

Fragmentation-aware service provisioning for advance reservation multicast in SD-EONs

Shengru Li,¹ Wei Lu,¹ Xiahe Liu,¹ and Zuqing Zhu^{1,*}

¹University of Science and Technology of China, Hefei, Anhui, 230027, China

[*zqzhu@ieee.org](mailto:zqzhu@ieee.org)

Abstract: In this paper, we study the service provisioning schemes for dynamic advance reservation (AR) multicast requests in elastic optical networks (EONs). We first propose several algorithms that can handle the service scheduling and routing and spectrum assignment (RSA) of AR multicast requests jointly, including an integrated two-dimensional fragmentation-aware RSA (2D-FMA) that can alleviate the 2D fragmentation caused by light-tree provisioning. Then, we leverage the idea of software-defined EONs (SD-EONs) that utilizes OpenFlow (OF) in the control plane to demonstrate and evaluate the proposed algorithms. Specifically, we build an SD-EON control plane testbed, implement the algorithms in it, and perform control plane experiments on dynamic AR multicast provisioning. The results indicate that 2D-FMA achieves the best blocking performance and provides the shortest average setup delay.

© 2015 Optical Society of America

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

References and links

1. O. Gerstel, M. Jinno, A. Lord, and B. Yoo, "Elastic optical networking: a new dawn for the optical layer?" *IEEE Commun. Mag.* **50**, S12–S20 (2012).
2. 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 Network*, in press (2015).
3. L. Sahasrabudde and B. Mukherjee, "Light trees: optical multicasting for improved performance in wavelength routed networks," *IEEE Commun. Mag.* **37**, 67–73 (1999).
4. R. Libeskind-Hadas and R. Melhem, "Multicast routing and wavelength assignment in multihop optical networks," *IEEE/ACM Trans. Netw.* **10**, 621–629 (2002).
5. Q. Wang and L. Chen, "Performance analysis of multicast traffic over spectrum elastic optical networks," in *Proc. of OFC 2012*, pp. 1–3 (2012).
6. L. Gong, X. Zhou, X. Liu, W. Zhao, W. Lu, and Z. Zhu, "Efficient resource allocation for all-optical multicasting over spectrum-sliced elastic optical networks," *J. Opt. Commun. Netw.* **5**, 836–847 (2013).
7. N. Charbonneau and V. Vokkarane, "Static routing and wavelength assignment for multicast advance reservation in all-optical wavelength-routed WDM networks," *IEEE/ACM Trans. Netw.* **20**, 1–14 (2012).
8. T. Entel, A. Gadkar, and V. Vokkarane, "Dynamic advance reservation multicast overlay for slotted optical WDM networks," in *Proc. of GLOBECOM 2012*, pp. 3007–3012 (2012).
9. 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.* **31**, 15–22 (2013).
10. W. Lu and Z. Zhu, "Dynamic service provisioning of advance reservation requests in elastic optical networks," *J. Lightw. Technol.* **31**, 1621–1627 (2013).
11. 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.* **61**, 2970–2978 (2013).
12. 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 ECOC 2013*, pp. 1–3 (2013).

13. L. Liu, R. Munoz, R. Casellas, T. Tsuritani, R. Martinez, and I. Morita, "OpenSlice: an OpenFlow-based control plane for spectrum sliced elastic optical path networks," *Opt. Express* **21**, 4194–4204 (2013).
14. R. Casellas, R. Martinez, R. Munoz, R. Vilalta, L. Liu, T. Tsuritani, and I. Morita, "Control and management of flexi-grid optical networks with an integrated stateful path computation element and OpenFlow controller," *J. Opt. Commun. Netw.* **5**, A57–A65 (2013).
15. W. Lu, S. Ma, C. Chen, X. Chen, and Z. Zhu, "Implementation and demonstration of revenue-driven provisioning for advance reservation requests in OpenFlow-controlled SD-EONs," *IEEE Commun. Lett.* **18**, 1727–1730 (2014).
16. Z. Zhu, X. Chen, C. Chen, S. Ma, M. Zhang, L. Liu, and B. Yoo, "OpenFlow-assisted online defragmentation in single-/multi-domain software-defined elastic optical networks," *J. Opt. Commun. Netw.* **6**, 901–910 (2015).
17. X. Liu, L. Gong, and Z. Zhu, "Design integrated RSA for multicast in elastic optical networks with a layered approach," in *Proc. of GLOBECOM 2013*, pp. 2346–2351 (2013).
18. Y. Yin, M. Zhang, Z. Zhu, and B. Yoo, "Fragmentation-aware routing, modulation and spectrum assignment algorithms in elastic optical networks," in *Proc. of OFC 2013*, pp. 1–3 (2013).
19. 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 ICC 2013*, pp. 3894–3898 (2013).
20. Wireshark [Online]. Available: <http://www.wireshark.org/>.

