

Optimizing Deadline-Driven Bulk-Data Transfer to Revitalize Spectrum Fragments in EONs [Invited]

Wei Lu, Zuqing Zhu, and Biswanath Mukherjee

Abstract—This paper studies how to optimize deadline-driven bulk-data transfer to recycle spectrum fragments in elastic optical networks (EONs). We first formulate a mixed integer linear program (MILP) for offline optimization with one of two objectives: 1) maximize the average data-transfer percentage of data-oriented requests (DO-Rs), or 2) minimize the incompletable data transfers among the DO-Rs. Then, we propose two dynamic provisioning algorithms. One addresses the first objective by setting a threshold on the minimum transferred data per time slot, and the other considers both admission control and blocking-aware routing and spectrum allocation to optimize the service provisioning according to the second objective. Simulation results indicate that the proposed algorithms can effectively optimize the DO-Rs' data transfers according to their objectives, revitalize the spectrum fragments, and improve the network spectrum utilization significantly.

Index Terms—Bulk-data transfer; Deadline; Elastic optical networks (EONs); Spectrum fragments.

I. INTRODUCTION

Flexible-grid elastic optical networks (EONs) provide a promising infrastructure for future optical backbone networks [1,2], since they can facilitate flexible and dynamic bandwidth allocation in the optical layer. However, it is also known that spectrum fragmentation [due to the accumulation of small-sized, nonaligned, and isolated frequency-slot (FS) blocks] is intrinsic to dynamic EONs and can restrict spectrum utilization [3,4]. Spectrum fragmentation occurs due to the dynamic provisioning of flow-oriented requests (FO-Rs), which have strict quality-of-service (QoS) requirements on the setup delay and transmission bandwidth (e.g., video streaming). In general, FO-Rs can use either immediate reservation (IR) [5,6] or advance reservation (AR) [7,8] schemes. Basically, IR and AR have different requirements on setup delay, but they both reserve a fixed bandwidth for each request. Since these FO-Rs can come and

leave on-the-fly, they can divide the optical spectra into small spectrum fragments [9], which is similar to memory fragmentation in computers. What makes the situation worse is that an AR-based FO-R can reserve an FS block for a future duration and cause two-dimensional (2D) spectrum fragments, i.e., spectrum fragments existing in both time and spectrum domains in a correlated manner [7]. Fortunately, due to the flexibility of data-oriented requests (DO-Rs), which need to accomplish bulk-data transfers that are delay-tolerant and allow flexible bandwidth allocation, these spectrum fragments can be effectively recycled if we optimize the service provisioning scheme of DO-Rs carefully [10].

Advances in networking technologies have stimulated the fast development of data-oriented services in backbone networks, e.g., e-science with grid computing and datacenter backup and migration. Previously, inspired by dynamic bandwidth allocation to accommodate time-varying traffic in EONs [11,12], we proposed malleable reservation (MR) based bulk-data transfer to leverage spectrum retuning and transmission pausing for DO-Rs to recycle the 2D spectrum fragments generated by FO-Rs [10]. However, the MR scheme can only optimize the provisioning of one DO-R at a time. Later, we studied how to optimize the provisioning of multiple DO-Rs simultaneously in [13], and provided some preliminary results.

In this paper, we extend our work in [13] and optimize the deadline-driven bulk-data transfers in an EON with one of two objectives: 1) maximize the average data-transfer percentage of the DO-Rs,¹ or 2) minimize the incompletable data transfers among the DO-Rs.² Basically, for each bulk-data transfer, the network operator can use changeable routing and spectrum assignment (RSA) configurations over time, while the maximum number of RSA reconfigurations is constrained to limit the operational complexity. Note that, in today's optical backbone networks, the traffic would not be as dynamic as that considered in this work. However, due to the rise of cloud computing [14], big data applications [15], and datacenter networks [16], bandwidth demands in optical backbone networks are becoming more dynamic as an inevitable trend. Therefore, considering highly dynamic traffic can make the bandwidth provisioning

Manuscript received June 2, 2015; revised October 11, 2015; accepted October 13, 2015; published November 13, 2015 (Doc. ID 242151).

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

W. Lu is also a visiting Ph.D. student at the University of California, Davis, Davis, California 95616, USA.

B. Mukherjee is with the Department of Computer Science, University of California, Davis, Davis, California 95616, USA.

<http://dx.doi.org/10.1364/JOCN.7.00B173>

¹The data-transfer percentage is the ratio of the completed data transfer.

²We consider a DO-R as blocked if it is incompletable.

more adaptive and future-proof. We first formulate a mixed integer linear program (MILP) for offline optimization. Then, we design two online service provisioning algorithms. One aims to maximize the average data-transfer percentage of the DO-Rs by setting a threshold on the minimum transferred data per time slot, and the other considers both admission control and blocking-aware RSA to minimize the incompletable data transfers among the DO-Rs.

The rest of the paper is organized as follows. Section II provides a survey on related work. Section III describes the network model and problem formulation. In Section IV, the MILP model is formulated for offline optimization. Then, online provisioning algorithms are proposed in Section V. We present performance evaluation in Section VI. Finally, Section VII summarizes the paper.

II. RELATED WORK

Previous investigations have studied how to handle bulk-data transfer in various networks. In [17], the authors proved that the problem of finding the optimal routing path to minimize the data-transfer time with flexible bandwidth allocation is \mathcal{NP} -hard. The work in [18] tried to find the optimal data-transfer paths under the constraint that path switchings can only be performed a limited number of times. Reference [19] discussed the complexity of similar problems with and without path switchings, and proved that both of them are \mathcal{NP} -complete. References [20] and [21] proposed a multiple-interval resource reservation scheme, which divides a data-transfer window into several intervals and reserves a fixed amount of resources in each of them. The authors of [22] investigated the scheduling problem for multiple concurrent DO-Rs, reduced it to the maximum concurrent multi-commodity flow problem, and designed a store-and-forward scheme for it. Then, they extended the work to propose a scheme that uses the already-paid-for-off-peak capacity to serve DO-Rs globally [23]. To achieve deadline-guaranteed bulk-data transfer, Ref. [24] designed both admission control and scheduling algorithms for e-science. However, none of these studies considered optical networks as the physical infrastructure. As the resource allocation scheme in EONs is different from that in packet networks (used in the above approaches), these approaches cannot be used to optimize deadline-driven bulk-data transfers in EONs.

The studies in [25–27] have addressed bulk-data transfers in optical networks. In [25], the authors proposed an elastic bandwidth reservation scheme for LambdaGrids, which can dynamically adjust the bandwidth allocations of in-service DO-Rs to accommodate new requests. The authors of [26] investigated the problem of provisioning DO-Rs with flexible transmission rates in wavelength-division multiplexing (WDM) networks, aiming to achieve deadline-guaranteed transmissions. The routing and bandwidth scheduling of DO-Rs in WDM networks has been addressed in [27], where three heuristics were proposed to minimize the data-transfer time. Nevertheless, the aforementioned investigations were not based on EONs, and they did not consider the 2D spectrum fragmentation either. The

flexible spectrum assignment and 2D fragmentation in EONs can make the problem of optimizing deadline-driven bulk-data transfers more complex and bring new challenges to achieve efficient service provisioning.

III. PROBLEM DESCRIPTION

A. Network and Request Models

We model the EON as a directed graph $G(V, E)$, where V denotes the node set and E is the set of fiber links. Each fiber can accommodate B frequency slots (FSs). We assume there is no spectrum converter in the EON, and hence the spectrum-continuity constraint has to be satisfied when setting up a connection. Besides, the EON operates in a discrete-time manner; i.e., the time axis is divided into time slots (TSs) evenly, and network operations happen at the TS boundaries. For the service scheduling of DO-Rs, we define T as the maximum number of look-ahead TSs that the operator can observe. The r th DO-R is modeled as a tuple $R^r(s^r, d^r, \mathcal{F}^r, t_a^r, t_e^r)$, where s^r and d^r are the source and destination nodes, \mathcal{F}^r is the data to be transmitted in terms of the usage of FSs over TSs, t_a^r is the TS at which the request arrives, and t_e^r is the TS before when the request's data transfer should be completed, thereby capturing the deadline for R^r .

