needs centralized network control and management (NC&M).

Software-defined networking (SDN) makes the network programmable, dynamic and application-aware by decoupling its data and control planes [14]. As a possible implementation of SDN, OpenFlow (OF) [15] was developed as an open standard protocol that leverages flow-based switching and uses a centralized controller to facilitate software-defined routing, switching and network management. Implementing SDN in optical networks via OF provides the service providers more freedom to customize their infrastructure dynamically and adapt to new services quickly [16]. Inline of this, people have proposed a few OF extensions that identify an optical flow based on the input/output ports it occupying on an optical switch, central frequency of its optical carrier, the transmission bandwidth and etc. [17]–[19]. With these extensions, one can introduce OF into the optical domain and facilitate SDN-type lightpath setup and management. More specifically, the extended OF protocol is implemented in the control plane, while the data plane equipments (i.e., optical transponders and switches) are instructed by the OF-based control plane to manage the lightpaths for data transmission. Previously, inter-operation of GMPLS and OF were investigated for optical transport networks in [20], [21]. The experimental demonstrations of OF-controlled wavelength routing in semi-practical network environments were discussed in [22]. More recently, people have considered the coupling of EON and

Manuscript received April 8, 2014; revised September 11, 2014 and October 20, 2014; accepted October 20, 2014. Date of publication October 23, 2014; date of current version November 11, 2014. This work was supported in part by the Program for New Century Excellent Talents in University (NCET) under Project NCET-11-0884, the National Natural Science Foundation of China (NSFC) under Project 61371117, the Fundamental Research Funds for the Central Universities (WK2100060010), and the Strategic Priority Research Program of the Chinese Academy of Sciences (XDA06030902).

The authors are with the School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China (e-mail: chencen@mail.ustc.edu.cn; arabus@mail.ustc.edu.cn; zmy12@mail.ustc.edu.cn; massak@mail.ustc.edu.cn; lzzitc@mail.ustc.edu.cn; shaozi@mail.ustc.edu.cn; mshj@mail.ustc.edu.cn; zqzhu@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JLT.2014.2364515

SDN for software-defined EONs (SD-EONs), and accomplished a few interesting experiments [18], [19],[23]–[25].

The centralized NC&M in SD-EON fits well with the requirement of DF, and researchers have already started to implement DF systems in such networks [26], [27]. In [26], Liu *et al.* have demonstrated OF-controlled fragmentation-aware routing and spectrum assignment (FA-RSA) in SD-EONs. However, it is known that FA-RSA itself cannot eliminate spectrum fragmentation, when DF with lightpath reconfiguration is absent [6]. We also have implemented a DF system that could facilitate lightpath reconfiguration, and conducted proof-of-concept experiments to demonstrate online DF in a control plane testbed [27]. Nevertheless, the implementation in [27] has not considered the system optimization for minimizing the traffic disruptions and the operation complexity during DF. As each DF operation can invoke a few lightpath reconfigurations, it is desired that both the extended OF protocol and the DF algorithms are tailored accordingly to reduce the complexity of network control. Moreover, since the negative effects from spectrum fragmentation can accumulate during network operation, we may need to implement the adaptive DF scheme that triggers DF operations based on network status, for consistent network performance improvement.

In this work, we investigate how to achieve highly-efficient online DF in SD-EONs that utilize OF. Note that there are both proactive and reactive DF schemes discussed in the literature. For instance, previous studies have proposed a reactive scheme in [5], several proactive schemes in [6], and both the reactive and proactive schemes in [10]. Specifically, proactive DF monitors network status proactively, and consolidates the network-wide spectrum utilization when necessary, while the reactive one reconfigures some/all of the in-service lightpaths to accommodate an incoming request, which would be blocked otherwise. Basically, we think that for reactive DF, the service model can be unfair or in favor of certain requests, since it would reconfigure some/all of the in-service lightpaths just for accommodating certain requests [5]. Therefore, if we would like to ensure service fairness in SD-EONs, we should consider proactive DF in the system design and implementation.

We first consider sequential DF, i.e., the scenario that involves a sequence of lightpath reconfigurations to progressively consolidate the spectrum utilization in EONs. We modify our DF algorithm in [6] to ensure that the reconfigurations can be performed in batches and the "make-before-break" scheme can be applied to all of them. The modified DF algorithm is then implemented in a POX-based OF controller (OF-C), and we propose OF extensions to facilitate synchronous batch reconfiguration (SBR) for efficient DF. Then, in order to further simplify the DF operations, we design and implement a parallel DF algorithm that can accomplish all the DF-related reconfigurations in one batch. All these implementations are experimentally demonstrated in an SD-EON control plane testbed that consists of 14 stand-alone OF agents (OF-AGs) and one OF-C, which are all built with high-performance Linux servers. The experimental results show that our OF-controlled online DF performs well and can effectively improve network performance in an efficient way.

The contributions of this work are as follows,

1) We modify our DF algorithms to facilitate efficient online DF and implement them in a semi-practical SD-EON control plane testbed that utilizes OF.
2) We propose to conduct SBR-enabled DF operations and achieve effective reduction on the complexity of DF-related network control.
3) We implement and experimentally demonstrate adaptive DF that can handle time-variant traffic by invoking DF operations based on network status.
4) We implement and experimentally demonstrate parallel DF that can minimize the reconfiguration latency by performing all the lightpath reconfigurations simultaneously.
5) To the best of our knowledge, we accomplish the first experimental demonstrations for control plane operations of adaptive DF and parallel DF in SD-EONs.

The rest of the paper is organized as follows. Section II explains the network architecture of SD-EONs and our design for realizing efficient online DF. The experimental investigation on sequential DF, including the algorithms, system implementation and demonstrations of periodic DF and adaptive DF, is discussed in Section III. We then study parallel DF and show its experimental demonstrations in Sections IV. Finally, Section V summarizes the paper.

## II. DF IN SD-EONS

### A. Network Architecture

Fig. 1 shows the overall network architecture of an SD-EON. The data plane consists of edge routers (ERs) and BV wavelength selective switches (BV-WSS') for setting up lightpaths. The detailed structure of an ER is shown in Fig. 1(a), which includes several BV transponders (BV-Ts) and a wavelength multiplexer/de-multiplexer (MUX/DEMUX). The BV-Ts and BV-WSS' operate on a flexible-grid wavelength plan, e.g., the one defined in the G.694.1 recommendation [28] suggested by ITU-T. BV-Ts set up lightpaths for client traffic by assigning just-enough spectrum resources (right numbers of FS'), while BV-WSS' allow variable-size spectra to be switched correctly. Note that when setting up a lightpath, BV-T also needs to choose the right modulation-level to accommodate its quality of transmission (QoT) [1].

On top of the optical data plane, the SD-EON has an OF-based control plane for centralized and efficient resource management. Two elements are essential for the control plane, the OF-C and the OF-AGs. An OF-AG is locally attached to each ER or BV-WSS for controlling their operations. OF-C communicates with all the OF-AGs in the SD-EON using an extended OF protocol for lightpath management. Thanks to the centralized NC&M provided by OF, OF-C can dynamically collect, analyze and arrange FS usages on all the fiber links in the SD-EON.

