Dynamic and Adaptive Bandwidth Defragmentation in Spectrum-Sliced Elastic Optical Networks With Time-Varying Traffic

Mingyang Zhang, Changsheng You, Huihui Jiang, and Zuqing Zhu, *Senior Member, IEEE*

***Abstract*—Elastic optical networks (EONs) enable network operators to have agile bandwidth management in the optical layer. In this paper, we investigate dynamic and adaptive bandwidth defragmentation (DF) in EONs with time-varying traffic using connection reconfigurations. We consider how to design DF procedure in a systematic way, and study the problems that have not been fully explored so far. Basically, we divide the procedure design into four subproblems: "*How to reconfigure?,*" "*How to migrate traffic?,*" "*When to reconfigure?,*" and "*What to reconfigure?,*" and solve them sequentially. For "*How to reconfigure?,*" we investigate the combination of routing and spectrum assignment (RSA) algorithms for DF, i.e., the RSA algorithm that the connections are originally served with and the algorithm that they are re-optimized with. For "*How to migrate traffic?,*" we propose to construct a dependency graph to represent the relations among the selected connections and to use it to assist the best-effort traffic migration. A *move-to-vacancy* method is also proposed to further reduce the traffic disruptions. For "*When to reconfigure?*" and "*What to reconfigure?,*" we propose intelligent timing selection and adaptive DF ratio selection methods to tackle the tradeoff between the bandwidth blocking probability (BBP) performance and operational complexity. Simulation results show that the algorithm with both methods implemented (DF-AT-AR) achieves better tradeoff between BBP performance and operational complexity, when compared with existing algorithms.**

***Index Terms*—Adaptive defragmentation, bandwidth fragmentation, elastic optical networks (EONs), time-varying traffic.**

I. Introduction

RECENTLY, the agile bandwidth management in the spectrum-sliced elastic optical networks (EONs) has attracted intensive research interests. It is known that with new technologies such as the optical orthogonal frequency-division multiplexing, the bandwidth allocation granularity in EONs can be reduced down to 12.5 GHz or less [1]. Hence, a new era of optical networking can be foreseen [2]. Besides these advantages, EONs also bring control plane challenges to the network operators, since during bandwidth allocation, they need to manipulate blocks of contiguous subcarrier frequency slots (FS') instead of independent wavelength channels [3], [4]. Most importantly, due to the increased number of narrow-band FS' in the optical spectrum, the problem of bandwidth fragmentation becomes more severe than that in the wavelength-division multiplexing networks [5]. Generally, bandwidth fragmentation refers to the existing of non-aligned, isolated, and small-sized unused bandwidth blocks in the optical spectrum, due to the frequent setting up and tearing down of dynamic connections [6], [7]. Since these bandwidth blocks can hardly be used for future connections, bandwidth fragmentation leads to low spectrum utilization in EONs and hence increases the blocking probability of requests whose bandwidth requirements are relatively large [7].

In order to relieve bandwidth fragmentation in EONs, people have investigated both preemptive and responsive approaches. The preemptive approaches try to minimize bandwidth fragmentation at the time when a connection is originally served [8]–[11]. In [8], [9], we studied fragmentation-aware routing and spectrum assignment (RSA) methods that quantify bandwidth fragmentation with different fragmentation-ratios and utilized them as the metrics to avoid bandwidth fragmentation during connection serving. Wang and Mukherjee proposed a spectrum management method to reduce bandwidth fragmentation in EONs in [10]. The proposed method partitioned the spectrum resources for connections with different bandwidth requirements and their simulation results showed that by doing so, the fragmentation-ratio and the blocking probability of EONs could be reduced. The investigation in [11] indicated that by leveraging RSA with multipath routing, bandwidth fragmentation could be relieved with path-splitting.

Even though the aforementioned preemptive approaches can relieve bandwidth fragmentation in EONs, they still have limitations. More specifically, as we will show in this paper, bandwidth fragmentation can still exist and deteriorate network performance, even when fragmentation-aware RSA methods have already been applied. Therefore, it would be interesting to investigate the responsive approaches that can reduce bandwidth fragmentation with network reconfigurations. Since their working principle is similar to that of the memory defragmentation in computer systems, we usually refer to these approaches as "bandwidth defragmentation" (DF) [5]. In [5], Takagi *et al.* proposed a DF algorithm that invoked a network

Manuscript received July 19, 2013; revised November 18, 2013; accepted December 23, 2013. Date of publication January 1, 2014; date of current version January 17, 2014. This work was supported in part by the Program for New Century Excellent Talents in University under Project NCET-11-0884, the National Natural Science Foundation of China under Project 61371117, the Natural Science Foundation of Anhui Province under Project 1208085MF88, the Fundamental Research Funds for the Central Universities (WK2100060010), and the Strategic Priority Research Program of the Chinese Academy of Sciences (XDA06010301).

The authors are with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: zqzhu@ieee.org; youcs@mail.ustc.edu.cn; jhh654321@126.com; 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.2013.2296781

reconfiguration with make-before-break rerouting when a request would be blocked otherwise, targeting for provisioning more requests successfully. DF schemes with the aid of an auxiliary graph and the maximum independent sets in it were studied independently in [6] and [12], where the authors tried to minimize traffic disruptions in each DF. The algorithms for dynamic bandwidth allocation in EONs with time-varying traffic have been investigated in [13]–[16], where with spectrum retuning, the authors proposed spectrum expansion, contraction, and reallocation policies to reduce the blocking probability. The push-pull based spectrum retuning technique has been experimentally demonstrated in [17] for DF and the DF algorithms based on it were investigated in [18]. Meanwhile, Proietti *et al.* have proposed and experimentally demonstrated a hop-tuning based retuning technique in [19], which could support full-spectrum retuning for more flexible DF and only introduces a relatively short reconfiguration latency (<1 μs). Based on the hop-tuning technique, we have studied the algorithms for DF with retuning only and DF with both retuning and rerouting in [20] and [21], respectively. Note that the retuning here means that a DF reconfiguration can only change the spectrum assignment of a connection but the routing path stays unchanged, while the rerouting means that a reconfiguration can change both the routing path and the spectrum assignment of a connection.

Previous investigations on the responsive approaches (i.e., DF with network reconfigurations [5], [6], [12]–[14], [17]–[23]) have tackled the problem from different perspectives. However, since DF in EONs is a relatively new research topic, there are still open questions regarding it, which have not been fully explored yet. For instance, DF usually needs to be invoked multiple times during network operation, but the timing of DF (i.e., the question of "*When to reconfigure?*") still has not been investigated thoroughly. Previous work just proposed to invoke a DF operation when a request would be blocked otherwise [5], or in a periodic manner [18], [21]. But these DF timing selection schemes may not be properly designed, especially for the EONs with time-varying traffic. Moreover, since DF is essentially re-performing RSA for some/all of the existing connections, both the RSA algorithm that the connections are originally served with and the RSA algorithm that they are re-optimized with would affect the effectiveness of DF. However, the selection of these two RSA algorithms (i.e., the question of "*How to reconfigure?*") is still under-explored, and to the best of our knowledge, no previous work has covered how to optimize the combination of RSA algorithms for DF.