1. Introduction

Recent advances on elastic optical networks (EONs) have shown that with newly-developed optical transmission and switching technologies, people can simultaneously achieve over Tb/s super-channel capacity and flexible bandwidth adjustment with a fine granularity of 12.5 GHz or less [1]. Hence, EON can promote the integration of physical transmission and upper-layer applications, and provides backbone networks a promising physical-layer infrastructure [2].

In the Internet, a considerable proportion of applications, *e.g.*, e-Science, datacenter backup, *etc.* require multicast services. Previously, optical multicast in fixed-grid wavelength-division multiplexing (WDM) networks has already been investigated intensively [3, 4]. Wang *et al.* recently discussed how to serve multicast requests in EONs and designed two routing and spectrum assignment (RSA) algorithms in [5]. In [6], we also addressed optical multicast in EONs. Specifically, we formulated two integer linear programming (ILP) models for network planning and designed a time-efficient heuristic for network provisioning. Nevertheless, most of these previous studies only considered the multicast requests that tolerate zero setup-delay and need to be served immediately upon their arrivals.

Note that multicast services can also use the advance reservation (AR) scheme [7], in which requests allow certain setup-delay, as long as they are served before a preset deadline. For example, the scheduled one-to-many backup in datacenters is a typical AR multicast service. Previously, the authors of [7] studied AR multicast in WDM networks with a static traffic model, and they extended to consider dynamic traffic in [8]. However, AR multicast in EONs is still under-explored. EONs need more sophisticated network control and management (NC&M) mechanisms, since they have to manage blocks of spectrally-contiguous frequency slots (FS') instead of discrete wavelength channels [9]. Moreover, provisioning AR requests dynamically in EONs can cause two-dimensional (2D) fragmentation [10]. Here, fragmentation refers to the existing of non-aligned, isolated and small-sized FS-blocks in the optical spectra due to dynamic service provisioning [11]. Since the FS-blocks only have very small sizes, they can hardly be used to serve future requests that have relatively large bandwidth requirements.

In this work, we expand our previous work in [12] and study how to efficiently serve dynamic AR multicast requests in EONs with consideration of 2D fragmentation. We first develop a 2D-fragmentation-aware RSA (2D-FMA) algorithm for dynamic AR multicast provisioning, which handles the service scheduling and RSA of requests jointly to alleviate the 2D fragmentation caused by light-tree provisioning. Then, we leverage the idea of software-defined EONs (SD-EONs), which utilize OpenFlow (OF) in the control plane [13–16], to demonstrate and evaluate the proposal. Specifically, we build an SD-EON control plane testbed, implement the proposed

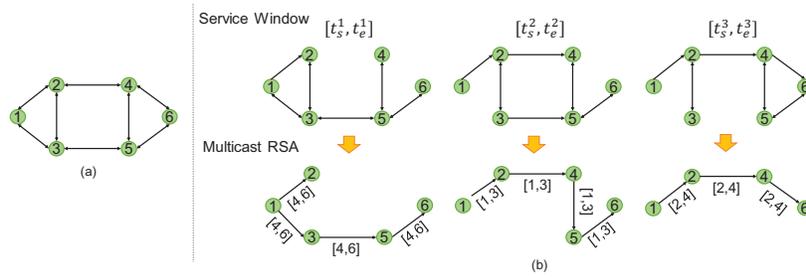


Fig. 1. (a) Six-node topology, and (b) Example on provisioning an AR multicast request.

algorithms in it, and perform NC&M experiments on dynamic AR multicast provisioning.

The rest of the paper is organized as follows. In Section 2, we formulate the problem of dynamic AR multicast provisioning in EONs. The 2D-FMA algorithm is proposed in Section 3. Section 4 shows the details on the SD-EON control plane testbed, and the experimental results are presented in Section 5. Finally, Section 6 summarizes the paper.

2. Problem formulation

We consider the physical EON topology as $G(V, E)$, where V is the set of nodes each of which includes a multicast-capable bandwidth-variable optical cross-connect (MC-BV-OXC) and E is the fiber link set. Here, we assume that similar to their counterparts in WDM networks [3], the MC-BV-OXCs can perform light-splitting to facilitate optical multicast. Each fiber link $e \in E$ accommodates F frequency slots (FS') whose capacity is $b_{slot} = 12.5$ Gb/s. In the context of this work, we consider optical multicast and assume that there is no spectrum converter in each light-tree. Therefore, the multicast service uses the same-spectrum scheme [6].