B. Bulk-Data Transfer to Recycle Spectrum Fragments

Similar to our work in [10], we consider a dynamic scenario, in which FO-Rs and DO-Rs arrive and leave on-the-fly. Considering the strict QoS requirements of FO-Rs, we serve them with high priority such that, at each TS, the pending FO-Rs are served before the DO-Rs. Since DO-Rs are delay-tolerant and allow flexible bandwidth allocation, they can utilize the 2D spectrum fragments left over by FO-Rs. To record the spectrum fragments, we define a set $C_{e,t}$ to denote the set of available FS blocks on link e during TS t . The size of the i th FS block on link e during t is $w_{e,t}^i$ in FSs. For the sake of referring to all the available FS blocks in the EON, we define a variable $\pi_{e,t}^i$ whose value is the globally unique index of the i th available FS block on e during TS t .

To serve $R^r(s^r, d^r, \mathcal{F}^r, t_a^r, t_e^r)$, we need to determine the RSA scheme,³ i.e., find a routing path $p_{s^r, d^r, t}$ and assign an FS block $[f_{s,t}^r, f_{e,t}^r]$ on it, where $f_{s,t}^r$ and $f_{e,t}^r$ are the indices of the start FS and end FS to satisfy

$$[f_{s,t}^r, f_{e,t}^r] \in \bigcap_{e \in p_{s^r, d^r, t}} C_{e,t}. \quad (1)$$

Note that, if necessary, the data transfer of R^r can also be paused in a certain TS $t \in [t_a^r, t_e^r]$. If the data of R^r have been transferred completely, we have

³To save bandwidth-variable transponders, we assume that each request only consumes a pair of BV-Ts; i.e., we do not consider spectrum splitting.

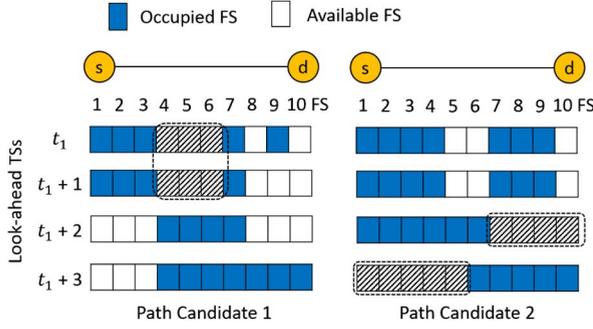


Fig. 1. Bulk-data transfer to recycle spectrum fragments in an EON.

$$\sum_{\{t:t \in [t_a^r, t_e^r], \vartheta_t^r = 1\}} f_{e,t}^r - f_{s,t}^r + 1 \geq \mathcal{F}^r, \quad (2)$$

where ϑ_t^r is a Boolean variable that equals 1 if there is data transfer for R^r at TS t , and 0 otherwise. For a TS t that has $\vartheta_t^r = 1$, if the $p_{s^r, d^r, t}$ and/or $[f_{s,t}^r, f_{e,t}^r]$ in it differ from those in $t-1$, an RSA reconfiguration is needed at the beginning of t . Hence, we define another Boolean variable ϕ_t^r that equals 1 if there is an RSA reconfiguration at t for R^r , and 0 otherwise. As frequent RSA reconfigurations introduce a high operational cost, we restrict the RSA reconfiguration times for each DO-R below M ; i.e., a complete bulk-data transfer should satisfy⁴

$$\sum_{\{t:t \in [t_a^r, t_e^r], \vartheta_t^r = 1\}} \phi_t^r \leq M + 1. \quad (3)$$

Figure 1 illustrates an example of bulk-data transfer to recycle 2D spectrum fragments in an EON. The DO-R comes in at t_1 and needs to accomplish the data transfer of 15 FSs before $t_1 + 3$. On the two path candidates, the FSs occupied by the DO-R are marked with the striped rectangles, i.e., FS block [4, 6] on path candidate 1 during $[t_1, t_1 + 1]$, and FS blocks [7, 10] and [1, 5] on path candidate 2 at $t_1 + 2$ and $t_1 + 3$, respectively. Since the DO-R's data can be transferred with the accumulated FSs of the marked schemes, it can be completed before the deadline. Specifically, the initial RSA configuration is performed at t_1 , and then two RSA reconfigurations are conducted at $t_1 + 2$ and $t_1 + 3$.

C. Optimization Objectives

Due to the constraints from t_e^r and M , it is possible that only a part of the data of DO-R R^r ($s^r, d^r, \mathcal{F}^r, t_a^r, t_e^r$) can be transferred within $[t_a^r, t_e^r]$. We have two options to deal with this situation: 1) use more time to send the remaining data of the DO-R, or 2) mark the DO-R as incompletable and do not serve it. In this work, we consider both options and design two separate optimization objectives: 1) maximize the average data-transfer percentage of the DO-Rs, and 2) minimize the incompletable data transfers among the DO-Rs, respectively.

⁴Here, M does not include the initial RSA configuration for a DO-R.

IV. MILP FORMULATION

In this section, we formulate a link-based MILP model for offline optimization, where the network utilization after serving all the pending FO-Rs (i.e., the set of available FS blocks $\{C_{e,t}: e \in E, t \in [1, T]\}$) is obtained and all the pending DO-Rs are known. Note that the parameters and variables defined in Section III are directly used below.

Parameters:

- D : Set of DO-Rs.
- O_v : Set of links that originate from node $v \in V$.
- I_v : Set of links that end at node $v \in V$.
- V^r : Set of intermediate nodes that can be used for the r th DO-R, i.e., $V^r = V \setminus \{s^r, d^r\}$.
- T_m^r : Set of TSs during which the data transfer of the r th DO-R can be scheduled.

Variables:

- η^r : Real variable ranges within $[0, 1]$ to indicate the data-transfer percentage of the r th DO-R.
- ξ^r : Boolean variable that equals 1 if the data of the r th DO-R can be transferred completely, and 0 otherwise.
- $x_{e,t}^{i,r}$: Boolean variable that equals 1 if the i th FS block on e during t is assigned to the r th DO-R, and 0 otherwise.

Objectives:

$$\text{Maximize } \frac{1}{|D|} \sum_{r \in D} \eta^r, \quad (4)$$

or

$$\text{Minimize } 1 - \frac{1}{|D|} \sum_{r \in D} \xi^r, \quad (5)$$

where $|D|$ is the total number of pending DO-Rs. Depending on the options discussed in Subsection III.C, Eq. (4) aims to maximize the average data-transfer percentage of the DO-Rs, while Eq. (5) aims to minimize the incompletable data transfers among the DO-Rs.

Constraints:

$$\sum_{e \in I_{s^r}} \sum_{i \in C_{e,t}} x_{e,t}^{i,r} = 0, \quad \forall r, t \in T_m^r, \quad (6)$$

$$\sum_{e \in O_{d^r}} \sum_{i \in C_{e,t}} x_{e,t}^{i,r} \leq 1, \quad \forall r, t \in T_m^r, \quad (7)$$

$$\sum_{e \in I_{d^r}} \sum_{i \in C_{e,t}} x_{e,t}^{i,r} \leq 1, \quad \forall r, t \in T_m^r, \quad (8)$$

$$\sum_{e \in O_{d^r}} \sum_{i \in C_{e,t}} x_{e,t}^{i,r} = 0, \quad \forall r, t \in T_m^r, \quad (9)$$

$$\sum_{e \in I_v} \sum_{i \in C_{e,t}} x_{e,t}^{i,r} \leq 1, \quad \forall r, t \in T_m^r, v \in V^r, \quad (10)$$

$$\sum_{e \in O_v} \sum_{i \in C_{e,t}} x_{e,t}^{i,r} \leq 1, \quad \forall r, t \in T_m^r, v \in V^r. \quad (11)$$

Equations (6)–(11) ensure that the assigned FS block of each DO-R is on the fiber links that are in the right direction, i.e., $s^r \rightarrow d^r$, and one DO-R occupies at most one FS block during each TS.

$$\sum_{r \in D} x_{e,t}^{i,r} \leq 1, \quad \forall e, t, i \in C_{e,t}, \quad (12)$$

$$\sum_{e \in O_v} \sum_{i \in C_{e,t}} x_{e,t}^{i,r} \cdot \pi_{e,t}^i = \sum_{e \in I_v} \sum_{i \in C_{e,t}} x_{e,t}^{i,r} \cdot \pi_{e,t}^i, \quad \forall r, t \in T_m^r, v \in V^r, \quad (13)$$

$$\sum_{t \in T_m^r} \phi_t^r \leq M + 1, \quad \forall r. \quad (14)$$

Equations (12) and (13) ensure that the spectrum non-overlapping and continuity constraints are satisfied, while Eq. (14) ensures that the RSA reconfigurations of each DO-R are within M .

