

Adaptive Bandwidth Allocation for Virtual Network Embedding in Optical Data Center Networks

Swarnalatha Madanantha, Tram Truong-Huu, Mohan Gurusamy

Department of Electrical and Computer Engineering

National University of Singapore, Singapore

E-mail: swarna.3989@gmail.com, eletht@nus.edu.sg, elegm@nus.edu.sg

Abstract—Wavelength division multiplexed optical networks have become an attractive candidate to meet the ever-growing traffic demands in cloud data centers due to the features of large capacity and dynamic reconfiguration capability. While the bandwidth does not affect the makespan of compute-intensive and content-delivery-network applications, it has an impact on data-intensive applications that therefore require guaranteed bandwidth beside computing and storage resources for predictable performance. Motivated by this, we consider the problem of dynamically adjusting bandwidth so as to increase the acceptance of virtual networks embedded in optical data centers. We first develop an optimization programming formulation for the problem. We then develop a heuristic algorithm that efficiently embeds and adaptively allocates bandwidth to virtual networks such that the applications complete and release resources for future requests. We evaluate our algorithm through extensive simulations. The results show that our algorithm outperforms baseline algorithms by reducing rejections by up to 25%.

Index Terms—Optical networks, virtual network embedding, adaptive bandwidth allocation, predictable performance.

I. INTRODUCTION

With the increasing traffic demands, the global data center traffic growth rate is estimated to roughly 25% per year and the global data center IP traffic is estimated to reach 7.7 zettabytes (10^{21} bytes) by the end of 2017. Among this traffic, 76% is estimated to remain within data centers [1]. The huge traffic demands force cloud providers to look for the means for not only increasing the network capacity but also improving bandwidth utilization. Wavelength division multiplexing (WDM) based optical networks have become an attractive candidate to meet the network capacity challenge. In comparison with electrical packet-switched networks, WDM-based optical networks have much larger bandwidth capacity with low power consumption and cabling complexity. Dynamic reconfiguration of lightpaths also brings in flexibility for network management and traffic engineering. However, in addition to the limitations of circuit-switching, the wavelength continuity constraint due to the dynamic arrival of resource requests and high degree of the optical switch, which interconnects multiple top-of-the-rack (ToR) switches, has an impact on resource usage efficiency, making the problem of request embedding more challenging. Thus, intelligent selection of ToRs and wavelengths is needed to embed resource requests so as to use the servers under ToRs and bandwidth (or wavelength) resources efficiently.

In addition to the specific characteristics of optical networks, the increase in complexity of users' resource requests drives

the need for intelligent algorithms to efficiently realize the resource request embedding and bandwidth allocation so as to improve resource efficiency. Cloud applications can be categorized into three classes based on the impact of bandwidth on their performance. The first class includes compute-intensive applications that do not change the *makespan*, i.e., the total running time of the application to complete the data processing, regardless of the amount of bandwidth allocated. The second class includes the applications such as content delivery network applications that change the quality of service but the makespan does not change. The third class includes the data-intensive applications such as workflow-based applications [2], data backup and virtual machine migration in data centers. The performance and makespan of such applications are significantly impacted by the network bandwidth.

Focusing on the third class, guaranteeing bandwidth is an important issue for a data center to deliver predictable performance to such applications [3]. While the cloud providers do not directly charge the users for the bandwidth usage, the amount of bandwidth affects the cost that users pay for providers in the way that low bandwidth makes the applications last longer, leading to the increase in the cost of computing and storage resources. From a provider's point of view, high bandwidth will shorten the makespan of applications that will release cloud resources earlier for future requests, thereby accommodating more requests and achieving economic benefits. An attractive request model could let the users specify their bandwidth requirement as a range with minimum and maximum values. The request is defined in the form of a virtual network (VN) with virtual machines (VMs) and their communication requirements. Given this, the challenge for the cloud providers is to choose an optimal bandwidth value for VNs so as to achieve both low blocking ratio and high quality of service for the users' applications.

In this paper, we address the problem of adaptive bandwidth allocation for virtual network embedding in optical data center networks by solving its two sub-problems: the VN embedding problem with guaranteed minimum bandwidth and the additional bandwidth allocation problem. We develop two optimization programming formulations for the sub-problems, respectively. The programming formulation of the embedding problem aims at maximizing the total bandwidth utilized in data centers given a set of VNs and the data center network state so as to improve the bandwidth efficiency and the

economic benefit for cloud providers. While the embedding problem guarantees the minimum bandwidth requirements of VNs, the programming formulation for the adaptive bandwidth allocation aims at allocating extra bandwidth for the admitted VNs so as to improve the performance of the applications running on the VMs in respective VNs. Solving these programming formulations is computationally prohibitive when the number of VNs is high and the size of the optical data network is large. We thus develop a heuristic algorithm that comprises two phases. The first phase tries to embed VNs in optical data centers while satisfying their minimum bandwidth requirements. The second phase tries to allocate extra bandwidth to the admitted VNs so as to improve the application performance. Both optimization programming models and heuristic algorithm have to satisfy the resource capacity constraint of the network: the residual bandwidth and the wavelength continuity constraint. We demonstrate the effectiveness of the proposed algorithm through comprehensive simulations.

The rest of the paper is organized as follows. We discuss the related works in Section II. We present the system model and the problem formulation in Section III. We then present our heuristic algorithm in Section IV. Section V presents the numerical simulations and analysis of results before we conclude the paper in Section VI.

II. RELATED WORK

Embedding of VM groups with bandwidth requests and virtual network requests in data centers has been studied in the literature [4], [5], [6]. Both works presented in [4] and [5] studied the problem of VM placement in data centers to reduce cross network traffic by embedding communicating VMs in the same server or rack. In [6], the problem of VM placement and bandwidth reduction for an application graph of VM clusters has been studied. In [7], the authors addressed the problem of allocation of computing and network resources in multi-tenant data centers with the assumption of knowledge of VM to VM bandwidth requirements, represented as a traffic matrix. These works considered fixed amount of bandwidth requirement between VMs for the entire lifetime of the request whereas time-predictable requests have flexible bandwidth requirements. To the best of our knowledge, the problem of embedding and bandwidth allocation of time-predictable requests has not been addressed before. We take the first step towards addressing the problem with this work.