An AR multicast request is denoted as $MR(s, D, bw, t_a, d_{max}, t_h)$, where s is the source node, D is the set of destination nodes, bw is the bandwidth requirement in Gb/s, t_a is the arrival time, d_{max} is the maximum setup-delay, and t_h is the demanded service duration. For simplicity, we assume that the EON operates according to discrete time units. We define $[t_s, t_e]$ as the request's service window, where t_s and t_e are its service start- and end-time, respectively. Hence, we have $t_e = t_s + t_h$ and t_s can slide as $t_s \in [t_a, t_a + d_{max}]$. To provision MR , we need to determine its service window in the time domain and its RSA in the spectral domain as follows

- Service scheduling: We need to find a valid service window $[t_s, t_e]$, during which the network has enough spectrum resources for MR .
- Multicast routing: We need to construct a light-tree $\mathcal{T}_{s,D}$ that roots at s and covers all the destinations $d_i \in D, i = 1, 2, \dots, |D|$.
- Spectrum assignment: According to the spectrum utilization during $[t_s, t_e]$, we need to assign enough FS' on the fiber links in $\mathcal{T}_{s,D}$ to satisfy bw . Specifically, with bw and b_{slot} , the number of FS' to be assigned on each link in $\mathcal{T}_{s,D}$ is $n = \lceil \frac{bw}{b_{slot}} \rceil$.

Figure 1 shows an illustrative example on provisioning an AR multicast request with service scheduling and multicast RSA. The EON's topology is shown in Fig. 1(a), and the AR multicast request is $MR(Node 1, \{Node 2, Node 6\}, 3 FS', t_a, 2, t_h)$. As $d_{max} = 2$, MR can be provisioned in three service windows starting at $t_s^1 = t_a$, $t_s^2 = t_a + 1$, and $t_s^3 = t_a + 2$, respectively. To find a feasible RSA for MR , we check the FS usage on fiber links during each service window $[t_s^i, t_e^i]$. If a link does not have enough available FS', we cut it as in Fig. 1(b). Then, a light-tree is built

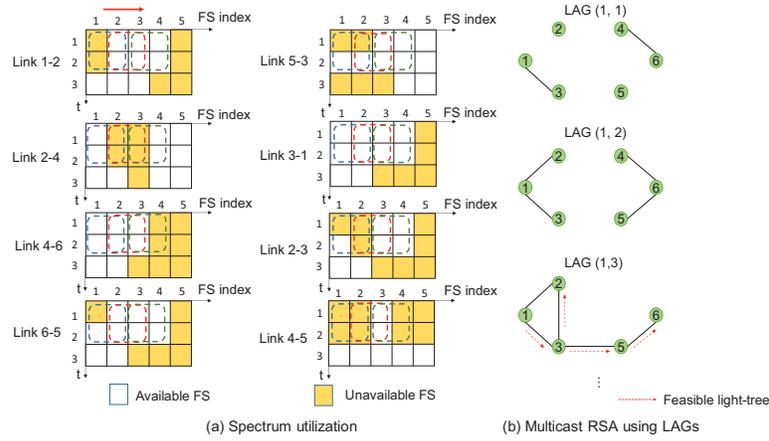


Fig. 2. Example on integrated multicast RSA using LAGs.

for each $[t_s^i, t_e^i]$ and we assign 3 FS' on it to accommodate MR . Figure 1(b) shows all the feasible provisioning schemes for MR . Here, the label $[4, 6]$ means that MR uses the FS-block from FS 4 to FS 6 on the links in the light-tree.

3. Dynamic AR multicast provisioning algorithms

3.1. Basic AR multicast provisioning

We first describe two basic service scheduling strategies to find service windows for AR multicast requests. They are referred as the least time to wait (LTW) and smallest FS start-index (SFSSI) strategies [10]. LTW aims to serving each request with the shortest setup delay. For instance, with the feasible provisioning schemes in Fig. 1(b), LTW will schedule the request to use the service window $[t_s^1, t_e^1]$. On the other hand, SFSSI schedules each request to use the service window, during which the RSA provides the smallest FS start-index. With the example in Fig. 1(b), SFSSI will instruct the request to use the service window $[t_s^2, t_e^2]$.

Then, the multicast RSA can be obtained with the integrated RSA using layered auxiliary graphs (LAGs), which is extended from the approach we developed in [17]. We represent the spectrum utilization on a link e with a boolean matrix \mathbf{B}_e of size $T \times F$, where $T = t_h + d_{max}$ and F is the number of FS' on e . Each element $b_{t,f}^e$ in \mathbf{B}_e is a boolean variable, where $b_{t,f}^e = 0$ if FS f on link e is used and unavailable at time t , and $b_{t,f}^e = 1$ otherwise. For a request $MR(s, D, bw, t_a, d_{max}, t_h)$, we first obtain n , i.e., the number of FS' to be assigned with bw and b_{slot} . Then, we utilize a boolean matrix $\mathbf{R} \in \{0, 1\}^{t_h \times n} = \mathbf{1}$ to obtain the provisioning scheme of MR .