$$\begin{cases} \phi_t^r \geq x_{e,t}^{i,r}, & \pi_{e,t}^i \notin \{\pi_{e,t-1}^j : j \in C_{e,t-1}\}, \\ \phi_t^r \geq x_{e,t}^{i,r} - x_{e,t-1}^{j,r}, & \pi_{e,t}^i = \pi_{e,t-1}^j, \end{cases} \quad \forall r, e, t \in T_m^r \setminus \{t_a^r\}, i \in C_{e,t}, \quad (15)$$

$$\phi_t^r \geq x_{e,t}^{i,r}, \quad \forall r, e, t = t_a^r, i \in C_{e,t}, \quad (16)$$

$$\xi^r \cdot \mathcal{F}^r \leq \sum_{t \in T_m^r} \sum_{e \in O_{s^r}} \sum_{i \in C_{e,t}} x_{e,t}^{i,r} \cdot w_{e,t}^i, \quad \forall r, \quad (17)$$

$$\eta^r \leq \frac{\sum_{t \in T_m^r} \sum_{e \in O_{s^r}} \sum_{i \in C_{e,t}} x_{e,t}^{i,r} \cdot w_{e,t}^i}{\mathcal{F}^r}, \quad \forall r. \quad (18)$$

Equations (15) and (16) determine whether an RSA configuration/reconfiguration is needed for a DO-R, Eq. (17) indicates whether the DO-R has been served completely, and Eq. (18) limits the value of η^r .

V. DYNAMIC SERVICE PROVISIONING ALGORITHMS

Due to its complexity, the MILP can become intractable when the EON has a relatively large size and/or there are many DO-Rs to be served. Hence, in this section, we propose several online algorithms to handle the dynamic network scenario.

A. Dynamic Service Provisioning for Hybrid FO-Rs/DO-Rs

Algorithm 1 provides the overall procedure for achieving dynamic service provisioning of FO-Rs and DO-Rs. Lines 1–6 are for initialization. Lines 7–20 show the details of dynamic network operation. Specifically, at each TS $t_c \in$

[1, T], we first serve all the pending FO-Rs, and get the spectrum fragments $\{C_{e,t} : e \in E, t \in [t_c, T]\}$, as shown in Lines 9 and 10. Line 11 sorts the pending DO-Rs including both the old and new ones in ascending order of their deadlines, i.e., $\{t_c^r\}$. Here, the “old” DO-Rs refer to those that arrive at time instants earlier than t_c but with incomplete data transfers. The for-loop that covers Lines 12–18 performs RSA to find the routing path and available FS block at t_c for each DO-R. Note that, in Line 13, in order to find $\{\theta_{t_c}^r, p_{s^r, d^r, t_c}^r, [f_{s, t_c}^r, f_{e, t_c}^r]\}$ (i.e., the RSA scheme at t_c) for DO-R r , we propose two DO-R provisioning algorithms with different optimization objectives, which will be discussed later. Then, if a DO-R has been served completely or we find that it is incompletable, it is removed from the pending DO-R set for the next TS in Line 16.

Algorithm 1 Dynamic Service Provisioning for Hybrid FO-Rs/DO-Rs

```

1 Phase I: (Network Initialization)
2   for all  $s$ - $d$  pairs in  $G(V, E)$ ,  $s, d \in V$  do
3     calculate  $K$ -shortest routing path candidates;
4     store them in set  $\{p_{s,d}^k : k = 1, 2, \dots, K\}$ ;
5   end
6 End
7 Phase II: (Dynamic Network Operation)
8   for  $t_c \in [1, T]$  do
9     serve all FO-Rs arriving at  $t_c$ ;
10    get set  $\{C_{e,t} : e \in E, t \in [t_c, T]\}$ ;
11    sort pending DO-Rs, including both old and
12    new ones in ascending order of  $\{t_c^r\}$ ;
13    for each DO-R  $r$  do
14      find  $\{\theta_{t_c}^r, p_{s^r, d^r, t_c}^r, [f_{s, t_c}^r, f_{e, t_c}^r]\}$  at  $t_c$ ;
15      configure lightpath for data transfer;
16      if DO-R  $r$  is fully served or blocked then
17        remove DO-R  $r$  from pending DO-R set;
18      end
19    end
20 End

```

B. Minimum-Transferred-Data-Guaranteed Algorithm

For the situation in which the data transfer of a DO-R cannot be completed before the specified deadline, if we choose to extend the DO-R's deadline on data transfer and use more time to send the remaining data, we try to achieve the optimization objective in Eq. (4), i.e., maximize the average data-transfer percentage of the DO-Rs. For this scenario, we propose a minimum-transferred-data-guaranteed algorithm (MTDG). Note that MTDG does not address the problem of extending the data-transfer deadlines of DO-Rs directly. But as it maximizes the average data-transfer percentage of DO-Rs, we can treat incomplete DO-Rs as new ones and apply it repeatedly to finish the data transfers. We will study the data-transfer deadline extension problem further in future work.

Algorithm 2 MTDG Service Provisioning for DO-R r

Input: $t_e^r, t_c, \{C_{e,t}: e \in E, t \in [t_c, T]\}, N_{\min}^r, m^r, \mathcal{F}_u^r, \{\vartheta_{t_c-1}^r, p_{s^r, d^r, t_c-1}^r, [f_{s, t_c-1}^r, f_{e, t_c-1}^r]\}, \{p_{s^r, d^r}^k\};$
Output: $m^r, \mathcal{F}_u^r, \{\vartheta_{t_c}^r, p_{s^r, d^r, t_c}^r, [f_{s, t_c}^r, f_{e, t_c}^r]\};$

- 1 **if** $m^r < t_e^r - t_c + 1$ **then**
- 2 **if** $\vartheta_{t_c-1}^r = 1$ **and** $\{p_{s^r, d^r, t_c-1}^r, [f_{s, t_c-1}^r, f_{e, t_c-1}^r]\}$ **is still available at** t_c **then**
- 3 $\vartheta_{t_c}^r = 1, m^r = m^r;$
- 4 $p_{s^r, d^r, t_c}^r = p_{s^r, d^r, t_c-1}^r;$
- 5 $[f_{s, t_c}^r, f_{e, t_c}^r] = [f_{s, t_c-1}^r, f_{e, t_c-1}^r];$
- 6 $\mathcal{F}_u^r = \max(\mathcal{F}_u^r - (f_{e, t_c-1}^r - f_{s, t_c-1}^r + 1), 0);$
- 7 **else**
- 8 **if** $m^r > 0$ **then**
- 9 find the largest available FS block over all the path candidates with Eq. (20);
- 10 **if** the FS block's size is larger than $\min(N_{\min}^r, \mathcal{F}_u^r)$ **then**
- 11 tailor a just-enough FS block with first fit and use it for the data transfer;
- 12 set $\{p_{s^r, d^r, t_c}^r, [f_{s, t_c}^r, f_{e, t_c}^r]\}$ accordingly;
- 13 $\vartheta_{t_c}^r = 1, m^r = m^r - 1;$
- 14 $\mathcal{F}_u^r = \mathcal{F}_u^r - (f_{e, t_c}^r - f_{s, t_c}^r + 1);$
- 15 **else**
- 16 pause the data transfer;
- 17 $\vartheta_{t_c}^r = 0, m^r = m^r, \mathcal{F}_u^r = \mathcal{F}_u^r;$
- 18 **end**
- 19 **else**
- 20 **if** $\mathcal{F}_u^r > 0$ **then**
- 21 $\vartheta_{t_c}^r = 0, m^r = m^r, \mathcal{F}_u^r = \mathcal{F}_u^r;$
- 22 report DO-R r as incomplete;
- 23 **return;**
- 24 **end**
- 25 **end**
- 26 **end**
- 27 **else**
- 28 choose the largest available FS block over all the path candidates with Eq. (20);
- 29 tailor a just-enough FS block with first fit and use it for the data transfer;
- 30 set $\{p_{s^r, d^r, t_c}^r, [f_{s, t_c}^r, f_{e, t_c}^r]\}$ accordingly;
- 31 $\vartheta_{t_c}^r = 1, m^r = m^r - 1;$
- 32 $\mathcal{F}_u^r = \mathcal{F}_u^r - (f_{e, t_c}^r - f_{s, t_c}^r + 1);$
- 33 **end**
- 34 **if** $\mathcal{F}_u^r = 0$ **then**
- 35 report the DO-R as served completely;
- 36 **return;**
- 37 **else**
- 38 **if** $t_c = t_e^r$ **then**
- 39 report the DO-R as incomplete;
- 40 **return;**
- 41 **end**
- 42 **end**