Virtual network embedding in optical networks has been recently studied [8], [9], [10], [11], [12]. While [9] and [10] studied the VNE problem in all-optical networks but not data centers, [8] and [11] considered this problem in hybrid optical-electrical data centers. The work presented in [11] additionally considered the capability of dynamic wavelength grouping of optical network architecture. These works however do not consider the flexibility of bandwidth requirements and variation of application makespan with the change in bandwidth allocated. This is the focus of our work. Our work also differs from the works presented in [13], [14], which focus more on

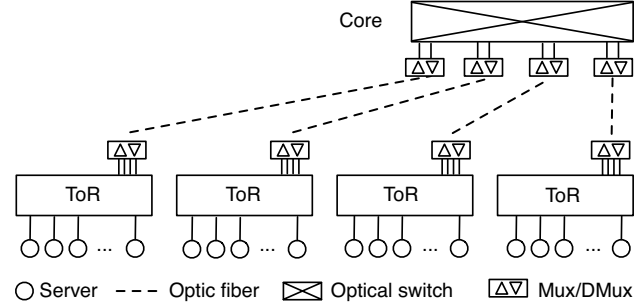


Fig. 1: Two-tier data center architecture.

the architecture design of optical networks rather than on the virtual network embedding problem.

III. SYSTEM MODEL

A. Problem Context and Assumptions

We consider a two-tier data center architecture based on WDM optical network as shown in Fig. 1. The bottom level contains servers that host multiple VMs. Multiple servers are connected to a ToR switch through server-ToR links. The ToRs are connected to the core (optical) switch in the next level by optical fibers. Each fiber can carry up to W wavelengths, meaning that a ToR switch can simultaneously reach up to W other ToR switches through optical paths or lightpaths. In the absence of wavelength converters, the lightpath between two ToR switches traversing through the optical switch needs to be wavelength-continuous. We assume that there are M ToRs in the data center and the number of ToRs is usually larger than the number of wavelengths that are carried by an optical fiber. The embedding and bandwidth allocation algorithm therefore needs to efficiently utilize bandwidth as well as wavelengths to avoid disconnectivity among ToRs, thus leading to rejection of future requests and revenue loss of providers.

Input requests are modeled as VNs: the nodes are VM-clusters and the edges represent data communication between the nodes. We assume that bandwidth requirement of a VN is the same for all the edges. We consider flexible bandwidth request model wherein users specify the required bandwidth as a range $(B^{\min}, B^{\max})$ driven by performance and price. The providers leverage on this flexibility and adjust the bandwidth dynamically so as to accommodate increased number of users while guaranteeing the minimum required performance. We assume that a node of a VN can be embedded in a ToR switch such that the data communication between VMs in the same cluster is realized through electrical links that connect the physical servers to the respective ToR switch. We also assume that no more than one node of a VN should be embedded in the same ToR. Otherwise, these two nodes can be merged to form a bigger node in the VN. The connectivity of nodes is represented by a symmetric matrix V where $V_{i,j} = V_{j,i} = 1$ indicates that node i and node j are connected. Otherwise, $V_{i,j} = V_{j,i} = 0$. Obviously, $V_{i,i} = 0$ for every node i in a VN.

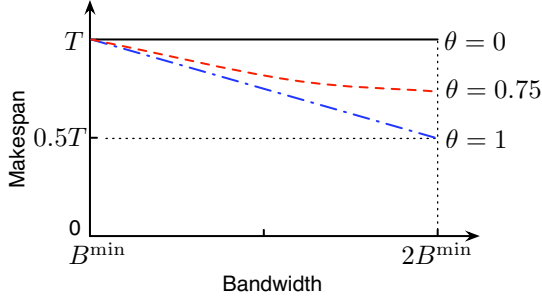


Fig. 2: Effect of bandwidth on application makespan with different variation ratios (θ).

While the providers may not know the actual application makespan when its VN request is submitted, we assume that based on a prediction algorithm, the providers are able to estimate the variation in the application makespan with respect to the amount of bandwidth allocated. The variation ratio can be defined as $\theta = (T_1/T_2)/(B_2/B_1)$ where T_1 and T_2 is the application makespan when running with the bandwidth B_1 and B_2 , respectively. In reality, most data-intensive applications running on VNs will complete the execution in a shorter duration if more bandwidth is allocated to their VNs. For instance, a data backup application can have a variation ratio as 1 when the amount of bandwidth is doubled, i.e., the makespan is reduced by half. Fig. 2 illustrates the variation ratio in the makespan of three classes of applications: a class of non-varying applications ($\theta = 0$), a class of moderate-varying applications, and a class of high-varying applications ($\theta = 1$).

Given such knowledge, providers have to solve a joint problem that composes of two sub-problems. The first sub-problem of VN embedding decides the admission of a VN in the data center with a guaranteed minimum bandwidth. If the VN is admitted, the second sub-problem of additional bandwidth allocation determines how much additional bandwidth will be allocated to improve the application performance. In the next sections, we present the optimization programming formulations for these two sub-problems.

B. Formulation of Optimal Embedding and Admission Control

Let $\mathcal{R}$ be the set of VNs submitted to the data center, where VN $r \in \mathcal{R}$ is represented by a tuple $(V^r, B_r^{\min}, B_r^{\max})$. The set of nodes in VN r is denoted by $\mathcal{V}_r$. We also denote $\mathcal{S}$ as the set of ToR switches in the data center such that $|\mathcal{S}| = M$. The capacity state of the optical data center network is represented by matrix C where $C_{u,v}$ is the residual bandwidth of the lightpath connecting ToR u and ToR v for $\forall u, v \in \mathcal{S}$. Initially, $C_{u,v} = 0$ since the lightpath is not established yet. Once a lightpath has been created, its capacity will be the maximum capacity of a wavelength, denoted as C^{wl} . The logical topology of the optical data center network is represented by symmetric matrix L where $L_{u,v}$ is the number of lightpaths that have been established between ToR u and ToR v . Obviously, $L_{u,u} = 0, \forall u \in \mathcal{S}$. The number of lightpaths that have been

created from ToR u , is then computed as $\mathcal{L}_u = \sum_{v \in \mathcal{S}} L_{u,v}$. Obviously, $\mathcal{L}_u \leq W$. If two ToRs u and v have $\mathcal{L}_u$ and $\mathcal{L}_v$, which are less than W , they can create a new lightpath between them to provide more bandwidth. To formulate the problem, we define $\pi : \mathcal{V}_r \rightarrow \mathcal{S}$, a mapping from the nodes in VN r to the ToR switches in the data center. The decision variable of the problem will be then $\chi_{i,\pi(i)}^r$, which indicates whether node i of VN $r \in \mathcal{R}$ is embedded in ToR $\pi(i)$ or not. For the admission decision of VN r , we define γ_r , a binary variable that indicates whether VN r is admitted or not. We additionally define $\xi_{u,v}$ as a decision variable that indicating the number of lightpaths that can be additionally established between ToR u and ToR v to satisfy bandwidth demands. The objective of the embedding problem is to maximize the total bandwidth utilized by the admitted VNs with minimum bandwidth requirements. The nonlinear optimization programming of the embedding problem is formulated as follows:

