Dynamic Continuous and Non-Continuous Advance Reservation in SLICE Networks

Bharath H. Ramaprasad
NetApp, Inc., Waltham, MA
Email: bharath.ramaprasad@netapp.com

Thilo Schöndienst and Vinod M. Vokkarane
ECE Department, University of Massachusetts, Lowell, MA
Email: thilo_schoendienst@student.uml.edu, vinod_vokkarane@uml.edu

Abstract—Today, large amounts of data, growing from terabytes to petabytes, need to be transferred across the globe in a timely manner. To accommodate this efficiently, network operators are augmenting their networks to 100 Gbps links and beyond. The spectrum-sliced elastic optical path network (SLICE) architecture enables accommodation of variable-rate data traffic in a highly spectrum-efficient manner. The blocking in SLICE networks observed in literature is much lower than on traditional fixed grid WDM networks. Still, the current routing and spectrum allocation (RSA) algorithms have potential for blocking performance improvements. To increase the spectral efficiency, we introduce a concept called routing, spectrum, and segment allocation (RSSA). We propose several novel continuous and non-continuous time heuristics based on the concept of RSSA. We consider dynamic advance reservation traffic requests over SLICE networks and evaluate the performance of the proposed RSSA heuristics. Through extensive simulations, we observe that our continuous-time, non-contiguous spectrum RSSA heuristics outperform existing heuristics in terms of both blocking probability and spectral utilization.

Index Terms—WDM, OFDM, SLICE, Flexible Spectrum Assignment, Routing and Spectrum Assignment (RSA).

I. INTRODUCTION

Currently, wavelength-routed networks require allocation of a full bandwidth per connection request even when the request’s bandwidth demand is not sufficient to fill the entire capacity of the wavelength. Wavelength-level granularity leads to inefficient capacity utilization, a problem expected to become even more significant with the deployment of higher capacity WDM networks of 100 Gbps per channel and more. To solve this problem, recently, a spectrum efficient and scalable optical network architecture called spectrum-sliced elastic optical path network (SLICE) has been proposed. SLICE [1] uses optical orthogonal frequency-division multiplexing (OFDM) based bandwidth-variable (BV) transponders and BV wavelength cross-connects (WXC) as its enabling technologies. When a connection request arrives at a SLICE network, the request must be routed over the physical topology based on some routing policy and also assigned a spectrum sufficient in bandwidth along the entire route. The spectrum is defined in terms of frequency or subcarrier slots. This is known as the NP-complete routing and spectrum allocation (RSA) problem [2]. The spectrum slots reserved for a single request are generally required to be adjoining, this is referred to as contiguous spectrum reservation. Additionally, no two requests can use the same spectrum on the same link at the same time. As dynamic requests arrive in the network new spectrum-paths (frequencies on selected physical paths) must be allocated. If an arriving request cannot find a spectrum-path, the request is rejected or blocked.

We investigate advance reservation (AR) [3] requests for SLICE for a number of applications. The most time critical applications include real time video and telecommunication. The start and end time as well as stringent bandwidth requirements must be met. For data transfers, the critical point is meeting the transmission deadline. Bandwidth and start time, as well as interruptions do not affect the outcome. In addition to allowing flexible start times, we allow non-continuous data transmission. That is, the sending application may pause transmission for a period of time and then resume at some point in the future. The request further specifies a bandwidth requirement which is constant throughout the transmission.

SLICE breaks the request into one or more segments, which can be transmitted on different spectrum-paths, instead of provisioning one large request over a single spectrum-path. Traditional advance reservation can lead to resource fragmentation where small voids in the frequency domain can occur and are unlikely to be utilized efficiently. Therefore we propose effective bin-packing policies in the continuous-time transmission RSA algorithms to help reduce fragmentation and fill these voids: routing, spectrum, and segment allocation (RSSA). We propose novel heuristics for both continuous and non-continuous fixed and flexible advance reservation RSSA mechanisms for contiguous/non-contiguous spectrum constraints. We show that the proposed heuristics outperform existing continuous and non-continuous fixed and flexible advance reservation RSA heuristics. To the best of our knowledge, we are the first to analyze the performance of continuous and non-continuous time routing and spectrum assignment for dynamic traffic. We also consider both contiguous and non-contiguous spectrum constraints for bandwidth-sensitive, duration-based requests for SLICE networks.

