On the Parallel Reconfiguration of Virtual Networks in Hybrid Optical/Electrical Datacenter Networks

Sicheng Zhao†, Xiaoqin Pan†‡, and Zuqing Zhu†
†School of Information Science and Technology, University of Science and Technology of China, Hefei, China
‡Engineering Technology Center, Southwest University of Science and Technology, Mianyang, China
†Email: {zqzhu}@ieee.org

***Abstract*—Recently, hybrid optical/electrical datacenter networks (HOE-DCNs) have been considered as a promising DCN architecture, because they merge the merits of electrical packet switching (EPS) and optical circuit switching (OCS). This paper considers the reconfiguration of virtual networks (VNTs) in an HOE-DCN to address the dynamic nature of emerging network services. Specifically, we study the problem that given the original and new virtual network embedding (VNE) schemes of several VNTs, how to schedule parallel virtual machine (VM) migrations in batches to realize the VNT reconfiguration within the shortest time. We design two algorithms to reconfigure the inter-rack network in an HOE-DCN in steps, schedule VMs to migrate accordingly, and allocate bandwidth to the VM migrations. The first algorithm uses the one-shot approach, where all the VM migrations are conducted in parallel within the shortest possible time. We formulate a linear programming (LP) to solve the bandwidth allocations in it exactly. Next, to relieve the bandwidth competition introduced by the one-shot approach, we propose the second algorithm by leveraging the multi-shot approach, *i.e.*, invoking multiple batches of parallel VM migrations such that the reconfiguration time can be further reduced. Extensive simulations verify the effectiveness of our proposals.**

***Index Terms*—Hybrid optical/electrical datacenter networks, Network virtualization, VM migration, Parallel reconfiguration.**

I. Introduction

Over the past decades, due to the raising of 5G, Big Data analytics, and other bandwidth-hungry applications [1], the Internet traffic that is related to datacenters (DCs) has been increasing exponentially [2]. Therefore, DC networks (DCNs) are facing substantial challenges from architecture scalability, energy efficiency, and management agility [3]. This motivates people to not only reconsider the architecture of DCNs, but also design novel network orchestration techniques for DCNs. From the perspective of DCN architecture, one of the most promising improvements is to consider the hybrid optical/electrical DCN (HOE-DCN) [4, 5], which inserts one or more optical switches in the inter-rack network of a DCN and combines optical circuit switching (OCS) together with electrical packet switching (EPS). This is because the advantages of OCS, especially those on bandwidth capacity and energy efficiency [6–11], and can be integrated seamlessly with the benefits of EPS, to make the DCN more scalable, cost-effective, and adaptive for emerging applications with various quality-of-service (QoS) requirements.

On the other hand, network orchestration can leverage new techniques, such as network virtualization [12–14], software-defined networking (SDN) [15–19], and machine learning based artificial intelligence (AI) [20–23], to allocate the IT and bandwidth resources in a DCN in the timely, flexible and application-aware manner. For instance, in [22, 24], we combined SDN, AI and network virtualization to experimentally demonstrate the knowledge-defined network orchestration (KD-NO) based on predictive analytics for HOE-DCNs, which not only realized timely and precise resource allocations to support highly-dynamic DC applications (*e.g.*, Hadoop MapReduce), but also further squeezed the well-known energy-latency tradeoff. Specifically, each DC application was treated as a virtual network (VNT), which consists of virtual machines (VMs) for computing/storage tasks and virtual links (VLs) to enable the data transfers among the VMs, and we designed KD-NO schemes to adjust the virtual network embedding (VNE) scheme of the VNT adaptively to avoid the performance degradations due to resource competition.

Although the proposals regarding KD-NO in HOE-DCNs [22, 24] were promising, the VNT reconfiguration involved in them has to be further studied to ensure the scalability and practicalness. More specifically, two additional problems should be tackled, which are 1) how to calculate the reconfiguration schemes of VNTs based on the network state in an HOE-DCN, and 2) given a VNE reconfiguration scheme, how to design the procedure to accomplish it quickly in the HOE-DCN. We have considered the first problem in [25], while to the best of our knowledge, the second one has not been addressed in the literature yet.