Basically, we build a series of LAGs based on the topology $G(V, E)$ and the spectrum utilization on each link along the time axis. Each LAG is denoted as $G^{t,k}(V^{t,k}, E^{t,k})$, where t is the corresponding service start-time and k is the start-index of the corresponding FS-block. More specifically, in order to build $G^{t,k}(V^{t,k}, E^{t,k})$, we scan the spectrum usage on each link, and insert a direct link $e^{t,k}$ in $G^{t,k}(V^{t,k}, E^{t,k})$ if there exists an all-ones matrix $\mathbf{R} \in \{0, 1\}^{t_h \times n}$ in \mathbf{B}_e starting from $b_{t,k}^e$. Or in other words, we have $b_{\tau,f}^e = 1, \forall \tau \in [t, t + t_h - 1], \forall f \in [k, k + n - 1]$. Note that when searching \mathbf{B}_e for \mathbf{R} , if we want to apply LTW for service scheduling, we increase k first, otherwise, if we increase t first, SFSSI is applied. Finally, if we can obtain a light-tree in $G^{t,k}$ to cover all nodes in $\{s, D\}$, the AR request MR can be served with the light-tree in the original topology G with service start-time as t and FS-block as $[k, k + n - 1]$. Therefore, the algorithm gets the service window, light-tree and spectrum assignment for MR in one integrated step.

We refer to this basic AR multicast provisioning algorithm as I-RSA-LTW or I-RSA-SFSSI depending on the actual service scheduling strategy.

Figure 2 shows an example of the integrated multicast RSA using LAGs. We still use the six-node topology in Fig. 1(a), and the AR multicast request is $MR(Node\ 1, \{Node\ 2, Node\ 3, Node\ 6\}, 2\ FS', t_a = 1, d_{max} = 1, t_h = 2)$. Since $d_{max} = 1$, we only consider two feasible service windows. The spectrum utilization on each link is shown in Fig. 2(a), and the dashed rectangle indicates the FS' that MR will occupy in each LAG. Suppose we use I-RSA-LTW, three LAGs are first built as shown in Fig. 2(b). Since we can obtain a feasible light-tree in LAG (1,3) to cover $\{Node\ 1, Node\ 2, Node\ 3, Node\ 6\}$, MR is provisioned with the light-tree in Fig. 2(b) and the corresponding spectrum assignment in Fig. 2 (a).

3.2. 2D fragmentation-aware AR multicast provisioning

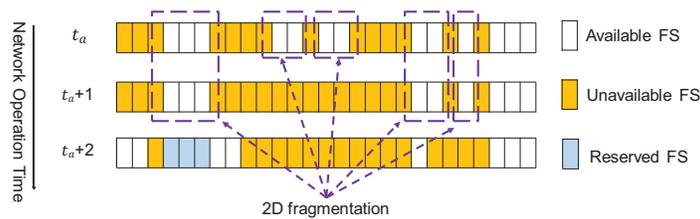


Fig. 3. Example for 2D fragmentation in an EON.

It is known that provisioning AR requests dynamically can cause 2D fragmentation in EONs [10]. Figure 3 illustrates an example on 2D fragmentation. Here, for simplicity, we only plot the spectrum usage on a link. It can be seen that since AR provisioning can reserve FS' from a future time instant, certain FS-blocks will become isolated in the time domain. As these FS-blocks will only be available for a relatively short period of time, they can hardly be used by other requests that ask for longer service durations and cause 2D fragmentation. In order to address this issue, we design an integrated 2D fragmentation-aware RSA (2D-FMA) algorithm for dynamic AR multicast provisioning.

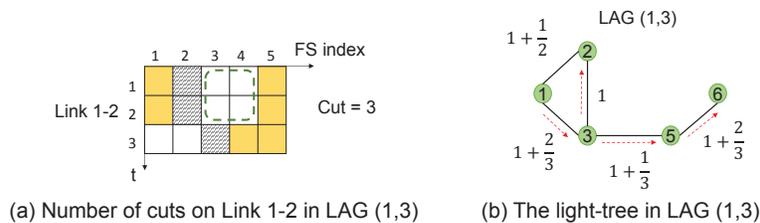


Fig. 4. Example of cut calculation and light-tree calculation

The proposed 2D-FMA considers the fragmentation in the time and spectral domains jointly. Basically, we leverage our idea in [18, 19] to measure the fragmentation that could be caused by provisioning an AR multicast request. Here, one spectral “cut” means that the AR multicast provisioning will break the contiguousness of an FS-block on a link in the time or spectral domain. We still use the case in Fig. 2 as the example. If we provision MR with the scheme obtained in LAG (1,3), the number of cuts caused on *Link* 1-2 is three as shown in Fig. 4(a). Then, the weight of each link in the LAG is calculated as $1 + \frac{c}{c_{max}}$, where c is the number of