During dynamic network operation, the SD-EON realizes spectrum DF by using the push-pull type of service automation. More specifically, OF-C collects and analyzes FS usage on each link proactively (i.e., pulling the network status). When a DF operation is necessary, it invokes lightpath reconfigurations to consolidate the spectrum utilization by instructing all the

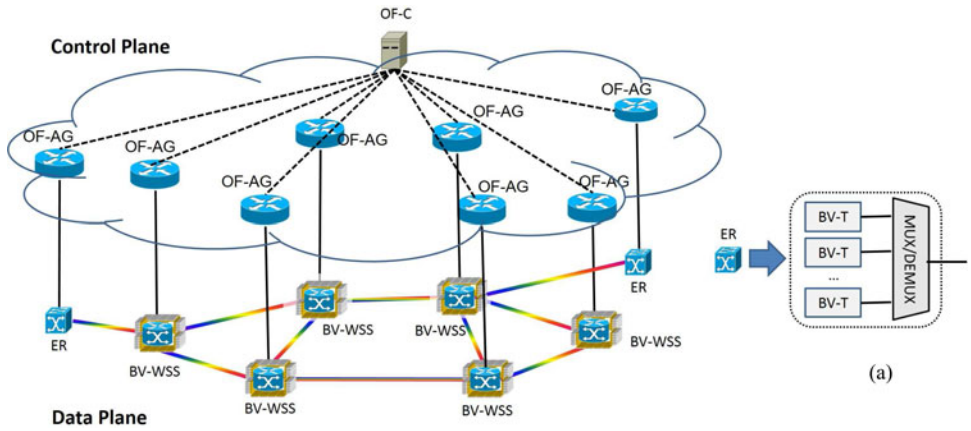


Fig. 1. Network architecture of an SD-EON, OF-C: OpenFlow controller, OF-AG: OpenFlow agent, ER: Edge router, BV-WSS: Bandwidth-variable wavelength-selective switch, BV-T: Bandwidth-variable transponder, MUX/DEMUX: Wavelength multiplexer/de-multiplexer.

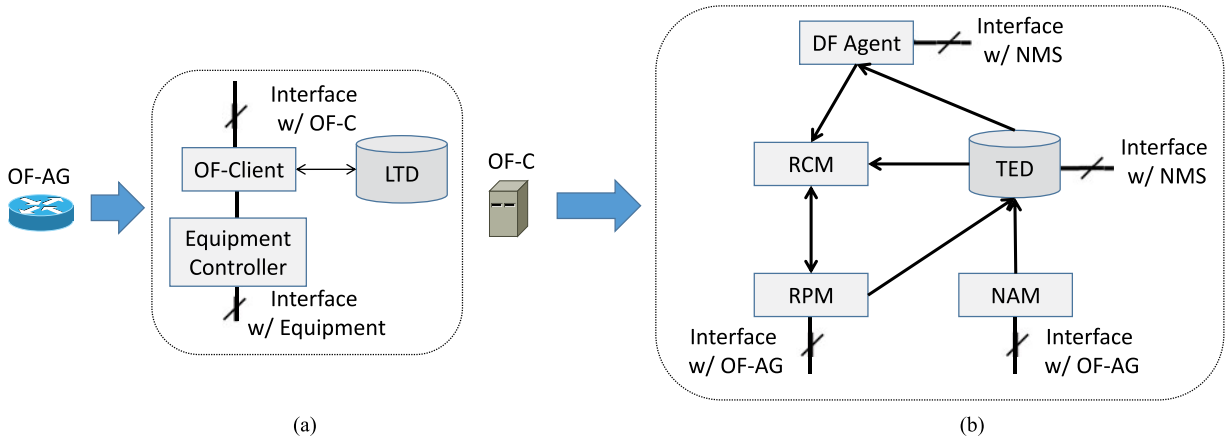


Fig. 2. Functional designs for (a) OpenFlow agent (OF-AG) and (b) OpenFlow controller (OF-C) to facilitate online spectrum DF, LTD: Local traffic database, OF-Client: OpenFlow client, DF Agent: Defragmentation agent, NMS: Network management system, RCM: Resource computation module, TED: Traffic engineering database, RPM: Resource provisioning module, NAM: Network abstraction module.

related OF-AGs to change the working status of their data plane equipments (i.e., pushing the network re-optimization). In order to minimize traffic disruption, the "make-before-break" scheme should be applied to each lightpath reconfiguration. Specifically, we first establish the lightpath on its new routing path and spectrum location (i.e., new RSA), and then disassemble it on the original RSA.

### *B. System Functional Design*

Fig. 2 shows the functional designs of OF-AG and OF-C. OF-AG configures data plane equipments (i.e., BV-Ts and BV-WSS') according to the flow-entries from OF-C. The components in OF-AG are shown in Fig. 2(a), where the OF-client uses an extended OF to talk with OF-C, the local traffic database stores flow-entries, and the equipment controller implements flow-entries in data plane equipment.

OF-C works as the "brain" of the SD-EON, and its architecture is illustrated in Fig. 2(b). The details are as follows.

1) *Resource Provision Module (RPM):* It interacts with OF-AGs and the resource computation module (RCM) for handling OF messages. RPM also instructs the traffic engineering database (TED) to update the records of in-service lightpaths in the SD-EONs.
2) *RCM:* It receives the computation tasks from RPM or DF Agent and calculates the corresponding RSA solutions. Upon receiving the tasks, RCM requests the current network status from TED, and when the tasks are done, it instructs RPM to build flow-entries for the related OF-AGs.
3) *TED:* It stores the current FS usage of each link, and the information of in-service lightpaths. TED is updated by RPM in real time to include the most-updated information of the SD-EON.
4) *DF Agent:* It invokes a DF operation, selects in-service lightpaths to reconfigure, and instructs RCM to calculate the new RSA solutions for them. Note that DF Agent can either invoke a DF operation automatically by monitoring the network status in TED or be instructed to do so by the network operator through the external network management system.
5) *Network Abstraction Module (NAM):* It communicates with OF-AGs, collects the SD-EON's topology

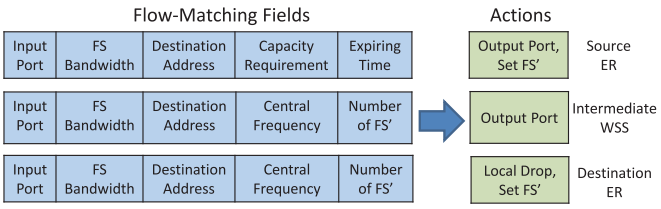


Fig. 3. Flow-matching examples in an SD-EON.

information (i.e., active nodes and the connectivity among them), and abstracts all the data plane equipments for TED.

### C. Extended OF Protocol

According to the working principle of OF, the SD-EON identify each lightpath as an optical flow with the flow-entry that consists of matching fields, actions and related counters [19]. We implement our SD-EON based on OF v1.0 [15] because it is a stable version that is widely supported by various OF systems. We propose two extensions to support DF, which can be easily accommodated in subsequent OF versions.

*1) Flow-Matching Fields:* Fig. 3 shows flow-matching examples, where the important fields are defined as follows.

1) *Frequency_Slot_Bandwidth:* It represents the bandwidth of an FS, i.e., the grid $f_{\rm grid}$, which is set as $f_{\rm grid} = 12.5$ GHz to compliant with ITU-T G.694.1 [28].
2) *Central_Frequency:* It uses an integer to represent the central frequency of a lightpath. We use 193.1 THz as the reference frequency [28] and calculate the central frequency as "$193100 + $ *Central_Frequency* $\times 6.25$" GHz.
3) *Number_of_Frequency_Slots:* It uses a positive integer to express the bandwidth of a lightpath, as "*Number_of_Frequency_Slots* $\times f_{\rm grid}$" GHz.
4) *Modulation_Format:* It tells the modulation-level of a lightpath, and our implementation currently supports BPSK, QPSK, 8-QAM, and 16-QAM, with the *Modulation_Format* values as 1, 2, 3, and 4, respectively.

*2) Flow-Matching Actions:* As shown in Fig. 3, the flow-matching actions are different for the source and destination ERs, and intermediate BV-WSS' of each lightpath.

1) *Source ER:* A lightpath is identified by the flow-entry that includes input port, FS bandwidth, destination address, capacity requirement, and expiring time. The corresponding flow-matching actions are to operate the related BV-T on an FS block with the assigned modulation-level and to send the lightpath's signal out on an output port.
2) *Intermediate WSS':* A lightpath is identified by the combination of input port, FS bandwidth, destination address, central frequency, and number of FS'. The flow-matching action is to switch it to an output port.
3) *Destination ER:* A lightpath is identified by the same combination as that for intermediate WSS', while the flow-matching action is to drop the lightpath locally and to adjust the related BV-T to receive it correctly.