A. Related work

There are two defining characteristics of advance reservation requests. First, the holding time must be declared or must be able to be derived from the supplied information. Second, the deadline, the time the request must be completed by, must be later than then request’s arrival time plus the holding time. In [3] the authors present a survey on solutions to the RWA problem in AR capable WDM networks. Variable bandwidth advance reservation (VBAR) has been previously proposed in [4]. Although the work presented is not directly related to SLICE networks, the authors give a perspective of volume based advance reservations on an overlay network. Their work primarily involves reserving time variant bandwidth over the duration of a connection. Christodouloupolos et al. propose routing and spectrum allocation (RSA) problem in an OFDM-based elastic
We discuss our proposed heuristics in Section III. We present a time-slotted network. We assume that the limits how far ahead advance reservations can be made in the network. The value of the horizon, \( U \), is updated after each request is processed. The state information consists of which timeslots are used on all subcarrier frequencies on all edges: \( U [E, F, H] \).

2) Advance Reservation Requests: We consider duration-based requests (\( R \)) which are typically found in large-scale real-world networks. These requests demand a definite amount of bandwidth for a definite amount of time. These requests typically stem from delay sensitive applications, like video broadcast and video-on-demand, and have a specific bandwidth demand for a fixed duration. The user request, \( R \), can be defined as a 6-tuple, as shown in Table I: \( s \in V \) is the source node, \( d \in V \) is the destination, \( \alpha, \omega \in H \) denotes the earliest start time and deadline of the request, \( f_{\text{s slots}} \) specifies the required bandwidth in terms of frequency slots, \( \tau \) is the required duration in timeslots. A frequency slot is the smallest unit of bandwidth defined as slots of the spectrum. We assume \( \alpha \) is inclusive and \( \omega \) is exclusive, i.e., we can use slots \([\alpha, \omega)\) for transmission. We assume that the data is readily available at the source at the start time. We also interpret the deadline as the time when the transmission has to be completed, thus it is excluded. A specified time and specified duration (STSD)-fixed reservation has \( \omega = \alpha + \tau \), meaning the request must start at \( \alpha \) to finish by \( \omega \). We define the flexibility of a request as \( \frac{\omega - \alpha}{\tau} - 1 \). Other works have defined STSD-flexible as \( (s, d, \ell, \ell', r, \tau) \), where \( s, d, \tau \) carry the same meanings, but \([\ell, r] \) represent the window of possible start times. In our notation, \( \ell = \alpha \) and \( r = \omega - \tau \). Instead of generating a single route and spectrum for a given request as in traditional RSA problems, our heuristics generate a schedule of one or more segments. We refer to this as the routing, spectrum, and segment assignment (RSSA) problem.

Depending on the traffic type, we need to choose an appropriate heuristic. As shown in Table I the requests are defined by the same tuple, but their provisioning constraints differ.

a) Continuous: For continuous advance reservation for a request \( R \), the first constraint states that the number of segments in the schedule cannot exceed the duration implying that there can be a maximum of \( \tau \) segments in \( S \). The second constraint states that the number of subcarrier frequencies for each segment in the schedule should match the number of frequency slots requested. The final constraint specifies that the segment start time can fall anywhere between the left and the right windows of the request.

b) Non-continuous: For non-continuous advance reservation for a request \( R \), the first constraint states that the request can start any time after the start time and before the end of the window. The second constraint states that any segment found cannot violate the deadline. The third constraint states that all the segments duration must sum up to the requested duration specified in the request. The fourth constraint states that the found segments should increment in time. The final constraint states that the number of subcarrier frequencies for each segment in the schedule should match the number of frequency slots requested.