Basically, for each DO-R r , we first define a threshold on the minimum assigned FSs per TS as

$$N_{\min}^r = \lceil \gamma \cdot \frac{\mathcal{F}_u^r}{t_e^r - t_a^r + 1} \rceil, \quad (19)$$

where γ is the control factor ranging within $[0, 1]$. Then, we use Algorithm 2 to find the RSA scheme of each DO-R at t_c ,

i.e., $\{\vartheta_{t_c}^r, p_{s^r, d^r, t_c}^r, [f_{s, t_c}^r, f_{e, t_c}^r]\}$. Here, for DO-R r , we define its remaining RSA reconfiguration times as m^r ($m^r \in [0, M]$), and the size of uncompleted data transfer as \mathcal{F}_u^r . Note that the largest available FS block over all the path candidates at t_c is

$$\max \left(\left\{ \bigcap_{e \in p_{s^r, d^r}^k} C_{e, t_c} : k = 1, 2, \dots, K \right\} \right). \quad (20)$$

At the beginning of t_c , if we have $m^r < t_e^r - t_c + 1$ for DO-R r (i.e., the number of remaining RSA reconfigurations is less than the remaining TSs for data transfer), we first check whether its RSA at $t_c - 1$ is still available. If yes, Lines 3–6 set $\{\vartheta_{t_c}^r, p_{s^r, d^r, t_c}^r, [f_{s, t_c}^r, f_{e, t_c}^r]\}$ (i.e., the RSA scheme at t_c) as $\{\vartheta_{t_c-1}^r, p_{s^r, d^r, t_c-1}^r, [f_{s, t_c-1}^r, f_{e, t_c-1}^r]\}$ (i.e., the RSA scheme at $t_c - 1$) to avoid unnecessary reconfiguration. Otherwise, in the case of $m^r > 0$ (i.e., there are remaining RSA reconfigurations), we find the largest available FS block with Eq. (20), and check whether the FS block's size can reach up to $\min(N_{\min}^r, \mathcal{F}_u^r)$, as shown in Line 9. If yes, Lines 11–14 conduct an RSA reconfiguration to use the new RSA scheme. Otherwise, Lines 16–17 pause the data transfer of DO-R r , i.e., have $\vartheta_{t_c}^r = 0$. On the other hand, if $m^r \geq t_e^r - t_c + 1$, which means that the constraint on the RSA reconfiguration times is not an issue anymore, we just choose the largest available FS block over all the path candidates, and tailor a just-enough RSA scheme for the DO-R, as shown in Lines 28–32. Finally, if \mathcal{F}_u^r (i.e., the size of uncompleted data transfer) becomes 0, Lines 35–36 report DO-R r as served completely. Otherwise, we report DO-R r as incomplete, if there is no remaining RSA reconfiguration (i.e., $m^r = 0$) or the deadline has arrived (i.e., $t_c = t_e^r$), as shown in Lines 21–23 and 39–40.

C. Blocking-Aware RSA Algorithm With Admission Control

On the other hand, if we choose to mark a DO-R as incompletable and do not serve it when it cannot be completed before the specified deadline, we need to consider the optimization objective in Eq. (5), i.e., minimize the incompletable data transfers among the DO-Rs. For this scenario, we propose a blocking-aware RSA algorithm with admission control (AC + BA). Note that the dynamic programming method (DPM) proposed in our previous work [10] can calculate the exact maximum amount of data $\varpi_{t_s, t_e, m}$ to be transferred during period $[t_s, t_e]$ in polynomial time, when the remaining RSA reconfiguration times m and the network utilization $\{C_{e,t}: e \in E, t \in [t_s, t_e]\}$ are known. Hence, we use the DPM in both admission control and blocking-aware RSA.

1) *Admission Control*: Algorithm 3 shows the detailed procedure of admission control (AC). Here, set $\{\varpi_{t_c, t_e, m^r}: t \in [t_c, t_e]\}$ is precalculated with DPM as an input, and set Λ stores all the possible values for the amount of data that can be transferred during period $[t_c, t_e]$. Line 1 stores ϖ_{t_c, t_e, m^r} in Λ , the value of which depends on the remaining RSA reconfiguration times m^r and the spectrum utilization

$\{C_{e,t}: e \in E, t \in [t_s, t'_e]\}$. Lines 2–16 consider the extra RSA schemes $\{p^{s,d^r,t_c-1}, [f_{s,t_c-1}^r, f_{e,t_c-1}^r]\}$ that are available at t_c and later TSs, as they do not need an RSA reconfiguration but still can contribute to the data transfer. The possible values for the amount of data that can be transferred with them are also stored in Λ , and their number depends on the availability of the RSA scheme $\{p^{s,d^r,t_c-1}, [f_{s,t_c-1}^r, f_{e,t_c-1}^r]\}$ along the time axis, as shown in Lines 3–15. Finally, if we have $\max(\Lambda) \geq \mathcal{F}_u^r$, Line 18 accepts the DO-R; otherwise, it is marked as incompletable at t_c , as shown in Line 20.

Algorithm 3 Admission Control for DO-Rs

Input: $t'_a, t'_e, t_c, \{\vartheta_{t_c-1}^r, p^{s,d^r,t_c-1}, [f_{s,t_c-1}^r, f_{e,t_c-1}^r]\}, \{\varpi_{t,t'_e,m^r}: t \in [t_c, t'_e]\}, \mathcal{F}_u^r$;

- 1 $\Lambda = \{\varpi_{t_c,t'_e,m^r}\}$;
- 2 **if** $t_c > t'_a$ **then**
- 3 **for** $t \in [t_c, t'_e]$ **do**
- 4 **if** $\vartheta_{t_c-1}^r = 1$, and $\{p^{s,d^r,t_c-1}, [f_{s,t_c-1}^r, f_{e,t_c-1}^r]\}$ is still available at t **then**
- 5 $t' = t + 1$;
- 6 $\omega = (f_{e,t_c-1}^r - f_{s,t_c-1}^r + 1) \cdot (t' - t_c)$;
- 7 **if** $t' \leq t'_e$ **then**
- 8 $\Lambda \leftarrow \varpi_{t',t'_e,m^r} + \omega$;
- 9 **else**
- 10 $\Lambda \leftarrow \omega$;
- 11 **end**
- 12 **else**
- 13 **break**;
- 14 **end**
- 15 **end**
- 16 **end**
- 17 **if** $\max(\Lambda) \geq \mathcal{F}_u^r$ **then**
- 18 report the DO-R as accepted;
- 19 **else**
- 20 report the DO-R as incompletable;
- 21 **end**

2) *Blocking-Aware RSA*: Once DO-R r is accepted, we need to find its RSA $\{\vartheta_{t_c}^r, p^{s,d^r,t_c}, [f_{s,t_c}^r, f_{e,t_c}^r]\}$. Here, we define the maximum redundancy ratio as the ratio of the maximum amount of data that can be transferred in the future to the amount of remaining data, and use it as the metric to evaluate the goodness of an RSA candidate. We denote the i th RSA candidate for DO-R r at t_c as $\{\vartheta_{t_c,i}^r, p^{s,d^r,t_c,i}, [f_{s,t_c,i}^r, f_{e,t_c,i}^r]\}$, its remaining RSA reconfigurations as m_i^r , the size of its remaining data as $\mathcal{F}_{u,i}^r$, and its maximum redundancy ratio as Δ_i^r . Note that m_i^r equals $m^r - 1$ if one RSA reconfiguration is needed, and m^r otherwise. Meanwhile, $\mathcal{F}_{u,i}^r$ equals $\max(\mathcal{F}_u^r - (f_{e,t_c,i}^r - f_{s,t_c,i}^r + 1), 0)$ if $\vartheta_{t_c,i}^r = 1$ (i.e., there is data transfer), and \mathcal{F}_u^r otherwise.