Note that, given the original and new VNE schemes of a VNT, we need to reconfigure the mapping schemes of both the affected VMs and VLs [26, 27]. This means that certain VMs need to be migrated to new servers, while all the affected VLs need to be embedded in the EPS-and-OCS-combined inter-rack network according to the new VNE scheme. It is known that in DCNs, a VM migration usually takes tens of seconds or even minutes [28], which will be much longer than the time used for remapping VLs. Hence, for the second problem about the reconfiguration procedure, we can ignore the latency of VL remappings, but concentrate on the scheduling of VM migrations to minimize the reconfiguration latency.

Although the design of VNT reconfiguration procedure can be reduced to solving the scheduling of related VM migrations, the second problem is still fundamentally different from the conventional scheduling of VM migrations in DCNs [29, 30]. The reasons are two-fold. Firstly, even though we ignore the

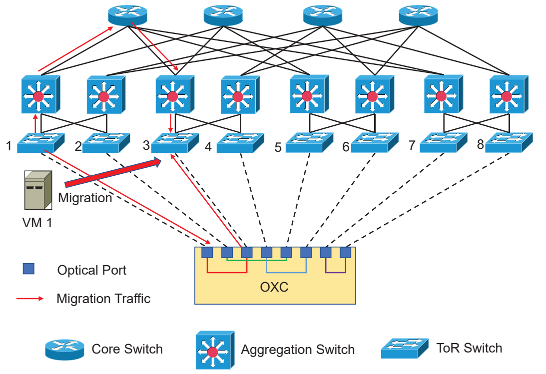


Fig. 1. Architecture of an HOE-DCN and VM migration in it.

latency of VL remappings, the bandwidth usages of VLs should still be considered during the VNT reconfiguration because we would like to use live VM migrations to minimize service interruptions. In other words, the reconfigurations of VMs and VLs are still correlated, *i.e.*, each VM migration can change the bandwidth usages of certain VLs, which will in turn affect the bandwidth available for other VM migrations. This is more complex than the pure scheduling of VM migrations that does not consider the bandwidth usages of the traffic among VMs [29, 30]. Secondly and more importantly, all the existing studies on VM migration scheduling in DCNs did not consider HOE-DCNs. Note that, as shown in Fig. 1, the optical cross-connects (OXCs) used in HOE-DCNs normally provide the "1-to-1" connectivity, *i.e.*, an OXC reconfiguration will change the physical connections between the input and output ports of an OXC [25]. Therefore, in addition to bandwidth usages, a VM migration can also change the physical topology of the inter-rack network in an HOE-DCN. This makes the scheduling of VM migrations even more complex.

In this paper, we investigate the problem that given the original and new VNE schemes of several VNTs, how to schedule parallel VM migrations in batches to realize the VNT reconfiguration within the shortest time. We design two parallel VM migration algorithms to reconfigure the inter-rack network in an HOE-DCN in steps, schedule VMs to migrate accordingly, and allocate bandwidth to the VM migrations. The first algorithm is referred to as the one-shot approach, where we first conduct all the VM migrations in parallel within the shortest possible time, and then reconfigure the OXC and related VLs. We formulate the bandwidth allocation in the one-shot approach as a linear programming (LP) model, and solve it exactly to obtain the shortest migration time. The second algorithm (*i.e.*, the multi-shot approach) leverages multiple batches of parallel VM migrations and reconfiguration of OXC and related VLs to relieve the bandwidth competition caused by the one-shot approach, and thus the reconfiguration latency can be further reduced. Extensive simulations verify the effectiveness of our proposals.

The rest of the paper is organized as follows. Section II provides the problem description. The two VNT reconfiguration algorithms are designed in Section III. In Section IV, we conduct numerical simulations to evaluate our proposals. Finally, Section V summarizes the paper.