*3) DF-Related Extensions:* We propose two DF-related extensions to facilitate efficient online spectrum DF. We first insert a *Defragmentation_Flag* in related OF messages and use it to indicate whether a message is for assembling a new lightpath (normal, with *Defragmentation_Flag* $= 0$), or reconfiguring an existing lightpath in DF (DF, with *Defragmentation_Flag*$= 1$). As in a DF operation, an OF-AG usually needs to reconfigure multiple in-service lightpaths on its data plane equipment, we leverage the bundle reconfiguration scheme proposed in [29] and design DF-related *Flow-Mod* messages that each includes more than one flow-entries to facilitate SBR. We refer them as SBR-enabled *Flow-Mod* messages. Such a message instructs an OF-AG to reconfigure multiple lightpaths in one shot, for reducing reconfiguration latency and operation complexity. Our DF implementation supports these messages and makes sure that an OF-AG would invoke SBR for multiple lightpaths upon receiving them. Fig. 4 illustrates the Wireshark capture of an SBR-enabled *Flow-Mod* message in our implementation. It can be seen that the message includes two flow-entries, whose *Defragmentation_Flags* are both turned on for DF.

## III. Demonstrations of Sequential DF

In this work, we leverage the proactive DF scheme discussed in [6] for the system design and implementation of DF in the control plane of SD-EON, since it can provide better service fairness than the reactive scheme in [5], which reconfigures some/all of the in-service lightpaths to accommodate a particular incoming request that would be blocked otherwise. Moreover, as the reconfiguration in proactive DF is network-wide and prepares the network better to accommodate more future requests, proactive DF can be more efficient and effective for control plane operations in SD-EONs.

Note that different from those DF system designs and demonstrations that were based on the push-pull-type spectrum retuning technique [5], [12], which only has limited spectrum retuning flexibility, we design our DF algorithms based on the hop-tuning technique proposed in [30], which could support full-spectrum retuning for more flexible DF and only introduces a relatively short reconfiguration latency (e.g., $<1$ $\mu$s in the experimental demonstration in [30]). Therefore, our design can potentially realize more flexible and faster DF operation.

### A. Modified DF Algorithm for SBR

We introduce the following notations to describe the DF algorithm that is implemented in OF-C.

1) $G(V, E)$: The physical topology, where $V$ and $E$ are the sets of switching nodes and fiber links, respectively.
2) $F$: The number of FS' on each link $e \in E$.
3) $C_i(s_i, d_i, n_i)$: The $i$th in-service lightpath, where $i$ is its unique index, $s_i$ and $d_i$ are the source and destination, respectively, and $n_i$ is the number of FS' that it uses.
4) $p_i$: The routing path that $C_i$ is using.
5) $a_i$: The $F$-bit bit-mask to represent the spectrum assignment of $C_i$ along the path $p_i$. If $a_i[j] = 1$, the $j$th FS is allocated to $C_i$, otherwise, $a_i[j] = 0$.

```
▽ Flow_entry_1
  ▽ Match1
      Inport: 1
      Match_Type
      Destination: Node_6 (6)
      Central_Frequency: 102
      Frequency_Slot_Bandwidth: 12.5GHz (0)
      Number_of_Frequency_Slots: 5
      Modulation_Format: QPSK (2)
      Defragmentation_Flag: Defragmentation (1)
      Command: Set_Up_New_Connection (0)
      IdleTimeout: 24
      HardTimeout: 0
      Priority: 32768
      BufferId: 4294967295
      OutPort: Any (65535)
      Flags: 0
  ▽ Action1
      action_type: OUTPUT (0)
      Output_Port: 3
▽ Flow_entry_2
  ▽ Match2
      Inport: 1
      Match_Type
      Destination: Node_7 (7)
      Central_Frequency: 176
      Frequency_Slot_Bandwidth: 12.5GHz (0)
      Number_of_Frequency_Slots: 3
      Modulation_Format: 8-QAM (3)
      Defragmentation_Flag: Defragmentation (1)
      Command: Set_Up_New_Connection (0)
      IdleTimeout: 18
      HardTimeout: 0
      Priority: 32768
      BufferId: 4294967295
      OutPort: Any (65535)
      Flags: 0
  ▽ Action2
      action_type: OUTPUT (0)
      Output_Port: 2
```

Fig. 4. An SBR-enabled *Flow-Mod* message in our implementation.

In a DF operation, if lightpath $C_i$ needs to be reconfigured, we first find an one-to-one RSA mapping,

$$M : \{p_i, a_i\} \mapsto \{p_i', a_i'\} \tag{1}$$

and then migrate it to the new RSA location. RSA mapping and traffic migration are correlated and an important concept regarding them is the reconfiguration dependency through dependency graph analysis [6]. Note that previous studies have also incorporated dependency graph analysis to investigate the lightpath reconfiguration in WDM networks [31], [32]. However, since in EONs, the spectrum assignment is more flexible and the bandwidth allocation granularity is smaller, the dependency graph can be more complicated. Therefore, it is necessary to revisit the problem by considering the uniqueness of EONs, e.g., partial spectrum overlapping, etc. Meanwhile, we also notice that there are existing investigations that used graph dependency

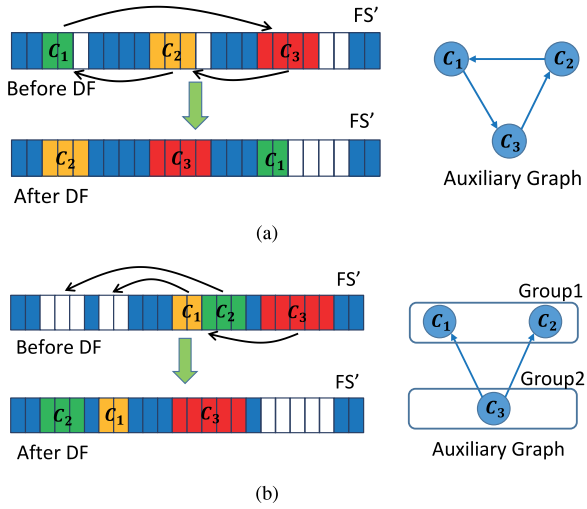


Fig. 5. Sequential DF and auxiliary graphs. (a) Scheme with cyclic dependencies. (b) Scheme without cyclic dependencies.

analysis for DF [5], [13]. Nevertheless, the dependency graphs discussed in them were undirected ones, which were only used to find the target spectrum locations of lightpath reconfiguration. In this work, we build directed dependency graphs in DF, and use them not only for the target spectrum locations of lightpath reconfiguration, but also for determining the non-disrupted traffic migration sequence.

In EONs, for two lightpaths $C_i$ and $C_j (i \neq j)$, if the new RSA of $C_i$ will use the FS' that are currently occupied by $C_j$, i.e., "$p_i' \cap p_j \neq \emptyset$" and "$\sum_{k=1}^{F} a_i'[k] \otimes a_j[k] \neq 0$" are both satisfied, we say that $C_i$ **depends on** $C_j$, denoted as $C_j \Leftarrow C_i$. When there are reconfiguration dependencies, OF-C has to carefully design the sequence for ensuring that "make-before-break" can be applied to all the reconfigurations in the DF operation. For example, if $C_j \Leftarrow C_i$, then $C_i$ cannot be reconfigured before $C_j$. Hence, the DF has to be performed sequentially.