Algorithm 4 explains how to calculate Δ_i^r (i.e., the maximum redundancy ratio of the i th RSA candidate for DO-R r at t_c), where Λ stores all the possible values for the amount of data that can be transferred during period $[t_c + 1, t'_e]$. In the case in which $t_c < t'_e$, the calculation of

Λ shown in Lines 3–16 is similar to that in Algorithm 3. Otherwise, if $t_c = t'_e$, the data transfer will be terminated at the end of t_c , and thus the maximum amount of data that can be transferred in the future is 0, as shown in Line 1. Finally, Line 18 calculates Δ_i^r as

$$\Delta_i^r = \frac{\max(\Lambda)}{\mathcal{F}_{u,i}^r}. \quad (21)$$

Note that $\Delta_i^r < 1$ means that DO-R r cannot finish the data transfer with the i th RSA candidate, while $\Delta_i^r = \infty$ indicates that the remaining data of DO-R r can be transferred completely at t_c with the i th RSA candidate (i.e., $\mathcal{F}_{u,i}^r = 0$). Hence, to minimize the blocking probability of DO-R r , blocking-aware RSA (BA) aims to find the RSA candidate that has the largest Δ_i^r .

Algorithm 4 Calculating Maximum Redundancy Ratio

Input: $t'_e, t_c, \{\vartheta_{t_c,i}^r, p^{s,d^r,t_c,i}, [f_{s,t_c,i}^r, f_{e,t_c,i}^r]\}, \mathcal{F}_{u,i}^r, \{\varpi_{t,t'_e,m_i^r}: t \in [t_c, t'_e]\}$;

Output: Δ_i^r ;

- 1 $\Lambda = \{0\}$;
- 2 **if** $t_c < t'_e$ **then**
- 3 $\Lambda \leftarrow \varpi_{t_c+1,t'_e,m_i^r}$;
- 4 **for** $t \in [t_c + 1, t'_e]$ **do**
- 5 **if** $\vartheta_{t_c,i}^r = 1$, and $\{p^{s,d^r,t_c,i}, [f_{s,t_c,i}^r, f_{e,t_c,i}^r]\}$ is still available at t **then**
- 6 $t' = t + 1$;
- 7 $\omega = (f_{e,t_c,i}^r - f_{s,t_c,i}^r + 1) \cdot (t' - t_c)$;
- 8 **if** $t' \leq t'_e$ **then**
- 9 $\Lambda \leftarrow \varpi_{t',t'_e,m_i^r} + \omega$;
- 10 **else**
- 11 $\Lambda \leftarrow \omega$;
- 12 **end**
- 13 **else**
- 14 **break**;
- 15 **end**
- 16 **end**
- 17 **end**
- 18 $\Delta_i^r = \frac{\max(\Lambda)}{\mathcal{F}_{u,i}^r}$;

Algorithm 5 shows the detailed procedure of BA. To obtain the RSA candidates for DO-R r at t_c , we consider three scenarios one-by-one. The first scenario (i.e., Lines 2–16) tries to use the RSA scheme at $t_c - 1$, i.e., $\{\vartheta_{t_c-1}^r, p^{s,d^r,t_c-1}, [f_{s,t_c-1}^r, f_{e,t_c-1}^r]\}$, to avoid RSA reconfiguration. If a feasible RSA solution can be obtained with this scenario, Lines 3–4 assign an index i to the RSA and store the remaining data after using it in $\mathcal{F}_{u,i}^r$. In the case in which $\mathcal{F}_{u,i}^r = 0$, which means that DO-R r can be transferred completely at t_c with the RSA scheme, we stop checking other scenarios, set $\{\vartheta_{t_c}^r, p^{s,d^r,t_c}, [f_{s,t_c}^r, f_{e,t_c}^r]\}$ as $\{\vartheta_{t_c-1}^r, p^{s,d^r,t_c-1}, [f_{s,t_c-1}^r, f_{e,t_c-1}^r]\}$, and report DO-R r as served completely, as shown in Lines 6–9. Otherwise, Lines 11–14 regard the RSA scheme as an RSA candidate $\{\vartheta_{t_c,i}^r, p^{s,d^r,t_c,i}, [f_{s,t_c,i}^r, f_{e,t_c,i}^r]\}$, and calculate its maximum redundancy ratio Δ_i^r with Algorithm 4.

Algorithm 5 Blocking-Aware RSA for DO-Rs

Input: t_c , $\{C_{e,t}: e \in E, t \in [t_c, T]\}$, t_a^r , t_e^r , m^r , $\{\vartheta_{t_c-1}^r \cdot p_{s^r, d^r}^r$, $d^r, t_c - 1, [f_{s,t_c-1}^r, f_{e,t_c-1}^r]\}$, \mathcal{F}^r , \mathcal{F}_u^r , $\{p_{s^r, d^r}^k\}$, $\{\varpi_{t_c, t_e^r, m^r}\}$, $\varpi_{t_c, t_e^r, m^r-1}: t \in [t_c, t_e^r]\}$;

Output: m^r , $\{\vartheta_{t_c}^r \cdot p_{s^r, d^r, t_c}^r, [f_{s,t_c}^r, f_{e,t_c}^r]\}$, \mathcal{F}_u^r ;

```

1  $i = 0$ ;
2 if  $t_c > t_a^r$ , and  $\vartheta_{t_c-1}^r = 1$ , and  $\{p_{s^r, d^r, t_c-1}^r, [f_{s,t_c-1}^r, f_{e,t_c-1}^r]\}$  is
   still available at  $t_c$  then
3    $i = i + 1$ ;
4    $\mathcal{F}_{u,i}^r = \max(\mathcal{F}_u^r - (f_{e,t_c-1}^r - f_{s,t_c-1}^r + 1), 0)$ ;
5   if  $\mathcal{F}_{u,i}^r = 0$  then
6      $\vartheta_{t_c}^r = 1$ ,  $p_{s^r, d^r, t_c}^r = p_{s^r, d^r, t_c-1}^r$ ;
7      $[f_{s,t_c}^r, f_{e,t_c}^r] = [f_{s,t_c-1}^r, f_{e,t_c-1}^r]$ ;
8     report DO-R  $r$  as served completely;
9     return;
10  else
11     $p_{s^r, d^r, t_c, i}^r = p_{s^r, d^r, t_c-1}^r$ ;
12     $[f_{s,t_c,i}^r, f_{e,t_c,i}^r] = [f_{s,t_c-1}^r, f_{e,t_c-1}^r]$ ;
13     $m_i^r = m^r$ ,  $\vartheta_{t_c,i}^r = 1$ ;
14    calculate  $\Delta_i^r$  with Algorithm 4;
15  end
16 end
17  $i = i + 1$ ;
18 set  $m_i^r$ ,  $\{\vartheta_{t_c,i}^r \cdot p_{s^r, d^r, t_c, i}^r, [f_{s,t_c,i}^r, f_{e,t_c,i}^r]\}$ ,  $\mathcal{F}_{u,i}^r$  according to the
   outputs of DPM when calculating  $\varpi_{t_c, t_e^r, m^r}$ ;
19 if  $\mathcal{F}_{u,i}^r = 0$  then
20    $\vartheta_{t_c}^r = 1$ ,  $p_{s^r, d^r, t_c}^r = p_{s^r, d^r, t_c, i}^r$ ;
21    $f_{s,t_c}^r = f_{s,t_c,i}^r$ ,  $f_{e,t_c}^r = f_{e,t_c,i}^r + \mathcal{F}_u^r - 1$ ;
22   report the DO-R as served completely;
23   return;
24 else
25   calculate  $\Delta_i^r$  with Algorithm 4;
26 end
27 if  $m^r > 0$  then
28   for each  $p_{s^r, d^r}^k$  in ascending order of hop count do
29     find all the available FS blocks on  $p_{s^r, d^r}^k$  and store
       them in set  $\{[f_{s,t_c,n}^k, f_{e,t_c,n}^k]\}$ ;
30     for each  $[f_{s,t_c,n}^k, f_{e,t_c,n}^k]$  do
31        $i = i + 1$ ;
32        $\mathcal{F}_{u,i}^r = \max(\mathcal{F}_u^r - (f_{e,t_c-1}^r - f_{s,t_c-1}^r + 1), 0)$ ;
33       if  $\mathcal{F}_{u,i}^r = 0$  then
34          $\vartheta_{t_c}^r = 1$ ,  $p_{s^r, d^r, t_c}^r = p_{s^r, d^r}^k$ ;
35          $f_{s,t_c}^r = f_{s,t_c,n}^k$ ,  $f_{e,t_c}^r = f_{s,t_c}^r + \mathcal{F}_u^r - 1$ ;
36         report the DO-R as served completely;
37         return;
38       else
39          $p_{s^r, d^r, t_c, i}^r = p_{s^r, d^r}^k$ ;
40          $[f_{s,t_c,i}^r, f_{e,t_c,i}^r] = [f_{s,t_c,n}^k, f_{e,t_c,n}^k]$ ;
41          $m_i^r = m^r - 1$ ,  $\vartheta_{t_c,i}^r = 1$ ;
42         calculate  $\Delta_i^r$  with Algorithm 4;
43       end
44     end
45   end
46 end
47 select the RSA candidate with the largest  $\Delta_i^r$ ;
48 set  $m^r$ ,  $\{\vartheta_{t_c}^r \cdot p_{s^r, d^r, t_c}^r, [f_{s,t_c}^r, f_{e,t_c}^r]\}$ ,  $\mathcal{F}_u^r$  accordingly;