cuts caused on the link and c_{max} is the maximum number of cuts that can be caused on the link. For instance, in Fig. 4(a), we have $c = 3$ and $c_{max} = 6$. After getting weights for all the links in an LAG (e.g., in Fig. 4(b)), we apply a minimum-weight Steiner tree (MWST) algorithm to find the light-tree. Basically, the link weight definition considers the link usage and the cuts on each link jointly. Hence, with the MWST algorithm, we can ensure that the procedure to obtain the light-tree is fragmentation-aware. *Algorithm 1* shows the overall procedure of 2D-FMA.

Algorithm 1 Integrated 2D Fragmentation-aware RSA (2D-FMA)

```

1: determine the number of requested FS'  $n$  for  $MR(s, D, bw, t_a, d_{max}, t_h)$ ;
2:  $\Psi = \emptyset$ ;
3: for  $t = t_a$  to  $(t_a + d_{max})$  do
4:   for  $k = 1$  to  $(F - n + 1)$  do
5:     for each  $e \in E$  do
6:       if there exists an all-ones matrix  $[\mathbf{R}]_{t_h \times n}$  in  $\mathbf{B}_e$  starting from  $b_{t,k}^e$  then
7:         calculate the weight of  $e$ ;
8:         insert  $e$  in  $G^{t,k}(V^{t,k}, E^{t,k})$  with the weight;
9:       end if
10:    end for
11:    if the source can reach all the destinations in  $G^{t,k}(V^{t,k}, E^{t,k})$  then
12:      apply the MWST algorithm in  $G^{t,k}(V^{t,k}, E^{t,k})$  to get the light-tree;
13:      calculate the cost of the provisioning scheme as the total weight on the light-tree;
14:      store the provisioning scheme together with its cost in  $\Psi$ ;
15:    end if
16:  end for
17:  if  $\Psi \neq \emptyset$  then
18:    serve  $MR$  with the provisioning scheme that has the minimum cost in  $\Psi$ ;
19:    break;
20:  end if
21: end for
22: if  $\Psi = \emptyset$  then
23:   mark  $MR$  as blocked;
24: end if

```

4. System implementation based on SD-EON

By decoupling the control and data planes of a network, software-defined networking (SDN) can make the network programmable and application-aware. Hence, previous studies proposed to realize software-defined EONs (SD-EONs) to further improve the flexibility of EONs [13, 14, 16]. Here, we leverage this idea and implement the proposed algorithms in an SD-EON control plane testbed to evaluate them. In such an SD-EON, the data plane consists of network elements such as edge routers (ERs) and MC-BV-OXC, which are used to set up lightpaths/light-trees. The SD-EON's control plane contains an OpenFlow controller (OF-C) and a few OF agents (OF-AGs) [13, 16]. Each OF-AG is locally attached to a network element and controls its operation. OF-C talks with all the OF-AGs using an extended OF protocol.

4.1. Functional modules

The functional modules in OF-AG and OF-C are shown in Fig. 5. The OF-AG in Fig. 5(a) has the same architecture as that in our previous work [16], and the OF-C in Fig. 5(b) is redesigned to facilitate AR multicast provisioning. Here, the traffic engineering database (TED) and the

network abstraction module (NAM) inherit the same functionalities as in [16], while the details of other functional modules are explained as follows.

AR-Q (Advance Reservation Queue): It is a priority queue to store the provisioning schemes of the AR multicast requests that have been successfully scheduled by the resource computation module (RCM). When the service window of a request starts, AR-Q pushes its provisioning scheme to the resource provision module (RPM), which instructs the related OF-AGs to build the light-tree accordingly.

RPM (Resource Provision Module): It encodes the requests' provisioning schemes in OF messages and interacts with OF-AGs to serve them with light-trees. RPM also instructs TED to update the status of in-service and scheduled requests.

RCM (Resource Calculation Module): It receives AR multicast requests from RPM and calculates provisioning schemes for them. Upon receiving the requests, RCM obtains the network status from TED. When the calculation is done, it pushes the provisioning scheme in AR-Q and instructs RPM to update TED with the scheduled request.

MGD (Multicast Group Database): It helps RCM to map a multicast group address to the corresponding destinations (*i.e.*, D). In this work, we use multicast group addresses to identify destination sets, and the mapping between them is pre-determined. By doing so, we avoid encoding a long list of destinations in OF messages and achieve enhanced scalability.