3) An Example: We present an example of the RSSA problem for duration-based requests, \( R \), in Fig. 1. Assume that there is a single link path available between a particular

---

### TABLE I

<table>
<thead>
<tr>
<th>Traffic Type</th>
<th>Request (( R ))</th>
<th>Constraints*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous</td>
<td>((s, d, \alpha, \tau, f_{\text{s slots}}, \omega))</td>
<td>( 1 \leq</td>
</tr>
<tr>
<td>Advance</td>
<td>Reservation</td>
<td>(</td>
</tr>
<tr>
<td>Non-Continuous</td>
<td>((s, d, \alpha, \tau, f_{\text{s slots}}, \omega))</td>
<td>( 1 \leq</td>
</tr>
</tbody>
</table>

---

*Assume the schedule vector \( S \) has \( n \) elements (from 1 to \( n \)).
source and destination in a given network. Fig. 1 (a) shows the current frequency availability for a snapshot of time from \( t_1 \) to \( t_3 \) for the only link on the path between source and destination. Let us consider that an AR request is submitted to the network. The request is as follows: \((s, d, t_1, 2, 2, t_3)\) which follows the request prototype for continuous AR as specified in Table I. As we can observe from Fig. 1 (a) there are no two frequency slots available which are both continuously available in time for 2 timeslots and having 2 contiguous subcarriers adjacent to each other in the frequency domain. Thus traditionally the request would be blocked. Introducing a continuous time and non-contiguous spectrum provisioning approach this request will not be blocked. The request will be accepted by selecting the non-contiguous subcarrier frequency set, \( \{ f \} = \{ f_0, f_2 \} \) while being continuous in time across \( t_1 \) and \( t_2 \) as shown in Fig. 1 (b). Now consider that another request of type \( R \) follows the just accepted request with the following demand: \((s, d, t_1, 2, 2, t_3)\). We can observe from Fig. 1 (c) that neither contiguous spectrum nor non-contiguous spectrum with continuous time transmission, can possibly, successfully accept this request. We tackle this challenge by introducing: non-contiguous transmission with contiguous or non-contiguous spectrum. We assign the subcarrier frequency set, \( \{ f \} = \{ f_3, f_4 \} \) while being non-contiguous in time at \( t_1 \) and \( t_3 \) as shown in Fig. 1 (d).

Consider Fig. 2(a). Assume that there is an AR request demanding 2 frequency slots for 3 duration timeslots between \( t_1 \) and \( t_3 \). We can observe from Fig. 2(a) that there are no two frequency slots continuously available across \( t_1 - t_3 \) and none of the above two approaches can be used to successfully accept the current request. Assuming that future SLICE hardware will be capable of flexible tuning, we suggest an extension to the above approaches: spectrum switching. As shown in Fig. 2(b) the request is successfully accepted by creating two segments, \( \{ f_2, f_3 \} \) from \( t_1 \) to \( t_2 \) and \( \{ f_3, f_4 \} \) at \( t_3 \).

4) Resource allocation/The schedule: Upon arrival of a request, the scheduler must allocate resources to the request. The scheduler will then return a vector \( S \) containing a number of segments, \( S \), called the schedule, is defined as \( \{(t_1, \tau, P_1, \{ f \}_1), (t_2, \tau, P_2, \{ f \}_2), \ldots, (t_i, \tau, P_i, \{ f \}_i)\} \). In this work, we use the term spectrum-path for SLICE networks which is a combination of a route and a spectrum (frequency range). Further, we define a segment as a spectrum-path used to transfer data between a specified start and end time. A segment can be defined as a four-tuple, \((t, d, P, \{ f \})\), where \( t \) is the start time, \( d \) is the duration, \( P \) is the path, and \( \{ f \} \) is a set consisting of subcarrier frequency ranges.

