

Delayed spectrum allocation in elastic optical networks with anycast traffic

PEGAH AFSHARLAR,^{1,2} ARASH DEYLAMSALEHI,^{1,3} AND VINOD M. VOKKARANE^{1,4}

¹Electrical and Computer Engineering Department, University of Massachusetts Lowell, MA 01854, USA

²Pegah_Afsharlar@student.uml.edu

³Arash_Deylamsalehi@uml.edu

⁴Vinod_Vokkarane@uml.edu

Abstract: We investigate the problem of dynamic routing and spectrum assignment for advance reservation anycast requests using a novel algorithm called Delayed Spectrum Allocation (DSA) which delays allocation of resources until immediately before transmission begins. We evaluate the flexible window STSD (demands that specify start-time and duration) to further improve request blocking. Through extensive simulation results, we show that anycast with flexible STSD using DSA approach can significantly reduce blocking probability of anycast and unicast requests compared to traditional Immediate Spectrum Allocation (ISA). The improvement is up to 50 percent in unicast traffic and even higher in anycasting. We also propose balanced routing for anycast algorithms to reduce the blocking.

© 2021 Optical Society of America under the terms of the [OSA Open Access Publishing Agreement](#)

1. Introduction

Due to unprecedented growth rates of data centers, ubiquitous cloud computing, and Content Delivery Networks (CDN), modern and future networking applications will necessitate fast, reliable, and efficient transfer of large datasets approaching the scale of petabytes and exabytes. Wavelength Division Multiplexed (WDM) optical networks, which employ a collection of finely spaced, non-interfering independent wavelength channels have long been touted as the communication technology that would support these applications. Unfortunately, even WDM networks are proving insufficient for emerging network application demands. As such, the field of *Elastic Optical Networks (EON)* has emerged. Proposed initially as the SLICE architecture [1], EON is a promising solution for next-generation high-speed optical transport that provides high levels of elasticity and efficiency to the spectral domain. EONs treat the available optical spectrum as an elastic resource and allocate portions of the spectrum range in a variable and adaptive manner, as needed by incoming demands, for more efficient consumption than fixed-grid WDM [2].

The three significant EON enabling technologies are Optical Orthogonal Frequency Division Multiplexing (OFDM), Bandwidth-Variable Cross-Connect (BV-OXC), and Bandwidth Variable Transponder (BV-T) [1]. OFDM enables adjacent subcarriers to overlap, and transmit data in parallel within a connection, thereby enabling better spectrum utilization than WDM. The inclusion of BV-OXCs offers further flexibility through the ability to dynamically adjust bandwidth capacity using variable modulation format and bit rate configurations. Spectrum guard bands are required between adjacent channels to prevent interference effects and increase the spectrum overhead that leads to efficient spectrum utilization. By contrast, WDM networks support only a fixed-grid of predetermined wavelength channels, which may be inefficiently consumed by sub-wavelength granularity demands, leading to resource wastage and starvation of subsequent sub-wavelength requests.

Given the unpredictable nature of real-world data transmissions (in terms of size, duration, repeatability, fiber traversal, etc.) efficient allocation of spectrum resources in EONs is a complex

task. The problem of assigning both a physical route and appropriate spectral resources to a request, or set of requests, is commonly known as *Routing and Spectrum Assignment (RSA)*. Practically, modulation formats must also be considered in large-scale networks, such as optical core mesh networks, when signal reachability becomes a limiting factor. This more specific view of allocation is the *Routing, Modulation, and Spectrum Allocation (RMSA)* problem [3]. RSA and RMSA are first proposed in [4], and are applied to stochastic traffic arrival patterns in [5–7]. Static demand sets are submitted for RMSA in [8], wherein a novel Integer Linear Programming (ILP) is formulated to describe the appropriate solution space, and is compared to related, but suboptimal RMSA heuristics. Furthermore, [9] considers the RMSA problem with a specific focus on efficient transponder allocation.

Reference [10] investigates the elastic spectrum allocation capability of elastic flexgrid optical networks and its effectiveness in terms of applicability. Moreover, [11] investigates and classifies RSA techniques in flexible optical network. Elastic size of contiguous subcarrier slots leads to misalignment of isolated and small-sized frequency slices which is called bandwidth fragmentation. Fragmentation increases the blocking probability of demands because using them for serving connection requests is difficult.

There are different approaches to confront fragmentation. One way is to allocate resources using efficient algorithms to minimize or avoid fragmentation. [12] explores the fragmentation management approaches in EON. The other solution is to reallocate assigned connection requests in a way to make bigger available blocks of resources [13].

This paper considers *Advance Reservation (AR)* traffic wherein transmission begins after some predetermined book-ahead time. AR differs from Immediate Reservation (IR) demands in that the latter require transmission to begin immediately upon arrival into the network. Networks employing AR services enjoy the efficiency offered through predictable scheduling and resource allocation among a set of competing or temporally conflicting lightpath demands [14]. There exist some delay tolerant services which are allocated in future time after a book-ahead time. Knowing the details of the request ahead of time can lead to more optimal solution for RSA and better quality of service [15], [16].

In this work, we assume that both the start-time and duration of dynamically arriving AR demands are explicitly given. We refer the reader to [17] for a comprehensive survey of AR work in optical networks, including coverage of additional AR classifications.

In this paper, we define Specified Start-Time Specified Duration (STSD) AR demand. STSD demands specify start-time and holding-time. The user can determine a fixed start-time, that request needs to start at specified time; otherwise it is blocked.

Another type of STSD demand is STSD with flexible window wherein the user determines a range of start-times. Typical use for this kind of request is during transfer of large file sets. Users can schedule any time in the future by determining a window of start-times. For example, in many applications transmission can start within some flexible window rather than a specific time of day. STSD with flexible window benefits from temporal flexibility and transmission start-time may slide earlier or later within the offered scheduling window as defined by application or network. In this work, we investigate STSD with flexible window.

Figure 1 shows an STSD request with a flexible start window. This request can be defined by its duration, ω , α_i , the service book-ahead time and L that is the length of the start window. The request must be able to start at any time within this window.