In this paper, we consider how to design DF procedure in a systematic way, investigate the problems that have not been fully explored so far, and solve them to achieve dynamic and adaptive DF in EONs with time-varying traffic. Basically, we divide the algorithm design into four subproblems: "*How to reconfigure?*," "*How to migrate traffic?*," "*When to reconfigure?*," and "*What to reconfigure?*," and solve them one by one. The rest of the paper is organized as follows. Section II studies the comprehensive procedure to realize dynamic and adaptive DF in EONs. The subproblem of "*How to reconfigure?*" is investigated in Section III, and Section IV discusses how to solve "*How to migrate traffic?*" with a best-effort traffic migration scheme. Then, Sections V and VI describe the intelligent timing selection and adaptive DF ratio selection to address "*When to reconfigure?*" and "*What to reconfigure?*," respectively. The performance of the proposed algorithms is evaluated in Section VII for EONs with time-varying traffic. Finally, Section VIII summarizes the paper.

II. Dynamic and Adaptive Bandwidth Defragmentation

In this section, we investigate the comprehensive procedure to realize dynamic and adaptive DF in EONs. It is known that DF is to reconfigure certain existing connections with the objective of making the network spectrum usage more consolidated, less fragmented, and less misaligned. Therefore, systematically, we can divide DF into four subproblems:

1) *How to reconfigure:* Optimize the combination of RSA algorithms for DF, i.e., the RSA algorithm that the connections are originally served with and the algorithm that they are re-optimized with.
2) *How to migrate traffic:* Design connection migration method for DF and try to minimize traffic disruptions.
3) *When to reconfigure:* Determine the timing of each DF operation.
4) *What to reconfigure:* Decide how many and which connections to be reconfigured in each DF operation.

Note that since the DF with hop-tuning retuning and rerouting [19] is flexible and can provide relatively good performance [20], [21], our DF algorithms are based on it. The difference between hop-tuning and push-pull retuning [17] is that push-pull retuning can only move a connection to contiguous and available FS' on the same routing path, while hop-tuning can move it to any available location in the spectrum domain. Hence, the subproblem of "*How to reconfigure?*" can be solved by designing a suitable RSA algorithm to re-optimize the resource allocations of the existing connections.

Besides the network performance improvement, we should also carefully consider the additional cost (i.e., operational expenditure) and operational complexity associated with the DF when designing the procedure. Specifically, the number of connection reconfigurations should be well controlled, otherwise, the DF procedure may become too expensive and complicated to be practical. Therefore, we need to solve the subproblems of "*What to reconfigure?*" and "*When to reconfigure?*," and try to optimize the tradeoff between the performance improvement and the cost and complexity increase. At last but not the least, traffic migration (i.e., "*How to migrate traffic?*") is also critical for DF. Specifically, we need to make sure that the connections are migrated from the original RSA's to new RSA's with minimum traffic disruptions.

Based on the discussions above, we design the comprehensive DF procedure for EONs as in Algorithm 1. Lines 1–2 are for network initialization, and the normal network operation is in Lines 3–14. In Lines 4–7, we serve the pending requests, monitor the network status (i.e., spectrum utilization, blocking probability, and etc) with a sampling method, and determine whether a DF operation is necessary. The detailed procedure of a DF operation are included in Lines 9–12, where we solve

Algorithm 1 Overall Defragmentation Procedure

```
determine the combination of RSA algorithms for request
serving and reconfiguring;
initialize the control parameters;
while the network is operational do
   serve pending requests;
   release the resources of expired requests;
   monitor network status;
   determine whether a DF operation is necessary;
   if a DF operation is necessary then
      determine how many connections to reconfigure;
      select the connections to reconfigure;
      re-perform RSA for the selected connections and
      determine their new RSA locations;
      migrate the traffic on selected connections;
   end if
end while
```

the subproblems of "*How to reconfigure?*," "*What to reconfigure?*," and "*How to migrate traffic?*" sequentially. The details for each part of Algorithm 1 and their performance evaluations are discussed in the following sections.

III. How to Reconfigure?

A. Selection of RSA Algorithm Combination

We consider an EON network topology as $G(V, E)$, where V and E are the sets of nodes and fiber links, respectively. Each link can accommodate N FS'. A connection can be modeled as $C_i(R_{s_i,d_i}, a_i)$, where i is the unique index of the connection, R_{s_i,d_i} is the routing path from s_i to d_i ($s_i, d_i \in V$), and a_i is a N-bit bit-mask for the spectrum assignment. When $a_i[j] = 1$, the j-th FS is allocated to C_i, otherwise, $a_i[j] = 0$. Note that we assume that there are no spectrum converters in the EON and therefore each connection has to be set up all-optically end-to-end, under the spectrum-continuity constraint [3], [4].

As we have mentioned above in Section II, in order to solve "*How to reconfigure?*," we need to optimize the combination of RSA algorithms, i.e., the RSA algorithm that the connections are originally served with (first RSA) and the algorithm that they are re-optimized with (second RSA). We choose the first and second RSA algorithms within the following ones, since they can achieve relatively good performance on BBP with low computational complexity.

1) K-shortest path (KSP) RSA algorithm [24], where the routing path is selected as the shortest one that is among the K pre-calculated shortest paths for having enough spectrum resources to provision the request.
2) Minimum maximum used slot index (MMUSI) [21], where the path that can provide the minimum MUSI for spectrum assignment is selected as the routing path.
3) Fragmentation and misalignment-aware (FMA) RSA algorithm [9], where the routing path that can provide the minimum fragmentation-related metric is selected.

For all of them, we pre-compute K shortest paths for the s_i-d_i pair of the connection and then choose the routing path among them. When the routing path is determined, the spectrum assignment is based on first-fit.

The first RSA obtains the set of $\{R_{s_i,d_i}, a_i\}$ for a connection C_i, and its objective is to minimize the bandwidth blocking probability (BBP), which is defined as the ratio of blocked to total requested bandwidth in the EON. In this paper, we introduce the DF ratio, which is defined as the ratio of reconfigured to total number of connections in the EON, for each DF operation. For the kth DF operation, we denote its DF ratio as γ_k. Then, with γ_k, we can select the most "critical" connections to reconfigure using the highest used slot-index first (HUSIF) strategy [21]. Specifically, the HUSIF strategy sorts the existing connections C_i based on $hidx(a_i)$ in descending order, selects the first $\gamma_k \cdot |\mathbb{C}|$ connections, and stores them in $\mathbb{C}_s$. Here, $\mathbb{C}$ is the set of existing connections, $\mathbb{C}_s$ is the set of selected connections, and $hidx(\cdot)$ returns the highest index of bit "1" in a bit-mask. The second RSA (i.e., the RSA for reconfiguration) gets the new set $\{R'_{s_i,d_i}, a'_i\}$ based on $\{R_{s_i,d_i}, a_i\}$ and the current network status for all $C_i \in \mathbb{C}_s$, while its objective is to move the spectrum usage to the lower end of the spectrum domain to the maximum extent possible. Then, we have several RSA combinations for DF, and their performance is evaluated in the next section.