II. Problem Description

A. Network Model

We model the topology of the inter-rack network in an HOE-DCN as a graph $G(V_s, E_s)$, where V_s and E_s are the sets of substrate nodes (SNs) and substrate links (SLs) for network virtualization, respectively. Here, each SN $v_s \in V_s$ represents a server rack that consists of a top-of-rack (ToR) switch and a few servers, and each SL $e_s \in E_s$ can be either an EPS-based or an OCS-based network connection. In the HOE-DCN, each pair of ToR switches are constantly connected through its EPS-based part (*i.e.*, a hierarchical topology based on Ethernet switches), while they can also talk through the OCS-based part if the OXC is properly configured. For instance, the OXC in Fig. 1 is configured to enable OCS-based connections between rack pairs 1-3, 2-5, 4-6, and 7-8. We assume that the EPS-based part is non-blocking (*e.g.*, it uses the k-ray fat-tree topology [31]). Hence, the bandwidth capacity to/from a ToR switch v_s through the EPS-based part is only limited by the ToR switch's port rate, which is denoted as B_{v_s}. Meanwhile, if the OXC is configured to bridge the communication between ToR switches v_s and u_s, the bandwidth capacity through the OCS-based part for the switch pair is referred to as $B_{(v_s,u_s)}$.

The topology of a VNT is modeled as $G_r(V_r, E_r)$, where V_r is the set of VMs and E_r is the set of VLs that interconnect the VMs. As the VNT reconfiguration involves VM migrations, we define the image size of a VM $v_r \in V_r$ as c_{v_r}. The bandwidth usage of a VL $(v_r, u_r) \in E_r$ is denoted as $b_{(v_r,u_r)}$.

B. VNT Reconfiguration

We focus on the problem that given the original and new VNE schemes of several VNTs, how to schedule the actions for remapping the related VMs and VLs such that the VNT reconfiguration can be accomplished within the shortest time.

$$\mathcal{M} = \begin{cases} \mathcal{M}_N: & V_r \mapsto V_s, \\ \mathcal{M}_L: & E_r \mapsto P_s, \end{cases} \quad \mathcal{M}' = \begin{cases} \mathcal{M}'_N: & V_r \mapsto V_s, \\ \mathcal{M}'_L: & E_r \mapsto P_s, \end{cases} \tag{1}$$

where $\mathcal{M}$ and $\mathcal{M}'$ are the original and new VNE schemes[1] of a VNT $G_r(V_r, E_r)$, respectively, the corresponding node and link mapping schemes are $\{\mathcal{M}_N, \mathcal{M}'_N\}$ and $\{\mathcal{M}_L, \mathcal{M}'_L\}$, respectively, and P_s denotes the set of pre-calculated substrate paths in $G(V_s, E_s)$. The VNT reconfiguration involves the remappings of related VMs and VLs (*i.e.*, $\mathcal{M}_N \rightarrow \mathcal{M}'_N$ and $\mathcal{M}_L \rightarrow \mathcal{M}'_L$), among which the VL remappings are performed when the related VMs have been migrated successfully, and because they take much shorter time than the VM migrations, we ignore their latencies. Therefore, our problem is reduced to how to schedule parallel VM migrations in batches to realize the VNT reconfiguration within the shortest time.

By analyzing the VM migrations of all the related VNTs, we can group the VMs whose source and destination racks are the same as one migration unit (MU), namely, $m \in M$, where M is set of all the obtained MUs. As shown in Fig. 1, the migration of an MU can leverage the bandwidth capacities in

[1]Note that, $\mathcal{M}$ and $\mathcal{M}'$ are the preset inputs to the algorithm developed in this work, and thus we will not discuss how to calculate them but concentrate on the procedure for realizing the VNT reconfiguration based on them.

both the EPS- and OCS-based parts of the HOE-DCN. More illustratively, we use the example in Fig. 2 to explain the VM migrations in an HOE-DCN. Here, we need to migrate three MUs, and based on the current OXC configuration, *MUs* 1 and 2 can use the bandwidth capacities of both the EPS- and OCS-based parts, while *MU* 3 can only be migrated through the EPS-based part. Hence, in order to minimize the overall migration latency, we need to schedule the MUs' migrations in parallel and allocate bandwidth resources in the HOE-DCN to them accordingly, which can be achieved with the algorithms designed in the next section.