```

The second scenario (i.e., Lines 17–26) tries to use the RSA scheme obtained with DPM when calculating ϖ_{t_c, t_e^r, m^r} (i.e., the maximum amount of data that can be transferred

during $[t_c, t_e^r]$ with m^r times of RSA reconfiguration). Basically, at t_c , we either invoke an RSA reconfiguration to continue the data transfer, or just pause it. Then, if the RSA scheme achieves $\mathcal{F}_{u,i}^r = 0$ (i.e., the remaining data of DO-R r can be transferred completely), we stop checking the third scenario, set $\{\vartheta_{t_c}^r \cdot p_{s^r, d^r, t_c}^r, [f_{s,t_c}^r, f_{e,t_c}^r]\}$ accordingly, and report the DO-R r as served completely, as shown in Lines 20–23. Otherwise, we regard the RSA scheme as an RSA candidate and calculate its maximum redundancy ratio in Line 25.

The third scenario (i.e., Lines 27–46) tries to find a new RSA scheme, which can continue the data transfer with an RSA reconfiguration, and this scenario is only considered when there are remaining RSA reconfigurations (i.e., $m^r > 0$). The for-loop that covers Lines 28–45 checks all the path candidates in $\{p_{s^r, d^r}^k\}$ to find feasible RSA candidates. For each p_{s^r, d^r}^k , Line 29 finds all the available FS blocks on p_{s^r, d^r}^k at t_c , each of which can be regarded as an RSA candidate. Then, the for-loop covering Lines 30–43 evaluates each FS block. For an available FS block $[f_{s,t_c,n}^k, f_{e,t_c,n}^k]$ on p_{s^r, d^r}^k , if it can accommodate the remaining data of DO-R r completely, i.e., $f_{e,t_c,n}^k - f_{s,t_c,n}^k + 1 \geq \mathcal{F}_u^r$, we stop checking the remaining RSA candidates, set $\{\vartheta_{t_c}^r \cdot p_{s^r, d^r, t_c}^r, [f_{s,t_c}^r, f_{e,t_c}^r]\}$ as $\{1, p_{s^r, d^r}^k, [f_{s,t_c,n}^k, f_{e,t_c,n}^k + \mathcal{F}_u^r - 1]\}$, and report DO-R r as served completely, as shown in Lines 34–37. Otherwise, Lines 39–42 regard it as an RSA candidate, assign an index i to it, and calculate its maximum redundancy ratio.

Finally, Lines 47–48 select the RSA candidate with the largest Δ_i^r to minimize the blocking probability of DO-R r , and set $\{\vartheta_{t_c}^r \cdot p_{s^r, d^r, t_c}^r, [f_{s,t_c}^r, f_{e,t_c}^r]\}$ accordingly.

D. Complexity Analysis

To serve each DO-R r , the time complexity of MTDG in Algorithm 2 is $O(K \cdot B \cdot |E| \cdot (t_e^r - t_a^r + 1))$, while the time complexity of the AC + BA algorithm is $O(K \cdot B \cdot |E| \cdot (t_e^r - t_a^r + 1) + (t_e^r - t_a^r + 1)^4)$, where the complexities of Algorithm 3 and Algorithm 4 are both $O(t_e^r - t_a^r + 1)$, and the time complexity of Algorithm 5 is $O(K \cdot B \cdot |E| \cdot (t_e^r - t_a^r + 1))$ with set $\{\varpi_{t_c, t_e^r, m^r}, \varpi_{t_c, t_e^r, m^r-1}: t \in [t_a^r, t_e^r]\}$ being precalculated by the DPM, whose time complexity is $O((t_e^r - t_a^r + 1)^4)$ [10].

VI. PERFORMANCE EVALUATION

In this section, we conduct numerical simulations to evaluate the proposed algorithms, which run on a computer with 2.60 GHz Intel Core i5-4300M CPU and 8 GB RAM. We use GLPK to solve the MILP and use MATLAB R2013a to implement MTDG and AC + BA algorithms.

A. Simulations With Dynamic Network Scenario

1) *Simulation Parameters:* We use the NSFNET in Fig. 2 as the EON's physical topology, which consists of 14 nodes and 44 directional fiber links. Each fiber link is assumed to accommodate 358 FSs, each of which supports

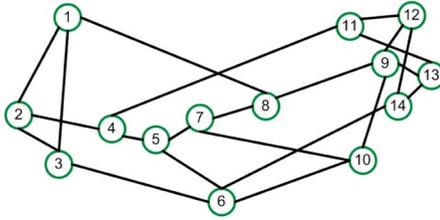


Fig. 2. NSFNET topology in dynamic network scenario.

12.5 Gbps.⁵ The number of look-ahead TSs that can be observed by the network operator is $T = 150$ TSs. The FO-Rs are generated according to a Poisson process; i.e., they arrive with an average rate of λ_f requests per TS and their durations follow the negative exponential distribution with an average of $\frac{1}{\mu_f}$ TSs. Then, their traffic load can be quantified with $\frac{\lambda_f}{\mu_f}$ in Erlangs. Note that we apply the IR-and-AR-mixed traffic model to the FO-Rs and use the specific starting time and specific duration (STSD) scheme for AR-based FO-Rs. Hence, there will be 2D spectrum fragments in the EON. The book-ahead time of the FO-Rs varies within $[0, 20]$ TSs. When its book-ahead time is 0, the FO-R is IR; otherwise, it is AR. We serve the IR-and-AR-mixed FO-Rs with the shortest-path routing and first-fit spectrum assignment algorithm (SPR-FFSA) [7].

The DO-Rs are also generated according to a Poisson process, and their traffic load is quantified as $\frac{\lambda_d}{\mu_d}$ in Erlangs, where λ_d and μ_d are the corresponding statistical parameters. For each DO-R r , its s^r - d^r pair is randomly selected, its data size \mathcal{F}^r is uniformly distributed with $[10, 100]$ FSs, and its service duration, i.e., the difference between t_a^r and t_e^r , follows the negative exponential distribution with an average of 10 TSs. Note that the maximum RSA reconfiguration time M is an important parameter that can affect the DO-R's incompleteness ratio. Therefore, we first focus on it and investigate its effect on the DO-Rs' incompleteness ratio. The maximum RSA reconfiguration times M range from 0 to 5. Table I summarizes the key simulation parameters.

2) *Simulation Results:* We first compare the performance of the MTDG and AC + BA algorithms in terms of several metrics under different maximum RSA reconfiguration times M . The traffic load of FO-Rs $\frac{\lambda_f}{\mu_f}$ is set as 300 Erlangs. In MTDG, both $\gamma = 0$ and $\gamma = 0.6$ are tested. Note that, for each data point, we run five independent simulations and average the results, while each simulation serves enough FO-Rs and DO-Rs to ensure that all the measurements can provide sufficient statistical accuracy.

Figure 3(a) shows the DO-Rs' incompleteness ratio. Here, the incompleteness ratio is defined as the ratio of the DO-Rs that experience incomplete data transfers to the total incoming DO-Rs. We notice that, when M increases from 0 to 5, the DO-R's incompleteness ratios from all the algorithms are reduced significantly. Moreover, it can be seen that AC + BA achieves a lower incompleteness ratio than MTDG when

⁵Supposing C-band is deployed in the network, each fiber link has ~ 4.475 THz bandwidth to allocate.