B. Performance Evaluations

To evaluate the performance of the RSA combinations, we design simulations with the 14-node NSFNET topology [4]. We assume that the bandwidth of each FS is 12.5 GHz and each fiber link can accommodate $N = 358$ FS', corresponding to the 4.475 THz bandwidth in the C-band. The dynamic traffic is generated using the Poisson traffic model and we select the source-destination pair of each request randomly. A request's bandwidth requirement is uniformly distributed within $[1, 16]$ FS'. In the simulations, we assume that γ_k is fixed and the DF operations are performed in equal time interval.

We first set the DF interval as ten time-units and the simulation results on BBP are in Fig. 1. In Fig. 1(a), we can see that when there are no DF operations, the blocking performance of the RSA algorithms degrades in the following order: FMA, MMUSI, KSP. By comparing the results in Fig. 1(b)–(d), we can conclude that KSP is not suitable for the second RSA for DF. For instance, in Fig. 1(b) and (d), when we use KSP as the second RSA, the results on BBP can even increase. On the other hand, MMUSI and FMA can improve the network's blocking performance no matter what algorithm is implemented as the first RSA. Meanwhile, it is interesting to notice that even when FMA is used as the first RSA, we can still reduce the BBP if a proper RSA algorithm is selected to reconfigure the connections afterwards. Specifically, the BBP results from FMA+FMA are lower than those from FMA without DF and FMA+MMUSI. These results suggest that even though a preemptive approach is effective to relieve bandwidth fragmentation, bandwidth fragmentation can still exist and deteriorate BBP performance if a responsive approach is absent. Among all the RSA combinations we test, FMA+FMA provides the best performance on BBP.

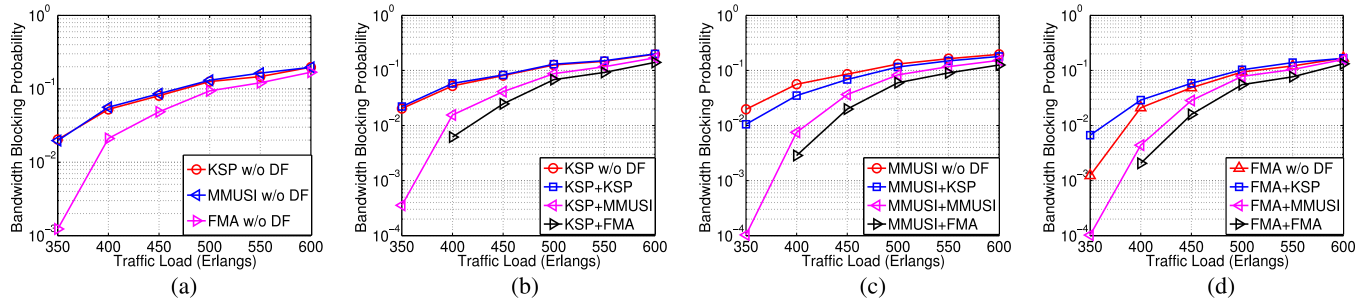

Fig. 1. Results on BBP from different RSA combinations for DF (DF interval is ten time-units, and $\gamma_k = 0.3$). (a) RSA without DF, (b) KSP as the first RSA, (c) MMUSI as the first RSA, and (d) FMA as the first RSA.

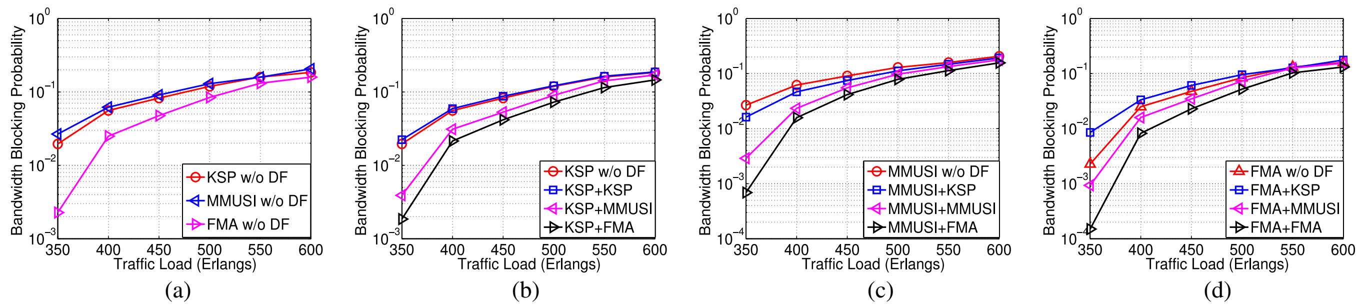

Fig. 2. Results on BBP from different RSA combinations for DF (DF interval is 20 time-units, and $\gamma_k = 0.3$). (a) RSA without DF, (b) KSP as the first RSA, (c) MMUSI as the first RSA, and (d) FMA as the first RSA.

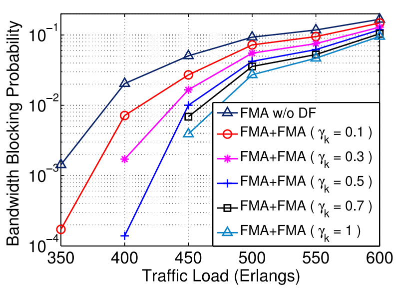

Fig. 3. Results on BBP from the RSA combinations of FMA+FMA with different γ_k (DF interval is ten time-units).

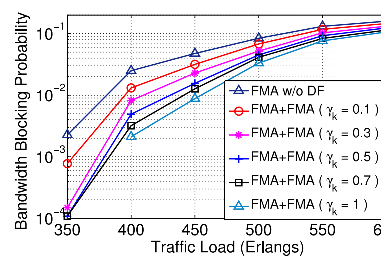

Fig. 4. Results on BBP from the RSA combinations of FMA+FMA with different γ_k (DF interval is 20 time-units).

Fig. 3 shows the simulation results on BBP from FMA+FMA with different γ_k, and we observe that by increasing the DF ratio γ_k, more BBP reduction can be achieved. Meanwhile, we should also mention that a larger γ_k leads to reconfiguring more connections in each DF operation, and hence increases the operational complexity of DF. Therefore, it would be interesting to investigate the subproblem of "*What to reconfigure?*" more thoroughly and obtain a better tradeoff. We will investigate this topic in Section VI.