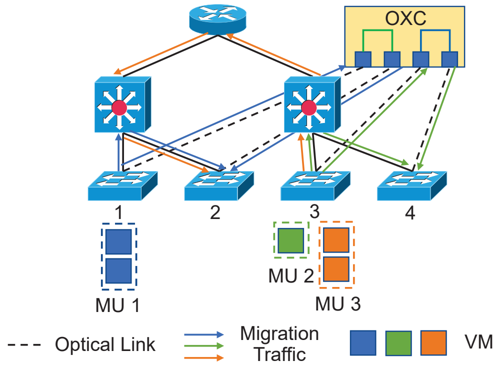


Fig. 2. Example on parallel VM migrations in an HOE-DCN.

III. Algorithm Designs

In this section, we design two algorithms to realize parallel VM migrations for the VNT reconfiguration in an HOE-DCN, based on the one-shot and multi-shot approaches, respectively.

A. One-Shot Approach

In the one-shot approach, we migrate all the VMs in parallel based on the current configuration of the HOE-DCN, and then reconfigure the OXC and related VLs. Then, the problem becomes how to allocate bandwidth to the VM migrations such that their overall migration time can be minimized. This can be described with the following linear programming (LP) model.

Notations:

- M: the set of MUs.
- $G(V_s, E_s)$: the inter-rack topology of the HOE-DCN.
- R_s: the set of rack pairs in the HOE-DCN.
- $b^{\text{in}}_{v_s}/b^{\text{out}}_{v_s}$: the available bandwidth capacities to/from rack $v_s \in V_s$ through the EPS-based inter-rack, respectively.
- $b_{(u_s,v_s)}$: the available bandwidth capacity from rack u_s to rack v_s through the OCS-based inter-rack, and it equals 0 if u_s cannot talk with v_s through the OXC.
- c_m: the total image size of all the VMs in MU $m \in M$.
- s_m: the source rack of MU m.
- d_m: the destination rack of MU m.
- $s^{v_s}_m$: the boolean parameter that equals 1 if rack v_s is the source rack of MU $m \in M$, and 0 otherwise.
- $d^{v_s}_m$: the boolean parameter that equals 1 if rack v_s is the destination rack of MU $m \in M$, and 0 otherwise.

Variables:

- x_m: the real variable that denotes the bandwidth allocated in the EPS-based inter-rack to migrate MU $m \in M$.
- y_m: the real variable that denotes the bandwidth allocated in the OCS-based inter-rack to migrate MU $m \in M$.
- ξ: the reciprocal of the overall migration time.

Objective:

As we invoke the VM migrations in parallel, the overall migration time is just the longest migration time of an MU. For each MU, its migration time can be obtained by dividing its total image size with the allocated bandwidth. Hence, to avoid nonlinearity, we set the optimization objective as to maximize the reciprocal of the overall migration time, which is equivalent to minimizing the overall migration time.

$$\text{Maximize} \quad \xi. \tag{2}$$

Constraints:

$$\sum_{m\in M} x_m \cdot s^{v_s}_m \le b^{\text{out}}_{v_s}, \quad \forall v_s \in V_s, \tag{3}$$

$$\sum_{m\in M} x_m \cdot d^{v_s}_m \le b^{\text{in}}_{v_s}, \quad \forall v_s \in V_s. \tag{4}$$

Eqs. (3)-(4) ensure that the bandwidths allocated in the EPS-based inter-rack for MUs migrated from/to rack $v_s \in V_s$ do not exceed the available bandwidth capacities, respectively.

$$y_m \le b_{(s_m,d_m)}, \quad \forall m \in M. \tag{5}$$

Eq. (5) ensures that the bandwidth allocated in the OCS-based inter-rack for migrating MU m from s_m to d_m does not exceed the available bandwidth capacity.

$$\begin{aligned} &x_m \ge 0, \ \forall m \in M, \\ &y_m \ge 0, \ \forall m \in M. \end{aligned} \tag{6}$$

Eq. (6) ensures that the allocated bandwidths are non-negative.

$$\xi \le \frac{x_m + y_m}{c_m}, \quad \forall m \in M. \tag{7}$$

Eq. (7) ensures that the value of ξ is set as the minimum reciprocal of an MU's migration time.

The aforementioned LP model can be solved exactly in polynomial-time. For example, if we solve it with the well-known interior point method [32], the time complexity is $O(W^{3.5} \cdot N)$, where W is the number of variables in the LP and N is the total number of bits of the input.