$$\text{Maximize: } \mathcal{B} = \sum_{r \in \mathcal{R}} \gamma_r \sum_{i,j \in \mathcal{V}_r} \chi_{i,\pi(i)}^r \chi_{j,\pi(j)}^r V_{i,j} B_r^{\min}, \quad (1)$$

subject to:

$$\sum_{r \in \mathcal{R}} \gamma_r \sum_{i,j \in \mathcal{V}_r} \chi_{i,u}^r \chi_{j,v}^r V_{i,j} B_r^{\min} \leq C_{u,v} + \xi_{u,v} C^{\text{wl}}, \forall u, \forall v \in \mathcal{S}, \quad (2)$$

$$\mathcal{L}_u + \sum_{v \in \mathcal{S}, v \neq u} \xi_{u,v} \leq W, \forall u \in \mathcal{S}, \quad (3)$$

$$\gamma_r \sum_{i \in \mathcal{V}_r} \chi_{i,u}^r \leq 1, \forall r \in \mathcal{R}, \forall u \in \mathcal{S}, \quad (4)$$

$$\gamma_r \sum_{u \in \mathcal{S}} \chi_{i,u}^r \leq 1, \forall i \in \mathcal{V}_r, \forall r \in \mathcal{R}, \quad (5)$$

$$\gamma_r \in \{0, 1\}, \forall r \in \mathcal{R}, \quad (6)$$

$$\chi_{i,u}^r \in \{0, 1\}, \forall i \in \mathcal{V}_r, \forall r \in \mathcal{R}, u \in \mathcal{S}, \quad (7)$$

$$\xi_{u,v} \in \{0, \dots, W\}. \quad (8)$$

The constraint in (2) ensures that a link in the optical data center network has sufficient bandwidth to satisfy the demands of all the edges of the VNs that are embedded in this link. The right hand side of Eq. (2) is the total residual bandwidth of the link connecting ToR u and ToR v , given the number of lightpaths that can be established as new lightpaths between these ToRs, denoted by $\xi_{u,v}$. The constraint in (3) ensures that the maximum number of lightpaths that can be established from a ToR should be less than the capacity of the fiber, i.e., the number of wavelengths carried by a fiber. The constraint in (4) ensures that if VN $r \in \mathcal{R}$ is admitted, a ToR in the data center can embed at most 1 node in the VN. The constraint in (5) ensures that if VN $r \in \mathcal{R}$ is admitted, each node must be embedded in only one ToR. The last three constraints indicate the value range of the decision variables in the problem. It is worth mentioning that with the constraint in (4) and the maximum number lightpaths that can be created from a ToR, i.e., the reachability factor, a VN with the highest degree of nodes greater than W cannot be embedded in the data center. Such VNs will be then rejected.

Solving the above problem is computationally hard even for a small data center and a small number of VNs. Indeed, the problem of determining which VNs will be admitted among the submitted VNs is a variant of the knapsack problem [15] while the problem of embedding belongs to the class of well-known combinatorial optimization problem called *Quadratic Assignment Problem* (QPA) [16]. The size of the problem, i.e., the number of decision variables, increases exponentially with respect to the size of the optical data center network, the number of VNs and the number of nodes in each VN. Achieving the optimal solution of the problem is therefore extremely hard due to its high complexity. We thus advocate for heuristic algorithm that solves the embedding problem in a shorter running time and results in an acceptable solution.

C. Formulation of Additional Bandwidth Allocation

Given the admission decision as a set of admitted VNs with their embedding solutions, we now present the optimization problem of additional bandwidth allocation to the admitted VNs so as to improve the bandwidth efficiency and performance of the applications running on the VMs. To ensure the fairness among the previously admitted VNs and current VNs, the additional bandwidth allocated to the VNs, which have been admitted previously but they are still running, may be revised. Algorithmically, additional bandwidth for the previously admitted VNs will be reset to zero just as the case of currently admitted VNs before solving the following optimization problem.

Let $\mathcal{R}^{\text{at}}$ denote the set of *active* VNs that include the VNs admitted in the previous time slots and those admitted in the current time slot. We denote θ_r as the variation ratio in the makespan of VN $r \in \mathcal{R}^{\text{at}}$ as defined in the previous section. We define δ_r as the additional bandwidth allocated to VN $r \in \mathcal{R}^{\text{at}}$. The optimization problem aims at maximizing the total running time that can be reduced for all the active VNs by determining the amount of bandwidth, δ_r , allocated to VN $r \in \mathcal{R}^{\text{at}}$. The linear optimization programming problem is formulated as follows:

$$\text{Maximize: } \mathcal{T} = \sum_{r \in \mathcal{R}^{\text{at}}} \theta_r \delta_r, \quad (9)$$

subject to:

$$\sum_{r \in \mathcal{R}^{\text{at}}} \sum_{i,j \in \mathcal{V}_r} \chi_{i,u}^r \chi_{j,v}^r V_{i,j} \delta_r \leq C_{u,v}, \forall u, \forall v \in \mathcal{S}, \quad (10)$$

$$\delta_r \geq 0, \forall r \in \mathcal{R}^{\text{at}}, \quad (11)$$

$$\delta_r \leq B_r^{\text{max}} - B_r^{\text{min}}, \forall r \in \mathcal{R}^{\text{at}}. \quad (12)$$

The constraint in (10) ensures that the total additional bandwidth used on each lightpath should not exceed its residual bandwidth where $\chi_{i,u}^r$ is the embedding decision indicating that whether node i in VN r is embedded in ToR u or not, $\forall i \in \mathcal{V}_r, \forall r \in \mathcal{R}^{\text{at}}$ and $\forall u \in \mathcal{S}$. It is to be noted that after solving the admission problem presented in the previous section, the optical logical topology of the data center network is updated with the newly created lightpaths. The residual bandwidth of all the lightpaths in the optical network is also

Algorithm 1 Adaptive Bandwidth Allocation for Virtual Network Embedding (ADA-BA)

Input: $L_{u,v}, C_{u,v}, \forall u, \forall v \in \mathcal{S}$, and $\mathcal{R}$.