We then increase the DF interval to 20 time-units and the corresponding simulation results are shown in Figs. 2 and 4. Similar trends can be seen in these results and the BBP results from FMA+FMA are still the lowest. However, we notice that since we make the DF operations less frequent, the BBP results increase for all cases. Note that making DF operations less frequent also brings down the operational complexity that is associated with the DF. Therefore, the corresponding tradeoff should be studied and the subproblem of "*When to reconfigure?*" is worth being investigated, especially for the EONs with time-varying traffic.

IV. How to Migrate Traffic?

A. *Migration Strategies*

Since migrating the traffic on the selected connections involves retuning their optical transponders and reconfiguring the optical switches along their routing paths, traffic disruptions can happen if the procedure are not designed properly. In this paper, we propose a best-effort traffic migration scheme as in Algorithm 2 to minimize the traffic disruptions.

Basically, for each $C_i \in \mathbb{C}_s$, traffic migration needs to accomplish a one-to-one moving, $\{R_{s_i,d_i}, a_i\} \mapsto \{R'_{s_i,d_i}, a'_i\}$. For two selected connections C_{i_1} and C_{i_2}, if the target RSA of C_{i_1} will use the resources that are currently occupied by C_{i_2}, i.e., "$R'_{s_{i_1},d_{i_1}} \cap R_{s_{i_2},d_{i_2}} \neq \emptyset$" and "$a'_{i_1} \otimes a_{i_2} \neq 0$" are both satisfied, we say that C_{i_1} depends on C_{i_2}. More specifically, if C_{i_1} depends on C_{i_2}, we cannot migrate the traffic on C_{i_1} until the

Algorithm 2 Best-Effort Traffic Migration

1: Construct DG, $G^d(V^d, E^d)$, for all selected connections in $\mathbb{C}_s$ based on their original and target RSA's;
2: **if** $G^d(V^d, E^d)$ is not a DAG **then**
3: find the MFVS in $G^d(V^d, E^d)$;
4: delete all links from the nodes in the MFVS;
5: perform best-effort MTV for the connections whose nodes are in the MFVS;
6: **end if**
7: **while** $G^d(V^d, E^d)$ is not empty **do**
8: find the nodes whose out-degrees are 0 and store them in V_0^d;
9: perform traffic migration for the connections whose nodes are in V_0^d;
10: delete all links that are incident to V_0^d;
11: delete V_0^d from $G^d(V^d, E^d)$;
12: **end while**

traffic migration of C_{i_2} has been done, otherwise, there would be traffic disruption since the *make-before-break* scenario [25] cannot be applied. Therefore, in *Line* 1 of Algorithm 2, we construct a dependency graph (DG) to represent the relations among selected connections and use it to assist the best-effort traffic migration. We denote the DG as $G^d(V^d, E^d)$, where V^d represents the node set and E^d is the link set. Each node $v^d \in V^d$ corresponds to a selected connection in $\mathbb{C}_s$, and if there is a directed link from v_1^d to v_2^d, the connection that v_1^d represents depends on that of v_2^d.

If the DG is a directed acyclic graph (DAG), which means that there is no circle in it, we can obtain the traffic migration sequence by performing a topological sort for it. Lines 7–12 show the detailed procedure of the traffic migration, since in this case, the *make-before-break* scenario can be applied, traffic disruptions can be avoided [26]. However, there is no guarantee that the DG will always be a DAG. If the DG is cyclic, we need to convert it to a DAG by breaking all circles in it while only introducing minimum traffic disruptions. This is actually equivalent to searching for the minimum feedback vertex set (MFVS) in the DG, and can be solved with an efficient algorithm developed in [27]. Then, we employ a *move-to-vacancy* (MTV) method to further reduce the traffic disruptions. Specifically, for the connections whose nodes are in the MFVS, we try to set up them with other available resources temporarily, and restore them to their target RSA's when the rest selected connections it depends are reconfigured. Note that when the network is crowded due to high traffic load, we may not always be able to find MTV resources for the connections whose nodes are in the MFVS. Unfortunately, when MTV cannot be applied, the connections can undergo relatively long traffic disruption because they have to be torn down and wait until all the desired resources for reconfiguration are released by other connections. Therefore, the traffic migration with MTV can only reduce traffic disruption in a best-effort way but cannot eliminate traffic disruption, since there is no guarantee that MTV will always succeed. Lines 2–6 show the procedure of converting the DG to a DAG.

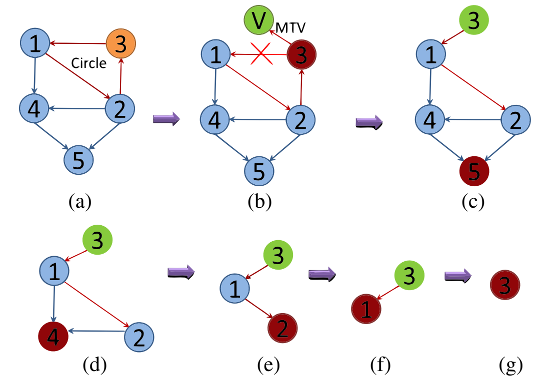

Fig. 5. An example of the traffic migration with MTV, under the assistance of the dependency graph (DG).

Fig. 5 shows an intuitive example of traffic migration with MTV. Here, we assume that there are five selected connections to be reconfigured and their initial dependence relation is shown in the DG in Fig. 5(a). Apparently, the DG is not a DAG since C_1, C_2 and C_3 form a circle. In order to convert the DG to a DAG, we find the MFVS in it, which is C_3. Therefore, with MTV, we reconfigure C_3 with some vacant spectrum resources temporarily and break the link from C_3 to C_1, as shown in Fig. 5(b). Then, in Fig. 5(c), the DG becomes a DAG and we can apply Lines 7–12 of Algorithm 2 to migrate the connections sequentially, as in Fig. 5(d)–(g). Here, the nodes in dark-red are for those that are reconfigured in the steps. Note that in the last step [in Fig. 5(g)], we restore C_3 to its target RSA when all the other connections are reconfigured. The algorithm stops when there is no node in the DG.

We evaluate the performance of Algorithm 2 with the simulations that use a similar setup as in Section III-B. We utilize FMA+FMA as the RSA algorithm combination for DF, fix the DF interval as ten time-units, and test the DF ratios $\gamma_k \in \{0.1, 0.3, 0.5, 0.7, 1\}$. Table I shows the simulation results on traffic disruption percentage, which is defined as the ratio of disrupted to total reconfigured connections in all DF operations. We observe that with MTV, the traffic disruption percentages drop dramatically for all simulated cases. Meanwhile, it is interesting to notice that the traffic disruption percentages of the cases without MTV do not change significantly for different traffic loads, when γ_k is fixed. We believe this can be explained as that when γ_k is fixed, the probability that the selected connections have cyclic dependencies is also fixed since their spectrum usages are randomly localized in certain spectrum range. On the other hand, when MTV is applied, the traffic disruption percentage increases with the traffic load, even when γ_k is fixed. This is due to the fact that when the traffic load is higher, the network spectrum becomes more crowed and hence the success rate of MTV gets smaller for less available spectrum resources.

