Virtual Network Mapping in Elastic Optical Networks with Advance Reservation

Juzi Zhao
Department of Electrical Engineering
San Jose State University
San Jose, USA
juzi.zhao@sjsu.edu

Suresh Subramaniam
Department of Electrical and Computer Engineering
The George Washington University
Washington, USA
suresh@gwu.edu

***Abstract*—Elastic optical networks are promising candidates for next-generation backbone networks due to their capability to efficiently and flexibly allocate optical bandwidth to demands with heterogeneous requirements. Scientific applications involving data processing, analysis, and transmission among multiple data centers can be modeled as virtual network requests. For delay-tolerant applications, the virtual network request has a specific deadline and is successfully mapped as long as sufficient resources are allocated to it before the deadline. In this paper, we investigate the dynamic virtual network mapping problem for advance reservation virtual network requests in elastic optical networks. We propose a stop-and-resume allocation scheme to map virtual network requests in order to satisfy deadline constraints and resource requirements, where the service duration of a request is divided into periods with different resources being allocated in different periods. Simulation results are presented to show the effectiveness of the proposed approach compared with a baseline algorithm in terms of request blocking ratio.**

***Index Terms*—Elastic optical networks, virtual network mapping, resource allocation, advance reservation.**

I. Introduction

Next generation backbone networks are required to efficiently support various applications with high traffic demands. The flexible resource provisioning and efficient bandwidth allocation make elastic optical networks (EONs) a promising candidate. In EONs, the huge optical bandwidth is divided into fine-grained frequency slots, and only the required number of frequency slots are allocated to each demand based on its bandwidth requirement. In addition, the most suited modulation format can be assigned to each demand, and this increases spectrum utilization [1], [2].

Network virtualization enables various virtual networks with different topologies and requirements to share resources of the same substrate physical network, resulting in more efficient resource utilization, low complexity of management, better reliability, and scalability. Each virtual network (VN) request can be represented as a logical graph including a set of virtual nodes and a set of virtual links. Each virtual node requires computing resources from physical nodes in the network (as well as storage resources, but we subsume storage under computing resources in this work), while each virtual link requires bandwidth from physical links. One crucial task in network virtualization is assigning spectrum and computing resources to the virtual network requests so that the total resources are minimized, which is called the virtual network mapping/embedding problem (VNM/VNE) and is proved to be NP-hard [3]. Scientific applications involving data processing, analysis, and transmission among multiple data centers can be modeled as VN requests. Some of these applications are delay-tolerant and can be scheduled at a future time as long as the required resources can be allocated before a specified deadline. Such VN requests are called Advance Reservation VN requests. Examples of such applications include database backup, VM migration, scientific data analysis (where service duration and deadline can be anticipated in advance), etc.

A. Related Work

VNM/VNE in EONs is a topic that has attracted much research attention recently. Static and dynamic versions of a VNM problem are investigated in [4]. The authors of [5] propose novel algorithms to minimize resource usage for virtual optical multicast tree mapping. In [6], an ILP formulation and two time-efficient algorithms are proposed to address cost-efficient VNM provisioning in a multidomain EON. The authors of [7] address the static version of survivable VNM problem in EONs with the objective of minimizing cost of working and backup resources. In [8], the VNM problem is investigated in software-defined EONs to reduce the blocking probability. A mixed-strategy gaming based approach is proposed in [9] to allocate resources to virtual network function service chains. The authors of [10] propose the flexible VNM by assigning different modulation formats, Forward Error Correction (FEC) overheads, and baud-rates to virtual network requests; in addition, virtual link splitting is considered.

There is also some research on advance reservation in EONs. The authors of [11] study dynamic version of advance reservation allocation for unicast requests. Delayed spectrum allocation scheme is proposed in [12] to allocate network resources to anycast requests. The authors of [13] propose ILP formulation and heuristics for both immediate reservation and advance reservation requests with survivability consideration.

B. Contributions

In this paper, we consider a dynamic version of the VNM problem for advance reservation VN requests in EONs. The requests arrive to and depart from the network randomly. Each

978-1-7281-5089-5/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: The George Washington University. Downloaded on September 30,2020 at 16:27:16 UTC from IEEE Xplore. Restrictions apply.

request includes a set of virtual nodes interconnected by a set of virtual links. It also specifies a deadline and a service duration (i.e., the duration the VN is active). Each virtual node has a computing resource requirement and each virtual link has a bit rate requirement. Virtual network request blocking ratio is considered as the network performance metric. Based on our best knowledge, this is the first work on dynamic advance reservation virtual network mapping in EONs. We propose a novel stop-and-resume allocation scheme, in which the service duration of each request can be partitioned into several periods with different sets of resources assigned in different periods. We show that this can result in improved performance even after accounting for the overhead for stopping and resuming a service. This is a follow-up work from our previous work [4] in which each VN request had a fixed start time slot, and therefore lacked the flexibility to provision the request at any start time slot as long as the service duration was satisfied.

The paper is organized as follows. In Section II, we present the network model and the detailed problem statement. Proposed algorithms and baseline heuristics are presented in Section III. Section IV presents and discusses the simulation results. We conclude the paper in Section V.