We assume that \( t \) refers to a specific timeslot and \( d \) is specified in number of timeslots. The start time is inclusive, so the segment transmits data from \([t, t + d - 1]\). Each segment either follows the spectrum continuity constraint or not on all links, based on whether the bandwidth variable (BV) transponders and BV WXCs are capable of transmitting and switching on non-contiguous subcarrier frequencies. If the hardware offers this possibility, we refer to them as spectrum path switch (SPS). Given this general description of the scheduler, we can define different traffic types for the duration-based request as shown in Table I.

**Definition:** RSSA: Given a network, \( G = (V, E, F, H) \), its current state, \( U[E, F, H] \), and an incoming request, \( R = (s, d, \alpha, \tau, f_{\text{slots}}, \omega) \) or a subset of this request in terms of number of parameters, we must return a schedule, \( S = \{(t_1, \tau, P_1, \{ f \}_1), (t_2, \tau, P_2, \{ f \}_2), \ldots, (t_i, \tau, P_i, \{ f \}_i)\} \), if the request can be accommodated, or reject the request and mark is as blocked otherwise. The schedule is subject to...
the constraints specified in Table I. The segments should be selected in a manner that reduces blocking of future requests.

The spectrum continuity constraint as discussed earlier, can be enforced or relaxed for duration-based requests depending on whether the underlying hardware (BV transponders and BV WXCs) support the transmission and switching of non-contiguous sub-carrier frequencies. This non-contiguous approach greatly improves flexibility but potentially requires more transponders to provision many small spectrum segments. Assuming that the response from the network is successful, the client application will transmit the data during the specified intervals on the specified route and spectrum according to the segments in the schedule returned by the network.

III. CONTINUOUS AND NON-CONTINUOUS TIME AND SPECTRUM TRANSMISSIONS HEURISTICS

A. SLICE-AR Network Model

In this section, we discuss our novel RSSA heuristics with static route computation. For each source-destination pair, we have pre-computed $k$ shortest paths using Yen’s algorithm, which finds $k$ loop-less paths while not being necessarily disjoint [15]. We have a total of $F \times k$ spectrum-paths for each request and a total of $\omega - \alpha$ timeslots that can be used on any of the spectrum-paths for any given request. Our heuristics presented in the following subsections compute the spectrum-path availability information. We store this spectrum-path availability information in a table, $A[P,F,H]$, where $P$ is a path, $F$ is a frequency set, and $H$ is the time-domain information. It is computed for each arriving request and each entry stores the number of timeslots available starting at the specific time for the given spectrum-path. Our heuristics scans all valid starting timeslots on all spectrum-paths and selects a segment based on a segment selection policy (SSP).

We have explored various segment-selection (bandwidth-packing) policies that assign a cost to each segment and choose the one with the least cost. These policies are taking into account combinations of frequency index and spread, temporal order of segments and frequencies already assigned, path congestion, starting time and duration of transmission. The Earliest Frequency First policy (EFF) assigns a cost by assigning the lowest index frequency subcarrier first. Thus the EFF policy is defined as $EFF(t,d,P,W) = f$.

The goal of the segment selection policies is to help select segments that will lower blocking probability for future requests entering the network. The complexity for all examined segment selection policies, is $O(Fk(\omega - \alpha - \tau))$. The computing function $A$ (line 6 in Algorithm 1) finds matching candidate schedules from the table $A$. The lookup takes $O(VFk(\omega - \alpha))$.