Output: $\mathcal{R}^{\text{ad}}, \chi_{i,u}^r, \forall i \in \mathcal{V}_r, \forall r \in \mathcal{R}^{\text{ad}}, \forall u \in \mathcal{S}, \delta_r, \forall r \in \mathcal{R}^{\text{at}}$.

```

1: for  $r \in \mathcal{R}$  do
2:   Run Algorithm 2 with  $L_{u,v}, C_{u,v}, \forall u, \forall v \in \mathcal{S}$  and  $\mathcal{R}$ 
   to obtain  $\gamma_r, \chi_{i,u}^r, \forall i \in \mathcal{V}_r, \forall u \in \mathcal{S}$ ;
3:   if  $\gamma_r = 1$  then
4:     Update  $L_{u,v}, C_{u,v}, \forall u, \forall v \in \mathcal{S}$  with  $r$ ;
5:     Add  $r$  to  $\mathcal{R}^{\text{ad}}$ ;
6:   end if
7: end for
8:  $\mathcal{R}^{\text{at}} \leftarrow \mathcal{R}^{\text{ad}} \cup \mathcal{R}^{\text{pread}}$ ;
9: Run Algorithm 3 with the updated  $L_{u,v}, C_{u,v}, \forall u, \forall v \in \mathcal{S}$ 
   and  $\mathcal{R}^{\text{at}}$  to obtain  $\delta_r, \forall r \in \mathcal{R}^{\text{at}}$ ;
10: return  $\mathcal{R}^{\text{ad}}, \chi_{i,u}^r, \forall i \in \mathcal{V}_r, \forall r \in \mathcal{R}^{\text{ad}}, \forall u \in \mathcal{S}, \delta_r, \forall r \in \mathcal{R}^{\text{at}}$ ;

```

updated by deducting the total bandwidth allocated to the admitted VNs for satisfying the minimum bandwidth requirement. The allocation of additional bandwidth is performed only based on the current state of the optical network without any new lightpath creation. The constraint in (11) ensures that the additional bandwidth should not be negative and the constraint in (12) indicates that it does not need to allocate additional bandwidth more than the requirement. Allocating more than the required bandwidth will not further reduce the application makespan as illustrated in Fig. 2 but it will waste bandwidth that may be useful for other concurrent VNs.

Even though the formulated problem is a linear optimization programming problem, an optimization solver requires long time to produce the optimal results. We thus propose a heuristic solution to allocate additional bandwidth to the active VNs based on a greedy approach.

IV. ALGORITHMS

A. Integrated Algorithm

Algorithm 1 describes the entire process that solves the problem of adaptive bandwidth allocation for VN embedding in optical data center networks. The input of the algorithm consists of the state of the optical data center network at the moment when the algorithm is run, and a set of VNs that are submitted to the data center. The network state is defined by the logical topology of the network and the residual bandwidth of the established lightpaths. The output of the algorithm consists of the set of admitted VNs, $\mathcal{R}^{\text{ad}}$, the embedding solution for each admitted VN, $\chi_{i,u}^r, \forall i \in \mathcal{V}_r, \forall r \in \mathcal{R}^{\text{ad}}, \forall u \in \mathcal{S}$, and the additional bandwidth allocated to the active VNs, $\delta_r, \forall r \in \mathcal{R}^{\text{at}} = \mathcal{R}^{\text{ad}} \cup \mathcal{R}^{\text{pread}}$ where the set of active VNs ($\mathcal{R}^{\text{at}}$) includes the set of admitted VNs at the current time slot ($\mathcal{R}^{\text{ad}}$) and the set of admitted VNs in previous time slots ($\mathcal{R}^{\text{pread}}$) whose applications are still running on the VMs.

As shown in Algorithm 1, we relax the admission control problem by assuming that VNs are submitted to the data center in a sequential order. This assumption still reflects realistic

scenarios in which users' requests are served based on the first-come-first-served basis. Therefore, the embedding algorithm (Algorithm 2) is invoked repeatedly to embed VNs one by one according to their arrival order. A VN will be rejected if it has a greater degree than the reachability factor, i.e., the maximum number of wavelengths carried by a fiber, or the optical network does not have sufficient residual bandwidth to guarantee the minimum bandwidth requirement of the VN. For each admitted VN, the algorithm updates the residual bandwidth of the lightpaths as well as the logical topology when new lightpaths are established during the embedding process of the VN (lines 3–6). After the embedding phase, the algorithm updates the set up active VNs with newly admitted VNs, $\mathcal{R}^{\text{at}}$, that is used as in the inputs for Algorithm 3 to determine the additional bandwidth for each active VN.

B. Iterative Virtual Network Embedding Algorithm

In this section, we present our proposed algorithm for solving the embedding problem. As mentioned in the previous section, this algorithm generates the embedding solution for a VN if it is possible to embed. The input of the algorithm consists of a VN, the current network state with optical logical topology formed by the lightpaths that have been established prior to this embedding. The residual bandwidth of the established lightpaths is also needed for the algorithm to check whether they are able to provide the minimum bandwidth for the VN. Below, we describe in detail the steps of the algorithm presented in Algorithm 2.

Given a VN, the general idea of the algorithm is to sequentially embed all the nodes in the VN until they are embedded or the algorithm cannot embed a certain node due to the topology constraint or bandwidth capacity constraint. To avoid unnecessary runs, the algorithm computes the degree of every node in the VN as shown in line 3. A VN with the highest node degree that is greater than the reachability factor of the optical data center network will be rejected with no further analysis. Otherwise, the algorithm starts the embedding process with the highest degree node, referred to the *entry* node. The rationale behind this decision is to avoid unnecessary runs for other nodes if the entry node cannot be embedded. Otherwise, it is likely that the VN is rejected only when the residual bandwidth is not sufficient.

Given that node e is the entry node with the highest degree, the algorithm computes a set of ToR candidates for node e (lines 9–12), denoted as $\mathcal{C}_e^{\text{ToR}}$. All the ToRs, which have the degree greater than the degree of the entry node, will be considered. It is to be noted that to compute the degree of a ToR to embed a node in a VN, we count only the lightpaths that have sufficient bandwidth to satisfy the bandwidth requirement of the VN (line 1). In the **for** loop from line 13 to line 35, the algorithm verifies the feasibility of embedding the entire VN when the entry node is embedded on a ToR candidate in set $\mathcal{C}_e^{\text{ToR}}$. Let ToR u be the current verified candidate to embed the entry node. The algorithm runs the **while** loop (see lines 16–30) to embed other nodes in the input VN. In each iteration, a node in the VN is selected

Algorithm 2 Virtual Network Embedding Algorithm

Input: $L_{u,v}, C_{u,v}, \forall u, \forall v \in \mathcal{S}, r$.