II. Model and Problem Statement

A. Network Model and Notation

The substrate EON topology is represented as a graph $G_p(V_p, E_p)$, where V_p is the set of physical nodes, and E_p is the set of physical links. There are H virtual machines (VMs) on each physical node $v \in V_p$. Each physical link $e \in E_p$ consists of two opposite directional fibers. The entire optical C-band of each fiber is divided into S frequency slots in EONs. The bandwidth of each frequency slot is standardized as 12.5 GHz. Each lightpath is allocated a particular number of frequency slots based on its spectrum requirement. There are three spectrum allocation constraints in EONs. Spectrum contiguity constraint requires that the frequency slots allocated to each lightpath be contiguous. The spectrum continuity constraint requires that in a transparent optical network without spectrum converters, the same band of frequency slots has to be allocated on each link of the path assigned to a lightpath. The spectrum clash constraint requires that each frequency slot on a fiber be allocated to at most one lightpath at a time. In this paper, time is divided into equal-sized time-slots (of an unspecified duration) [14]. The state of the physical network is changed only at the beginning of a time-slot. Due to imperfect switching operation at each physical node, a guardband of G consecutive frequency slots is required to be allocated between adjacent frequency slot bands assigned to different lightpaths if they share common physical links and have time-slots overlap, in order to avoid interference. There are K precomputed shortest distance paths for each pair of physical nodes, and M possible modulation formats can be assigned to lightpaths. According to the physical impairments, there is a maximum transmission reach limit R_m for each modulation format m.

In this paper, we consider dynamic traffic. The virtual network requests arrive to the network according to a Poisson process. Each virtual request i has an arrival time-slot A_i, and required service duration T_i time-slots, and a deadline time-slot D_i. If the system cannot allocate sufficient resources to a virtual network request by its deadline, the request will be blocked. Any T_i time-slots in the window $[A_i, D_i]$ can be allocated to request i. A virtual network request i can be modeled as a graph $G_l^i(V_l^i, E_l^i)$, where V_l^i is the set of virtual nodes, and E_l^i is the set of virtual links. There is a computation resources requirement for each virtual node $v^i \in V_l^i$ as h_v^i VMs. We assume that virtual node v^i also has a primary physical node n_v^i to be mapped to, i.e., virtual node v^i can be mapped to physical node n_v^i or an adjacent node n' of n_v^i in the physical network (since in practice a virtual node usually specifies a physical location to be mapped to); these physical nodes form a candidate physical node set to map to for virtual node v^i. We assume that no two virtual nodes of a request can be mapped to the same physical node due to reliability concerns. Each virtual link $k^i \in E_l^i$ has a bit rate requirement Λ_k^i for bidirectional communication between its two end virtual nodes. According to the candidate physical node sets of two end virtual nodes, virtual link k^i has a candidate path set P_k^i. If virtual link k^i is mapped to path $p \in P_k^i$, the bit rate requirement Λ_k^i of virtual link k^i can be converted into a frequency slot (including guardband) requirement $b_k^i(p)$ based on the modulation format m_p^k allocated to virtual link k as

$$b_k^i(p) = \lceil \Lambda_k^i / (s_p^k * 12.5) \rceil + G, \tag{1}$$

where s_p^k is the spectrum efficiency of m_p^k in bps/Hz. Here, m_p^k is the modulation format that achieves the minimum value of $b_k^i(p)$, and the transmission reach limit is satisfied based on the length of path p. (If there is more than one such modulation format, the lower one is allocated.)

B. Problem Statement

In this paper, we investigate the dynamic version of virtual network mapping problem, where each request will be blocked or accepted by the system. If virtual request i is accepted, in each allocated time-slot t (recall that request i requires total T_i time-slots), each virtual node in request i needs to be mapped to a single physical node with sufficient available computing resources (selected from its candidate physical node set). Meanwhile, each virtual link of request i has to be mapped to a single physical path p (according to its end virtual node mapping) in time-slot t, and a band of contiguous frequency slots is allocated to the virtual link based on its bit rate requirement. The objective is to minimize the request blocking ratio (defined as the total number of blocked requests divided by the total number of arrived requests).

III. Algorithms

In this section we present our proposed approaches, which select the possible allocation with minimum resource cost,

Authorized licensed use limited to: The George Washington University. Downloaded on September 30,2020 at 16:27:16 UTC from IEEE Xplore. Restrictions apply.

