OpenFlow-Controlled Revenue-Driven AR Service Provisioning in Software-Defined Elastic Optical Networks

Shoujiang Ma, Wei Lu, Cen Chen, Xiaoliang Chen, Zuqing Zhu*

University of Science and Technology of China, Hefei, Anhui 230027, China, Email: zqzhu@ieee.org

Abstract: We propose a control plane framework for realizing OpenFlow-controlled revenue-driven advance reservation (AR) provisioning in software-defined elastic optical networks (SD-EONs), and design two AR provisioning algorithms to be implemented in it.

OCIS codes: (060.1155) All-optical networks; (060.4251) Networks, assignment and routing algorithms.

1. Introduction

Recently, more and more optical networking researches have focused on flexible-grid elastic optical networks (EONs). With agile bandwidth management in the optical layer, EONs can adapt to highly-dynamic traffic from emerging applications easily [1]. Meanwhile, software-defined networking (SDN) that leverages OpenFlow (OF) has been proposed and demonstrated for efficient optical network control and management (NC&M) [2]. The combination of SDN and EON leads to software-defined EONs (SD-EONs), which can further improve the cost-effectiveness of service provisioning. Previously, service provisioning in EONs has been studied for both normal [3] and advance reservation (AR) requests [4]. Here, AR requests refer to those that allow certain delay during lightpath setup, as long as the resources are allocated before a deadline [4]. Previous work on AR provisioning only tried to reduce request blocking probability in the best-effort way. However, in reality, a service provider usually cares more about the revenue-gain from AR provisioning. In this work, we propose a control plane framework for realizing OF-controlled revenue-driven AR provisioning in SD-EONs. The proposed framework is implemented in an SD-EON testbed that consists of 14 standalone OF agents (OF-AGs) and one OF controller (OF-C), which are all realized with high-performance Linux servers. We conduct control plane experiments with different AR payment models and the experimental results indicate that the proposed framework can effectively increase the revenue-gain from AR provisioning.

2. Operation Principle

Fig. 1(a) shows the overall network architecture for realizing OF-controlled revenue-driven AR provisioning in SD-EONs. Basically, the data plane consists of several bandwidth-variable optical cross-connects (BV-OXCs) that each is controlled by an OF agent (OF-AG) attached locally. The OF-AGs and centralized OF controller (OF-C) belong to the control plane. The detailed architectures of OF-AG and OF-C are also shown in Fig. 1(a), which are similar to those in our previous work [5], except for the AR request queue (AR-Q). Basically, each OF-AG communicates with OF-C with an extended OF protocol, executes the instructions from OF-C for provisioning AR requests, and configures its BV-OXC accordingly. We first explain how a single AR request is provisioned with this OF system.

Step 1: The source node receives an AR request $LR(d,n,\Delta t,d_{max},p_{max})$, where d is the destination address, n is the bandwidth requirement in terms of frequency slots (FS'), Δt is the service duration, d_{max} is the maximum setup delay, and p_{max} is the maximum payment from the client for the service.

Step 2: The OF-AG on the source node includes source address s, and constructs a *Packet-In* message to forward the AR request's information $\{s,d,n,\Delta t,d_{max},p_{max}\}$ to OF-C.

Step 3: The resource provision module (RPM) in OF-C receives the *Packet-In* message, buffers the AR request in AR-Q, and instructs the resource computation module (RCM) to calculate the AR provisioning solution.

Step 4: RCM gets network status from the traffic engineering database (TED) and executes an AR provisioning algorithm. If a feasible solution can be obtained, RCM instructs RPM to encode the corresponding flow-entries for all the OF-AGs along the routing path. Each flow-entry includes information such as input and output ports, starting frequency, number of FS', and *etc*. Otherwise, if the AR provisioning fails due to insufficient resource, RCM will instruct RPM to communicate with the OF-AG on the source node for blocking the request.

Step 5: RPM sends the *Flow-Mod* messages to the OF-AGs on the source and destination and all the intermediate nodes along the path. Each OF-AG parses the *Flow-Mod* message and configures its data-plane equipment accordingly.

3. AR Provisioning Algorithms

In the proposed framework, RCM runs an AR provisioning algorithm based on the information in AR-Q and TED, and instructs RPM to provision the AR request by allocating proper spectrum resources staring from the right time instant.

Source	Destination	Protocol	Length	Info
Node-1	Controller	OF-Extension_AR	196	37987 > 2009 [Type:PacketIn]
Controller	Node-1	OF-Extension_AR	154	2009 > 37987 [Type:FlowMod]
Controller	Node-9	OF-Extension_AR	162	2009 > 47942 [Type:FlowMod]
Controller	Node-8	OF-Extension_AR	162	2009 > 47170 [Type:FlowMod]
Controller	Node-9	OF-Extension_AR	74	2009 > 47942 [Type:Barrier_Request]
Controller	Node-8	OF-Extension_AR	74	2009 > 47170 [Type:Barrier_Request]
Controller	Node-1	OF-Extension_AR	178	2009 > 37987 [Type:FlowMod]
Node-9	Controller	OF-Extension_AR	74	47942 > 2009 [Type:Barrier_Reply]
Controller	Node-1	OF-Extension_AR	74	2009 > 37987 [Type:Barrier_Request]
Node-8	Controller	OF-Extension_AR	74	47170 > 2009 [Type:Barrier_Reply]
Node-1	Controller	OF-Extension_AR	74	37987 > 2009 [Type:Barrier_Reply]

(a) (b) (c)

Fig. 1. (a) Network architecture of SD-EON to realize OF-controlled revenue-driven AR provisioning, (b) Control plane testbed built with high-performance Linux servers, and (c) Wireshark capture for provisioning an AR request.