Several works have studied AR for truly continuous traffic demands, however most contemporary investigations are fixed STSD. For simplicity, we assume any unit of work that alters the state of the network must occur at the beginning of a time-slot and must finish at an integer multiple of the time-slot span. For example, consider the set of requests given in Table 1 wherein the time-slot span is exactly 15 minutes long. Reservations may only begin and end in 15-minute intervals, and must have a duration measured in some whole number of time-slots. Requests are

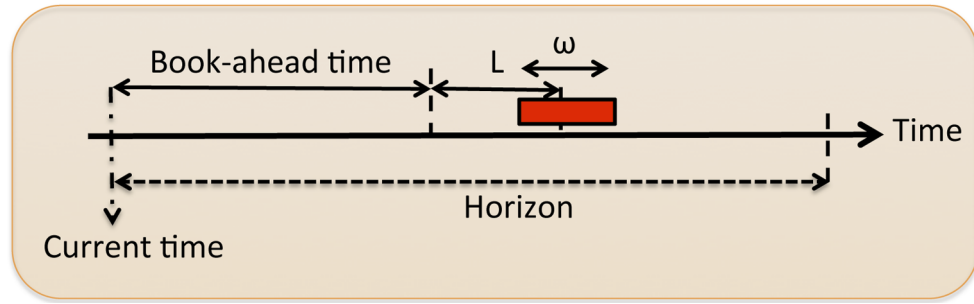


Fig. 1. STSD based AR demand with a flexible scheduling service window.

allowed to be served after Book-ahead time and before the deadline. Note that although R_1 and R_3 are pair-wise independent in the time-domain, R_2 overlaps temporally with both, and in such a case, the requests are in competition for available network resources during their respective periods of overlap. However flexible STSD solves the problem and make allocation possible for all three requests.

Table 1. Example set of requests (time-slot = 15 min)

Request	Book-ahead time	Deadline	Duration	Slot-Duration
R_1	9:15 AM	11:29 AM	2 hr, 15 min	9 time-slots
R_2	10:30 AM	11:59 AM	1 hr, 30 min	6 time-slots
R_3	11:30 AM	13:14 PM	1 hr, 30 min	6 time-slots

As EONs gain popularity among the networking community, a related need to perform synchronization or distribution among collections of replicated data is moving to the fore. Cloud computing, CDN, distributed computing environments, and distributed storage systems are on the verge of ubiquity. With these flexible and convenient data replication and synchronization services come the ability to not only perform reliable networking at high speeds, but to also ensure efficient communications with one or more of these replicated data repositories. For example, a CDN is system of distributed servers that precipitates the delivery of web content by storing data in numerous replicas to be available for users in their nearest server with high performance. Cloud computing is similarly, a network of servers that help user to use processing resources over the Internet instead of maintaining local computing infrastructures [18]. As an example, Akamai owns the largest CDN in the world with more than 175,000 servers distributed globally [19]. Consequently, *anycast* transmission services have become indispensable.

In anycast services, a transfer demand can be satisfied by any of the available replicated data repositories. Thus, the destinations of an end-to-end connection can vary flexibly over time, or throughout some period, thus enabling an intelligent resource scheduler the opportunity to select and reach the destination which is cheapest, nearest, least-loaded, etc., based on an adoptive and regularly updated network state. Anycasting is known to improve network resource allocation efficiency when compared to inflexible pre-determined fixed unicast lightpaths for many applications [20]. Anycast has been addressed in the context of both WDM [21] and EON [22]. The work in [23] has modeled the EON planning problem with anycast demands to minimize spectrum resources required throughout the network.

In [24] the authors investigate dynamic routing of both anycast and unicast traffic in EONs, and obtain the best trade-off between blocking probability and usage of regenerators by enabling the selection of different modulation formats. Distance-adaptive modulation is found to reduce spectrum usage by using higher modulation levels, while shorter paths lead to cost-efficiency

because of savings in regenerators throughout the network. The authors in [25] proposes an adaptive regenerator aware algorithm that enables modulation to change along the lightpath. In [26], the authors have formulated an ILP model for problem of anycast flow restoration in EONs. The survivability of joint anycast and unicast demands in EON is discussed in [27]. An ILP formulation describing the RSA problem with dedicated path protection is considered for this mixed traffic. The key goal of this work is to improve the network performance under dynamic anycast traffic scenario in terms of blocking probability using our proposed algorithm.

Unprecedented interest in both EONs and the anycast paradigm motivate investigation of anycast oriented RMSA. In this work, we apply a novel technique called *Delayed Spectrum Allocation (DSA)* to sets of stochastically arriving time-delayed AR anycast lightpath requests in EONs. DSA delays the allocation of spectrum resources to a demand until immediately prior to the start of transmission. Our hypothesis is that by supporting DSA, requests which arrive to the system late, but start before some of the previously scheduled demands, will experience less starvation than traditional ISA approach, thereby lowering overall blocking rates during a long-term schedule and increasing network-wide resource efficiency. The difference between this work and our previous one is that in this work resources are not assigned to demand upon start-time of the request. In the new algorithm resource availability is checked after arrival of request and remain in the schedule list to allocate upon the start time.

The objective of this paper is thus to increase overall resource efficiency and minimize total blocking demands for unicast and anycast applications in EONs by employing DSA.

2. Network and scheduling assumptions

Here we describe the scope of, and assumed inputs to the considered RMSA problem with DSA support. First, we describe the time, space, and spectrum resource assumptions, and then detail the AR lightpath request specification including anycast parameters.

We consider a network topology as a connected graph $G = (\bar{V}, \bar{E}, \bar{F}, H, \bar{T})$, where \bar{V} represents the set of vertex nodes, and \bar{E} is the set of bi-directional fiber links between nodes in \bar{V} . Let V and E therefore denote the number of network nodes and links, respectively. All nodes in \bar{V} have been equipped with a BV-OXC to enable EON functionality and dynamic channel tuning. We divide the spectral granularity of the transmitters and OXC into a set of sub-carriers frequency-slots \bar{F} , with each $f_i \in \bar{F}$ supporting a capacity C_{slot} . C_{slot} is tunable to different modulation levels. A guard band of N_g slots is required for demodulation (in BV-OXC) to separate adjacent spectra. The horizon, H , denotes some time in the future beyond which no demand nor network information is currently known. H therefore represents the cumulative period between the current time and some time in the future for which demands may be assigned resources by the network allocator. We assume that H is sufficiently large to prevent any call blocking caused by long holding-times or slow service rates. For simplicity, the range between the t_{now} and H is discretized into a set of fixed-duration time-slots, \bar{T} . Several works have studied AR for truly continuous traffic demands, however most contemporary investigations use time-slots of a fixed and predetermined size [28,29]. For simplicity, we assume any unit of work that alters the state of the network must occur at the beginning of a time-slot and must finish at an integer multiple of the time-slot span.