Note that MGD and TED have interfaces with an external network management system (NMS). Hence, a network operator can retrieve and update the information in them.

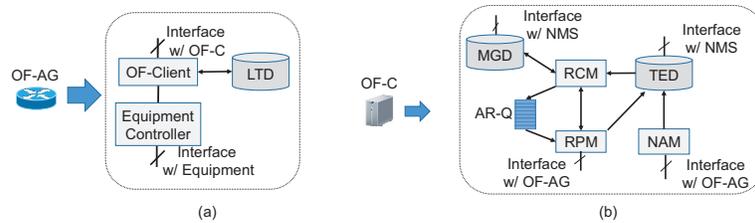


Fig. 5. Functional modules in (a) OF-AG and (b) OF-C.

5. Performance evaluation

We implement an SD-EON control plane testbed with high-performance Linux servers. Each OF-AG is implemented with Open-vSwitch running on a stand-alone server and the 14 OF-AGs are connected according to the NSFNET topology in Fig. 6(a). OF-C is programmed based on the POX platform and it also runs on an independent server. The real arrangement of the OF-C and OF-AGs is shown in Fig. 6(b). Similar to our previous work in [16], we only focus on the control plane operations and emulate the data plane. In other words, each OF-AG configures a virtual software-emulated network element but not a real ER or MC-BV-OXC.

We assume that each fiber link accommodates $F = 200$ FS^{*}. In the experiments, the testbed operates on discrete time-units, each of which is 1 second. The AR multicast requests are generated using the Poisson traffic model with an average service duration of 10 seconds. For each request, the source s and destinations D are randomly selected, and the number of destinations is uniformly distributed in $[2, 5]$. The bandwidth requirement bw is uniformly distributed in $[12.5, 125]$ Gb/s, and the maximum setup-delay d_{max} is randomly chosen from $[1, 5]$ seconds.

We first monitor control plane operations in the testbed and verify that the system can correctly provision AR multicast requests as expected. The AR multicast request is $MR(Node\ 1, \{Node\ 7, Node\ 9, Node\ 13\}, 8\ FS', t_a, d_{max}, t_h)$, where $d_{max} = 5$ seconds. This

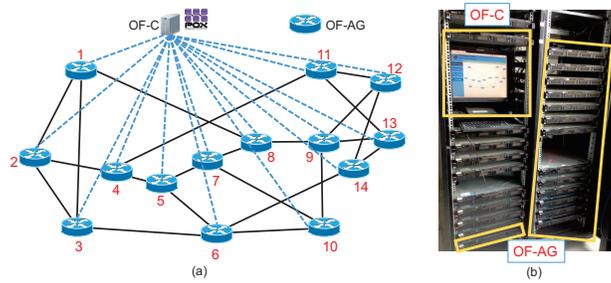


Fig. 6. Experimental setup: (a) control plane topology according to NSFNET, and (b) testbed built with high-performance servers.

means that the maximum setup-delay that the request can tolerate is 5 seconds. RCM gets the light-tree in Fig. 7 and selects FS' [0, 7] on the links to serve MR . The control plane operations for serving MR are also depicted in Fig. 7 and we explain them below.

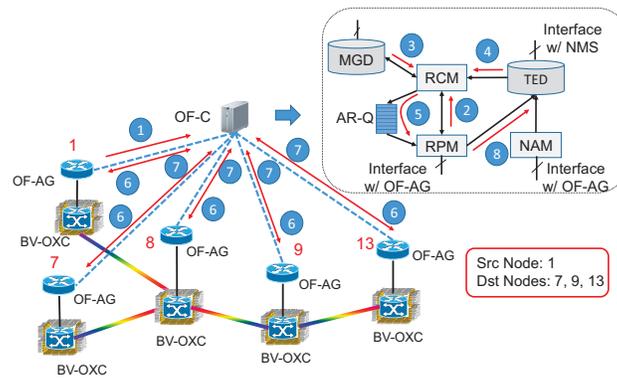


Fig. 7. Procedure for provisioning an AR multicast request (the blue circles contains step numbers).

Step 1: The OF-AG on *Node 1* receives MR , encodes its information in a *PacketIn* message to send to OF-C.

Step 2: RPM in OF-C receives the *PacketIn* message, parses it, and forwards the information to RCM for calculating the provisioning scheme of MR .

Step 3: RCM gets the multicast group address and uses it to request for the actual destination addresses from MGD.

Step 4: RCM obtains the network status from TED, calculates the provisioning scheme of MR , and pushes it in AR-Q.

Step 5: When the service window of MR is about to start, AR-Q pushes its provisioning scheme to RPM. Then, RPM builds the flow-entries for the all related OF-AGs.