TABLE I
NOTATIONS

Symbol	Meaning
i	the newly arrived VN request
T_i	service duration (in time-slots) of request i
A_i	arrival time-slot of request i
D_i	deadline time-slot of request i
v	a virtual node of request i
k	a virtual link of request i
C_i	resource cost of request i
α	weight parameter for computing resources
β	weight parameter for networking resources
h_v^i	number of VMs required by virtual node v
$\hat{h_i}$	the total number of VMs required by request i ($\sum_{v \in V_l^i} h_v^i$)
$\hat{s_i}$	the total number of frequency slots allocated to all virtual links of request i ($\sum_{k \in E_l^i} b_k^i(p) l(p)$, where p is the physical path assigned to virtual link k, and $l(p)$ is the number of physical links on path p)
j	an arbitrary service period of request i
τ_j	the duration (in time-slots) of service period j
x_j	the starting time-slot of service period j
y_j	the ending time-slot of service period j ($y_j = x_j + \tau_j - 1$)
θ_v^j	the number of time-slots used to create the VM image of virtual node v after service period j
I_v^j	the size of VM image (in Megabytes) of virtual node v after service period j
IO	the amount of I/O resources (in Megabytes per time slot) to generate VM images
π_v^j	the number of time-slots used to transfer the VM image of virtual node v after service period j from current physical node to next physical node
μ_v^j	the number of time-slots used to store the VM image of virtual node v at current physical node before VM image transmission
ρ_v^j	the number of time-slots used to store the VM image of virtual node v at next physical node after VM image transmission
ω_v^j	the number of time-slots used for VM image reloading of virtual node v at the next physical node after VM image transmission of service period j
$\hat{s_i}^j$	the total number of frequency slots allocated to all virtual links of request i in service period j ($\sum_{k \in E_l^i} b_k^i(p_j) l(p_j)$, where p_j is the physical path assigned to virtual link k in service period j, and $l(p_j)$ is the number of physical links on path p_j)
$\tilde{s_i}^j(v)$	the total number of frequency slots allocated to transfer VM image of virtual node v after service period j (the value of $\tilde{s_i}^j(v)$ is calculated as the number of physical links on the path for VM image transfer multiplied by the number of frequency slots needed for VM image transfer per physical link)
γ	weight parameter for storage resources

and baseline algorithms to address the dynamic advance reservation VNM problem. Before describing the details of these algorithms, we first present the normal Non-stop allocation scheme and our proposed Stop-and-Resume allocation scheme, which are applied in the algorithms. The resource cost calculation for these two schemes is also described. The notations used in the algorithms can be found in Table I.

A. Non-stop Scheme

This is the normal advance reservation allocation scheme. For a newly arrived request i with service duration T_i time-slots, arrival time-slot A_i, and deadline time-slot D_i, it can start at any time slot $t_i \in [A_i, D_i - T_i + 1]$ and finish at time slot $t_i + T_i - 1$. Each virtual node of VN request i is mapped to one physical node and each virtual link of VN request i is mapped to one physical path in the time period $[t_i, t_i + T_i - 1]$. For this scheme, the resource cost of mapping request i is calculated as $C_i = (\alpha \hat{h_i} + \beta \hat{s_i}) T_i$ (see Table I for details).

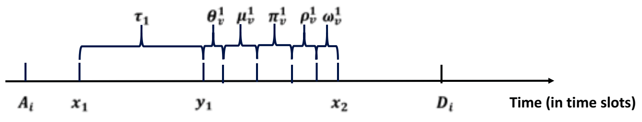

Fig. 1. Stop and Resume scheme illustration.

B. Stop-and-Resume Scheme

This is a new allocation scheme we propose for the advance reservation VNM problem that has not been studied before in optical networks. In this scheme, the total service duration T_i time-slots of request i is divided into multiple service periods, and each period j has length τ_j (time-slots) with starting time-slot x_j and finishing time-slot $y_j = x_j + \tau_j - 1$. To satisfy the service duration of request i, we need $T_i = \sum_j \tau_j$. Note that the entire virtual network mapping (for all virtual nodes and virtual links of request i) is stopped at time-slot $y_j + 1$, then resumed at time-slot x_{j+1}. Each virtual node v of request i is mapped to one physical node n; and each virtual link k of request i is mapped to one path (with sufficient frequency slots) in the time period $[x_j, y_j]$. For a particular virtual node v, it could be mapped to the same or different physical nodes in different periods. Similarly, a specific virtual link k could be mapped to the same or different physical paths in different periods. After each service period, there is a stop period before the next service period. It is to be noted that the stop and resume scheme incurs some overhead that cannot be neglected. In order to "stop" a virtual node and resume it on another physical node, the context of each virtual node must be saved as a VM image - which takes time and space at the original physical node - and then transferred to another physical node (to which the virtual node is mapped when it is "resumed") - which requires network bandwidth. We consider these overheads in our scheme, and will show in our results that this scheme nevertheless generally outperforms the non-stop scheme.