Output: $\gamma_r, \chi_{i,u}^r, \forall i \in \mathcal{V}_r, \forall u \in \mathcal{S}$.

```

1:  $D_u^{\text{ToR}} \leftarrow \sum_{v \in \mathcal{S}, C_{u,v} \geq B_r^{\min}} L_{u,v}, \forall u \in \mathcal{S};$  /*ToR degree*/
2:  $E_i \leftarrow 0, \forall i \in \mathcal{V}_r;$  /*node has been embedded or not*/
3:  $D_i^{\text{rq}} \leftarrow \sum_{j \in \mathcal{V}_r} V_{i,j}, \forall i \in \mathcal{V}_r;$  /*node degree*/
4: Pick first node  $e$  with  $D_e^{\text{rq}} = \max\{D_i^{\text{rq}}, \forall i \in \mathcal{V}_r\};$ 
5:  $\gamma_r \leftarrow 0;$ 
6: if  $D_e^{\text{rq}} > W$  then
7:   return  $\gamma_r;$  /*unable to embed*/
8: end if
9:  $\mathcal{C}_e^{\text{ToR}} \leftarrow \{u | u \in \mathcal{S}, D_u^{\text{ToR}} \geq D_e^{\text{rq}}\};$ 
10: if  $\mathcal{C}_e^{\text{ToR}} = \emptyset$  then
11:    $\mathcal{C}_e^{\text{ToR}} \leftarrow \{u | D_u^{\text{ToR}} = \max\{D_v^{\text{ToR}}, \forall v \in \mathcal{S}\}\};$ 
12: end if
13: for  $u \in \mathcal{C}_e^{\text{ToR}}$  do
14:    $\chi_{e,u}^r \leftarrow 1;$  /*embed node  $e$  in ToR  $u$ */
15:    $E_e \leftarrow 1; L^{\text{org}} \leftarrow L;$ 
16:   while  $\exists i \in \mathcal{V}_r | E_i = 0$  do
17:     Pick node  $i$  such that
18:      $D_i^{\text{rq}} = \max\{D_i^{\text{rq}} | E_{i'} = 0, \sum_{j \in \mathcal{V}_r} V_{i',j} E_j \geq 1\};$ 
19:      $\mathcal{C}_i^{\text{ToR}} \leftarrow \{v | \forall v, v' \in \mathcal{S}, C_{v,v'} \geq B^{\min},$ 
20:        $\exists j \in \mathcal{V}_r | \chi_{j,v'} V_{i,j} E_j = 1\};$ 
21:     if  $\mathcal{C}_i^{\text{ToR}} = \emptyset$  then
22:        $\mathcal{C}_i^{\text{ToR}} \leftarrow \{v | D_v^{\text{ToR}} = \max\{D_{v'}^{\text{ToR}}, \chi_{j,v'} = 0,$ 
23:          $\forall j \in \mathcal{V}_r, \forall v' \in \mathcal{S}\}\};$ 
24:     end if
25:     Pick ToR  $v \in \mathcal{C}_i^{\text{ToR}}$  with the least  $D_v^{\text{ToR}} \geq D_i^{\text{rq}}$ 
26:     if exists. Otherwise, pick the highest  $D_v^{\text{ToR}};$ 
27:     if Cannot create required lightpaths then
28:        $L \leftarrow L^{\text{org}};$  /*Reset the logical topology*/
29:        $\chi_{j,v'} \leftarrow 0, E_j \leftarrow 0, \forall j \in \mathcal{V}_r, v' \in \mathcal{S};$ 
30:       break;
31:     end if
32:     Create required lightpaths;
33:      $\chi_{i,v} \leftarrow 1, E_i \leftarrow 1;$ 
34:   end while
35:   if  $E_i = 1, \forall i \in \mathcal{V}_r$  then
36:      $\gamma_r \leftarrow 1;$ 
37:     break;
38:   end if
39: end for
40: return  $\gamma_r, \chi_{i,u}^r, \forall i \in \mathcal{V}_r, \forall u \in \mathcal{S};$ 

```

to be embedded and the algorithm determines a ToR for the selected node. As shown in line 17, among the nodes that have not been embedded, the algorithm selects the node with the highest degree and at least one of its neighboring nodes has been embedded. For instance, the first iteration will select one node from the neighboring nodes of the entry node to embed.

Given that node i is the current processed node, the algorithm computes the set of ToR candidates for node i , denoted

Algorithm 3 Proportional Bandwidth Allocation

Input: $L_{u,v}, C_{u,v}, \forall u, \forall v \in \mathcal{S}, \mathcal{R}^{\text{at}}$.**Output:** $\delta_r, r \in \mathcal{R}^{\text{at}}$.

```
1: for each  $(u, v) | L_{u,v} \neq 0, u, v \in \mathcal{S}$  do
2:   Pick all the requests that use link  $(u, v)$  and add them
   to set  $\mathcal{R}_{u,v}^{\text{at}}$ ;
3:    $\Theta \leftarrow \sum_{r \in \mathcal{R}_{u,v}^{\text{at}}} \theta_r$ ; /*total reduction ratio*/
4:    $\delta_r^{(u,v)} \leftarrow \theta_r C_{u,v} / \Theta, \forall r \in \mathcal{R}_{u,v}^{\text{at}}$ ;
5: end for
6: for  $r \in \mathcal{R}^{\text{at}}$  do
7:    $\delta_r \leftarrow \min\{\delta_r^{(u,v)} | \chi_{i,u} \chi_{j,v} V_{i,j} = 1, \forall u, v \in \mathcal{S}, i, j \in \mathcal{V}_r\}$ ;
8: end for
9: return  $\delta_r, r \in \mathcal{R}^{\text{at}}$ ;
```

as $\mathcal{C}_i^{\text{ToR}}$ (lines 18–21). This set includes all the ToRs that have not been used and they are neighboring with the ToRs, which have embedded the neighbors of node i (line 18). For instance, in the first iteration, since the algorithm embeds a neighboring node of the entry node, only the ToRs that are adjacent with the ToR embedding the entry node are considered. If there does not exist any such candidate, the highest degree ToR is selected. Let ToR v be the selected ToR for node i . The algorithm verifies whether new lightpaths need to be established or not and whether it is possible to establish these new lightpaths or not. If it is possible, then node i is embedded in ToR v and the algorithm will start new iteration for other nodes. Otherwise, it resets all parameters as shown in lines 23–27 and tries a new ToR candidate for the entry node. If all the nodes in the VN have been embedded, the algorithm breaks the **for** loop even though there still exists other ToR candidates for the entry node. If the **for** loop finishes without a successful embedding solution, the request is then rejected. Otherwise, the algorithm returns the embedding solution and exits.