Previously, we have proposed a DF algorithm that performs partial reconfiguration and determines the reconfiguration sequence with the assistance of an auxiliary graph $G^d(V^d, E^d)$ [6]. In $G^d(V^d, E^d)$, each node $v^d \in V^d$ is for a lightpath selected by the DF to reconfigure. If $C_j \Leftarrow C_i$ and $v_j^d$ and $v_i^d$ are their corresponding nodes in $G^d(V^d, E^d)$, there is a directed link $v_i^d \rightarrow v_j^d$ in $G^d(V^d, E^d)$. However, the algorithm in [6] cannot avoid the situation where there are cyclic dependencies in $G^d(V^d, E^d)$. As shown in Fig. 5(a), we have $C_1 \Leftarrow C_2$, $C_2 \Leftarrow C_3$ and $C_3 \Leftarrow C_1$. In order to address the cyclic dependencies, we need to either tolerant prolonged traffic disruptions or implement sophisticated state-machines in OF-C, which can increase reconfiguration latency. For instance, in the implementation in [27], we used a green-field reconfiguration scheme that disassembles all the related lightpaths first and then sets them up on new RSA locations. This scheme is simple in terms of system implementation, but it also causes the longest traffic disruption, i.e., in the worst case a lightpath can be unavailable for the whole DF operation.

In this work, we propose to eliminate cyclic dependencies by applying an "acyclic constraint" on the RSA mapping: the new

spectrum location of a lightpath should have a smaller FS start-index than the original one, i.e., the new spectrum location is in a lower frequency region, such that cycle dependencies will not happen. If a new RSA cannot be found under this constraint, the lightpath will not be reconfigured. Then, we ensure that there is no cyclic dependencies in $G^d(V^d, E^d)$. Meanwhile, OF-C can leverage the SBR-enabled *Flow-Mod* messages to realize lightpath reconfigurations in batches efficiently [as shown in Fig. 5(b)].

*Algorithm 1* shows the detailed procedure of the modified sequential DF algorithm for SBR. Here, in a DF operation, a partial reconfiguration is performed to consolidate the spectrum utilization in the SD-EON, which means that DF only selects a portion of the in-service lightpaths to reconfigure. As *Line* 3 indicates, the selection ratio is $\gamma \in (0, 1)$, which is defined as the ratio of reconfigured to total number of in-service lightpaths in the SD-EON. Then, with $\gamma$, DF Agent in OF-C selects the most "critical" lightpaths using the highest used slot-index first (HUSIF) strategy we developed in [33]. In *Lines* 4-5, RCM calculates new RSAs with the fragmentation-and-misalignment-aware RSA (FMA-RSA) algorithm in [34], and builds the auxiliary graph $G^d(V^d, E^d)$. *Lines* 6-15 show the detailed procedure of reconfiguring the lightpaths with SBR. Since $G^d(V^d, E^d)$ is acyclic, we select a batch of lightpaths that do not depend on others, and reconfigure them simultaneously using the SBR-enabled *Flow-Mod* messages.

**Algorithm 1:** Sequential DF algorithm for SBR

```
1  obtain network status from TED;
2  if a DF operation is needed then
3      select γ portion of in-service lightpaths using the
        HUSIF strategy [33];
4      calculate new RSAs for the selected lightpaths with
        FMA-RSA in [34] under acyclic constraint;
5      build the auxiliary graph G^d(V^d, E^d);
6      while G^d(V^d, E^d) is not empty do
7          run a topological sorting on G^d(V^d, E^d);
8          𝔾 = ∅;
9          for each v^d ∈ V^d that has no outgoing link do
10             find the corresponding lightpath C_i of v^d;
11             𝔾 = 𝔾 ∪ {C_i};
12             remove v^d and its incoming links from
                G^d(V^d, E^d);
13         end
14         invoke SBR for all the lightpaths in 𝔾;
15     end
16 end
```

### B. SD-EON Control Plane Testbed

We implement the aforementioned DF framework in a SD-EON control plane testbed that is built with high-performance servers (ThinkServer RD530), as shown in Fig. 6. We have 14 OF-AGs that each runs on an independent server, and they are connected according to the scheme in Fig. 7 to mimic

Fig. 6. Control plane testbed built with high-performance servers.

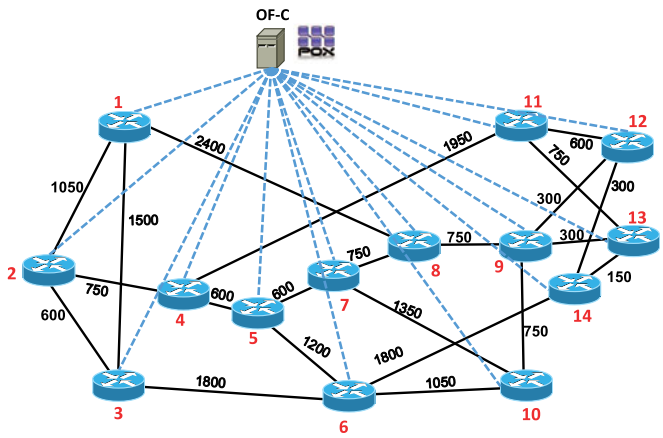


Fig. 7. Experimental setup to mimic the NSFNET topology (link lengths in km).

the NSFNET topology. Each OF-AG is programmed based on Open-vSwitch [35] running on Linux. The OF-C is implemented with the POX platform [36] and runs on another independent server that is directly connected to all the OF-AGs. In this work, we only focus on the control plane implementation for DF, and the data plane is emulated with Ethernet connections. We also assume that OF-C performs QoT-aware RSA and selects the signal' modulation-level adaptively based on the transmission distance of a lightpath [8], [37]. The fiber link lengths are as those marked in Fig. 7.

### C. OF Implementation for DF

In the testbed, OF-C manages two tasks for each lightpath, 1) setting it up with RSA when it initially arrives, and 2) re-configuring it in DF. Since the first task is performed with a similar approach as those in previous work [18], [19], [23],

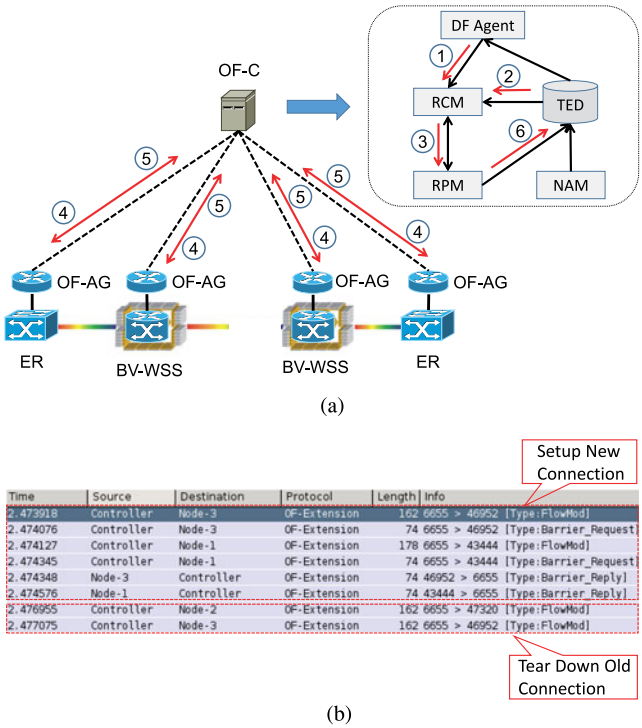


Fig. 8. Sequential DF implementation. (a) Detailed procedure for a DF operation in the SD-EON. (b) Wireshark capture of OF messages for a lightpath reconfiguration in DF.

[24], we omit the description on its operation details here. But we want to clarify that in our implementation, when handling the first task, RCM calculates RSA with the $K$-shortest path and balance-load spectrum assignment algorithm in [38] and adopts the adaptive modulation-level selection in [37]. Specifically, with $B_i$ as the lightpath's bandwidth requirement in Gb/s, the number of FS' we need to assign is

$$n_i = \left\lceil \frac{B_i}{m \cdot B_{\text{grid}}^{\text{BPSK}}} \right\rceil \tag{2}$$

where $B_{\text{grid}}^{\text{BPSK}}$ represents the transmission capacity that an FS with a bandwidth of $f_{\text{grid}}$ can provide with BPSK, and $m$ is the modulation-level, which is 1, 2, 3, and 4, standing for BPSK, QPSK, 8-QAM and 16-QAM, respectively. In the experiments, we set $f_{\text{grid}} = 12.5$ GHz and $B_{\text{grid}}^{\text{BPSK}} = 12.5$ Gb/s. For the topology in Fig. 7, the maximum transmission reaches of BPSK, QPSK, 8-QAM, and 16-QAM signals are 5000, 2500, 1250, and 625 km, respectively, according to the experimental results in [39]. Specifically, when the distance of the routing path permits, RCM always selects the highest modulation-level possible for a lightpath.