An example of this procedure is illustrated in Fig. 1 with only the first period completely shown. Please see Table I for the notation details. x_1 and y_1 are the starting and finishing time-slots of the first service period (applied to all virtual nodes and virtual links of request i), respectively. The size of the VM image of v for service period 1, I_v^1, is calculated as $h_v \cdot f(\tau_1)$, where h_v is the number of VMs required by virtual node v and $f(\tau_1)$ is a function of τ_1, since the longer the service period the larger the VM image size. θ_v^1 can be calculated as $\theta_v^1 = I_v^1 / IO$. If a virtual node is going to be mapped to a different physical node in its next service period, its corresponding VM image should be transferred to the next physical node, which takes both time (π_v^1 in the figure)

Authorized licensed use limited to: The George Washington University. Downloaded on September 30,2020 at 16:27:16 UTC from IEEE Xplore. Restrictions apply.

and network bandwidth. π_v^1 depends on the VM image size (I_v^1), the path allocated for transferring (i.e., which determines the modulation format), and the number of frequency slots assigned on the path. Note that if a virtual node will be mapped to the same physical node in the next service period, there is no need for VM image transmission. In addition, the VM image might be stored at the current physical node and the next physical node before and after the VM image transmission, which results in storage resources usage (I_v^1). At last, the VM image reloading at the next physical node takes some time ω_v^1, and storage resources (I_v) are required at the next physical node during this period. Note that the values of $\theta_v^1, \mu_v^1, \phi_v^1, \rho_v^1$ and ω_v^1 might be different for different virtual nodes; but x_2 is the next service period starting time-slot for all virtual nodes and virtual links of request i. In addition, $\theta_v^j, \mu_v^j, \phi_v^j, \rho_v^j$ and ω_v^j values are equal to zero if j is the last service period.

For this scheme, the cost of request i mapping is calculated as $C_i = \alpha \hat{h}_i T_i + \beta \sum_j \hat{s_i}^j \tau_j + \beta \sum_{v,j} \tilde{s_i}^j(v) \pi_v^j + \gamma \sum_{v,j} I_v^j(\theta_v^j + \mu_v^j + \pi_v^j) + \gamma \sum_{v,j} I_v^j(\pi_v^j + \rho_v^j + \omega_v^j)$, where $\theta_v^j + \mu_v^j + \pi_v^j$ is the total number of time-slots that I_v^j storage resources are required at the current physical node after service period j; and $\pi_v^j + \rho_v^j + \omega_v^j$ is the total time period that I_v^j storage resources are required at the next physical node after service period j (parameters defined in Table I).

C. Proposed Heuristics

1) NStopMC (Non-stop Scheme Minimum Cost Allocation): The NStopMC algorithm adopts Non-stop scheme to map the newly arrived request i, and it selects the possible allocation that minimizes the request's resource cost.

A minimum resource cost value Z is initialized to be infinite at the beginning of the algorithm. NStopMC tries all possible service windows one by one for request i, with a possible starting time-slot $A_i \leq t_i \leq D_i - T_i + 1$ and the corresponding finishing time-slot $w_i = t_i + T_i - 1$.

For each possible service window $[t_i, w_i]$, there are several rounds with each round including a virtual node mapping followed by some virtual links mapping. This procedure repeats until all virtual nodes and virtual links are successfully mapped. If a virtual node and/or a virtual link cannot be allocated sufficient resources, then the next service window is tried.

To select the virtual node in current round, a weight b_v is given to each unmapped virtual node v as follows. First, all the valid candidate physical nodes of unmapped virtual node v are found. Then, each valid candidate physical node n is assigned a weight g_n. After that, b_v is calculated as the difference between $\min_n g_n$ and the second minimum g_n (if the virtual node v has only one valid candidate physical node, b_v is set as ∞). Among all the unmapped virtual nodes, the one ($\hat{v}$) with maximum weight ($\max_v b_v$) is selected and mapped to physical node $\hat{n}_{\hat{v}}$ (the physical node achieves $\min_n g_n$). The reader is referred to the pseudo-code in Algorithm 1 for details of valid candidate physical nodes.

For the virtual link mapping in current round, a set F is created including all the unmapped virtual links whose two end

```
1  **foreach** unmapped virtual node $v$ in request $i$ **do**
2      **foreach** candidate physical node $n$ of $v$ **do**
3          Check it has $\geq h_v$ VM resources in period $[t_i, w_i]$
4          Check it has not been allocated to other virtual nodes of request $i$ in period $[t_i, w_i]$
5          Check if mapping virtual node $v$ to node $n$ won't make other virtual nodes having no available candidate physical nodes in period $[t_i, w_i]$
6          **if** all requirements satisfied **then**
7              node $n$ is a valid candidate physical node of $v$
8              assign a weight $g_n$ to $n$ as the minimum number of required frequency slots for all the virtual links with $v$ as an end virtual node when $v$ is mapped to $n$
9          **endif**
10     **end**
11 **end**
```

Algorithm 1: Virtual node mapping in NStopMC.

virtual nodes have been mapped already. For each unmapped virtual link in set F (the unmapped virtual links in set F are sorted in decreasing order of bit rate requirements), find all available candidate paths based on the current available frequency slots on each physical link, and assign each valid candidate path p a weight as the number of required frequency slots for virtual link k if it is mapped to path p (if multiple available bands with required frequency slots exist, the first band is selected (First Fit spectrum allocation [15])). Virtual link k is mapped to path $\bar{p}$ with the minimum weight.