B. Analysis on Traffic Disruption

In this section, we estimate the traffic disruption associated with the connection reconfigurations for DF. We investigate

TABLE I
RESULTS ON DF TRAFFIC DISRUPTION PERCENTAGES

Traffic Load (Erlangs)	$\gamma_k = 0.1$		$\gamma_k = 0.3$		$\gamma_k = 0.5$		$\gamma_k = 0.7$		$\gamma_k = 1$	
	w/o MTV	$w/$ MTV	w/o MTV	$w/$ MTV	w/o MTV	$w/$ MTV	w/o MTV	$w/$ MTV	w/o MTV	$w/$ MTV
350	7.76%	0.05%	8.82%	0.01%	9.82%	0.01%	10.16%	0.00%	10.41%	0.00%
400	7.43%	0.79%	8.87%	0.41%	10.03%	0.39%	10.81%	0.38%	10.82%	0.27%
450	7.03%	2.60%	9.26%	2.63%	10.99%	2.73%	11.73%	2.53%	11.76%	2.24%
500	7.82%	4.06%	9.23%	3.94%	11.16%	4.44%	12.07%	4.58%	12.12%	4.23%
550	7.80%	4.99%	9.67%	5.53%	10.97%	6.00%	12.04%	6.26%	12.02%	5.85%
600	7.39%	5.44%	9.56%	6.34%	11.60%	7.16%	12.26%	7.22%	12.49%	6.93%

TABLE II
THE LONGEST TRAFFIC DISRUPTION DUE TO DF ($\gamma_k = 0.3$)

Traffic Load (Erlangs)	Disruption Duration
350	$13 \cdot T_r$
400	$23 \cdot T_r$
450	$34 \cdot T_r$
500	$34 \cdot T_r$
550	$29 \cdot T_r$
600	$34 \cdot T_r$

the traffic disruption for three cases, because Algorithm 2 can reconfigure the connections in three different situations.

For the first case, we consider the situation where the make-before-break policy can be applied for a connection reconfiguration. Since the new connection is set up before the original one being broken, the traffic disruption only comes from the retuning process that is conducted by the transmitter and receiver at the two end-nodes. It is known that the hop-tuning technique demonstrated in [19] can accomplish the retuning process within 1 μs. Therefore, for this case, the traffic disruption's duration is within 1 μs, which can be easily compensated by inserting a small-sized buffer at the transmitter and will not cause excessive traffic loss.

The second case is the situation where the make-before-break policy cannot be applied for reconfiguring a connection because it has cyclic dependence with other connections, but the MTV policy can be applied. Hence, the connection needs to be reconfigured twice to achieve the "make-before-break" effect. Again, since for each of the two reconfigurations, the new connection is set up before the original one being broken, the traffic disruption's duration of this case is just double of that in the first case. Therefore, there will not be excessive traffic loss either, and the traffic loss can also be compensated by inserting a small-sized buffer at the transmitter.

The third case is the situation where both the make-before-break and MTV policies cannot be applied. Unfortunately, in this case, we will have much longer traffic disruption because the original connection has to be torn down before a new one can be set up, or when all the desired resources are released by other connections. Then, for the worst case, the traffic on a connection can be disrupted until all the other ones that occupy the spectrum resources it needs (i.e., those are on the same cycle of it in the DAG) are reconfigured. We denote the time required for reconfiguring a connection as T_r, which is in the order of μs according to the discussion above. We utilize simulations to investigate the traffic disruption, and record the longest traffic disruption durations for different traffic loads in Table II. The simulation scenario is similar as that in the previous section with $\gamma_k = 0.3$. It can be seen that the traffic disruption is still tolerable. The disruption duration increases with the traffic load until it reaches 450 Erlangs, and then, the disruption duration stabilizes at around $34 \cdot T_r$. Since the simulations generate all the connection requests randomly, the cycles in the DGs for DF operations are formed randomly too. Therefore, even though we simulate 10,000 requests for each traffic load, it is still possible that the size of the largest cycle in the DGs at a higher traffic load is smaller. This explains why we have a longest disruption duration as $29 \cdot T_r$ when the traffic load is 550 Erlangs.

In all, we can see that the traffic disruption associated with the connection reconfigurations for DF will be tolerable for practical network operations and it can be easily overcome by incorporating a buffer in the transmitter. For the third situation, the duration of traffic disruption is the longest and we need to buffer more data per connection. However, according to the results in Table I, the percentage of the third situation is within 7.3% of all the connection reconfigurations. In the future, we will work on new algorithms to further reduce the traffic disruption duration for the third situation.

V. WHEN TO RECONFIGURE?

Until now, we has assumed that the DF operations would be invoked in equal time interval and the interval is arbitrarily selected. However, as we have commented in Section III-B, for DF, there is a tradeoff between the performance improvement and the operational complexity increase. If we set the interval too small and invoke the DF operations too frequently, we may introduce unnecessary complexity since the bandwidth fragmentation in the EON is not severe at all. Otherwise, if the DF operations are not invoked timely, we may have dramatic BBP increase due to severe bandwidth fragmentation. Therefore, it is desired that we have a timing selection algorithm to invoke the DF operations intelligently, and ensure that they are performed cost-effectively.

It is known that the traffic load in an EON can vary unpredictably but in a gradual manner [2], [13]. Here, time-varying traffic can be generated by changing the intensity of the arrival process of requests over time, as shown in Fig. 8. We assume that the bandwidth requirement of each individual request is fixed and stays unchanged throughout the service period. Thus the DF timing selection algorithm should determine the intervals between two adjacent DF operations intelligently, according to the network status. In order to achieve this, we define the following parameters and variables,

1) $U(t)$: Bandwidth utilization at time t, as the ratio of used to total number of FS' in the EON.

Algorithm 3 Intelligent DF Timing Selection

Inputs: T_c, T_{min}, B_{th}.

1: $k = 0$, $j = 1$;
2: $t = 1$, $\Delta t = 0$, $W_k^T = T_{min}$, $t_{next} = t + W_k^T$;
3: $U_{-1} = 0$, $U_0 = 0$;
4: **while** the EON is operational **do**
5: $t = t + 1$, $\Delta t = \Delta t + 1$;
6: **if** $(t = j \cdot T_c)$ **then**
7: calculate bandwidth utilization U_j;
8: **if** $(U_j > U_{j-1})$ **AND** $(U_{j-1} < U_{j-2})$ **then**
9: reset Δt, k, j, W_k^T, U_{-1} and U_0 to their initial values;
10: $t_{next} = t + W_k^T$;
11: **else**
12: $j = j + 1$;
13: **end if**
14: **end if**
15: **if** $(t = t_{next})$ **then**
16: calculate instant BBP $B(\Delta t)$;
17: **if** $(B(\Delta t) \geq B_{th})$ **then**
18: invoke a DF operation;
19: $T_k = \Delta t$, $B_k = B(T_k)$, $k = k + 1$;
20: calculate R_k^B and A_k^B based on T_k;
21: determine W_k^T based on W_{k-1}^T, R_k^B and A_k^B;
22: $t_{next} = t + W_k^T$, $\Delta t = 0$;
23: **else**
24: $t_{next} = t_{next} + 1$;
25: **end if**
26: **end if**
27: run the EON for 1 time-unit;
28: **end while**