For the DF task, Fig. 8(a) shows how the SD-EON works.

1) *Step 1:* DF Agent in OF-C invokes a DF operation, selects certain in-service lightpaths in TED to reconfigure, and instructs RCM to calculate RSAs for them.
2) *Step 2:* RCM requests current network status from TED, and re-optimizes the RSAs of the selected lightpaths with the objective to minimize spectrum fragmentation.
3) *Step 3:* Based on the new RSA solutions, RCM obtains the reconfiguration sequence with *Algorithm 1*, and then instructs RPM to build the flow-entries accordingly.
4) *Step 4:* For each lightpath bathes in sequence, RPM encodes the flow-entries in SBR-enabled *Flow-Mod* messages, turns on the *Defragmentation_Flags* in them, and sends the messages to the related OF-AGs for realizing the lightpath reconfiguration.
5) *Step 5:* Each related OF-AG parses the flow-entry, configures its data plane equipment accordingly with the make-before-break scheme, and returns the configuration result to OF-C using the *Barrier-Reply* message.
6) *Step 6:* In OF-C, RPM updates TED accordingly when the lightpath reconfigurations have been finished.
7) *Step 7:* Repeat Steps 4–6 for all the lightpath batches.

### D. Experimental Demonstrations of Periodic DF

We first perform a simple lightpath reconfiguration experiment to illustrate the make-before-break scheme. Fig. 8(b) shows the Wireshark capture of OF messages for reconfiguring one lightpath in DF. It can be seen that OF-C instructs the related OF-AGs to assemble the lightpath on the new RSA location before tearing it down on the original one. The lightpath is reconfigured from $1 \rightarrow 2 \rightarrow 3$ to $1 \rightarrow 3$.

We then consider the case that DF Agent invokes DF operations periodically, and conduct online DF experiments. Specifically, a new DF operation is invoked when the number of expired lightpaths since the last DF reaches 80. In the experiments, we set $\gamma = 0.3$ and assume that each fiber link in the SD-EON can accommodate $F = 358$ FS'. On each node in the testbed, the OF-AG generates lightpath requests according to the Poisson traffic model and selects their destination address randomly. Specifically, the dynamic requests arrive according to the Poisson process with an average rate of $\lambda$ requests per second, and the life-time of each request follows the negative exponential distribution with an average of $\frac{1}{\mu}$ seconds. Hence, the traffic load can be quantified with $\frac{\lambda}{\mu}$ in Erlangs. The bandwidth requirement of each request (i.e., $B_i$) is uniformly distributed within $[25, 500]$ Gb/s.

Fig. 9 shows the memory dumps of TED, which indicates the spectrum utilization before and after a DF. The $x$- and $y$-axes are for the indices of FS' and links, respectively, and a red slot indicates that the specified FS on a link is occupied. It can be seen clearly that the DF operation consolidates the spectrum utilization in the SD-EON effectively, by reducing the maximum used FS index from 333 to 295. Fig. 10 shows the results from the online sequential DF experiment when the traffic load is 400 Erlangs. The results on OF-C's CPU utilization indicate that the DF operations are invoked periodically as expected. In Fig. 10(a), we also observe that when there is no DF operations, the maximum used FS index reaches 358 (the maximum value) when the SD-EON has only operated for $\sim$85 s, and then never goes down. While for the scenario with DF, Fig. 10(b) shows that the maximum used FS index first reaches 358 after $\sim$100 s, and its value is reduced repeatedly by the DF operations since then.

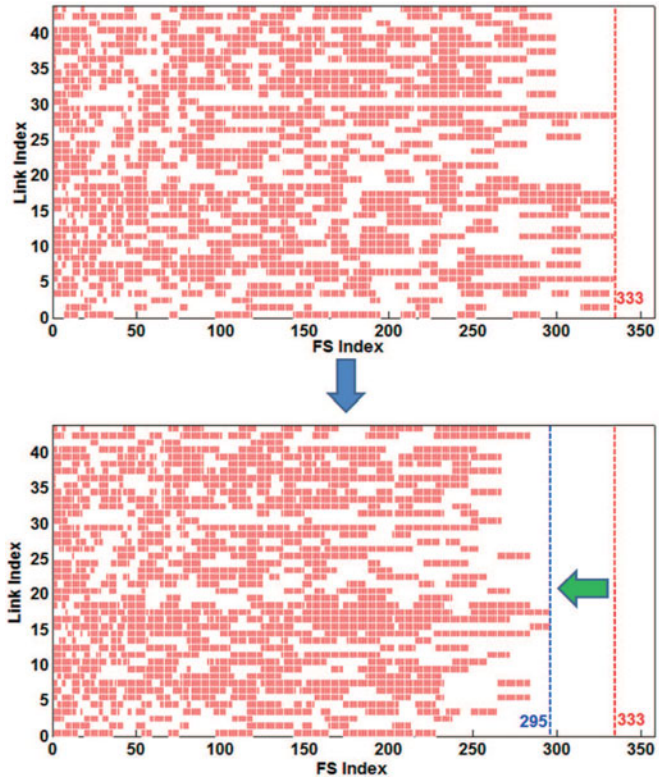


Fig. 9. Network spectrum utilization before and after a DF operation.

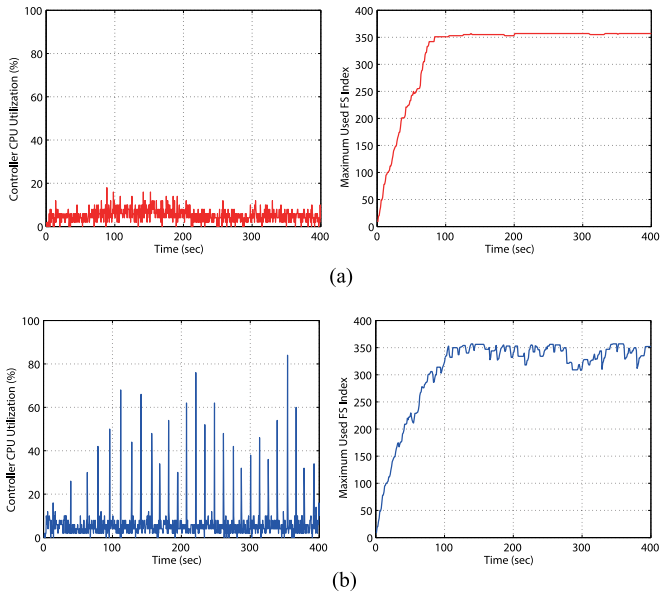


Fig. 10. Experimental results from online sequential DF when traffic load is 400 Erlangs. (a) CPU utilization of OF-C and maximum used FS index in the SD-EON without DF. (b) CPU utilization of OF-C and maximum used FS index in the SD-EON with DF.