If all virtual nodes and virtual links of request i can be successfully mapped in the service window $[t_i, w_i]$ and the resulting resources cost C_i^t with this window is less than Z, then update $Z = C_i^t$. At last, the request is accepted with the service window achieving the minimum resource cost value Z. If there is no valid mapping for request i for all possible service windows, request i is blocked. The worst case time complexity is $O(V^3N^2D^2 + EKSLD^2)$, where V is the number of virtual nodes of the VN request, N is the number of candidate physical nodes of each virtual node, D is the deadline of VN request, E is the number of virtual links in the VN request, K is the number of precomputed paths between each pair of physical nodes, S is the number of frequency slots on each fiber, and L is the number of physical links in the network.

2) SRMC (Stop and Resume Scheme Minimum Cost Allocation): For the newly arrived request i, SRMC algorithm tries the NStopMC algorithm first; if it fails, the Stop-and-Resume scheme is adopted and a possible allocation that minimizes the request's resource cost is selected. The pseudo-code for the SRMC algorithm in shown in Algorithm 2.

A variable χ is defined as the accumulated service period length and is initialized as 0. The minimum resource cost is

Authorized licensed use limited to: The George Washington University. Downloaded on September 30,2020 at 16:27:16 UTC from IEEE Xplore. Restrictions apply.

```
1  NStopMC algorithm, if mapping failed, continue the following steps
2  $\chi = 0$, $Z = \infty$
3  foreach $x_1 = A_i, A_i + 1, \ldots, D_i - T_i$ do
4      foreach $\tau_1 = T_i - 1, T_i - 2, \ldots, 1$ do
5          if all virtual nodes and virtual links are successfully mapped in first service period $[x_1, x_1 + \tau_1 - 1]$ then
6          endif
7          $\chi = \tau_1$, goto Line 9
8      end
9      $j = 2$
10     while $\chi < T_i$ do
11         Find the earliest starting time-slot $\hat{x}_j$ for service period $j$
12         foreach $x_j = \hat{x}_j, \hat{x}_j + 1, \ldots, D_i - (T_i - \chi) + 1$ do
13             foreach $\tau_j = T_i - \chi, T_i - \chi - 1, \ldots, 1$ do
14                 if all virtual nodes and virtual links are successfully mapped in service period $[x_j, x_j + \tau_j - 1]$ then
15                 endif
16                 $\chi = \chi + \tau_j$, $j = j + 1$, goto Line 10
17             end
18         end
19     end
20     if all virtual nodes and virtual links of $i$ are successfully mapped and $C < Z$ and $\chi = T_i$ then
21         $Z = C$
22     endif
23 end
24 if $Z < \infty$ then
25     Accept request $i$
26 endif
27 else
28     Block request $i$
29 endif
```

Algorithm 2: SRMC algorithm.

denoted as Z and is initialized to ∞. SRMC algorithm tries all possible first service period starting time-slot x_1 in range $[A_i, D_i - T_i]$ (Line 3). The duration of first service period τ_1 is in range $[1, T_i - 1]$ (Line 4). For a particular starting time-slot x_1 and length τ_1, the steps to map virtual nodes and virtual links in time period $[x_1, y_1 = x_1 + \tau_1 - 1]$ is similar to the mapping in NStopMC algorithm. The difference is that besides checking the VM availability on candidate physical node n (Line 3 in Algorithm 1), availability of storage resources (should be $\geq I_v^1$) at n is also checked for the VM image generation period $[y_1 + 1, y_1 + \theta_v^1]$ for virtual node v. If all virtual nodes and virtual links are successfully mapped in the first period $[x_1, y_1]$, then the next period will be considered (Lines 5-6) and χ is updated; otherwise, next value of τ_1 will be tried.

For the next service period j, the algorithm calculates the earliest possible starting time-slot $\hat{x}_j = \max_v(y_{j-1} + 1 + \theta_v^{j-1} + \omega_v^{j-1})$ (Line 11). For each starting time-slot x_j in range $[\hat{x}_j, D_i - (T_i - \chi) + 1]$ (Line 12) and period duration τ_j in range $[1, (T_i - \chi)]$ (Line 13), the steps to map virtual nodes and virtual links in time period $[x_j, y_j = x_j + \tau_j - 1]$ (Line 14) is similar to the mapping steps in NStopMC algorithm with the following additional checking in Algorithm 1. Virtual node v is mapped to physical node n_v^{j-1} in previous service period $j-1$. For a candidate physical node n for virtual node v in service period j, if $n = n_v^{j-1}$, then storage resources (with requirement I_v^{j-1}) will be checked on n_v^{j-1} in time period $[y_{j-1} + \theta_v^{j-1} + 1, x_j - 1]$. Otherwise, if $n \neq n_v^{j-1}$, storage resources will be checked at node n_v^{j-1} in time period $[y_{j-1} + \theta_v^{j-1} + 1, y_{j-1} + \theta_v^{j-1} + \pi_v^{j-1}]$, as well as at node n in time period $[y_{j-1} + \theta_v^{j-1} + \mu_v^{j-1}, x_j - 1]$; also, a path with sufficient frequency slots between n_v^{j-1} and n during time period $[y_{j-1} + 1 + \theta_v^{j-1} + \mu_v^{j-1}, y_{j-1} + \theta_v^{j-1} + \mu_v^{j-1} + \pi_v^{j-1}]$ is required for VM image transmission. In addition, storage resources with requirement I_v^j needs to be checked on node n for the time period $[y_j + 1, y_j + \theta_v^j]$. For all the possible scheduling of the VM transmission for virtual node v (with various values of $\mu_v^{j-1}, \pi_v^{j-1}, \rho_v^{j-1}, \omega_v^{j-1}$ and p, n), the best combination in terms of resources cost will be selected. If all virtual nodes and virtual links are successfully mapped in service period $[x_j, y_j]$, then the next service period $j+1$ will be considered (Line 15) and χ is updated.