C. Additional Bandwidth Allocation Algorithm

Given the set of active VNs, $\mathcal{R}^{\text{at}}$, we now present the algorithm that computes additional bandwidth allocated to the each active VN. Since the lightpaths are shared by different VNs and the edges of each VN are embedded on different lightpaths, the amount of additional bandwidth allocated to each edge may not be the same. The proposed algorithm simply takes the lowest value of additional bandwidth among the edges as the decision value for the VN. To bring the fairness among the active VNs, we propose to apply two simple allocation strategies: (i) proportional-bandwidth allocation, and (ii) even-bandwidth allocation. The proportional strategy allocates the residual bandwidth to the VNs proportionally to their reduction ratio in the makespan of the application running on the VMs of the respective VN. The even strategy evenly allocates the residual bandwidth on the lightpaths to all the active VNs. The pseudo code presented in Algorithm 3 describes the proportional bandwidth allocation strategy.

The input parameters of the algorithm are the state of the optical data center network and a set of active VNs. The output of the algorithm is the additional bandwidth allocated to each active VN. In the first **for** loop from line 1 to line 5, the algorithm computes the bandwidth that can be allocated to the VNs sharing the same link (u, v) in the optical data center network. It selects all the VNs that use link (u, v) and computes the total reduction ratio as shown in line 3. In line 4, the algorithm computes the additional bandwidth that an edge in the VN using link (u, v) can receive. The second **for** loop revises the additional bandwidth that can be allocated to a VN and takes the decision for final value, which is the minimum value among those obtained from the first **for** loop.

Adopting the above algorithm to the even-bandwidth allocation strategy is simple by removing line 3 and changing line 4. The additional bandwidth that an edge of a VN can receive is simply computed as $\delta_r^{(u,v)} \leftarrow C_{u,v} / |\mathcal{R}_{u,v}^{\text{at}}|, \forall r \in \mathcal{R}_{u,v}^{\text{at}}$ where $\mathcal{R}_{u,v}^{\text{at}}$ is the set of active VNs that use link (u, v) in the optical data center network. It is to be noted that with heterogeneous VNs, using the even-bandwidth allocation strategy may not be efficient since the applications of some VNs do not reduce the makespan even though the bandwidth allocated is increased.

D. Complexity of Algorithms

For embedding a virtual network, Algorithm 2 has a complexity of $\mathcal{O}(M|\mathcal{V}|)$ where M is the number of ToRs in the data center network and $|\mathcal{V}|$ is the number of nodes in the VN. This is due to the nested **while** loop in the **for** loop from line 13 to line 35 of Algorithm 2. Algorithm 3 has a complexity of $\mathcal{O}(MW)$ where M is the number of ToRs in the data center and W is the number of wavelengths that a fiber can carry so as MW is the maximum number of lightpaths can be established. The integrated algorithm shown in Algorithm 1 therefore has an overall complexity of $\mathcal{O}(|\mathcal{R}|M \max\{|\mathcal{V}_r|, \forall r \in \mathcal{R}\})$ since Algorithm 1 has to run for all VNs.

V. PERFORMANCE STUDY

A. Simulation Setting

We consider an optical data center with the architecture as shown in Fig. 1. The core (optical) switch connects 16 ToR switches. The number of wavelengths carried by a fiber is set to 8 and the capacity of each wavelength is set to 1 Gbps. The VNs are generated randomly with the number of nodes in each VN varying in the range of $[2, 5]$. The minimum bandwidth requirement per edge in each request is randomly chosen in the range of $[400, 600]$ Kbps and the maximum bandwidth requirement per edge in each VN is randomly chosen in the range of $[600, 800]$ Kbps. We use the following performance metrics to evaluate the performance of the proposed algorithm.

- Rejection ratio: The number of rejected VN requests over the total number of submitted VN requests;
- Average bandwidth allocated per VN per time slot: it is computed by taking the average bandwidth values obtained from the algorithm over the number of admitted VNs over the total number of time slots simulated;

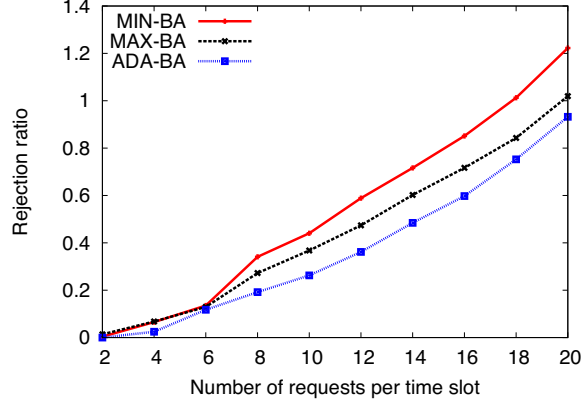


Fig. 3: Rejection ratio of the algorithms for varying arrival rates.

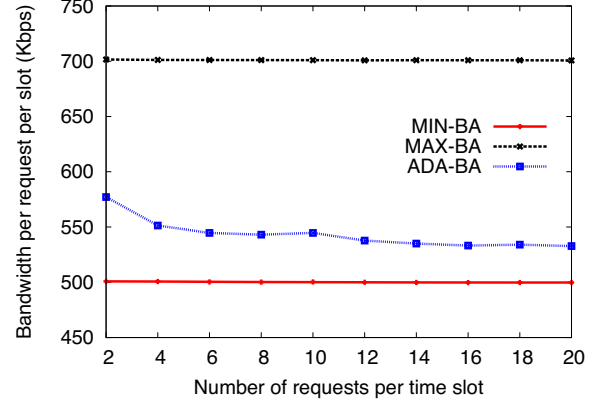


Fig. 4: Edgewise bandwidth per request per time slot.

- Total bandwidth used to complete the execution of all the admitted VNs: it is computed by summing the bandwidth consumed by each admitted VN in each time slot until its completion. The total bandwidth consumed by each admitted VN in each time slot is the sum of bandwidth allocated to all the edges.

Using the proposed embedding algorithm (Algorithm 2), we compare the performance of the adaptive bandwidth allocation algorithm (denoted as ADA-BA in all the figures) against two following baseline algorithms:

- Minimum Bandwidth Allocation (MIN-BA): the algorithm only satisfies the minimum bandwidth requirement of every admitted VN request;
- Maximum Bandwidth Allocation (MAX-BA): the algorithm allocates the maximum bandwidth requirement for every admitted VN request.