We assume k replica servers located at different nodes of network and anycast demand can be assigned to any of the replica servers. The replica selection for each anycast request is performed according to the balanced load path. X_p denotes the number of hops for path p and U_p is the spectrum slot usage of path p . We use adaptive routing to obtain a balanced path between source and destination. We propose a balanced path to trade-off between number of hops and spectrum availability of paths. Let X_{Max} be the maximum number of hops of the path and U_{Max} is the

maximum spectrum usage of the network in each path. The link weight metric is calculated as

$$W_p = \eta * \frac{X_p}{X_{\text{Max}}} + (1 - \eta) * \frac{U_p}{U_{\text{Max}}}, \quad (1)$$

where η is a weight factor on the range of $[0, 1]$. Decreasing η increases the impact of the spectrum utilization factor on the routing. Setting η to 0 leads choosing paths based on least spectrum usage and setting η to 1 results in the minimum-hop shortest path.

2.1. AR request assumptions

Each lightpath request is denoted by a seven-tuple: $R_i = (s_i, D_i, t_i, \alpha_i, \omega_i, L, C_i)$, where $s_i \in \bar{V}$ is the source node, and $D_i \subseteq \bar{V} - \{s_i\}$ is the destination node of request R_i . L is the length of the flexible window.

The requested bandwidth capacity, C_i (in Gbps) maps to some number of frequency-slots, S_i , according to Eq. (2), where M indicates the modulation format. In this work, we consider four modulation formats commonly associated with EON architectures [28]. Modulation level selected for a given path depends on the path length; each modulation format affects signal reachability. 8-QAM (3 bits per symbol quadrature amplitude modulation) and 16-QAM (4 bits per symbol quadrature amplitude modulation) are enabled for short-reach connections. QPSK (2 bits per symbol quadrature phase shift keying) and BPSK (1 bit per symbol binary phase shift keying) modulation support long-reach connections. Table 2 describes realistic values assumed for the described spectrum variables.

$$S_i = \left\lceil \frac{C_i}{C_{\text{slot}} * M} \right\rceil + N_g. \quad (2)$$

Shortest path routing is used between s_i and each $d_i \in \bar{D}_i$, such that a set of paths, \bar{P}_i , is computed for each request. The request arrives to the network scheduler at time-slot $t_i \in \bar{T}$. α_i is the service book-ahead time-slot, and ω_i is the reservation's holding-time. Any request must be assigned the same set of contiguous frequency-slots throughout its entire duration in order for successful reservation. R_i may also be trivially expressed with its implicitly known attributes: $R_i = (\bar{P}_i, S_i, \omega_i)$.

Table 2. Spectrum Assumptions

F, Number of frequency-slots per link	128
Spectrum efficiency of BPSK	1 bps/Hz
C_{slot} , bandwidth of a frequency per link	12.5 GHz
N_g , number of guard-band slots per request	1
Transmission reach of BPSK (M=1)	5000 km
Transmission reach of QPSK (M=2)	2500 km
Transmission reach of 8-QAM (M=3)	1250 km
Transmission reach of 16-QAM (M=4)	625 km
Average service duration Δt	100
Range of requested capacity (C_i)	12.5 - 200 Gbps

3. Delayed spectrum allocation (DSA) in flexible window STSD AR

In this section, we describe DSA on EON. Our previous work in [30] proposes the novel concept of Delayed Allocation for WDM-enabled networks. In the prior work, we propose the decomposition of network resource allocation into two distinct phases, the *Request Provisioning Phase (RPP)*

and the *Request Allocation Phase (RAP)*. We have previously proposed a novel DSA algorithm in [31] for decoupling the provisioning and allocation of resources to demand sets. In that work, the requests that are scheduled to start some time in the future will be flagged to indicate that resources have been promised for that future span. In case of insufficiently low spectral fragmentation but sufficient spectral capacity to support a request, the current request and all other requests in the provisioned set will enter the re-provisioning process. This process is engineered to determine if rearranging the requests according to their book-ahead times, after arrival will allow all requests, including the current request, to be provisioned successfully. This approach eliminates any possibility that provisioned demands that have already been promised resources will be blocked at allocation time. In this current work, our goal is to reduce complexity overhead comparing to previous work, as well as total blocking, and delay of transmission. Complexity reduction is obtained by eliminating the tagging and re-provisioning process that were applied in our previous work. Our new two phase DSA is decreasing the running time of algorithm, as well. Figure 2 present an illustrative comparison demonstrating how the proposed DSA algorithm can decrease blocking probability and spectral fragmentation over single phase RSA. For simplicity, we assume a single-link network, with 6 frequency-slots, upon which the four demands must be provisioned during the time-span of t_0 - t_3 . We simplify each request to the form $R_i = (\alpha_i, \omega_i, S_i, L = 1)$ to describe its start- and end-time and number of frequency-slots necessary to satisfy the request. The requests may be defined as: $R_1 = (t_3, t_3, 2, 1)$, $R_2 = (t_1, t_1, 4, 1)$, $R_3 = (t_2, t_3, 3, 1)$, and $R_4 = (t_0, t_2, 2, 1)$. Using a traditional first-fit band assignment policy, the first available spectrum will be reserved upon arrival of a request. Figure 2(a) depicts how R_1, R_2 , and R_3 are all assigned wavelengths in-order, thus, preventing R_4 from receiving service [32]. Figure 3(b) offers an alternative solution using DSA. Each demand is only *promised* resources at the time of arrival to the network scheduler in the RPP, and specific resources are allocated immediately prior to the start-time during the RAP. For example, since R_4 starts first, it will be the first to be assigned a final spectrum-band beginning from time t_0 . R_2 will be satisfied next at t_1 , and will be assigned to f_3 through f_6 since R_4 has already consumed f_1 and f_2 . R_3 and R_1 will be provisioned at t_2 and t_3 , respectively, thereby preventing blocking for this example, and lowering the fragmented use of the available wavelengths.