B. Multi-Shot Approach

Although the LP model can provide us the exact solution to schedule VM migrations in parallel, the one-shot approach might not always lead to the shortest migration time due to bandwidth competition. This motivates us to consider a multi-shot approach. Fig. 3 gives an illustrative example on the comparison between one-shot and multi-shot approaches. Here, for the three MUs, their image sizes are 1, 2 and 1 units, respectively, the bandwidth usages of their VMs are 1, 2 and 1 units/time-unit, respectively, and the available bandwidth capacities on their ToR switches are 3, 1 and 3 units/time-unit, respectively. Therefore, when using the one-shot approach, we cannot reduce the overall migration time due to the bandwidth competition on the ToR switch for *Rack* 2, *i.e.*, the overall migration time will be $\frac{1+2}{1} = 3$ time-units. On the other hand, if we first migrate *MUs* 1 and 3 in parallel and then handle *MU* 2, we can get a shorter overall migration time as $\frac{1}{1} + \frac{2}{2} = 2$ time-units. Note that, the bandwidth competition in Fig. 3 can happen when MUs to/from different racks are being migrated

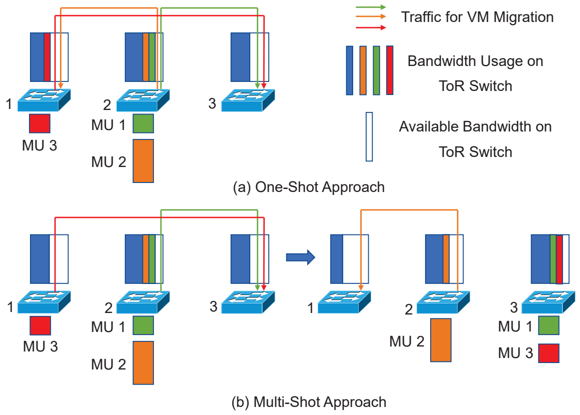


Fig. 3. Comparison between one-shot and multi-shot VM migrations.

from/to a same rack. This suggests that it only happens in the EPS-based part of the HOE-DCN, while since the OXC only supports "1-to-1" connectivity among the racks, the bandwidth competition will not occur in the OCS-based part.

Algorithm 1 shows the detailed procedure of our multi-shot approach. Specifically, it leverages a while-loop to schedule parallel VM migrations in batches. In each iteration, *Lines* 2-3 are for the initialization, where M_p will store the MUs selected to migrate in the current batch. The for-loop covering *Lines* 4-18 checks each rack to select the MUs that will be migrated in parallel. Here, we first use the for-loop that covers *Lines* 6-12 to evaluate each MU m on a rack. We get the destination rack of m as u_s (*Line* 7), hypothetically migrate m to get the resulting available bandwidths in the EPS-based part of the HOE-DCN (*Line* 8), and insert m in the temporary set M_t if the resulting bandwidths satisfy $b_{v_s}^{\text{out},t} < b_{u_s}^{\text{in},t}$ (*Lines* 9-11).

Then, if the temporary set M_t is not empty, we select the MU m^* from it with the expression in *Line* 14, where c_m is the total image size of all the VMs in MU m, and b_m^e is the total bandwidth usage (in the EPS-based part) of all the VMs in MU m. Note that, the bandwidth usage b_m^e is for the normal operation of the VMs but not for the VM migrations. The rationale behind *Lines* 5-17 is that if we first select the MUs whose migrations are from the racks whose output bandwidth usages are the highest to those whose input bandwidth usages are the lowest (*i.e.*, the usages are all in the EPS-based part), the bandwidth competition in the HOE-DCN can be relieved. The MUs selected to migrate in the current batch are stored in set M_p (*Line* 15). If M_p is not empty, we obtain the bandwidth allocations for migrating all the MUs in it in parallel by replacing M with M_p and solving the LP for the one-shot approach (*Lines* 19-21). Otherwise, we migrate all the remaining MUs in set M in parallel also by solving the LP (*Lines* 22-24). Since we need to solve the LP model in each iteration of the outer while-loop (*Lines* 1-26), the time complexity of *Algorithm* 1 is $O(|M| \cdot (|V_s| \cdot |M| + O(W^{3.5} \cdot N)))$.