### Algorithm 1: Continuous-Time RSSA

**Input:** $R = (s, d, \alpha, \tau, f_{\text{slots}}, \omega)$, Segment Selection Policy $SSP$, $G = (V, E, W, H)$

**Output:** Schedule, $S = \{(t, \tau, P, F)\}$

1. $schedule = \phi$
2. $tempSchedule = \phi$
3. for $k = 1$ to $K$
4.     for $t = \alpha$ to $\omega - \tau$
5.         for $f = 0$ to $F$
6.             if $A[P_k, f, t] \geq \tau$
7.                 and $SSP(t, \tau, P_k, f)$ is lowest cost then
8.                     Add segment $(t, \tau, P_k, f)$ to temp_schedule
9.     if $temp_schedule \geq 0$ then
10.        if Spectrum_Continuity_Check $\equiv$ true then
11.            for $s = 0$ to $|temp_schedule|$
12.                if $s > 0$ and $temp_schedule_s \{f\}$
13.                    $\equiv temp_schedule_{s-1}\{f\} + 1$ then
14.                        schedule_s = temp_schedule_s
15.                    break
16.            else
17.                clear schedule
18.        else
19.            for $s = 0$ to $f_{\text{slots}}$
20.                schedule_s $\equiv temp_schedule_s$
21. return schedule

As an area of future work, we will combine these two steps to make the heuristics faster. Each novel RSSA heuristic we discuss below, is designed in a way that it can handle two scenarios. The first being, when the spectrum-continuity constraint is enforced and the second when the spectrum-continuity constraint is relaxed which is referred to as non-contiguous spectrum throughout the text.

B. Continuous-Time RSSA (with Spectrum Switches) for Advance Reservation Requests

The problem formulation of this heuristic is as discussed in Table I. The approach for the continuous-time RSA for duration-based heuristic is the same as represented and explained earlier in Fig. 1(a) and (b). The algorithm for continuous-time RSA for duration-based advance reservation, is as shown in Algorithm 1. Initially in lines 3 to 7 we try to create a temporary schedule within the given constraints. In line 6 the segments are chosen according to the segment selection policy. After finding a feasible schedule we may chose to enforce the optional additional spectrum continuity constraint. This is done starting in line 9, by checking the candidate schedules in temp_schedule for continuity.

If we are assuming availability of spectrum switches (as explained in the example earlier and illustrated in Fig. 2) the algorithm is modified by allowing any frequency slot to be used (loop in line 5 and arguments in lines 6 and 7). The SSP
is responsible to prevent unnecessary spectrum switches, and therefore segment fragmentation.

C. Non-Continuous-Time RSSA (with Spectrum Switches) for Advance Reservation Requests

The problem formulation of this heuristic is as discussed in Table I. The Non-Continuous-Time RSA for duration-based heuristic is illustrated and explained earlier in Fig 1(c) and (d). The algorithm for non-continuous-time RSA for duration-based advance reservation is not shown for space considerations. The algorithm for this heuristic is similar to Algorithm 1 except that in the loop from $\omega$ to $\omega - \tau$ (line 4) we allow discontinuities: the request may be interrupted for single or multiple segments as long as it can finish within its given deadline. Another difference in the implementation is that we do not compute all the possible candidate segments in advance. Rather, we call a subroutine at every iteration of the loop, to generate segments to fill as much of the remaining duration as possible. Here too, the presence of spectrum switches nullifies the need for the segments to be of the same frequency slot.

IV. PERFORMANCE EVALUATION

We now discuss the performance of our proposed continuous and non-continuous time RSSA heuristics when compared to the baseline RSA heuristic. The selection of number of subcarrier frequencies assumed to provide the bandwidth in terms of frequency slots is based upon the underlying hardware and the average network traffic expected. Also the timeslot size largely depends on the type of traffic a network operator expects. The number of subcarrier frequencies dynamically added to the central-carrier frequency is also dependent upon the modulation format/level chosen which expands or contracts the bandwidth provided per subcarrier frequency. We plan to include per request-adapted modulation in our future work. To simplify the analysis, we assume that the flexibility ($\frac{\tau - \alpha}{\tau}$) for each request is set to one. This means that if a request has a duration of $\tau$ slots, it has $\tau$ valid start times beginning at $\alpha$. Our primary performance metric will be blocking probability, which is defined as the ratio of the number of blocked AR requests to the total number of AR requests. We also observe the average number of spectrum-path switches necessary for each successfully accepted request, averaged over one simulation instance. Lastly, we define the start delay which is essentially the time difference between the actual transfer time and the ideal transfer time. We simulate $10^6$ requests and present the average of six randomly generated traffic sets. We use a Poisson arrival process and holding time distribution is exponential. The book-ahead time as noted earlier is uniform for each request at uniformly random timeslots well within the horizon time range. The horizon is set to 2000 timeslots. We evaluate our heuristics over the 14-node, 21-link NSF network shown in Fig. 3.