B. Analysis of Results

1) *Overall Performance*: In the first simulation, we evaluate the overall performance of the proposed algorithm. All VNs are categorized into three classes:

- a non-varying class that includes the VNs on which the makespan of the applications do not change regardless of the bandwidth allocated ($\theta = 0$);
- a moderate-varying class that consists of the VNs on which the makespan of the applications varies at a slower rate than the bandwidth change ($\theta = 0.75$); and
- a high-varying class that includes the remaining VNs, i.e., the variation ratio in the makespan attains by $\theta = 1$ when the bandwidth is doubled.

In Fig. 3, we present the rejection ratio of algorithms with respect to the arrival rate, i.e., the number of VNs that are submitted to the data center per time slot. The results show that the ADA-BA algorithm outperforms other naive algorithms by reducing the rejection ratio by up to 25% compared to MIN-BA. In Fig. 4, we present the edgewise bandwidth allocated to a VN per time slot, i.e., the average of all δ_r 's values with $r \in \mathcal{R}^{\text{at}}$. While MIN-BA and MAX-BA consistently

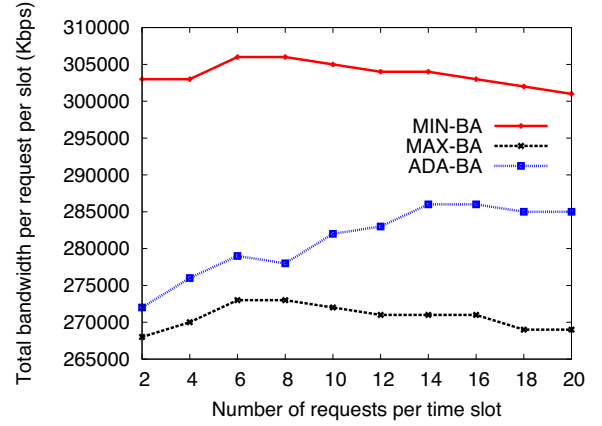


Fig. 5: Total bandwidth required per VN to complete execution of application running on VMs of virtual networks.

allocate minimum and maximum bandwidth to VNs, ADA-BA allocates more extra bandwidth to VNs when the arrival rate is low. The extra bandwidth is reduced gradually with respect to the increase in the arrival rate. Since residual bandwidth decreases when more VN requests are admitted.

The interesting observation is shown in Fig. 5 where we plot the total bandwidth required for a VN to complete the execution of the application running on the VMs. The results show that allocating minimum bandwidth to requests results in negative consequence such that the total bandwidth required for a VN during its lifetime in the data center is the highest. This is due to the fact that allocating minimum bandwidth makes the VNs remain in the data center longer, leading to the increase in the total utilized bandwidth. Conversely, allocating maximum bandwidth to VNs significantly reduces the total required bandwidth. However, it blocks others concurrent VNs due to insufficient residual bandwidth on the lightpaths. The adaptive bandwidth allocation (ADA-BA) uses less bandwidth than MIN-BA and accommodates more VNs than MAX-BA. This leads to better revenue for cloud providers. It is also

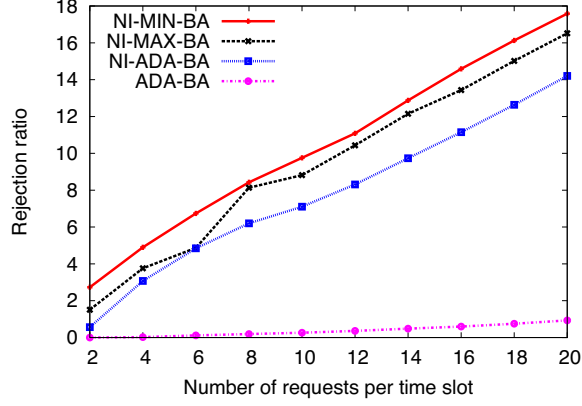


Fig. 6: Comparison against non-iterative embedding algorithm.

observed that when the arrival rate is high, the total bandwidth required for the entire running time of the VNs decreases since only the VNs with low bandwidth requirement are more likely to be accommodated with this high load.

2) *Iterative vs. Non-Iterative Virtual Network Embedding:* We also evaluate the performance of the VN embedding algorithm described in Section IV-B against the non-iterative VN embedding algorithm. It is worth recalling that the iterative VN embedding algorithm repeats through all the ToR candidates for the entry node of a VN until one ToR is found such that the entire VN is successfully embedded. On the other hand, the non iterative embedding algorithm rejects the VN if the first ToR candidate for the entry node does not lead to the successful embedding of the entire VN. In Fig. 6, we present the rejection ratio of the algorithms with respect to the arrival rate where the performance of all the non-iterative embedding algorithms are prefixed by the term “NI”. The results show that the iterative embedding algorithm significantly reduces the rejection ratio compared to the non-iterative algorithm. At the arrival rate of 20 VNs per time slot, all the non-iterative embedding algorithms reject more than 12% of VNs even with the adaptive bandwidth allocation. Combining the iterative embedding with the adaptive bandwidth allocation reduces the rejection ratio by up to 92% in the best case, thus demonstrating the effectiveness of the proposed algorithm.

3) *Proportional- vs. Even-Bandwidth Allocation:* We also run the simulation to compare the performance of the adaptive bandwidth allocation algorithms with two strategies presented in Section IV-C. We present the rejection ratio of the algorithms in Fig. 7 where the performance of all the algorithms, which use the even-bandwidth allocation for extra bandwidth, are post-fixed by the term “ED”. The results show that proportional-bandwidth allocation (ADA-BA) has slightly better performance than that of the algorithms with the even-bandwidth allocation. Since VN requests are heterogeneous with some of them having $\theta = 0$, allocating extra bandwidth to such VN requests as does the even-bandwidth allocation just wastes the bandwidth but does not improve the performance of the applications running on the VMs.

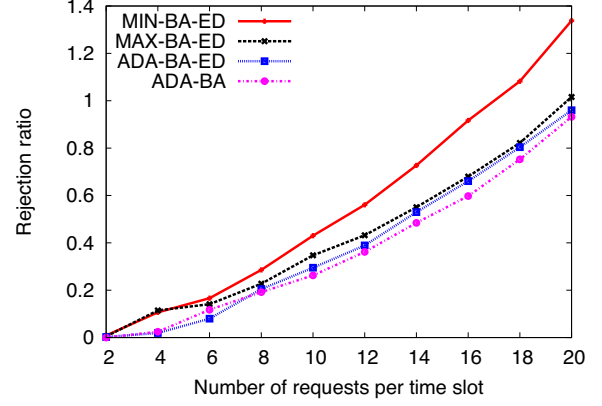


Fig. 7: Proportional- vs. even-allocation for extra bandwidth.