The network abstraction module (NAM) in OF-C abstracts the SD-EON as $G(V,E)$, where V and E are the sets for nodes and fiber links, respectively. The spectrum resource on each link is modeled as B FS' that each has a bandwidth of 12.5 GHz. If a LR is provisioned, we have $[t_s,t_e]$ as its service window, where t_s and t_e are the service start- and end-time. Hence, according to the working principle of AR, t_s should fall in $[t_a,t_a+d_{max}]$, where t_a is the time when the request comes in. In this work, we assume that the client's payment p for LR is related to the setup delay $d'=t_s-t_a$ and consider two payment models: 1) Fixed, as $p(d')=p_{max}, d'\leq d_{max}$, and 2) Linear, as $p(d')=p-\beta\cdot d', d'\leq d_{max}$.

We have $p(0)=p_{max}$ to ensure that the provider gets full payment if LR is served immediately upon arrival, while on the other extreme, we have $p(d')=0, d'>d_{max}$ to enforce the service-level agreement (SLA), *i.e.*, the provider gets nothing if the setup delay of LR is longer than d_{max}. Here, we assume that the SD-EON operates on discrete time intervals and hence the next service time after t_s is t_s+1. Apparently, for both payment models, we have $p(d_{max}+1)=0$. On the other hand, the actual cost of serving LR can be determined by looking at the spectrum resources it occupies. Let $R_{s,d}$ denote the routing path of LR and $hop(R_{s,d})$ return the hop count of $R_{s,d}$, and we define the cost of provisioning LR as

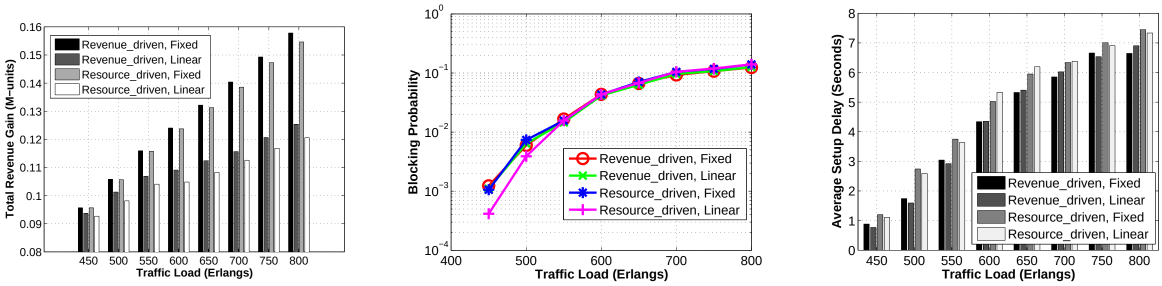
$$q(LR)=n\cdot hop(R_{s,d})\cdot\Delta t\cdot q_{unit}, \tag{1}$$

where q_{unit} is the unit cost for occupying a FS on a link for a time interval. Therefore, for LR, after determining t_s and the routing and spectrum assignment (RSA), the provider can calculates its revenue gain as

$$g=p(d')-q(LR). \tag{2}$$

We first adopt a *Resource-Driven* AR provisioning algorithm from our previous work in [4], and it tries to minimize the resource occupation of each AR request with the following procedure.
Step 1: At a service time, OF-C stores new pending AR requests in AR-Q and sort them in ascending order of $n\cdot\Delta t$.
Step 2: RCM performs scheduling and RSA for the sorted requests in AR-Q one-by-one, with the assistance of a resource matrix $[\mathbf{RS}]_{K\times T}$, where K is the number of shortest path candidates, and T is the number of service start-times that the request allows. Here, we pre-calculate K shortest paths for each s-d pair in $G(V,E)$. Each element $s_{k,j}$ in $[\mathbf{RS}]$ represents the spectrum resources required for provisioning the request over the k-th path candidate from the j-th service start-time. More specifically, $s_{k,j}$ equals to the product of n and the k-th path candidate's hop-count. If the k-th path candidate is unavailable from the j-th service start-time due to insufficient available FS', we set $s_{k,j}=+\infty$.
Step 3: RCM searches $[\mathbf{RS}]$ to find the minimum element s_{k_m,j_m}. If $s_{k_m,j_m}<+\infty$, RCM instructs RPM to schedule the AR request with the k_m-th path from the j_m-th start-time. Otherwise, RPM invokes the procedure to block the request.
Step 4: OF-C repeats **Steps** 2-3 until all the AR requests in AR-Q are processed.