As a baseline comparison we consider duration-based continuous-time, contiguous spectrum constraint RSSA with first fit spectrum assignment. We compare this baseline heuristic with the proposed continuous-time, contiguous and non-contiguous spectrum with both spectrum switches and spectrum-path switches (allowing a change in spectrum and path in the segment selection process) RSSA heuristics. For $k = 2$ the proposed RSSA heuristics show a significant performance benefit in terms of reduced blocking probability as shown in Fig. 4(a). The blocking performance increases by about 20% with flexibility when compared to the proposed heuristics without flexibility. The improvement over single shortest path routing ($k=1$) was found to be between 85% and 100% (not shown). Fig. 4(b) shows that due to flexibility the average starting delays of the request is not at 0, which would be the case if it was STSD-fixed window. The small values are due to the fact that we simulate $10^6$ requests and then calculate the average; most requests can efficiently be scheduled at the original start time. We observe from Fig. 4(b) that the non-contiguous spectrum variants have the least amount of start delay as they scout each and every frequency slot available before scanning the next path or timeslot. This is reflective of why the network does not get congested (low blocking probability) even at higher loads for the non-contiguous spectrum variants as shown in Fig. 4(a).

Now, we discuss the performance of the non-continuous-time heuristics. From Fig. 5(a), we observe that the NonContinuous-Time RSSA with Spectrum-Path Switches with Non-Contiguous Spectrum heuristic has 65 to 80% lower blocking when compared to that of its Contiguous spectrum variant for both $k = 1$ and $k = 2$ considered in pairs. We can also observe that NonContinuous-Time RSSA with Spectrum-Path Switches with NonContiguous Spectrum and $k = 2$ is 80-100% better in terms of blocking probability when compared
to the baseline (extended by Spectrum Path switching capability for fair comparison). From Fig. 5(b) we can observe that the NonContinuous-Time RSA with Spectrum-Path Switches with NonContiguous Spectrum and \( k = 2 \) has the least spectrum-path switches with nearly 10 times lower spectrum-path switching when compared to NonContinuous-Time RSA with Spectrum-Path Switches with Contiguous Spectrum and \( k = 1 \).

V. CONCLUSION

The proposed continuous and non-continuous heuristics for both contiguous and non-contiguous spectrum constrained duration-based requests performed significantly better than their baselines by about 50 - 60%. The heuristics for duration-based AR requests show nearly 60% - 70% blocking performance improvement when the contiguous spectrum constraint is relaxed when compared to when the contiguous spectrum constraint is enforced. Overall, the proposed RSA heuristics significantly outperforms the existing RSA heuristics in terms of blocking performance to an extent of about 70% - 100%. For the benefit of non contiguous transmission more transponders are needed for provisioning a large number of narrow bandwidth segments. The effects of using expensive and complex spectrum-path switches for the proposed RSA heuristics are negligible. The proposed RSA heuristics reduces fragmentation of bandwidth and thereby increases spectrum utilization in terms of frequency slots significantly.

The next steps in our research will be to extend and study the performance of the heuristics to handle immediate reservation requests. We will also address the important topics of quality of service, survivability and route adaptive modulation. The latter will be part of our efforts to reduce network energy consumption, an important step to improve the already efficient technology SLICE.

ACKNOWLEDGMENT

This work has been supported by the Department of Energy (DOE) COMMON project under grant DE-SC0004909 and the NSF CARGONET project under grant CNS-1406370.

REFERENCES


Fig. 5 – Results for Non Continuous-Time RSA with Spectrum-Path Switches for EFF.