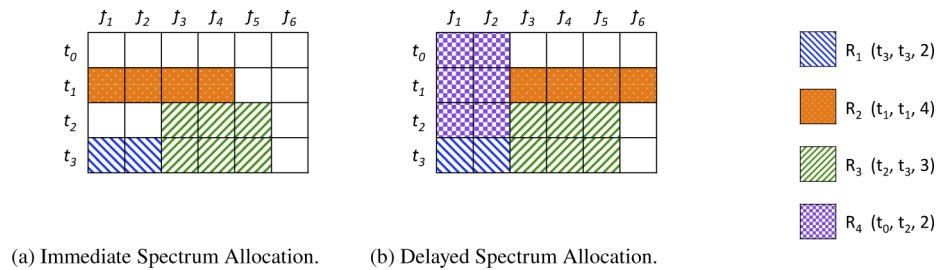


Fig. 2. First-fit RSA with single-phase provisioning and allocation leads to spectral fragmentation at t_6 and cannot satisfy all five requests. DSA resolves this issue and eliminates blocking.

Figure 3 offers an illustrative example of DSA given flexible window STSD. Three requests arrive to the network in-order and all require transmission between the same source-destination pair. In flexible window the size of sliding window is equal to 3 time slots for all requests. The requests may be defined as: $R_1 = (t_1, 4, 2, 3)$, $R_2 = (t_3, 3, 3, 3)$ and $R_3 = (t_2, 2, 4, 3)$. Using immediate spectrum allocation Fig. 3(a), R_1 and R_2 , are assigned spectrum slots in-order, thus preventing R_3 from receiving service. R_3 is able to delay its start time to t_3, t_4 or t_5 but there are not enough available contiguous slots and R_3 is blocked.

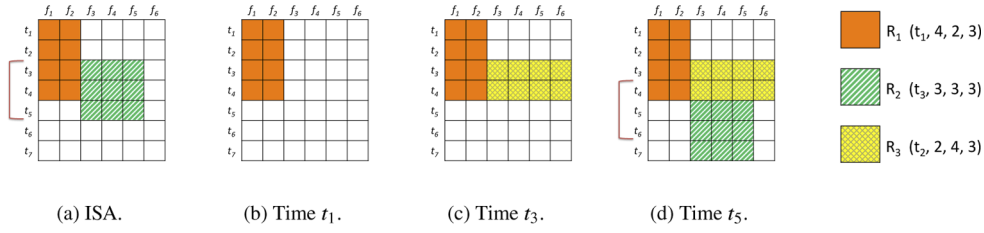


Fig. 3. (a) Immediate spectrum allocation with flexible window of 3 time slots RSA with single-phase provisioning and allocation cannot satisfy all three requests. (d) DSA resolves this issue and eliminates blocking.

Figures 3(b) to 3(d) offers an alternative solution using DSA. Each demand is only *promised* resources at the time of arrival to the network scheduler in the RPP, and specific resources are allocated immediately prior to the book-ahead-time during the RAP. For example, since R_1 starts first, it will be the first to be assigned beginning from time t_1 . R_3 will be satisfied next at t_2 , and will be assigned $f_3 - f_6$ since R_1 has already consumed f_1 and f_2 . R_2 has the book-ahead equal to 3, so it can be scheduled to begin at any time from t_3 to t_5 . R_2 will be provisioned at t_5 , thereby preventing blocking for this example. In terms of minimizing the delay, the algorithm tries to allocate each request with minimum delay in the flexible window.

3.1. Proposed DSA algorithms

In this section, we present a DSA algorithm for both unicast and anycast RMSA in EONs. The main goal of the algorithm is to minimize total blocking probability. The input to the DSA algorithm, given in Algorithm 1, is the set of known time-slots until the horizon, \bar{T} , as well as a set of (possibly dynamic) requests with specified arrival times, book-ahead and holding-times, specified requested bandwidth capacity, source and the k replicas which comprise the candidate destination set D_i . L represents the size of flexible window and η is a weight factor on the range of $[0, 1]$. All network assumptions are also treated as inputs.

The algorithm loops through each time-slot in \bar{T} and examines every newly arrived request in turn. All requests are inserted into a temporary holding list called $\bar{R}_{reserve}$. We assume that a given request, R_i , may be scheduled if and only if its arrival time parameter t_i is equivalent to the current $t \in \bar{T}$. This means that if the book-ahead time of the request, α_i , is not equal to the current time (e.g. it is scheduled to begin at a future time-slot), the demand will remain in $\bar{R}_{reserve}$. Thus, the allocation of resources to satisfy the demand is delayed until the current time reaches the book-ahead of the request. Paths are calculated to reach each replica using η and Eq. (1). Minimum-hop paths are calculated to reach each replica using Dijkstra's algorithm (Line 9). D_k is the set of available replicas. In k replica scenarios, k nodes with data replicas are available and the algorithm chooses best replica and route according to lowest weight factor. Election of one destination among k replicas is denoted by $k/1$.

A specific modulation level maps to each path on Line 18 depending on the reachability limits specified in Table 2. The number of frequency-slots that are required to satisfy the request to transfer to each destination node on Line 19 is calculated from Eq. (2).

The largest span of contiguous frequency-slots for the duration of the request is computed by a call to the subroutine on Line 18. If the computation on Line 19 finds that there is not enough spectrum available, then it will try the next time-slot. After trying whole flexible window and not finding enough available spectrum to support the entire request to transfer to a specified replica, then the next replica will be tried. If there is indeed enough total spectrum capacity and sufficient contiguous frequency to completely support the request, request will be served. We use first-fit spectrum assignment for allocation. If the current demand cannot however find appropriately

Algorithm 1: Flexible STSD with Delayed Spectrum Allocation (DSA)

```

input :  $\bar{R}_{demand} \mid R_i \in \bar{R}_{demand} = (s_i, D_i, C_i, t_i, \alpha_i, \omega_i)$ 
        Topology  $G = (V, E, F, H, T)$ 
         $N_g$ 
         $C_{slot}$ 
         $L$ 
         $k$ 
         $\eta$ 

1  $\bar{R}_{reserve} \rightarrow \emptyset$ 
2  $\bar{U}_s \rightarrow \emptyset$ 
3 forall  $t \in T$  do
4   foreach  $R_i \in \bar{R}_{demand}$  do
5      $\bar{R}_{reserve}.add(R_i)$ 
6   foreach  $R_j \in \bar{R}_{reserve}$  do
7     if  $\alpha_j == t$  then
8       foreach  $d_j \in \bar{D}_i$  do
9          $P_j \leftarrow \text{Dijkstra}(s_j, d_j)$ 
10         $h_j \leftarrow P_j.length - 1$ 
11         $w_j \leftarrow \text{findWeightFactor}(h_j, U_j, \eta)$  (Equation(1))
12         $\bar{Destinations}.add(d_j, h_j)$ 
13       $\bar{Destinations}.sort(h_j)$ 
14      foreach  $l \in \bar{L}$  do
15         $x = 0$ 
16        foreach  $d_x \in \bar{D}_l$  do
17           $x = x + 1$ 
18           $M_x \leftarrow \text{findModulationFormat}(P_x)$  (Table 2)
19           $S_x \leftarrow \text{findNumFrequencySlots}(C_i, N_g,$ 
20             $C_{slot}, M_x)$  (Equation (2))
21           $g_x \leftarrow \text{countSpectrumContiguity}(P_x, \alpha_x + l, \omega_x)$ 
22          if  $g_x \geq S_x$  then
23             $\text{ALLOCATE}(R_j, \bar{F}, \bar{T})$ 
24             $\bar{U}_s.update()$ 
25            break
26          else if  $(t == \alpha_l + L) \ \&\& \ (x == k)$  then
27             $\text{BLOCK } R_i$ 
28            break