TABLE I
SIMULATION PARAMETERS FOR DYNAMIC NETWORK OPERATIONS

T , Number of look-ahead TSs observed by operator	150
B , Number of FSs on each fiber link	358
$\frac{\lambda_f}{\mu_f}$, Flow-oriented traffic load	[300, 750] Erlangs
$\frac{\lambda_d}{\mu_d}$, Data-oriented traffic load	120 Erlangs
Average service duration of DO-Rs	10 TSs
\mathcal{F}^r , Data size of a DO-R	[10, 100] FSs
M , Maximum RSA reconfiguration times	[0, 5]
K , Number of alternative path candidates	5

$M \leq 3$, which verifies the effectiveness of the proposed AC scheme. Specifically, the AC scheme rejects those DO-Rs that are incompletable before even serving them and leaves more spectrum resources for other DO-Rs. But, when $M \geq 4$, we observe the opposite results. This is because, for these cases, the incompleteness ratios are around 10^{-3} , which means that the resource competitions among DO-Rs become much less. This makes the shortage of remaining RSA reconfiguration times the main reason for a DO-R to be incomplete. Since the two MTDG schemes try to avoid unnecessary reconfigurations as much as possible when serving a DO-R, they can serve more DO-Rs in full. These observations suggest that AC + BA works effectively, especially when the DO-R's incompleteness ratio is relatively high in a network. On the other hand, when comparing the two MTDG schemes, we find that when $\gamma = 0.6$, the algorithm performs better than that with $\gamma = 0$ for most cases except for $M = 0$. This is because when γ is larger, MTDG tends to choose larger spectrum fragments in the future TSs, and thus the probability for MTDG to get enough FSs to complete a bulk-data transfer is higher. Note that, in the case of $M = 0$, as no RSA reconfiguration is allowed during a data transfer, "MTDG, $\gamma = 0$ " may in turn complete more bulk-data transfer due to the lowest threshold on the minimum assigned FSs per TS.

Figure 3(b) shows the average percentage of transferred data among all the DO-Rs. Again, we observe that, when M increases from 0 to 5, the results achieved by all the algorithms increase significantly. It is interesting to note that the two MTDG schemes achieve higher average percentages than AC + BA, especially when $M = 0, 1$. This is because AC + BA can purposely reject certain DO-Rs and make $\eta^r = 0$ for them, which brings down the average percentage. When comparing the two MTDG schemes, we can see that "MTDG, $\gamma = 0.6$ " achieves higher average η^r than "MTDG, $\gamma = 0$ " for most cases except for $M = 0$. This observation verifies that, with a larger γ , MTDG can balance the cost on RSA reconfigurations and the data-transfer amount better, and thus achieve higher values on η^r . Again, in the case of $M = 0$, due to the same reasons explained for Fig. 3(a), "MTDG, $\gamma = 0$ " achieves a higher average percentage.

Figure 3(c) shows the average RSA reconfiguration times when provisioning each DO-R. As expected, the average RSA reconfiguration times increase with M . It is interesting to find that "MTDG, $\gamma = 0.6$ " requires fewer RSA reconfigurations than "MTDG, $\gamma = 0$," which verifies that, with a

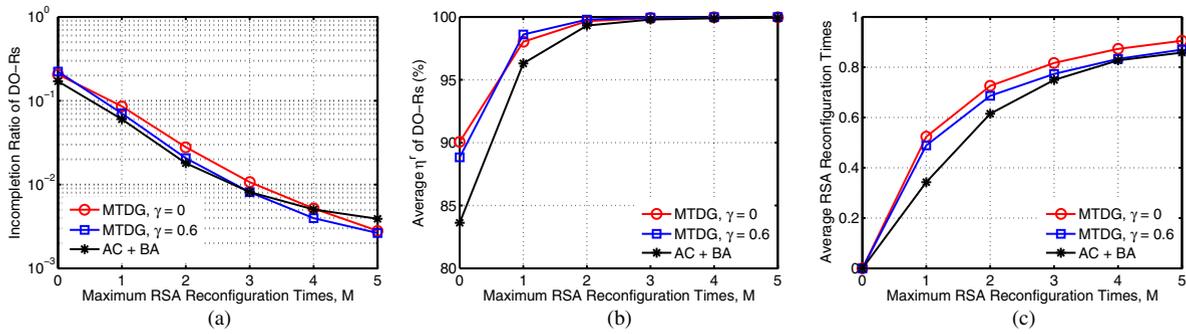


Fig. 3. Effect of maximum RSA reconfiguration times M on the provisioning of DO-Rs.

larger γ , MTDG can balance the cost on RSA reconfigurations and the data-transfer amount better again. Moreover, we note that AC + BA requires fewer RSA reconfigurations than the two MTDG schemes for all the cases. This is because AC + BA can purposely drop those DO-Rs that are incompletable to avoid unnecessary FS utilization and RSA reconfigurations.

We then investigate the effect of the provisioning of DO-Rs on that of FO-Rs. In this case, the FO-Rs' traffic load ranges from 300 to 750 Erlangs, and the maximum RSA reconfiguration times M is set as 5 for each DO-R. Figure 4(a) shows the blocking ratio of FO-Rs, when the DO-R traffic load $\frac{\lambda_d}{\mu_d}$ is 0 and 120 Erlangs. We can see that the blocking ratio of FO-Rs stays unchanged when the DO-R traffic load increases. This observation verifies that the proposed algorithms can effectively revitalize the spectrum

fragments generated by FO-Rs for bulk-data transfer while not affecting the provisioning of FO-Rs.

Figure 4(b) shows the average network spectrum utilization. We notice that, with the injection of DO-Rs, all the proposed algorithms can improve the network spectrum utilization significantly. This again confirms that the spectrum fragments are effectively recycled. Moreover, AC + BA achieves the lowest spectrum utilization, even though it can serve the most DO-Rs completely as shown in Fig. 3(a). This observation suggests that AC + BA can achieve the highest spectrum efficiency by only focusing on the DO-Rs that can be served completely while purposely dropping the others. "MTDG, $\gamma = 0.6$ " achieves lower average spectrum utilizations than "MTDG, $\gamma = 0$." Together with its lower incompletion ratio in Fig. 3(a) and higher average η' in Fig. 3(b), this observation suggests that "MTDG, $\gamma = 0.6$ " can achieve higher spectrum efficiency than "MTDG, $\gamma = 0$."

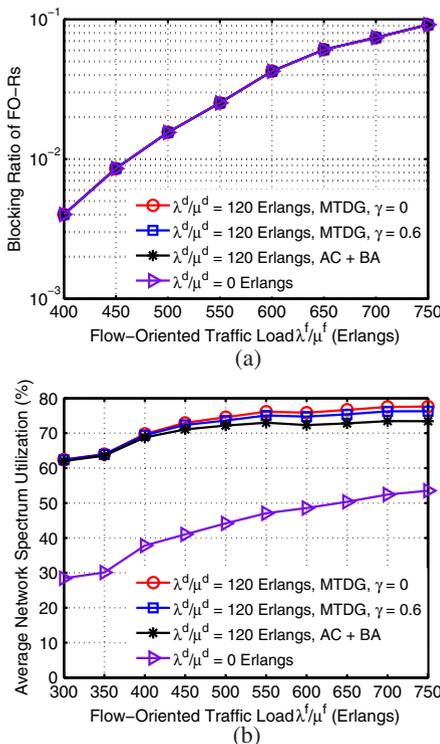


Fig. 4. Effect of the provisioning of DO-Rs on that of FO-Rs.

B. Simulations With Static Network Scenario

We also perform simulations with the static network scenario to compare the performance of MILP and proposed heuristics. To obtain the optimum solutions within a limited amount of time, we use the four-node topology in Fig. 5 and assume that each fiber link accommodates five FSs. The FO-Rs are generated similarly as for the dynamic network operations, but are served with SPR-FFSA in advance to generate the time-varying network spectrum utilization over the period of [1, 30] TSs. For each DO-R r , its s^r - d^r pair is randomly chosen, the arrival time t_a^r is uniformly distributed with [15, 21] TSs, and the service duration is uniformly distributed within [3, 5] TSs. The maximum RSA reconfiguration time is set as $M = 1$. Table II shows the results for $|D| = 1, 2, 3, 4$. We can see

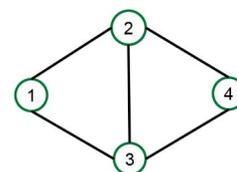


Fig. 5. Four-node topology in static network scenario.