Fig. 11(a) compares the results on bandwidth blocking probability (BBP) from the experiments with and without DF. The results on reconfiguration latency per DF (in the control plane) are plotted in Fig. 11(b), which indicates that the latency per DF operation is 650.4 msec when the traffic load is as high as 500

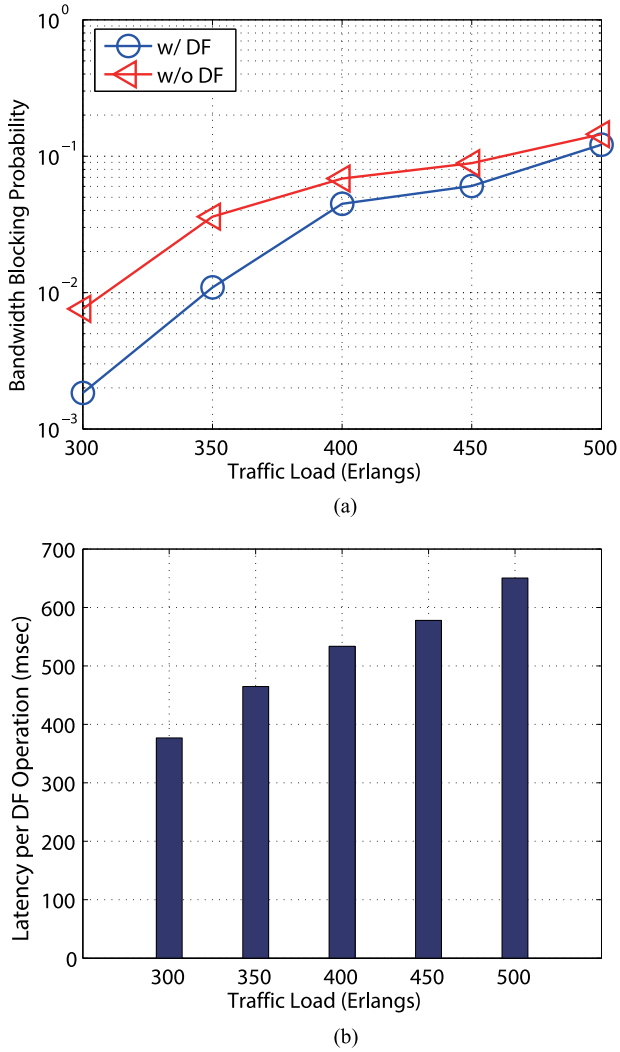


Fig. 11. Experiment results on (a) bandwidth blocking probability, and (b) reconfiguration latency per DF.

Erlangs. In the experiments, when the traffic load increases, the number of lightpaths to reconfigure in each DF is also larger, and this is the reason why the latency increases with traffic load in Fig. 11(b). In order to obtain each data point in Fig. 11(a) and (b), OF-C serves 4000 incoming requests from the 14 OF-AGs in the SD-EON testbed.

Note that the latency here is longer than that presented in [27], and this is because the implementation in [27] used the green-field reconfiguration scheme. Basically, OF-C first sends out *Flow-Mod* messages to disassemble all the selected lightpaths and then uses another set of *Flow-Mod* messages to set them up on new RSA locations. Therefore, as the reconfigurations are not performed in batches, the latency is shorter. However, since the green-field scheme does not apply "make-before-break", the lightpaths have prolonged traffic disruptions during DF. In other words, the green-field scheme in [27] provided a shorter reconfiguration latency per DF, but the traffic disruption can be as long as the latency. While even though the SBR-based scheme in this work provides slightly longer reconfiguration latency,

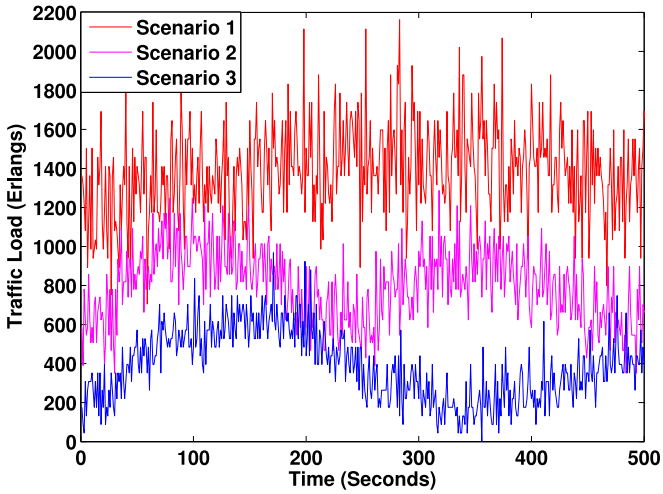


Fig. 12. Timing-varying traffic scenarios for adaptive DF experiments.

it applies "make-before-break" to all the lightpath reconfigurations to make sure that the traffic disruption is much shorter than the latency.

### *E. Experimental Demonstrations of Adaptive DF*

The experimental demonstration in the previous section invokes a DF operation when a fixed number of lightpath have expired and the selection ratio $\gamma$ is fixed too. However, it is known that for DF, there exists a tradeoff between the performance improvement and operation complexity increase [6]. If the two DF parameters mentioned above were not selected properly, we might either end up with introducing unnecessarily-high operation complexity or not invoking DF operations timely. Therefore, it is desired that we implement adaptive DF in SD-EONs to ensure cost-effectiveness, especially for the situation where the traffic is time-variant. In line of this, we extend the function of DF-Agent to facilitate adaptive DF, i.e., it invokes DF operations according to instant BBP and selects $\gamma$ adaptively based on the network status [6]. Basically, we modify and implement our adaptive DF algorithm in [6], and make sure that SBR is still supported for lightpath reconfigurations.

The adaptive DF experiments use the same setup as that in Section III-D, and the parameters are also similar except for the traffic loads, which is generated as time-variant. We design three traffic scenarios as *Scenarios* 1, 2 and 3 in Fig. 12, each of which covers a period of 500 s, with bandwidth requirement uniformly distributed within $[25, 150]$, $[25, 400]$, and $[25, 250]$ Gb/s, respectively. The experiments compare the performance of non-adaptive and adaptive DF schemes.

1) *Non-Adaptive DF:* A DF operation is invoked every 32 s, and each DF has $\gamma = 0.5$.
2) *Adaptive DF:* Both the timing and $\gamma$ of each DF is determined adaptively based on network status.

The two DF schemes are evaluated with three metrics, i.e., overall BBP, total number of lightpath reconfigurations, and reconfiguration latency per DF. Table I summarizes the experimental results, which show that the adaptive DF provides lower BBP with less number of reconfigurations for *Scenarios* 1 and 2, while for *Scenario* 3, the adaptive DF's BBP is still lower but it needs slightly larger number of reconfigurations.

TABLE I
RESULTS FROM ADAPTIVE AND NON-ADAPTIVE DF SCHEMES

| | | Number of Reconfigurations | Latency per DF (msec) | BBP |
|---|---|---|---|---|
| Scenario 1 | Adaptive | 5953 | 682.9 | 0.0171 |
| | Non-Adaptive | 6378 | 779.1 | 0.0204 |
| Scenario 2 | Adaptive | 1019 | 469.6 | 0.0166 |
| | Non-Adaptive | 1320 | 348.1 | 0.0216 |
| Scenario 3 | Adaptive | 2736 | 585.7 | 0.0172 |
| | Non-Adaptive | 2610 | 475.2 | 0.0287 |

## IV. DEMONSTRATIONS OF PARALLEL DF

The experimental demonstrations in the previous section are for sequential DF, and the main reason for invoking reconfigurations in batches sequentially is the dependencies among them. Note that for practical DF implementations in SD-EONs, it is desired that the related network control in OF-C is as simple as possible to minimize the reconfiguration latency. Hence, in this section, we investigate how to realize the DF-related reconfigurations in a purely parallel manner, i.e., all of them are accomplished simultaneously in one shot. In such a scenario, reconfiguration latency is reduced to the maximum extent. Moreover, parallel DF reduces the operation complexity of OF-C. In sequential DF, OF-C needs to maintain a state-machine to perform the SBRs group by group as shown in Fig. 5(b), while since parallel DF is stateless, the state-machine is not required any more.

### *A. Parallel DF Algorithm*

Previously, in [40], we have proposed an algorithm to parallelize the lightpath reconfigurations in DF. However, same as the cases of our sequential DF algorithms, it has to be tailored to fit into the DF implementation we discuss in this paper. Therefore, without losing generality, we provide the detailed procedure of parallel DF in this section.