```

contiguous frequency-slots available with flexible start-time in the flexible window, or along the route to any of the replicas throughout the request's entire duration period, the request will be blocked (Line 23).

3.2. Runtime complexity analysis

Let R , F , T , E , k , L represent the number of requests, frequency-slots, time-slots, number of bi-directional fiber links, number of replicas in the system and size of the flexible window, respectively. The worst-case runtime complexity of the ALLOCATE subroutine is $O(F^2TE)$. Path computation may be calculated with complexity: $O(E + V \log V)$. The calculation of sorting request on Line 11 can be computed for a single request in $O(k \log k)$. Higher-order terms are introduced by the loops initiated on Line 3: $O(T)$, Line 4: $O(R)$, Line 6: $O(R)$, Line 8: $O(k)$, Line 14: $O(L)$, Line 16: $O(k)$. Accounting for these loops, the highest degree of complexity is introduced by the call to ALLOCATE on Line 22: $O(T^2F^2ERLk)$, the sorting of *Destinations*: $O(TRk \log k)$, and the path computation on Line 9: $O(RTk(E + V \log V))$. Simplifying terms, the worst-case runtime complexity of the anycast DSA algorithm can be expressed as $O(RTk(TF^2EL + V \log V + \log k))$.

4. Performance analysis

In this section, we quantitatively evaluate DSA for realistic traffic sets on the 14-node NSFnet topology in Fig. 4. We consider dynamically arriving AR traffic sets of 50,000 demands generated according to a Poisson distribution process with an average arrival rate of λ and an average reservation holding time of $1/\mu$. The network load is measured in Erlang (λ/μ). All source and destinations are uniformly distributed throughout the network. All other assumptions correspond to those listed in Table 2. All results given represent average values across 40 unique request sets with 95% confidence interval and are compared to a baseline Fixed ISA heuristic with no balanced routing capability. We have explored several categories of algorithmic variants. We have compared ISA and DSA with both fixed and flexible STSD. We have also investigated anycast and unicast requests and proposed balanced routing for anycast algorithm to reduce blocking.

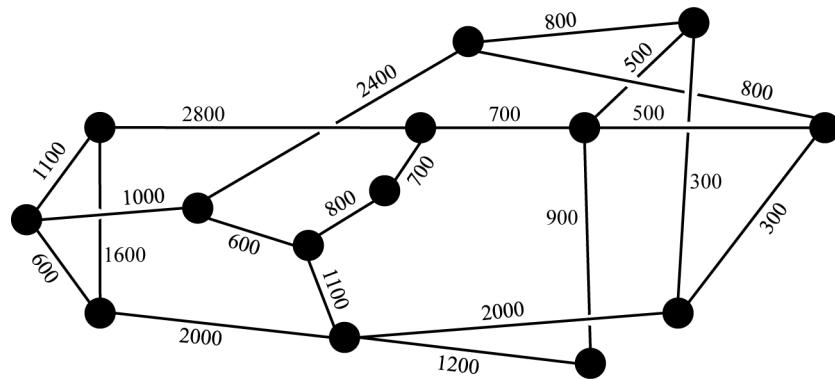


Fig. 4. 14-node NSFnet topology, with link weights given in km.

Figure 5(a) illustrates blocking probability of DSA and ISA with fixed and flexible STSD, when the size of flexible window is 10 time-slots. We consider the same window flexibility for starting the transmission that is applied in the DSA algorithm for ISA as our baseline to compare them fairly. We observe that DSA outperforms in both scenarios of fixed and flexible STSD requests. DSA with flexible window can decrease blocking that is due to fragmentation dramatically up to 66%. Flexible window of 10 time-slot leads to up to 31% blocking improvement in ISA. In

flexible STSD if the request is not able to be served after its book-ahead time, it can stay in the queue in hopes of finding available resources before the delay tolerance expires. We also investigated the impact of the flexible window's size on blocking probability. In Fig. 5(b) size of the flexible STSD window is increased to 50 time-slots. This figure shows blocking probability of DSA and ISA for 50 time-slots flexible STSD requests. Increasing the size of flexible window offers blocking reduction up to 99% in DSA and up to 85% in ISA, as giving more flexibility to requests to be started increases available resources and reduces blocking.

Figures 5(c) to 5(d) display average initial delay of DSA and ISA algorithms in flexible STSD with window size of 10 and 50 time slots, respectively. In the scenario of $L = 10$, the average initial delay in DSA is almost one unit of time longer than ISA. Note that the higher blocking improvement in $L = 50$ scenario leads to longer initial delay since requests are allowed to have up to 50 time slots delay.

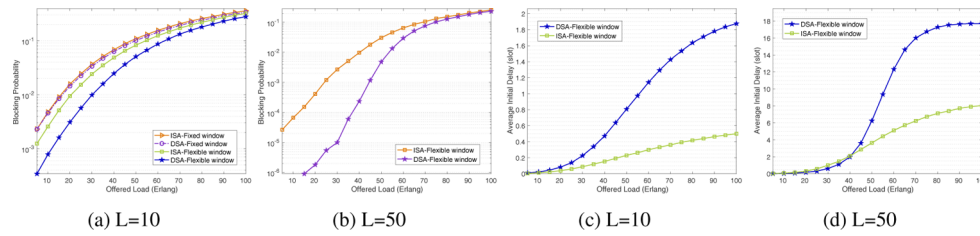


Fig. 5. Quantitative comparison of DSA and ISA scenarios of unicast traffic on the 14-node NSFnet.