Step 6: RPM encodes the *FlowMod* messages and sends them to the related OF-AGs.

Step 7: Each related OF-AG parses the *FlowMod* message and configures its network element accordingly.

Step 8: RPM updates the TED with the information of MR .

Figure 8 shows the Wireshark [20] capture for the OF messages involved in provisioning MR . The first line shows the *PacketIn* message that is sent from the OF-AG on the source node (*i.e.*, *Node 1*) to OF-C. The service window of MR starts from the second line, where we can

Time	Source	Destination	Protocol	Length	Info
10.877665000	Node-1	Controller	OF-Extension_ARMulticast	196	56206 > 6655 [Type:packet_in]
15.590534000	Controller	Node-13	OF-Extension_ARMulticast	162	6655 > 38720 [Type:flow_mod]
15.590757000	Controller	Node-13	OF-Extension_ARMulticast	74	6655 > 38720 [Type:barrier_request]
15.590967000	Node-13	Controller	OF-Extension_ARMulticast	74	38720 > 6655 [Type:barrier_reply]
15.591333000	Controller	Node-9	OF-Extension_ARMulticast	170	6655 > 58892 [Type:flow_mod]
15.591543000	Controller	Node-9	OF-Extension_ARMulticast	74	6655 > 58892 [Type:barrier_request]
15.591913000	Controller	Node-7	OF-Extension_ARMulticast	74	58892 > 6655 [Type:barrier_reply]
15.593657000	Controller	Node-7	OF-Extension_ARMulticast	162	6655 > 48195 [Type:flow_mod]
15.593992000	Controller	Node-7	OF-Extension_ARMulticast	74	6655 > 48195 [Type:barrier_request]
15.594212000	Node-7	Controller	OF-Extension_ARMulticast	74	48195 > 6655 [Type:barrier_reply]
15.595617000	Controller	Node-8	OF-Extension_ARMulticast	170	6655 > 42761 [Type:flow_mod]
15.595847000	Controller	Node-8	OF-Extension_ARMulticast	74	6655 > 42761 [Type:barrier_request]
15.596168000	Controller	Node-8	OF-Extension_ARMulticast	74	42761 > 6655 [Type:barrier_reply]
15.596554000	Controller	Node-1	OF-Extension_ARMulticast	178	6655 > 56206 [Type:flow_mod]
15.596814000	Controller	Node-1	OF-Extension_ARMulticast	74	6655 > 56206 [Type:barrier_request]
15.597077000	Node-1	Controller	OF-Extension_ARMulticast	74	56206 > 6655 [Type:barrier_reply]
15.658129000	Controller	Node-1	OF-Extension_ARMulticast	90	6655 > 56206 [Type:packet_out]

Fig. 8. OF messages involved in provisioning an AR multicast request.

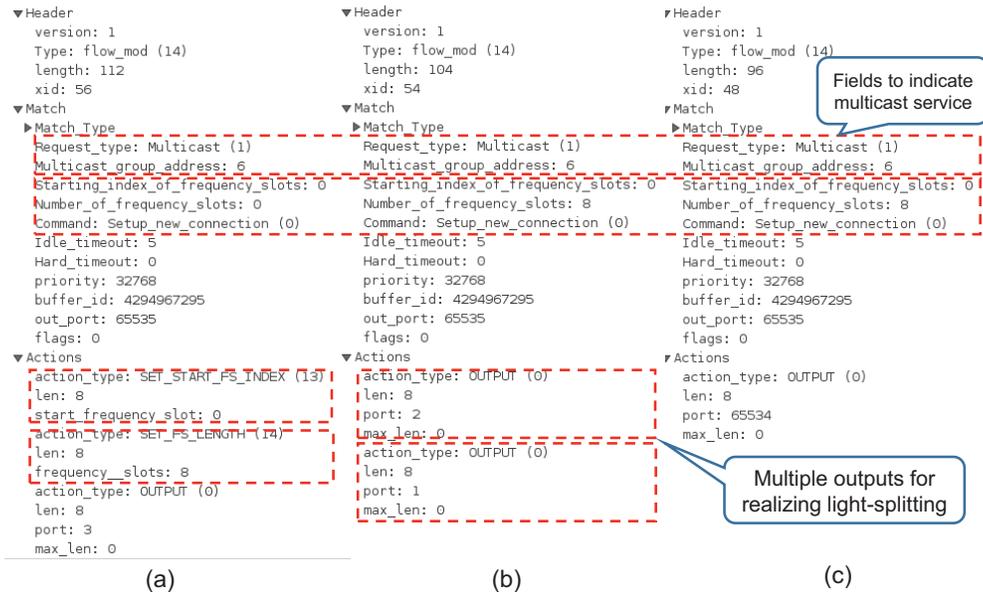


Fig. 9. FlowMod messages received by (a) the source node, (b) an intermediate light-splitting node, and (c) a destination node for setting up a light-tree.