2) T_c: Constant time interval for monitoring the bandwidth utilization periodically.
3) U_j: The j-th monitored bandwidth utilization, as $U_j = U(t = j \cdot T_c)$.
4) $T_{\min}$: Shortest interval between adjacent DF operations.
5) T_k: Time interval between the $(k-1)$th and kth DF operations.
6) $B(\Delta t)$: Instant BBP, as the ratio of blocked to total requested bandwidth during the period Δt.
7) B_k: Instant BBP during T_k, as $B_k = B(T_k)$.
8) B_{th}: Instant BBP threshold, and if $B(\Delta t) < B_{\text{th}}$, a DF operation would not be triggered after Δt.
9) W_k^T: Time window set after the k-th DF operation.
10) R_k^B: Average speed of the instant BBP's change during $T_k + T_{k-1}$, as $R_k^B = \frac{2(B_k - B_{k-1})}{T_k + T_{k-1}}$.
11) A_k^B: Average acceleration of the instant BBP's change, as $A_k^B = R_k^B - R_{k-1}^B$.

The detailed procedure are shown in Algorithm 3. The algorithm operates with the three parameters, T_c, $T_{\min}$, and B_{th}, which can be set by the network operator. It determines the time interval between two adjacent DF operations (i.e., T_k) based on instant BBP and its characteristics (i.e., the average speed R_k^B and the acceleration A_k^B of its change). Lines 1–3 are for parameter initialization, and t_{next} stores the time instant to check instant BBP.

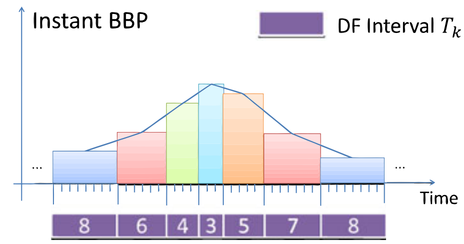

Fig. 6. An example of intelligent timing selection.

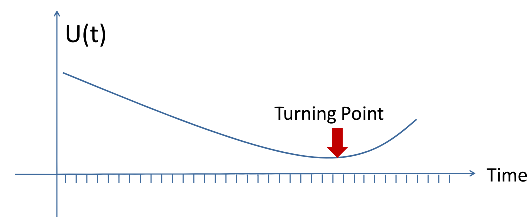

Fig. 7. The scenario of bandwidth utilization $U(t)$ starts to increase after a long decreasing period.

The key procedure of the algorithm are Lines 15–26, which show the mechanism of the DF timing selection. Specifically, a DF operation is invoked when and only when the preset time window W_k^T is reached **and** the instant BBP since the last DF operation (i.e., $B(\Delta t)$) is above the threshold B_{th}. When a DF operation has been invoked, we update the variables as in Lines 19–22. The time window W_k^T is set according to W_{k-1}^T, R_k^B and A_k^B. When $R_k^B > 0$, the network is becoming more crowded and requires more frequent DF operations, and we should decrease W_{k-1}^T to get W_k^T, otherwise, W_k^T should be changed to the opposite direction. Then, A_k^B determines the actual change step from W_{k-1}^T to W_k^T. When $A_k^B > 0$, the instant BBP is changing faster, and we should set a larger step, otherwise, we use a smaller one. When W_k^T is obtained, we set $t_{\text{next}} = t + W_k^T$ as the next time instant to check instant BBP. If a DF operation has not been invoked due to $B(\Delta t) < B_{\text{th}}$, we set t_{next} as the next time instant, as shown in *Line* 24. An example of this intelligent DF timing selection is shown in Fig. 6. Note that T_k and W_k^T are different in the algorithm. T_k is the actual time interval between two adjacent DF operations, while W_k^T is the time interval for checking the instant BBP. More specifically, when $B(\Delta t = W_{k-1}^T) \geq B_{\text{th}}$, we have $T_k = W_{k-1}^T$, otherwise, $T_k > W_{k-1}^T$.

For the intelligent DF timing selection mentioned above, we would prefer larger T_k when the bandwidth utilization $U(t)$ decreases, as the network becomes less crowded and frequent DF operations are not necessary. However, since the traffic load can vary unpredictably, we may have the scenario as shown in Fig. 7, i.e., $U(t)$ starts to increase after a long decreasing period. For this scenario, the algorithm may take a long time to adapt, for the reason that T_k has already become large and makes the algorithm under-sample the instant BBP for a while. In order to address this issue, we introduce a mechanism as shown in Lines 6–14 to monitor $U(t)$ periodically, and reset the variables to their initial values when the scenario in Fig. 7 is observed. Note that when $k = 0$, W_0^T is set as $T_{\min}$, which is the smallest interval permitted between two adjacent DF operations. Therefore, the problem of under-sampling can be solved.

VI. What to Reconfigure?

With Algorithm 3, we can invoke the DF operations intelligently according to the network status. However, how to determine the DF ratios $\{\gamma_k\}$ for all DF operations is still an open question. In this section, we describe the algorithm for selecting the DF ratios adaptively.

As we have explained in Section III-A, in the kth DF operation, the connections for reconfiguration should be selected with the HUSIF strategy [21], according to γ_k. It is known that a larger BBP reduction can be achieved with a larger γ_k [21], but a larger γ_k also increases the complexity of the DF operation. Therefore, we would like to optimize the tradeoff between the performance improvement and the complexity increase. Ideally, γ_k should be chosen based on the corresponding reductions on future BBP. However, since the traffic in a dynamic EON is unpredictable, it will be difficult to map γ_k to future BBP reduction. To this end, we leverage the compactness and consecutiveness of the EON's spectrum utilization [28] and use them as the indirect indicators of future BBP reduction. We first define a parameter as follows:

$$\eta(\gamma_k) = \frac{1}{2|E|} \sum_e \sum_l w_l^e(\gamma_k)(w_l^e(\gamma_i) + 1), \quad \forall e \in E, l \quad (1)$$

where E is the link set of the topology, $e \in E$ is a fiber link, $w_l^e(\gamma_k)$ is the size of the l-th available block of FS' on link e after the DF operation with γ_k. With $\eta(\gamma_k)$, we can quantify the benefit of a DF operation with γ_k as

$$\zeta(\gamma_k) = (\eta(\gamma_k) - \eta(0))(1 - \gamma_k). \quad (2)$$

Here, $\eta(0)$ corresponds to the case before the DF. Eq. (2) can balance the tradeoff between the performance improvement and the complexity increase. Since trying all possible values of γ_k continuously within $[0, 1]$ is not practical, we assume that in each DF operation, the network operator chooses γ_k from a finite set S of discrete values, e.g., $S = \{0.1, 0.3, 0.5, 0.7, 1\}$. With each value of γ_k, we emulate a DF operation with the current network status, obtain its benefit with (2), and implement the γ_k whose benefit is the largest.