The next category explored is adding anycast flexibility as another combinatorial option. In Figs. 6(a) and 6(b) we evaluate the blocking, assuming there are two and three replicas in the network, respectively. The availability of terminal flexibility via anycast conclusively has a great impact on lightpath establishment. Blocking is decreased as the number of anycast candidates grows. In all type of traffic, unicast and anycast 2/1 & 3/1 DSA-flexible has lower blocking comparing to ISA-flexible and also, DSA-fix performs better than ISA-fix window.

Figures 6(c) and 6(d) display the average initial delay of DSA and ISA algorithm with flexible 10 time-slots window when two and three replicas, respectively. Note that by increasing the number of replicas in the network, initial delay will be longer since initial delay is calculating for successful demands which is higher with more replicas in the network.

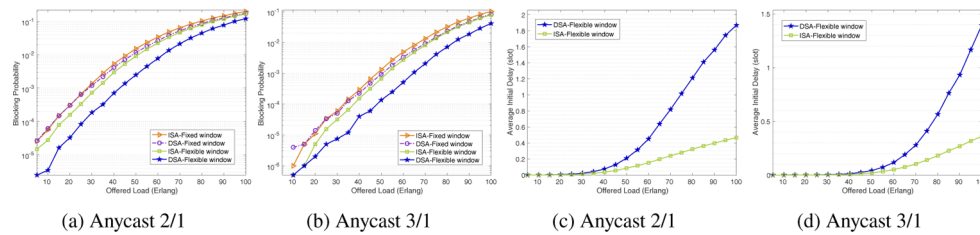


Fig. 6. Quantitative comparison of DSA and ISA scenarios (a), (b) Blocking, (c), (d) Delay of anycast traffic on the 14-node NSFnet..

Figures 7(a) to 7(c) show the blocking probability improvement of DSA and ISA in fixed and flexible window requests with $L = 10$ units of time for unicast, anycast 2/1 and anycast 3/1. We can observe that with greater number of replica DSA provides higher improvement in terms of blocking probability compared to ISA in both fixed and flexible scenarios. The reason for this is increasing the number of replicas improves network resource allocation efficiency. One additional replica leads up to 23% blocking improvement; however, increasing replicas from two

to three causes only about 5% blocking improvement. Since the results follow the similar trend for different value of L , we only plot those for $L = 10$ here.

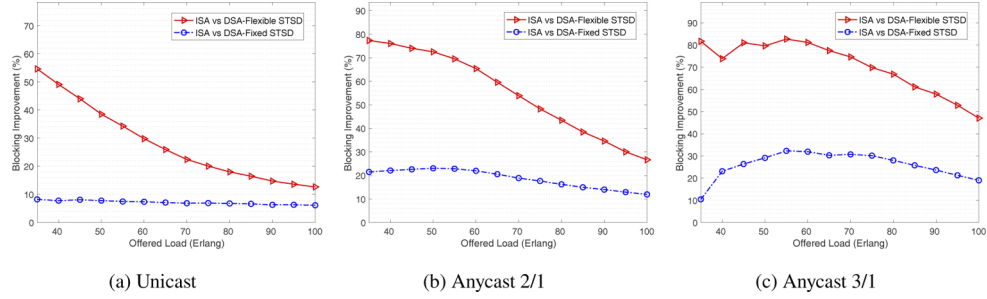


Fig. 7. Quantitative blocking improvement comparison of DSA vs ISA in fixed and flexible STSD ($L=10$).

We investigated the dynamic selection of least cost path to trade off between resource availability and number of hops of paths in anycast scenarios. Figures 8(a) to 8(d) show that blocking probability changes with various values of weight factor η . Figure 8(a) illustrates blocking probability of delayed spectrum allocation algorithm with fixed STSD, assuming there are two replicas in the network. Figure 8(b) displays blocking of DSA algorithm fixed STSD when three replicas are available in the network. Figure 8(c) and 8(d) show blocking of flexible STSD DSA when two and three replicas placed in the network, respectively.

We observe that when $\eta=0.8$, blocking probability is lowest for all these positions. Considering Eq. (1) η equal to 1 means that algorithm has picked the server with minimum hop path and η value equals to 0 means that the algorithm picks the replica with most available spectrum slots in its path. Therefore, the best result is obtained when we choose a data center between available slots in the minimum hop path.

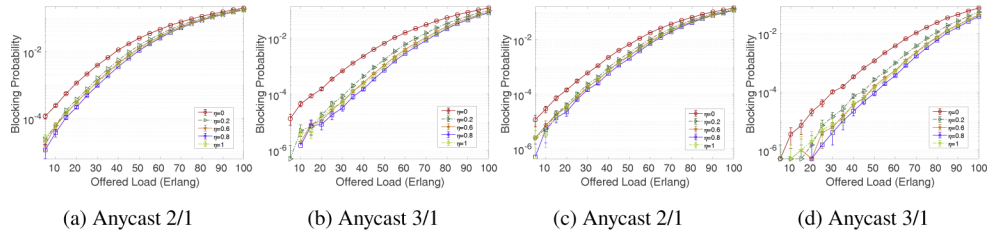


Fig. 8. Blocking probability of flexible and fixed STSD with balanced routing (a), (b) Fixed (c), (d) Flexible $L=10$ in DSA.

Figures 9(a) and 9(b) show average initial delay of DSA algorithm with flexible 10 time-slot window considering balanced routing when two and three replicas, respectively. Note that when $\eta=0.8$ initial delay will be shorter than the minimum hop scenario. The reason for this comes from consideration of link usage that helps in reducing total blocking and less requirement to postpone sending a request in order to prevent blocking of that request. In least spectrum usage ($\eta=0$) the initial delay will be longer for the same reason.

Figures 10(a) and 10(b) show the average number of hops traversed to reach from source to destination in two and three destinations anycast scenario with $L=10$. Least spectrum usage policy cause the higher number of hops that leads to higher delay as well. Minimum hop path and balanced routing with $\eta=0.8$ have very close numbers of hops. The difference between their number of hops is at most 0.12%.

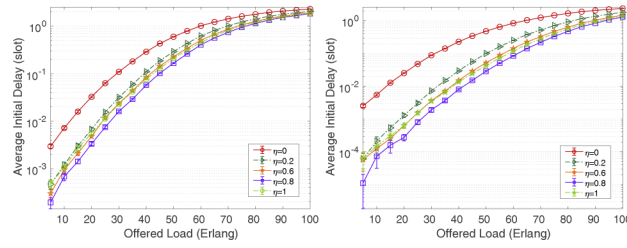


Fig. 9. Comparison of time delay in Flexible STSD DSA $L=10$ with balanced routing.