In order to eliminate reconfiguration dependencies and realize parallel DF, we introduce a "move-to-vacancy constraint": a in-service lightpath can only be reconfigured to use an unoccupied FS block, and apply it together with the "acyclic constraint" discussed in Section III-A to the RSA mapping. Fig. 13(a) shows an example of finding the feasible FS blocks for two lightpaths $C_1$ and $C_2$, under the two constraints. We then build a conflict graph (CG), as shown in Fig. 13(b), to describe the conflicts among the feasible FS blocks of all the selected lightpaths that will be reconfigured. Specifically, for each selected lightpath, we insert nodes into the CG, which each represents a feasible FS block. Here, the implementation of parallel DF only considers the available FS blocks on the lightpath's current routing path to limit the size of the CG.

The nodes in the CG are then connected as follows. First of all, if any two nodes are for the same lightpath, we connect them

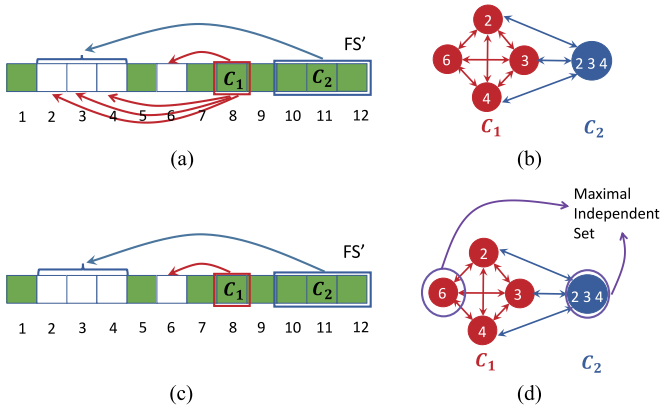


Fig. 13. An example of our parallel DF.

to indicate that a lightpath can only be reconfigured to one new FS block. Then, if two feasible FS blocks have overlapping, we connect the nodes that represent them to indicate that the FS blocks cannot be allocated to two different lightpaths simultaneously. We find the maximal independent set (MIS) in the CG, and since all the nodes in the MIS are not connected with each other, they provide us the largest set of lightpath reconfigurations that can be performed with SBR. For instance, Fig. 13(d) shows the MIS, and the corresponding parallel DF is illustrated in Fig. 13(c). *Algorithm 2* shows the detailed procedure of parallel DF in our implementation.

**Algorithm 2:** Parallel DF Implementation with SBR

```
1  DF Agent obtains network status from TED;
2  if a DF operation is needed then
3      DF Agent selects γ portion of in-service lightpaths in
       TED using the HUSIF strategy [33];
       // Lines 4-12 are executed in RCM:
4      get feasible FS blocks for all the selected lightpaths;
5      build the CG with all the feasible FS blocks;
6      find the MIS in the CG;
7      𝔾 = ∅;
8      for each node in the MIS do
9          find the lightpath C_i that it represents;
10         𝔾 = 𝔾 ∪ {C_i};
11     end
12     instruct RPM to encode SBR-enabled Flow_Mod
       messages for all the lightpaths in 𝔾;
       // End of the processing in RCM.
13     RPM invokes SBR for all the lightpaths in 𝔾;
14 end
```

### B. Experimental Demonstrations of Parallel DF

We conduct parallel DF experiments still with the setup in Section III-D, while the parameters are also similar except for $\gamma = 0.5$. The experiments compare the performance of sequential DF and parallel DF. Here, sequential DF refers to the periodic DF in Section III-D. Fig. 14 shows the experimental results on

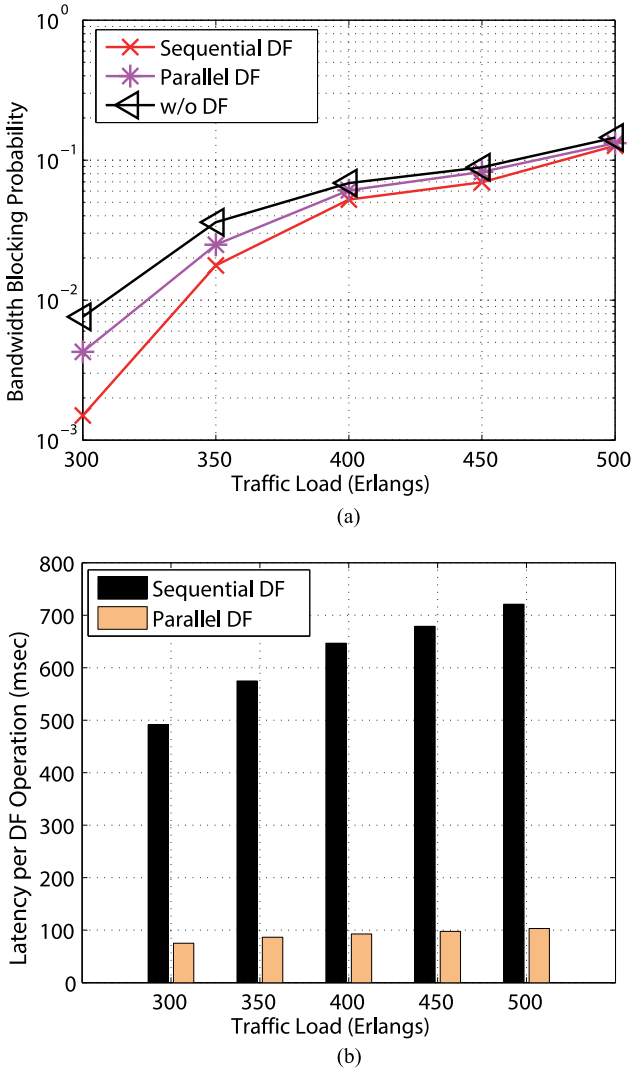


Fig. 14. Results on (a) bandwidth blocking probability and (b) reconfiguration latency per DF, from sequential DF and parallel DF experiments.

BBP and reconfiguration latency per DF. In Fig. 14(a), we observe that parallel DF cannot achieve as good BBP performance as that of sequential DF. This is due to the additional "move-to-vacancy constraint", i.e., in parallel DF, we sacrifice certain re-optimization margin for consolidating spectrum utilization to eliminate the reconfiguration dependencies. The results on reconfiguration latency per DF are plotted in Fig. 14(b), which indicate that parallel DF finishes each DF operation with much shorter time. Moreover, the results from parallel DF are almost constant, while the sequential DF's latency increases dramatically with the traffic load. This is because in sequential DF, OF-C needs to handle more reconfiguration batches for a higher traffic load, but parallel DF always finishes all the reconfigurations in one batch no matter how high the traffic load is.

## V. Conclusion

This paper investigated how to achieve highly-efficient online DF in SD-EONs that utilize OF. We first considered sequential

DF and modified our previous DF algorithm to make sure that the reconfigurations can be performed in batches and the "make-before-break" scheme can be applied to all of them. The modified algorithm was implemented in an OF-C, and we proposed OF extensions to facilitate SBR. Then, we further simplified the DF operations by designing and implementing parallel DF that can accomplish all the DF-related lightpath reconfigurations simultaneously. All the DF implementations were experimentally demonstrated in an SD-EON control plane testbed that consists of 14 stand-alone OF-AGs and one OF-C. The experimental results indicated that our OF-controlled online DF performed well and could improve network performance in an efficient way.

## REFERENCES


[1] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka, "Spectrum-efficient and scalable elastic optical path network: Architecture, benefits, and enabling technologies," *IEEE Commun. Mag.*, vol. 47, no. 11, pp. 66–73, Nov. 2009.

[2] O. Gerstel, M. Jinno, A. Lord, and B. Yoo, "Elastic optical networking: A new dawn for the optical layer?" *IEEE Commun. Mag.*, vol. 50, no. 2, pp. S12–S20, Feb. 2012.

[3] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, "Elastic bandwidth allocation in flexible OFDM-based optical networks," *J. Lightw. Technol.*, vol. 29, no. 9, pp. 1354–1366, May 2011.

[4] 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, no. 1, pp. 15–22, Jan. 2013.