VII. Performance Evaluations of Adaptive DF in EONs with Time-Varying Traffic

In this section, we evaluate the performance of the proposed adaptive DF algorithms. The simulation setup is similar to that in Section III-B, and we utilize FMA+FMA as the RSA algorithm combination for DF. The dynamic traffic is generated using the Poisson model with a time-varying load. In order to verify that the proposed algorithms can be applied to different traffic variations, we design three traffic scenarios as *Scenarios* 1, 2, and 3 in Fig. 8. Each scenario is assumed to cover a 24-h period, and the requests' bandwidth requirements are uniformly distributed within $[1, 6]$, $[1, 10]$, and $[1, 16]$ FS', for *Scenarios* 1, 2, and 3, respectively. In the simulations, we assume that one

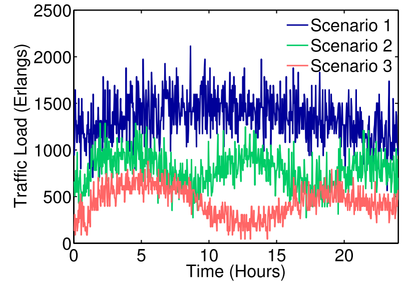

Fig. 8. Time-varying traffic scenarios in the simulations.

time-unit, i.e., $t = 1$, corresponds to 2 min. We set $T_{\min} = 8$ (i.e., 16 min), $B_{\text{th}} = 0.015$, and $T_c = 20$ (i.e., 160 min).

In order to evaluate the algorithms proposed in Sections V and VI, we implement five algorithms:

1) DF-AT-AR: When both the timing and DF ratio selections are adaptive, we refer to the algorithm as DF with adaptive timing and adaptive ratio selections.
2) DF-AT-FR: When only the timing selection is adaptive, we refer to the algorithm as DF with adaptive timing and fixed ratio selections.
3) DF-FT-FR: When both the timing and DF ratio selections are fixed, we refer to the algorithm as DF with fixed timing and fixed ratio selections.
4) DF-FNR-FR: When a DF operation is triggered after fixed number of expired requests [21], we refer to the algorithm as DF after fixed number of expired requests with fixed ratio selection.
5) DF-RB-FR: When a DF operation is triggered by the incident that a request would be blocked otherwise [5], we refer to the algorithm as DF triggered by request blocking with fixed ratio selection.

The above algorithms are evaluated with three performance metrics:

1) *Metric* 1: The percentage of the time intervals whose instant BBP is above 0 (i.e., $B(\Delta t = 1) > 0$).
2) *Metric* 2: The overall BBP for the whole 24-h period.
3) *Metric* 3: The total number of connection reconfigurations within the 24-h period.

Metric 1 indicates the volume of the EON's serious congestions in the time domain, and therefore can be used to judge whether the DF operations are invoked timely or not. *Metric* 2 shows the long-term performance of the DF algorithms, and *Metric* 3 measures the DF algorithms' complexities.

In the simulations, we have $\gamma_k \in \{0.1, 0.3, 0.5, 0.7, 1\}$ for DF-AT-AR, and for the rest of the DF algorithms (i.e., DF-FT-FR, DF-FNR-FR, DF-AT-FR, and DF-RB-FR), we set $\gamma_k = 0.5$. Fig. 9 shows that the algorithms with adaptive timing selection (i.e., DF-AT-FR and DF-AT-AR) provide smaller *Metric* 1 than other algorithms, except for DF-RB-FR. More specifically, compared with DF-FT-FR and DF-FNR-FR, DF-AT-AR reduces *Metric* 1 by 15.9% and 12.6% on average, respectively. This verifies that the adaptive timing selection can invoke the DF operations timely. Meanwhile, if we exclude DF-RB-FR, DF-AT-FR and DF-AT-AR also achieve the smallest overall BBP (*Metric* 2) in Fig. 10. More specifically, compared with DF-FT-FR

TABLE III
RESULTS ON TOTAL NUMBER OF CONNECTION RECONFIGURATIONS WITHIN THE 24-H PERIOD (*METRIC* 3).

	w/ o DF	DF-FT-FR	DF-FNR-FR	DF-AT-FR	DF-AT-AR	DF-RB-FR
Scenario 1	0	22291	23368	28771	23204	390868
Scenario 2	0	11380	11750	13790	11130	128270
Scenario 3	0	7084	7536	8405	6839	76408

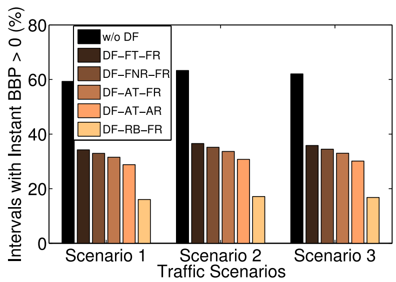

Fig. 9. Results on percentages of the time intervals whose instant BBP is above 0 (*Metric* 1, for DF-AT-AR, $\gamma_k \in$ [0.1, 1], and for the rest of the DF algorithms, $\gamma_k = 0.5$).

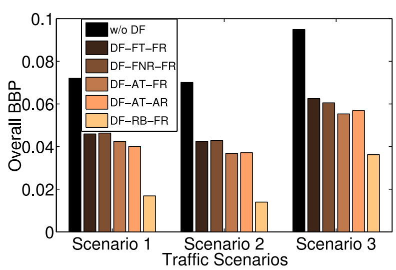

Fig. 10. Results on overall BBP for the 24-h period (*Metric* 2, for DF-AT-AR, $\gamma_k \in$ [0.1, 1], and for the rest of the DF algorithms, $\gamma_k = 0.5$).

and DF-FNR-FR, DF-AT-AR reduces *Metric* 2 by 11.5% and 11.0% on average, respectively. It is interesting to observe that DF-RB-FR can achieve smaller *Metrics* 1 and 2 than DF-AT-FR and DF-AT-AR. We believe this is due to the fact that DF-RB-FR invokes DF operations very frequently, i.e., whenever a request would be blocked otherwise. However, this kind of timing selection brings excessive operational complexity. From Table III, we can see that the total numbers of connection reconfigurations (*Metric* 3) from DF-RB-FR are always ten times or more than those from DF-AT-AR. By comparing the results from DF-AT-FR and DF-AT-AR, we can see that DF-AT-AR always provides smaller *Metrics* 1 and 3. The results on *Metric* 2 from DF-AT-AR are smaller than, comparable to, and slightly larger than those from DF-AT-FR, for traffic *Scenarios* 1, 2, and 3, respectively. This is because for DF-AT-FR, we fixed γ_k at a relatively large value as 0.5, while the adaptive DF ratio selection in DF-AT-AR can select smaller γ_k to tackle the tradeoff between performance improvement and complexity increase. Note that compared with DF-AT-FR, DF-AT-AR reduces *Metric* 3 up to 19.3%.