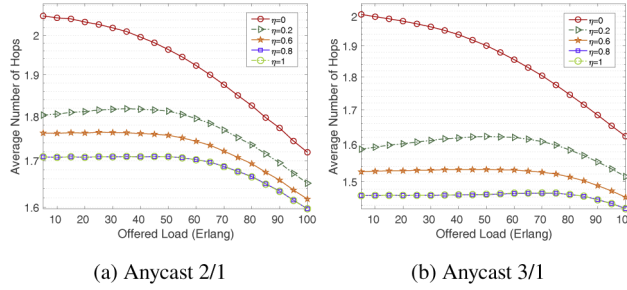


Fig. 10. Comparison of hop number in Flexible STSD DSA $L=10$ with balanced routing.

Figure 11 illustrates the comparison between average number of hops in balanced routing with $\eta=0.8$ in fixed and flexible window STSD with $k=1,2,3$. Increasing the number of replicas leads to fewer hop counts. Selected path traverses more hops, and use more resources in flexible window STSD. The reason is that in flexible window total blocking decreases and more requests are able to be served. Increasing the total requests in the network increases the number of hops for traversing.

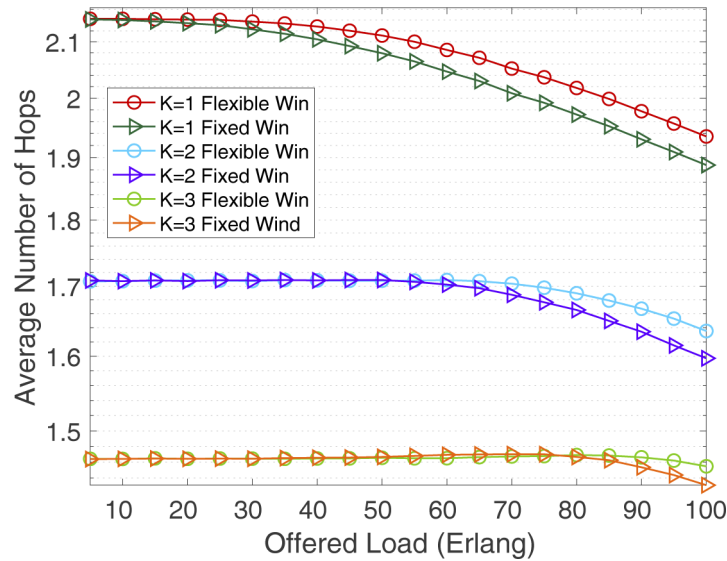


Fig. 11. Comparison of hop number in balanced routing when $\eta=0.8$ with Fixed and Flexible window ($L = 10$) DSA.

Figure 12 shows fragmentation of DSA in fixed and flexible window requests with $L = 10$ units of time for unicast and anycast 2/1. with greater number of replica DSA leads to lower fragmentation in both fixed and flexible scenarios. The reason for this is increasing the number of replicas improves network resource allocation efficiency. Only one additional replica leads up to 29% fragmentation improvement in average across the entire range of considered load values. Since the results follow the similar trend for different value of L , we only plot those for $L = 10$ here.

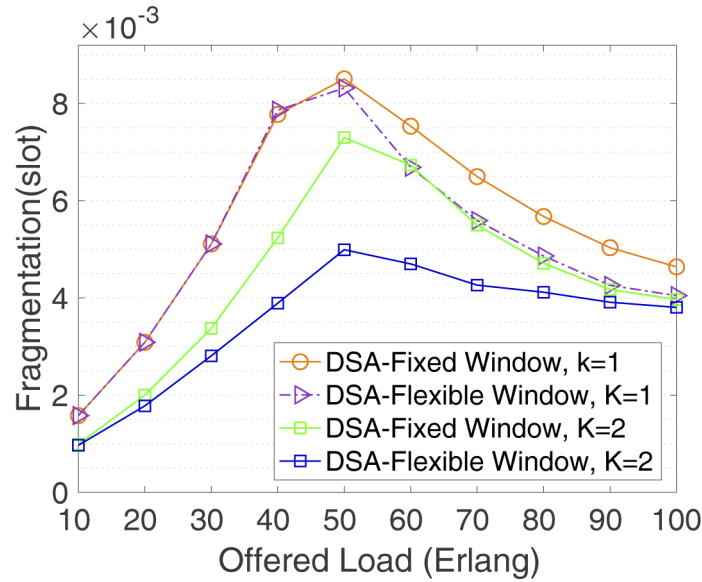


Fig. 12. Comparison of Fragmentation in balanced routing unicast and anycast 2/1 when $\eta=0.8$ with Fixed and Flexible window ($L = 10$) DSA.

5. Conclusion

In this work, we have investigated the problem of allocation of unicast and anycast traffic demands using the novel approach of DSA in elastic optical networks with fixed and flexible window STSD with balanced routing. We illustrate how the DSA technique can improve blocking of the network over traditional ISA. This improvement depends on the number of data centers in the network and increases by increasing the number of data centers. We can demonstrate that by using DSA delay of transmission increases however by decreasing the size of the flexible window the initial delay decreases. We show that by using a balanced routing approach we can decrease blocking without increasing delay. Also, we reach the best trade-off between minimum shortest hop count and spectrum availability in each path. In future work, we plan to apply modeling to optimize anycast flow in EON in the context of DSA.

Acknowledgment. This work has been supported by the NSF CARGONET project under grant CNS-1406370. Some portions of this work have been published in [33].

Disclosures. The authors declare that there are no conflicts of interest related to this article.

References

1. M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka, "Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies," *IEEE Commun. Mag.* **47**(11), 66–73 (2009).
2. C. Devellder, M. D. Leenheer, B. Dhoedt, M. Pickavet, D. Colle, F. D. Turck, and P. Demeester, "Optical networks for grid and cloud computing applications," *Proc. IEEE* **100**(5), 1149–1167 (2012).