see OF-C distributes *FlowMod* messages to the related OF-AGs for installing the light-tree. We can see that there are 4.713 seconds in between the first and second lines. This is due to the AR scheme, and the result verifies that the request is successfully provisioned before its maximum setup-delay, i.e., $d_{max} = 5$ seconds. The details of the *FlowMod* messages are presented in Fig. 9. Here, we still use Wireshark to parse all the fields in the messages. Figure 9(a) shows the *FlowMod* message sent to the source node. OF-C uses *Request_type* to indicate that the message is for multicast, and encodes *Multicast_group_address* as 6 to represent the destination set {Node 7, Node 9, Node 13}. The *FlowMod* message received by an intermediate light-splitting node, i.e., Node 8, is in Fig. 9(b). The message includes two output ports in the actions, which means that the OF-AG needs to configure underlayer MC-BV-OXC to perform a light-splitting for two light-tree branches. Figure 9(c) shows the *FlowMod* message to a destination.

We then conduct experiments with dynamic network operations to evaluate the performance of the proposed algorithms. The algorithms discussed in Section 3 are all implemented in OF-C,

which are I-RSA-LTW, I-RSA-SFSSI, and 2D-FMA. The control plane experiments evaluate their performance with two metrics, *i.e.*, request blocking probability and average setup delay. For each request, the setup-delay here is the discrete delay time obtained by the RCM in OF-C.

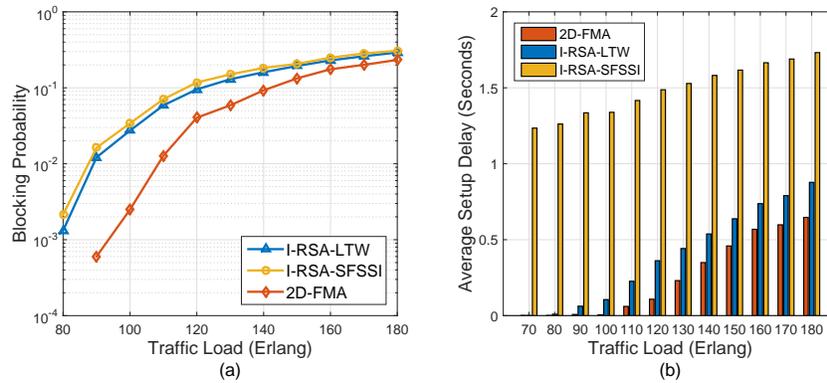


Fig. 10. Results on (a) request blocking probability and (b) average setup delay.

Figure 10 (a) shows the results on blocking probability, and for each traffic load, OF-C serves 3000 requests on average. We observe that 2D-FMA achieves best blocking performance among three algorithms, which suggests that when serving dynamic AR multicast requests, only considering the information in the time and spectrum domains jointly is not enough, and we should address how to effectively relieve the 2D fragmentation carefully. Meanwhile, I-RSA-SFSSI provides higher blocking probability than I-RSA-LTW. This is because SFSSI may lead to worse fragmentation in the time domain. Specifically, I-RSA-SFSSI may provision a request with relatively long setup delay for obtaining the smallest FS start-index. This, however, can make certain FS-blocks become isolated in the time domain during $[t_a, t_s]$ and leave them in the situation that can hardly be allocated to the requests whose service durations satisfy $t_h > t_s - t_a + 1$. Hence, the FS-blocks would be wasted. The results on average setup delay in Fig. 10(b) also confirm our analysis. We can see that I-RSA-LTW provides much shorter setup delays than I-RSA-SFSSI, while 2D-FMA obtains the shortest setup delays.

6. Conclusions

In this work, we developed a 2D-fragmentation-aware RSA (2D-FMA) algorithm for dynamic AR multicast provisioning, which handles the service scheduling and RSA of requests jointly to alleviate the 2D fragmentation caused by light-tree provisioning. Then, we built an SD-EON control plane testbed with high-performance servers, implemented the algorithms, and performed NC&M experiments to evaluate their performance. In the experiments, we monitored the procedure for provisioning AR multicast in the SD-EON, and compared the proposed algorithms in terms of blocking probability and request setup delay. The results showed that 2D-FMA achieved the best blocking performance and the shortest average setup delay.

Acknowledgments

This work was supported in part by the NSFC Project 61371117, the Fundamental Research Funds for the Central Universities (WK2100060010), and the Strategic Priority Research Program of the CAS (XDA06011202).