After finding sufficient service periods ensuring $\chi = T_i$, if resource cost with current mapping C_i is less than Z, then Z is updated (Lines $20, 21$). At last, request i is mapped to the valid allocation achieving minimum resource cost Z (Line 25). If no valid allocation is found, the request is blocked. The worst case time complexity is $O(V^3N^2T^3D + EKSLT^3D)$, where T is the service duration of the VN request.

D. Baseline Heuristics

1) NStopFF (Non-stop Scheme First Fit Allocation): The NStopFF algorithm adopts the Non-stop scheme and First Fit mapping selection. It first maps all virtual nodes in request i then maps all the virtual links. In the node mapping, each virtual node v is mapped to the first valid candidate physical node (with smallest index). In the virtual link mapping, for a particular virtual link k with its end virtual node v_1 mapped to n_1 and v_2 mapped to n_2, the algorithm first checks whether the shortest path between n_1 and n_2 has sufficient frequency slots to satisfy the virtual link k's bit rate requirement; if yes, k is mapped to the shortest path, otherwise, the next shortest path will be checked. The first valid allocation (with smallest value of starting time-slot t_i) for all virtual nodes and virtual links is selected for request i. If no valid allocation exist, request i is blocked. The worst-case time complexity is $O(V^2N^2D^2 + EKSLD^2)$.

2) SRFF (Stop and Resume Scheme First Fit Allocation): SRFF algorithm first tries the NStopFF algorithm, if failed, the Stop-and-Resume scheme is utilized. The steps to find the service periods are similar to the SRMC algorithm. For each

Authorized licensed use limited to: The George Washington University. Downloaded on September 30,2020 at 16:27:16 UTC from IEEE Xplore. Restrictions apply.

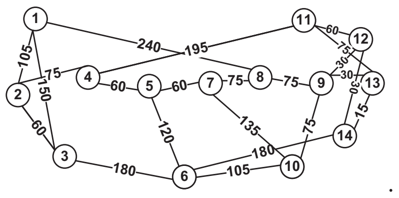

Fig. 2. 14-node NSF Network. The number along each link is link distance in km.

service period, SRFF algorithm first maps all virtual nodes then maps all the virtual links with the First Fit mapping as in NStopFF algorithm (with additional consideration of storage resources and frequency slot resources for VM image transmissions). The first valid allocation found for all virtual nodes and virtual links is selected for request i. If no valid allocation exists, request i is blocked. The worst-case time complexity is $O(V^2N^2T^3D + EKSLT^3D)$.

IV. Simulation Results

In this section, we present simulation results for the NSF network topology with 14 nodes and 22 links as shown in Fig. 2. Each physical node n has 100 VMs (H_n) and 5000 MB storage resources *dedicated* to store VM images that are transferred from current physical node to next physical node in Stop-and-Resume Scheme. The I/O resources for saving VM images (IO) is set as 20 MB/second. The duration of each time slot is set as 1 hour. Since the VM image reloading time duration (ω_v^j) is in range 40-60 seconds [16], it is omitted. There are $S = 320$ frequency slots on each fiber. The number of frequency slots used as guardband (G) is set as 1. There are 3 precomputed shortest paths for each pair of nodes. Each lightpath can be assigned one of 4 modulation formats: BPSK, QPSK, 8QAM, and 16QAM with transmission reach limits as 3000 km, 1500 km, 750 km, and 375 km, respectively [17]. 5000 virtual requests randomly arrive to the network following a Poisson distribution. The number of required service duration (T_i) of each request is geometrically distributed with mean 10 time-slots. Each virtual request consists of either 3 or 4 virtual nodes (selected randomly). The number of VMs required by each virtual node v (h_v) is randomly selected between 3-5 VMs. The deadline for each request i (D_i) is set as A_i+T_i+50 (where A_i is the request's arrival time-slot). The probability that there is a virtual link between one pair of virtual nodes is set as 0.8. The bit rate requirement for each virtual link k (Λ_k) is randomly selected in the set $[100, 105, \ldots, 395, 400]$ Gbps (the number of frequency slots required by each virtual link can be calculated according to Equation (1) on page 2). For the resources cost (C_i) calculation, the parameters values are $\alpha = 1$, $\beta = 14*100/(22*320)$ (14 physical nodes, 100 VMs per node, 22 physical links and 320 frequency slots per link), $\gamma = 100/5000$ (since there are 100 VMs and 5000 Megabytes on each physical node). The VM image size of virtual node v with τ_j time-slots as length of a service period j can be calculated as $I_v^j = h_v * 143\log_{10}(\tau_j)$ [18]. The results shown below are averaged over 10 simulation trials.