[5] Y. Yin, K. Wen, D. Geisler, R. Liu, and S. J. B. Yoo, "Dynamic on-demand defragmentation in flexible bandwidth elastic optical networks," *Opt. Exp.*, vol. 20, pp. 1798–1804, Jan. 2012.

[6] 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, no. 9, pp. 1014–1023, Mar. 2014.

[7] R. Skoog, J. Gannett, K. Kim, H. Kobrinski, M. Rauch, A. Lehmen, and B. Wilson, "Analysis and implementation of a 3-way handshake signaling protocol for highly dynamic transport networks," in *Proc. Opt. Fiber Commun. Conf. Exhib.*, Mar. 2014, pp. 1–3.

[8] T. Takagi, H. Hasegawa, K. Sato, Y. Sone, A. Hirano, and M. Jinno, "Disruption minimized spectrum defragmentation in elastic optical path networks that adopt distance adaptive modulation," in *Proc. 37th Eur. Conf. Exhib. Opt. Commun.*, Sep. 2011, pp. 1–3.

[9] X. Chen, A. Jukan, and A. Gumaste, "Multipath de-fragmentation: Achieving better spectral efficiency in elastic optical path networks," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2013, pp. 1–9.

[10] R. Wang and B. Mukherjee, "Provisioning in elastic optical networks with non-disruptive defragmentation," *J. Lightw. Technol.*, vol. 31, no. 15, pp. 2491–2500, Aug. 2013.

[11] Y. Yin, H. Zhang, M. Zhang, M. Xia, Z. Zhu, S. Dahlfort, and S. J. B. Yoo, "Spectral and spatial 2D fragmentation-aware routing and spectrum assignment algorithms in elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. A100–A106, Oct. 2013.

[12] F. Cugini, F. Paolucci, G. Meloni, G. Berrettini, M. Secondini, F. Fresi, N. Sambo, L. Poti, and P. Castoldi,"Push-pull defragmentation without traffic disruption in flexible grid optical networks," *J. Lightw. Technol.*, vol. 31, no. 1, pp. 125–133, Jan. 2013.

[13] S. Shakya and X. Cao, "Spectral defragmentation in elastic optical path networks using independent sets," in *Proc. Opt. Fiber Commun. Conf.*, Mar. 2013, pp. 1–3.

[14] G. Goth, "Software-defined networking could shake up more than packets," *IEEE Internet Comput.*, vol. 15, no. 4, pp. 6–9, Jul./Aug. 2011.

[15] OpenFlow [Online]. Available: http://www.openflow.org/

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

[17] V. Gudla, S. Das, A. Shastri, N. Mckeown, L. Kazovsky, and S. Yamashita, "Experimental demonstration of OpenFlow control of packet and circuit switches," in *Proc. Opt. Fiber Commun. Conf.*, Mar. 2010, pp. 1–3.

[18] L. Liu, D. Zhang, T. Tsuritani, R. Vilalta, R. Casellas, L. Hong, I. Morita, H. Guo, J. Wu, R. Martinez, and R. Munoz, "Field trial of an OpenFlow-based unified control plane for multilayer multigranularity optical switching networks," *J. Lightw. Technol.*, vol. 31, no. 4, pp. 506–514, Feb. 2013.

[19] 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.

[20] M. Shirazipour, Y. Zhang, N. Beheshti, G. Lefebvre, and M. Tatipamula, "Openflow and multi-layer extensions: Overview and next steps," in *Proc. Eur. Workshop Softw. Defined Netw.*, Oct. 2012, pp. 13–17.

[21] S. Azodolmolky, R. Nejabati, E. Escalona, R. Jayakumar, N. Efstathiou, and D. Simeonidou, "Integrated OpenFlow-GMPLS control plane: An overlay model for software defined packet over optical networks," in *Proc. 37th Eur. Conf. Exhib. Opt. Commun.*, Sep. 2011, pp. 1–3.

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

[23] 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 *Proc. Opt. Fiber Commun. Conf.*, Mar. 2013, pp. 1–3.

[24] 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.

[25] N. Cvijetic, A. Tanaka, P. Ji, K. Sethuraman, S. Murakami, and T. Wang, "SDN and OpenFlow for dynamic flex-grid optical access and aggregation networks," *J. Lightw. Technol.*, vol. 32, no. 4, pp. 864–870, Feb. 2014.

[26] L. Liu, Y. Yin, M. Xia, M. Shirazipour, Z. Zhu, R. Proietti, Q. Xu, S. Dahlfort, and S. J. B. Yoo, "Software-defined fragmentation-aware elastic optical networks enabled by OpenFlow," in *Proc. Eur. Conf. Exhib. Opt. Commun.*, Sep. 2013, pp. 1–3.

[27] S. Ma., C. Chen, S. Li, M. Zhang, S. Li, Y. Shao, Z. Zhu, L. Liu, and S. J. B. Yoo, "Demonstration of online spectrum defragmentation enabled by openflow in software-defined elastic optical networks," in *Proc. Opt. Fiber Commun. Conf.*, Mar. 2014, pp. 1–3.

[28] "Spectral grids for WDM applications: DWDM frequency grid," *ITU-T Rec. G.694.1*, Feb. 2012.

[29] A. Giorgetti, F. Paolucci, F. Cugini, and P. Castoldi, "Fast restoration in SDN-based flexible optical networks," in *Proc. Opt. Fiber Commun. Conf.*, Mar. 2014, pp. 1–3.

[30] R. Proietti, C. Qin, B. Guan, Y. Yin, R. Scott, R. Yu, and S. J. B. Yoo, "Rapid and complete hitless defragmentation method using a coherent RX LO with fast wavelength tracking in elastic optical networks," *Opt. Exp.*, vol. 20, pp. 26958–26968, Nov. 2012.

[31] N. Jose, and K. Somani, "Connection rerouting/network reconfiguration," in *Proc. 4th Int. Workshop Design Rel. Commun. Netw.*, Oct. 2003, pp. 23–30.

[32] J. Ahmed, F. Solano, P. Monti, and L. Wosinska, "Traffic re-optimization strategies for dynamically provisioned WDM networks," in *Proc. 15th Int. Conf. Opt. Netw. Design Model.*, Feb. 2011, pp. 1–6.

[33] M. Zhang, W. Shi, L. Gong, W. Lu, and Z. Zhu, "Bandwidth defragmentation in dynamic elastic optical networks with minimum traffic disruptions," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2013, pp. 1–5.

[34] Y. Yin, M. Zhang, Z. Zhu, and S. Yoo, "Fragmentation-aware routing, modulation and spectrum assignment algorithms in elastic optical networks," in *Proc. Opt. Fiber Commun. Conf.*, Mar. 2013, pp. 1–3.

[35] Open vSwitch [Online]. Available: http://www.openvswitch.org/

[36] POX Wiki [Online]. Available: http://openflow.stanford.edu/display/ONL/POX+Wiki

[37] L. Gong, X. Zhou, L. Wei, and Z. Zhu, "A two-population based evolutionary approach for optimizing routing, modulation and spectrum assignments (RMSA) in O-OFDM networks," *IEEE Commun. Lett.*, vol. 16, no. 9, pp. 1520–1523, Sep. 2012.

[38] Y. Wang, X. Cao, and Y. Pan, "A study of the routing and spectrum allocation in spectrum-sliced elastic optical path networks," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2011, pp. 1503–1511.

[39] A. Bocoi, M. Schuster, F. Rambach, M. Kiese, C. Bunge, and B. Spinnler, "Reach-dependent capacity in optical networks enabled by OFDM," in *Proc. Opt. Fiber Commun. Conf.*, Mar. 2009, pp. 1–3.

[40] C. You, M. Zhang, and Z. Zhu, "Reduce spectrum defragmentation latency in EONs with effective parallelization of connection reconfigurations," in *Proc. Opt. Fiber Commun. Conf.*, Mar. 2014, pp. 1–3.


Authors' biographies not available at the time of publication.