We then compare the simulation results from DF-FT-FR, DF-FNR-FR, DF-AT-FR, and DF-AT-AR. In Figs. 9 and 10, we observe that DF-AT-AR and DF-AT-FR achieve smaller *Metrics* 1 and 2 than DF-FT-FR and DF-FNR-FR. However, the *Metric* 3 from DF-AT-FR is larger than those from DF-FT-FR and DF-FNR-FR in Table III. This is due to the fact that in order to adapt to traffic variations, the intelligent timing selection in DF-AT-FR invokes more frequent DF operations than DF-FT-FR and DF-FNR-FR, while as the time interval between DF operations becomes smaller, a relatively large γ_k as 0.5 is not necessary any more. The adaptive DF ratio selection of DF-AT-AR can overcome this drawback. In the situation when frequent DF operations are invoked, the benefit from increasing γ_k gets smaller, and therefore DF-AT-AR will select a relatively small γ_k to avoid reconfiguring connections unnecessarily. In all, the simulation results indicate that among all algorithms, DF-AT-AR achieves the best tradeoff between BBP performance and operational complexity as it can stabilize and reduce BBP with a reasonably small number of connection reconfigurations.

VIII. CONCLUSION

In this paper, we considered how to design DF procedure in a systematic way, and designed procedure to achieve dynamic and adaptive DF in EONs with time-varying traffic. We divided the procedure design into four subproblems: "*How to reconfigure?*," "*How to migrate traffic?*," "*When to reconfigure?*," and "*What to reconfigure?*," and solved them one by one.

For "*How to reconfigure?*," we investigated the combination of RSA algorithms for DF, i.e., the RSA algorithm that the connections were originally served with and the algorithm that they were re-optimized with. Our simulation results indicated that the selections of both RSA algorithms could affect the BBP performance significantly and among all the RSA algorithm combinations we tested, FMA+FMA provided the best BBP performance. For "*How to migrate traffic?*," we proposed to construct a dependency graph (DG) to represent the relations among the selected connections and to use it to assist the best-effort traffic migration. An MTV method was also proposed to further reduce the traffic disruptions. The simulation results showed that our traffic migration method only introduced 12.49% traffic disruptions even when the traffic load was as high as 600 Erlangs and a full reconfiguration ($\gamma_k = 1$) was invoked in each DF operation. The results also showed that with MTV, the traffic disruption percentage of that case could be further reduced to 6.93%. For "*When to reconfigure?*" and "*What to reconfigure?*," we proposed intelligent timing selection and adaptive DF ratio selection methods to tackle the tradeoff between BBP performance improvement and operational complexity increase, and investigated their performance in simulations that used time-varying traffic load. The simulation results showed that the algorithm with both methods implemented (DF-AT-AR) achieved better tradeoff between BBP performance and operational complexity, when compared with several existing algorithms. Hence, the results confirmed that DF-AT-AR could provide a feasible solution for cost-effective DF in dynamic EON with time-varying traffic.

References

[1] W. Shieh, X. Yi, and Y. Tang, "Transmission experiment of multi-Gigabit coherent optical OFDM systems over 1000 km SSMF fibre," *IEE Electron. Lett.*, vol. 43, no. 3, pp. 183–185, Feb. 2007.

[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] 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. Eur. Conf. Exhib. Opt. Commun.*, Sep. 2011, pp. 1–3.

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

[7] 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, no. 7, pp. 2970–2978, Jul. 2013.

[8] M. Zhang, W. Lu, Z. Zhu, Y. Yin, and B. Yoo, "Planning and provisioning of elastic O-OFDM networks with fragmentation-aware routing and spectrum assignment (RSA) algorithms," in *Proc. Asia Commun. Photon. Conf.*, Nov. 2012, pp. 1–3.

[9] Y. Yin, H. Zhang, M. Zhang, M. Xia, Z. Zhu, S. Dahlfort, and 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.

[10] R. Wang and B. Mukherjee, "Spectrum management in heterogeneous bandwidth networks," in *Proc. Global Commun. Conf.*, Dec. 2012, pp. 2907–2911.

[11] X. Chen, A. Jukan, and A. Gumaste, "Multipath de-fragmentation: Achieving better spectral efficiency in elastic optical path networks," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 390–394.

[12] 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.

[13] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, "Dynamic bandwidth allocation in flexible OFDM-based networks," in *Proc. Opt. Fiber Commun. Conf.*, Mar. 2011, pp. 1–3.

[14] E. Palkopoulou *et al.*, "Dynamic cooperative spectrum sharing in elastic networks," in *Proc. Opt. Fiber Commun. Conf.*, Mar. 2013, pp. 1–3.

[15] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, "Time-varying spectrum allocation policies and blocking analysis in flexible optical networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 1, pp. 13–25, Jan. 2013.

[16] M. Klinkowski, M. Ruiz, L. Velasco, D. Careglio, V. Lopez, and J. Comellas, "Elastic spectrum allocation for time-varying traffic in flexgrid optical networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 1, pp. 26–38, Jan. 2013.

[17] 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.

[18] 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.

[19] R. Proietti, C. Qin, B. Guan, Y. Yin, R. Scott, R. Yu, and 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.

[20] M. Zhang, Y. Yin, R. Proietti, Z. Zhu, and B. Yoo, "Spectrum defragmentation algorithms for elastic optical networks using hitless spectrum retuning techniques," in *Proc. Opt. Fiber Commun. Conf.*, Mar. 2013, pp. 1–3.

[21] 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. 3894–3898.

[22] J. Zhang, Y. Zhao, X. Yu, and J. Zhao, "Priority-based defragmentation scheme in spectrum-efficient optical networks," in *Proc. Int. Conf. Opt. Internet*, Oct. 2012, pp. 16–17.

[23] A. Patel, P. Ji, J. Jue, and T. Wang, "Defragmentation of transparent flexible optical WDM (FWDM) networks," in *Proc. Opt. Fiber Commun. Conf.*, Mar. 2011, pp. 1–3.

[24] 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 INFOCOM*, Apr. 2011, pp. 1503–1511.

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

[26] N. Jose and K. Somani, "Connection rerouting/network reconfiguration," in *Proc. Int. Conf. Design Reliable Commun. Netw.*, Oct. 2003, pp. 23–30.

[27] H. Lin and J. Jou, "Computing minimum feedback vertex sets by contraction operations and its applications on CAD," in *Proc. IEEE Int. Conf. Comput. Design*, Oct. 1999, pp. 364–369.

[28] Y. Sone, A. Hirano, A. Kadohata, M. Jinno, and O. Ishida, "Routing and spectrum assignment algorithm maximizes spectrum utilization in optical networks," in *Proc. Eur. Conf. Exhib. Opt. Commun.*, Sep. 2011, pp. 1–3.

Authors' biographies not available at the time of publication.