The *Resource-Driven* mentioned above only tries to minimize FS usages in SD-EONs, but may not achieve the maximum revenue gain for providers. Hence, we propose a *Revenue-Driven* AR provisioning algorithm as follows.
Step 1: At a service time, OF-C sorts new pending AR requests in AR-Q in descending order of their predicted revenue gains, which are calculated with Eq. (2) by assuming each request is routed over the shortest path. Specifically, RCM gets the shortest path for LR, calculates $p(0)$ and $q(LR)$, and then the predicted revenue-gain is $g=p(0)-q(LR)$.
Step 2: RCM performs scheduling and RSA for the sorted requests in AR-Q one-by-one, with the aid of a revenue gain matrix $[\mathbf{RG}]_{K\times T}$. Each element $g_{k,j}$ in $[\mathbf{RG}]$ represents the revenue-gain for provisioning the request over the k-th path candidate from the j-th service start-time. If the k-th path candidate is unavailable from the j-th service start-time



(a) Total revenue gain (b) Blocking probability (c) Average setup delay

Fig. 2. Experimental results from OF-controlled AR provisioning in an SD-EON.

due to insufficient available FS', we set $g_{k,j} = 0$.

Step 3: RCM searches $[\mathbf{RG}]$ to find the maximum element g_{k_m,j_m}. If $g_{k_m,j_m} > 0$, we schedule the request with the k_m-th path candidate from the j_m-th service start-time. Otherwise, RPM invokes the procedure to block the request.

Step 4: OF-C repeats **Steps** 2-3 until all the AR requests in AR-Q are processed.

4. Experimental Setup and Results

We evaluate the performance of the proposed AR provisioning framework in an SD-EON control plane testbed built with high-performance Linux servers as shown in Fig. 1(b), which consists of 14 stand-alone OF-AGs that are connected according to the NSFNET topology, and a centralized OF-C. Each OF-AG is programmed based on Open-vSwitch running on Linux, while the OF-C is implemented with the POX platform. OF-C is directly connected to all the OF-AGs. Here, the SD-EON's data plane is emulated, where we assume that each fiber link can accommodate $B = 358$ FS'. The AR requests are generated according to the Poisson traffic model. The *s*-*d* pair of each request is randomly chosen, the bandwidth requirement n is uniformly distributed within $[2,40]$ FS', the maximum setup delay d_{max} is within $[30,50]$ seconds, and the service duration Δt follows the negative exponential distribution with a mean of 100 seconds. Here, OF-C processes AR requests according to discrete time intervals, and a service time interval is set as 10 seconds to save the time for experiments. Note that in practical SD-EONs, the service time interval can be much longer, and our system can take arbitrary service time intervals. The maximum payment p_{max} of each request is within $[20,50]$ units, and we have $q_{unit} = 1$ to normalize the cost. Fig. 1(c) shows the wireshark capture for provisioning an AR request with the OF-controlled scheme, which includes all the related OF messages in time order.

Fig. 2(a) shows the experimental results on total revenue gain (1 M-unit = 10^6 units) from all the provisioned AR requests. We observe that when traffic load is below 500 Erlangs, the results from all the algorithms are similar due to the low blocking probability, *i.e.*, the network is relatively empty and most requests can be provisioned immediately upon arrival. However, when the traffic load keeps increasing, the revenue gains from *Revenue-Driven* are larger than those from *Resource-Driven* for both payment models. This is because *Revenue-Driven* tries to maximize the revenue gain with a more effective approach. Fig. 2(b) illustrates the results on blocking probability and we can see that the results from both algorithms are similar when the fixed model is used. For the linear model, the blocking probability from *Revenue-Driven* is slightly higher when the traffic load is lower than 550 Erlangs. Fig. 2(c) compares the average setup delay from the two algorithms, and *Revenue-Driven* provides shorter setup delay than *Resource-Driven*.

5. Summary

We proposed a control plane framework for realizing OF-controlled revenue-driven AR provisioning in SD-EONs, and designed two AR provisioning algorithms to be implemented in it. The experimental demonstrations evaluated the framework and algorithms in an SD-EON control plane testbed with two payment models.

Acknowledgment

This work was supported in part by the Projects NCET-11-0884, NSFC 61371117, WK2100060010, and the Strategic Priority Research Program of CAS (XDA06010302).

References

[1] B. Yoo, *et al.*, in *Proc. of ONDM 2012*, pp. 1-6, Apr. 2012.
[2] V. Gudla, *et al.*, in *Proc. of OFC 2010*, paper OTuG2, Mar. 2011.
[3] M. Jinno, *et al.*, *IEEE Commun. Mag.*, vol. 48, pp. 138-145, Aug. 2010.
[4] W. Lu, *et al.*, *J. Lightw. Technol.*, vol. 31, pp. 1621-1627, May 2013.
[5] S. Ma, *et al.*, in *Proc. of OFC 2014*, paper W4A.2, Mar. 2014.