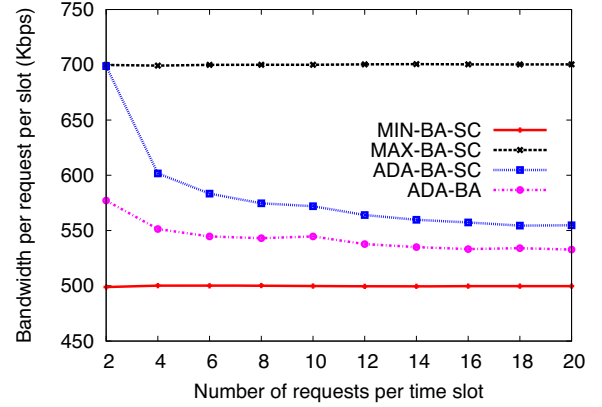


Fig. 8: Single class vs. Multiple classes of virtual networks.

4) *Single class vs. Multiple classes of Virtual Networks:* In this simulation, we show the performance difference when running with single class of VNs and multiple classes of VNs. As shown in previous simulations, the multiple classes of VNs include three categories with one category whose the makespan of applications does not change regardless of the bandwidth. The single class of VNs includes only the VNs that belong to the third category described above. Since we obtain the same behaviors for rejection ratio and total bandwidth utilized during the running time of a VN, we present only the edgewise bandwidth per VN per time slot for the algorithms in Fig. 8 where the performance of the algorithms with single class of VNs are denoted with postfix “SC”. It is observed that the edgewise bandwidth per VN per time slot with single class of VNs is higher than that of multi classes of VNs by up to 22%. It is because all the VNs in single class of VNs always receive additional bandwidth to shorten the makespan of the applications. When the arrival rate is low, all the requests can receive maximum bandwidth while several VNs in multiple classes of VNs do not receive any additional bandwidth, thus the average edgewise bandwidth per VN per time slot is smaller. This demonstrates the efficiency of the proposed algorithm that will not allocate more bandwidth for the VNs with no gain in performance.

VI. CONCLUSION

In this paper, we addressed the problem of adaptive and flexible bandwidth allocation for virtual network embedding in optical data center networks. We considered the broad category of cloud applications whose makespan depends on the bandwidth allocated for the communicating virtual machines. We developed the optimization programming formulations for the problem, which are computationally prohibitive. We therefore developed a heuristic algorithm that efficiently embeds the virtual networks in optical data center networks and adaptively allocates bandwidth to them so as to yield required performance for the applications while accommodating increased number of users. We carried out performance study through comprehensive simulations to demonstrate the effectiveness of the proposed algorithm. The results show that the proposed algorithm significantly reduces the rejection ratio by up to 25% compared to the baseline algorithms that do not consider the flexibility in bandwidth allocation.

ACKNOWLEDGMENT

This work was supported by Singapore Ministry of Education, Academic Research Fund Tier 2, Grant No. MOE2013-T2-2-135, NUS WBS No. R-263-000-B11-112.

REFERENCES

- [1] "Cisco Global Cloud Index: Forecast and Methodology, 2013-2018," White Paper, Cisco, Oct. 2015.
- [2] T. Truong-Huu, G. Koslovski, F. Anhalt, J. Montagnat, and P. Vicat-Blanc Primet, "Joint Elastic Cloud and Virtual Network Framework for Application Performance-cost Optimization," *Grid Computing*, 2011.
- [3] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards Predictable Datacenter Networks," in *ACM SIGCOMM 2011*, Toronto, Ontario, Canada, Aug. 2011, pp. 242–253.
- [4] X. Meng, V. Pappas, and L. Zhang, "Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement," in *INFOCOM 2010*, San Diego, California, USA, 2010, pp. 1154–1162.
- [5] J. Jiang, T. Lan, S. Ha, M. Chen, and M. Chiang, "Joint VM Placement and Routing for Data Center Traffic Engineering," in *IEEE INFOCOM 2012*, Orlando, FL, USA, 2012, pp. 2876–2880.
- [6] J. Lee, M. Lee, L. Popa, Y. Turner, S. Banerjee, P. Sharma, and B. Stephenson, "CloudMirror: Application-Aware Bandwidth Reservations in the Cloud," in *HotCloud 2013*, San Jose, CA, USA, 2013.
- [7] D. M. Divakaran, N. T. Le, and M. Gurusamy, "An Online Integrated Resource Allocator for Guaranteed Performance in Data Centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 6, pp. 1382–1392, June 2014.
- [8] S. Hegde, R. Srinivas, D. M. Divakaran, and M. Gurusamy, "Dynamic Embedding of Virtual Networks in Hybrid Optical-Electrical Datacenters," in *IEEE ICCCN 2014*, Shanghai, China, Aug. 2014, pp. 1–8.
- [9] L. Gong, W. Zhao, Y. Wen, and Z. Zhu, "Dynamic Transparent Virtual Network Embedding over Elastic Optical Infrastructures," in *IEEE ICC 2013*, Budapest, Hungary, June 2013, pp. 3466–3470.
- [10] A. Pagès, J. Perelló, S. Spadaro, J. García-Espín, J. Riera, and S. Figuerola, "Optimal Allocation of Virtual Optical Networks for the Future Internet," in *ONDM 2012*, Colchester, UK, Apr. 2012, pp. 1–6.
- [11] R. Srinivas, S. Hegde, D. M. Divakaran, and M. Gurusamy, "Virtual Network Embedding in Hybrid Datacenters with Dynamic Wavelength Grouping," in *CloudCom 2014*, Singapore, Dec. 2014, pp. 905–910.
- [12] V. Girisagar, T. Truong-Huu, and M. Gurusamy, "Mapping Workflow Resource Requests for Bandwidth Efficiency in Data Centers," in *ICDCN 2016*, Singapore, Jan. 2016, pp. 1–10.
- [13] N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat, "Helios : A Hybrid Electrical/Optical Switch Architecture for Modular Data Centers," in *ACM SIGCOMM 2010*, New Delhi, India, 2010, pp. 339–350.
- [14] K. Chen, A. Singla, A. Singh, K. Ramachandran, L. Xu, Y. Zhang, X. Wen, and Y. Chen, "OSA : An Optical Switching Architecture for Data Center Networks with Unprecedented Flexibility," *IEEE/ACM Transactions on Networking*, vol. 22, no. 2, pp. 498–511, 2014.
- [15] L. Caccetta and A. Kulanoort, "Computational aspects of hard Knapsack problems," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 47, no. 8, pp. 5547–5558, 2001.
- [16] E. Cela, *The Quadratic Assignment Problem: Theory and Algorithms*. Springer US, 1998.