3. J. Zhao, H. Wymeersch, and E. Agrell, "Nonlinear impairment-aware static resource allocation in elastic optical networks," *J. Lightwave Technol.* **33**(22), 4554–4564 (2015).
4. M. Jinno, B. Kozicki, H. Takara, A. Watanabe, Y. Sone, T. Tanaka, and A. Hirano, "Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network," *IEEE Commun. Mag.* **48**(8), 138–145 (2010).
5. X. Wan, N. Hua, and X. Zheng, "Dynamic routing and spectrum assignment in spectrum-flexible transparent optical networks," *J. Opt. Commun. Netw.* **4**(8), 603–613 (2012).
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.* **31**(1), 15–22 (2013).
7. X. Wang, K. Kuang, S. Wang, S. Xu, H. Liu, and G. Liu, "Dynamic routing and spectrum allocation in elastic optical networks with mixed line rates," *J. Opt. Commun. Netw.* **6**(12), 1115–1127 (2014).
8. K. Christodouloupoulos, I. Tomkos, and E. Varvarigos, "Corrections to "elastic bandwidth allocation in flexible ofdm-based optical networks" [may 11 1354-1366]," *J. Lightwave Technol.* **29**(12), 1899 (2011).
9. K. Christodouloupoulos, P. Soumplis, and E. Varvarigos, "Planning flexible optical networks under physical layer constraints," *J. Opt. Commun. Netw.* **5**(11), 1296–1312 (2013).
10. 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. Select. Areas Commun.* **31**(1), 26–38 (2013).
11. B. C. Chatterjee, N. Sarma, and E. Oki, "Routing and spectrum allocation in elastic optical networks: A tutorial," *IEEE Commun. Surv. Tutorials* **17**(3), 1776–1800 (2015).
12. B. C. Chatterjee, S. Ba, and E. Oki, "Fragmentation problems and management approaches in elastic optical networks: A survey," *IEEE Commun. Surv. Tutorials* **20**(1), 183–210 (2018).
13. A. Castro, L. Velasco, M. Ruiz, M. Klinkowski, J. P. Fernández-Palacios, and D. Careglio, "Dynamic routing and spectrum (re) allocation in future flexgrid optical networks," *Comput. Netw.* **56**(12), 2869–2883 (2012).
14. J. Kuri, N. Puech, M. Gagnaire, E. Dotaro, and R. Douville, "Routing and wavelength assignment of scheduled lightpath demands," *IEEE J. Select. Areas Commun.* **21**(8), 1231–1240 (2003).
15. C. Lee and J. K. Rhee, "Efficient design and scalable control for store-and-forward capable optical transport networks," *J. Opt. Commun. Netw.* **9**(8), 699–710 (2017).
16. X. Lin, W. Sun, M. Veeraraghavan, and W. Hu, "Time-shifted multilayer graph: A routing framework for bulk data transfer in optical circuit-switched networks with assistive storage," *J. Opt. Commun. Netw.* **8**(3), 162–174 (2016).
17. N. Charbonneau and V. M. Vokkarane, "A survey of advance reservation routing and wavelength assignment in wavelength-routed WDM networks," *IEEE Commun. Surv. Tutorials* **14**(4), 1037–1064 (2012).
18. L. Contreras, V. Lopez, O. De Dios, A. Tovar, F. Munoz, A. Azanon, J. Fernandez-Palacios, and J. Folgueira, "Toward cloud-ready transport networks," *IEEE Commun. Mag.* **50**(9), 48–55 (2012).
19. <https://www.akamai.com>.
20. D.-R. Din, "A hybrid method for solving arwa problem on wdm network," *Comput. Commun.* **30**(2), 385–395 (2007).
21. A. Shaikh, J. Buysse, B. Jaumard, and C. Devellder, "Anycast routing for survivable optical grids: Scalable solution methods and the impact of relocation," in *IEEE/OSA Journal of Optical Communications and Networking*, vol. 3 (2011), pp. 767–779.
22. K. Walkowiak, A. Kasprzak, and M. Klinkowski, "Dynamic routing of anycast and unicast traffic in elastic optical networks," *Communications (ICC), 2014 IEEE International Conference on* pp. 3313–3318 (2014).
23. K. Walkowiak and M. Klinkowski, "Joint anycast and unicast routing for elastic optical networks: Modeling and optimization," in *Communications (ICC), 2013 IEEE International Conference on*, (2013), pp. 3909–3914.
24. M. Aibin and K. Walkowiak, "Dynamic routing of anycast and unicast traffic in elastic optical networks with various modulation formats ; trade-off between blocking probability and network cost," in *High Performance Switching and Routing (HPSR), 2014 IEEE 15th International Conference on*, (2014), pp. 64–69.
25. M. Aibin and K. Walkowiak, "Adaptive modulation and regenerator-aware dynamic routing algorithm in elastic optical networks," in *Communications (ICC), 2015 IEEE International Conference on*, (2015), pp. 5138–5143.
26. K. Walkowiak, M. Kucharzak, P. Kopeć, and A. Kasprzak, "ILP model and algorithms for restoration of anycast flows in elastic optical networks," in *Reliable Networks Design and Modeling (RNDM), 2014 6th International Workshop on*, (2014), pp. 102–108.
27. R. Goscienn, K. Walkowiak, and M. Klinkowski, "Joint anycast and unicast routing and spectrum allocation with dedicated path protection in elastic optical networks," in *Design of Reliable Communication Networks (DRCN), 2014 10th International Conference on the*, (2014), pp. 1–8.
28. W. Lu and Z. Zhu, "Dynamic service provisioning of advance reservation requests in elastic optical networks," *J. Lightwave Technol.* **31**(10), 1621–1627 (2013).
29. H. Chen, Y. Zhao, J. Zhang, R. He, W. Wang, J. Wu, Y. Wang, Y. Ji, H. Zheng, Y. Lin, and B. Hou, "Time-spectrum consecutiveness based scheduling with advance reservation in elastic optical networks," *IEEE Commun. Lett.* **19**(1), 70–73 (2015).
30. A. Somani, V. M. Vokkarane, and B. H. Ramaprasad, "Dynamic advance reservation with delayed allocation over wavelength-routed networks," in *Transparent Optical Networks (ICTON), 2011 13th International Conference on*, (IEEE, 2011), pp. 1–5.
31. P. Afsharlar, A. Deylamsalehi, J. M. Plante, J. Zhao, and V. M. Vokkarane, "Routing and spectrum assignment with delayed allocation in elastic optical networks," in *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9 (2017), pp. B101–B111.

32. We assume that the same contiguous span of frequency-slots must be provisioned at each time-slot of the request duration for successful reservation.
33. P. Afsharlar, A. Deylamsalehi, and V. M. Vokkarane, "Delayed spectrum allocation for anycast advance reservation with flexible window in elastic optical networks," *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)* pp. 1–6 (2016).