TABLE II
RESULTS FROM SIMULATIONS FOR OFFLINE OPTIMIZATION

D	Incompletion Ratio of DO-Rs				Average Percentage η^T				Average Running Time (s)			
	MILP	AC + BA	MTDG		MILP	AC + BA	MTDG		MILP	AC + BA	MTDG	
			$\gamma = 0$	$\gamma = 0.6$			$\gamma = 0$	$\gamma = 0.6$			$\gamma = 0$	$\gamma = 0.6$
1	0.00%	0.00%	0.00%	100.00%	100.00%	100.00%	100.00%	72.14%	0.2467	0.0374	0.0128	0.0119
2	50.00%	50.00%	50.00%	50.00%	89.29%	50.00%	89.29%	75.00%	0.5461	0.0459	0.0143	0.0136
3	33.33%	33.33%	33.33%	33.33%	82.98%	66.67%	79.14%	77.57%	2.3383	0.0464	0.0163	0.0146
4	50.00%	50.00%	75.00%	100.00%	72.46%	50.00%	67.46%	60.55%	428.6296	0.0646	0.0171	0.0182

that AC + BA achieves the same incompletion ratios as the MILP, and MTDG with $\gamma = 0$ and the MILP provide similar results on average percentage η^T . Meanwhile, the average running time of the heuristics is significantly less than that of the MILP. Note that, in the case of $|D| = 1$, the 100% incompletion ratio from “MTDG, $\gamma = 0.6$ ” means that the data transfer cannot be completed, even though there is only one DO-R, which can be confirmed by the 74.14% average percentage η^T .

VII. CONCLUSION

This paper investigated deadline-driven bulk-data transfers to revitalize spectrum fragments in EONs. We first formulated a MILP model for offline optimization, with two objectives: 1) maximize the average data-transfer percentage of the DO-Rs, or 2) minimize the incompletable data transfers among the DO-Rs. Then, we proposed two dynamic provisioning algorithms, each of which aims to optimize one of the above objectives. Simulation results verified that the proposed algorithms could effectively optimize DO-Rs’ data transfers according to their objectives, revitalize spectrum fragments in EONs, and improve network spectrum utilization significantly.

ACKNOWLEDGMENT

This work was supported in part by the NSFC Project 61371117, the Fundamental Research Funds for the Central Universities (WK2100060010), the Natural Science Research Project for Universities in Anhui (KJ2014ZD38), the Strategic Priority Research Program of the CAS (XDA06011202), and the UC Davis Networks Lab.

REFERENCES

- [1] O. Gerstel, M. Jinno, A. Lord, and S. Yoo, “Elastic optical networking: A new dawn for the optical layer?” *IEEE Commun. Mag.*, vol. 50, no. 2, pp. s12–s20, Feb. 2012.
- [2] P. Lu, L. Zhang, X. Liu, J. Yao, and Z. Zhu, “Highly-efficient data migration and backup for big data applications in elastic optical inter-datacenter networks,” *IEEE Netw.*, vol. 29, pp. 36–42, Sept./Oct. 2015.
- [3] A. Pages, J. Perello, and S. Spadaro, “Lightpath fragmentation for efficient spectrum utilization in dynamic elastic optical networks,” in *Proc. of Optical Network Design and Modeling (ONDM)*, Apr. 2012, pp. 1–6.
- [4] Y. Yin, H. Zhang, M. Zhang, M. Xia, Z. Zhu, S. Dahlfort, and S. 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.
- [5] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, “Elastic bandwidth allocation in flexible OFDM-based optical networks,” *J. Lightwave Technol.*, vol. 29, pp. 1354–1366, May 2011.
- [6] Z. Zhu, W. Lu, L. Zhang, and N. Ansari, “Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing,” *J. Lightwave Technol.*, vol. 31, pp. 15–22, Jan. 2013.
- [7] W. Lu and Z. Zhu, “Dynamic service provisioning of advance reservation requests in elastic optical networks,” *J. Lightwave Technol.*, vol. 31, pp. 1621–1627, May 2013.
- [8] S. Shen, W. Lu, X. Liu, L. Gong, and Z. Zhu, “Dynamic advance reservation multicast in data center networks over elastic optical infrastructure,” in *Proc. of European Conf. and Exhibition on Optical Communication (ECOC)*, Sept. 2013, pp. 1–3.
- [9] M. Zhang, W. Shi, L. Gong, W. Lu, and Z. Zhu, “Bandwidth defragmentation in dynamic elastic optical networks with minimum traffic disruptions,” in *Proc. of Int. Conf. on Communications (ICC)*, June 2013, pp. 3894–3898.
- [10] W. Lu and Z. Zhu, “Malleable reservation based bulk-data transfer to recycle spectrum fragments in elastic optical networks,” *J. Lightwave Technol.*, vol. 33, pp. 2078–2086, May 2015.
- [11] 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, pp. 13–25, Jan. 2013.
- [12] 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, pp. 26–38, Jan. 2013.
- [13] W. Lu, Z. Zhu, and B. Mukherjee, “Data-oriented malleable reservation to revitalize spectrum fragments in elastic optical networks,” in *Optical Fiber Communications Conf. (OFC)*, Mar. 2015, pp. 1–3.
- [14] C. Develder, M. De Leenheer, B. Dhoedt, M. Pickavet, D. Colle, F. De Turck, and P. Demeester, “Optical networks for grid and cloud computing applications,” *Proc. IEEE*, vol. 100, pp. 1149–1167, May 2012.
- [15] J. Yao, P. Lu, L. Gong, and Z. Zhu, “On fast and coordinated data backup in geo-distributed optical inter-datacenter networks,” *J. Lightwave Technol.*, vol. 33, pp. 934–942, July 2015.
- [16] L. Gong, W. Zhao, Y. Wen, and Z. Zhu, “Dynamic transparent virtual network embedding over elastic optical infrastructures,” in *Proc. of IEEE Int. Conf. on Communications (ICC)*, June 2013, pp. 3466–3470.
- [17] R. Guerin and A. Orda, “Networks with advance reservations: The routing perspective,” in *Proc. of INFOCOM*, Mar. 2000, pp. 118–127.

- [18] S. Ganguly, A. Sen, G. Xue, B. Hao, and B. Shen, "Optimal routing for fast transfer of bulk data files in time-varying networks," in *Proc. of Int. Conf. on Communications (ICC)*, June 2004, pp. 1182–1186.
- [19] Y. Lin and Q. Wu, "Path computation with variable bandwidth for bulk data transfer in high-performance networks," in *Proc. of INFOCOM Workshops*, Apr. 2009, pp. 1–6.
- [20] B. Chen and P. Primet, "Supporting bulk data transfers of high-end applications with guaranteed completion time," in *Proc. of Int. Conf. on Communications (ICC)*, June 2007, pp. 575–580.
- [21] B. Chen and P. Primet, "Scheduling deadline-constrained bulk data transfers to minimize network congestion," in *Proc. of Cluster Computing and the Grid (CCGRID)*, May 2007, pp. 410–417.
- [22] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter bulk transfers with netstitcher," in *Proc. of ACM SIGCOMM*, Aug. 2011, pp. 74–85.
- [23] N. Laoutaris, G. Smaragdakis, R. Stanojevic, P. Rodriguez, and R. Sundaram, "Delay-tolerant bulk data transfers on the Internet," *IEEE/ACM Trans. Netw.*, vol. 21, pp. 1852–1865, Dec. 2013.
- [24] K. Rajah, S. Ranka, and Y. Xia, "Advance reservations and scheduling for bulk transfers in research networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, pp. 1682–1697, Nov. 2009.
- [25] S. Naiksatam and S. Figueira, "Elastic reservations for efficient bandwidth utilization in LambdaGrids," *Future Gener. Comput. Syst.*, vol. 23, pp. 1–22, Jan. 2007.
- [26] D. Andrei, M. Tornatore, M. Batayneh, C. Martel, and B. Mukherjee, "Provisioning of deadline-driven requests with flexible transmission rates in WDM mesh networks," *IEEE/ACM Trans. Netw.*, vol. 18, pp. 353–366, Apr. 2010.
- [27] A. Patel and J. Jue, "Routing and scheduling for variable bandwidth advance reservation," *J. Opt. Commun. Netw.*, vol. 3, pp. 912–923, Dec. 2011.