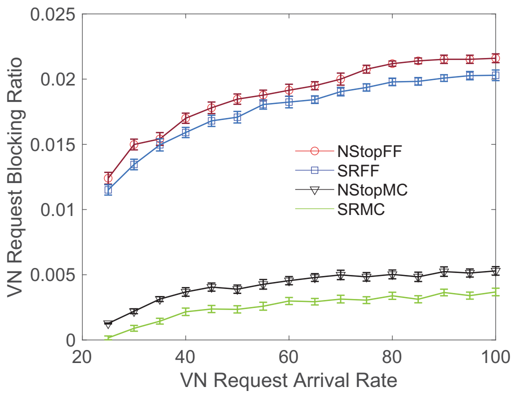

Fig. 3. VN Request Blocking Ratio vs. VN Requests Arrival Rate.

We compare the blocking ratio performance of the 4 algorithms with various request arrival rates (number of requests per time-slot) as shown in Fig. 3. We see that the proposed algorithms outperform the baseline algorithms significantly. In addition, Stop-and-Resume allocation scheme reduces the blocking ratio by 6.13% and 42.04% on average for the NStopFF and NStopMC algorithms, respectively. On average, 204.14 and 421.91 VN requests are accepted with allocations including more than one service periods in SRMC and SRFF algorithms, respectively; in addition, such allocations include, on average, 3.09 and 3.15 service periods.

Each VN request may be blocked due to VM shortage, frequency slots shortage for Non-stop scheme based algorithms. For Stop-and-Resume scheme based algorithms, storage shortage is another reason. Fig. 4 shows the number of blocked VN requests due to various reasons for different algorithms when the arrival rate is 50 VN requests per time-slot. For each blocked VN request, multiple service periods/windows might be tried, and the results show how many trials were failed due to each blocking reason. For NStopMC and NStopFF, it was difficult to find sufficient VM resources in consecutive required time-slots. For baseline algorithms (NStopFF and SRFF), since virtual node mapping steps failed to take virtual link requirements into account, this results in a lot of blocked VN requests due to frequency slots shortage (there is not enough frequency slot resources after virtual node mapping). For other arrival rate values, the result trends are similar for various algorithms and the three blocking reasons.

We limit the number of service periods in SRMC algorithms to at most $2, 3, \ldots, 10$ service periods per VN request for different arrival rates. The corresponding VN request blocking ratios are shown in Fig. 5, where results with 1 service period limit is the result of NStopMC algorithm and ∞ means there is no limit on the number of service periods (the original SRMC algorithm). It can be seen that as the limit increases, the blocking ratio first decreases and then increases; the reason is that requests with many service periods occupy more frequency slots and storage resources, which results in more blocked future requests. In addition, it shows that even with at most 2 service periods, the blocking ratio is reduced by

Authorized licensed use limited to: The George Washington University. Downloaded on September 30,2020 at 16:27:16 UTC from IEEE Xplore. Restrictions apply.

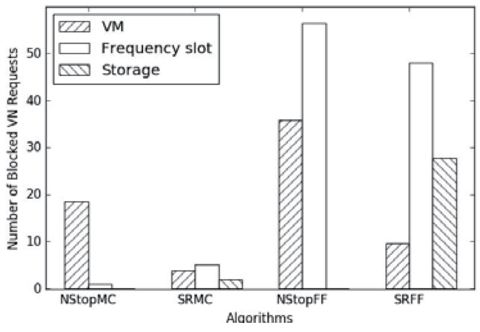

Fig. 4. Number of Blocked VN Requests for 4 algorithms.

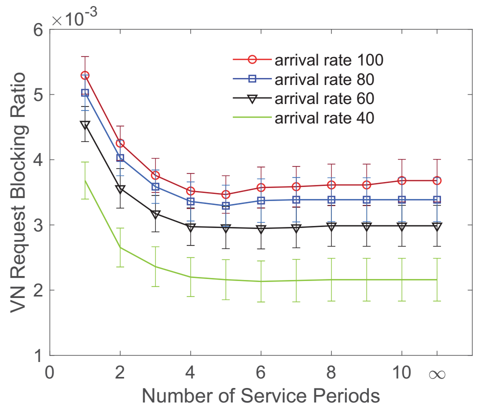

Fig. 5. VN Request Blocking Ratio vs. Number of Service Periods.

22.29% on average.

V. Conclusions

The scheduling and resource allocation for dynamic advance reservation virtual network requests is addressed in elastic optical networks. A Stop-and-Resume allocation scheme is proposed. Two algorithms based on Non-stop allocation and Stop-and-Resume allocation schemes are proposed. Simulation results show that comparing with baseline algorithms, the proposed algorithms achieve better performance in terms of virtual network request blocking ratio. In addition, the proposed Stop-and-Resume allocation scheme outperforms the conventional Non-stop allocation scheme but with storage cost to save VM images at physical nodes and time cost due to VM image transmission process between different physical nodes. In the future, we plan to investigate the advance reservation allocation scheme for static virtual network requests. Another extension we plan to make in the future is to explore the case with both delay tolerant requests and delay-sensitive requests.

Acknowledgment

This work was supported in part by NSF grants CNS-1813617 and CNS-1813772.

References

[1] I. Tomkos, S. Azodolmolky, J. Sole-Pareta, D. Careglio, and E. Palkopoulou, "A tutorial on the flexible optical networking paradigm: State of the art, trends, and research challenges," Proceedings of the IEEE, vol. 102, pp. 1317–1337, Sep. 2014.

[2] J. Masahiko, T. Hidehiko, K. Bartlomiej, T. Yukio, S. Yoshiaki, and M. Shinji, "Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies," IEEE Communications Magazine, vol. 47, pp. 66–73, Nov. 2009.

[3] N. M. M. K. Chowdhury, M. R. Rahman and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," IEEE INFOCOM 2009, Rio de Janeiro, 2009, pp. 783–791.

[4] J. Zhao, S. Subramaniam and M. Brandt-Pearce, "Virtual topology mapping in elastic optical networks," IEEE International Conference on Communications (ICC), Budapest, 2013, pp. 3904–3908.

[5] E. Guler, D. Zheng, G. Luo, L. Tian and X. Cao, "Virtual multicast tree embedding over elastic optical networks," GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, 2017, pp. 1–6.

[6] Y. Wang, P. Lu, W. Lu and Z. Zhu, "Cost-efficient virtual network function graph (vNFG) provisioning in multidomain elastic optical networks," in Journal of Lightwave Technology, vol. 35, no. 13, pp. 2712–2723, July 2017.

[7] W. Xie, J.P. Jue, Q. Zhang, X. Wang, Q. She, P. Palacharla, and M. Sekiya, "Survivable impairment-constrained virtual optical network mapping in flexible-grid optical networks," in IEEE/OSA Journal of Optical Communications and Networking, vol. 6, no. 11, pp. 1008–1017, November 2014.

[8] A. N. Patel, Z. Ye and P. N. Ji, "Cloud service embedding in software-defined flexible grid optical transport networks," OFC, San Francisco, CA, 2014, pp. 1–3.

[9] X. Chen, Z. Zhu, J. Guo, S. Kang, R. Proietti, A. Castro, and S. J. B. Yoo, "Leveraging mixed-strategy gaming to realize incentive-driven VNF service chain provisioning in broker-based elastic optical inter-datacenter networks," in IEEE/OSA Journal of Optical Communications and Networking, vol. 10, no. 2, pp. A232–A240, Feb. 2018.

[10] N. Shahriar, S. Taeb, S. R. Chowdhury, M. Tornatore, R. Boutaba, J. Mitra, and M. Hemmati,"Achieving a fully-flexible virtual network embedding in elastic optical networks, IEEE INFOCOM 2019, Paris, France, 2019, pp. 1756–1764.

[11] W. Lu and Z. Zhu, "Dynamic service provisioning of advance reservation requests in elastic optical networks," in Journal of Lightwave Technology, vol. 31, no. 10, pp. 1621–1627, May 2013.

[12] 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, no. 3, pp. B101–B111, March 2017.

[13] E. E. Moghaddam, H. Beyranvand and J. A. Salehi, "Routing, spectrum and modulation level assignment, and scheduling in survivable elastic optical networks supporting multi-class traffic," in Journal of Lightwave Technology, vol. 36, no. 23, pp. 5451–5461, 1 Dec. 2018.

[14] W. Ni, M. Schlosser, Q. Li, Y. Guo, H. Zhang and X. Zheng, "Achieving Optimal Lightpath Scheduling in Survivable WDM Mesh Networks," OFC 2008, San Diego, CA, 2008, pp. 1–3.

[15] B. C. Chatterjee, N. Sarma and E. Oki, "Routing and Spectrum Allocation in Elastic Optical Networks: A Tutorial," in IEEE Communications Surveys and Tutorials, vol. 17, no. 3, pp. 1776–1800, 2015.

[16] L. Cui, Z. Hao, L. Li, H. Fei, Z. Ding, B. Li, and P. Liu, "Lightweight virtual machine checkpoint and rollback for long-running applications", in Algorithms and Architectures for Parallel Processing, 2015, pp. 577–596.

[17] L. Gong and Z. Zhu, "Virtual Optical Network Embedding (VONE) Over Elastic Optical Networks," in Journal of Lightwave Technology, vol. 32, no. 3, pp. 450–460, Feb. 2014.

[18] J. Zhao, Y. Xiang, T. Lan, H. H. Huang and S. Subramaniam, "Elastic Reliability Optimization Through Peer-to-Peer Checkpointing in Cloud Computing," in IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 2, pp. 491-502, Feb. 2017.

Authorized licensed use limited to: The George Washington University. Downloaded on September 30,2020 at 16:27:16 UTC from IEEE Xplore. Restrictions apply.