IV. Performance Evaluation

A. Simulation Setup

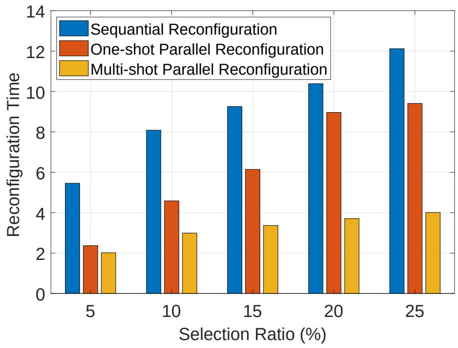
In this section, we perform simulations to evaluate the performance of our proposed algorithms for realizing parallel

Algorithm 1: Multi-Shot Parallel VM Migrations

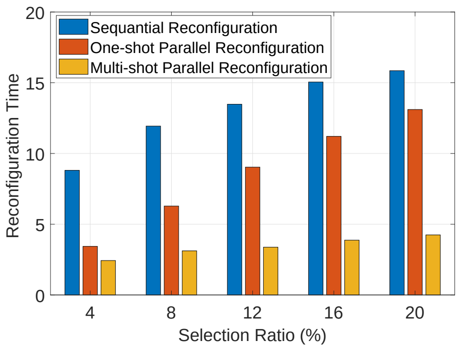
1 **while** $M \neq \emptyset$ **do**
2 $\quad M_p = \emptyset$;
3 $\quad$ store values of $\{b_{v_s}^{\text{in}}, b_{v_s}^{\text{out}}\}$ and $\{b_{(u_s,v_s)}\}$ in temporary variables $\{b_{v_s}^{\text{in},t}, b_{v_s}^{\text{out},t}\}$ and $\{b_{(u_s,v_s)}^t\}$;
4 $\quad$ **for** *each rack $v_s \in V_s$ that has VMs to migrate* **do**
5 $\quad\quad M_t = \emptyset$;
6 $\quad\quad$ **for** *each MU m on rack v_s* **do**
7 $\quad\quad\quad$ get destination rack of m as u_s;
8 $\quad\quad\quad$ migrate m hypothetically to get the resulting $b_{v_s}^{\text{out},t}$ and $b_{u_s}^{\text{in},t}$;
9 $\quad\quad\quad$ **if** $b_{v_s}^{out,t} < b_{u_s}^{in,t}$ **then**
10 $\quad\quad\quad\quad$ insert m in M_t;
11 $\quad\quad\quad$ **end**
12 $\quad\quad$ **end**
13 $\quad\quad$ **if** $M_t \neq \emptyset$ **then**
14 $\quad\quad\quad$ $m^* = \underset{m \in M_t}{\text{argmin}} \left(\frac{c_m}{b_m^e}\right)$;
15 $\quad\quad\quad$ insert m^* in M_p and remove m^* from M;
16 $\quad\quad\quad$ migrate m^* hypothetically to update the values of related $b_{v_s}^{\text{out},t}$, $b_{u_s}^{\text{in},t}$ and $b_{(v_s,u_s)}^t$;
17 $\quad\quad$ **end**
18 $\quad$ **end**
19 $\quad$ **if** $M_p \neq \emptyset$ **then**
20 $\quad\quad$ migrate all the MUs in M_p in parallel;
21 $\quad\quad$ reconfigure the OXC and related VLs and update the status of HOE-DCN;
22 $\quad$ **else**
23 $\quad\quad$ migrate all the MUs in M in parallel;
24 $\quad\quad$ **break**;
25 $\quad$ **end**
26 **end**

reconfiguration of VNTs in an HOE-DCN. The simulations use the k-ray fat-tree topology [31] to architect the EPS-based inter-rack network. Specifically, the k-ray fat-tree topology evenly distributes $\frac{k^2}{2}$ racks/ToR switches in k points-of-delivery (PoDs). Each ToR switch has $\frac{k}{2}$ Ethernet ports to connect to the EPS-based inter-rack, and it also equips an optical port to the OXC. We consider the 6-ray and 8-ray fat-tree topologies in our simulations, *i.e.*, there are 18 and 32 racks/ToR switches, respectively. The bandwidth capacity of each Ethernet port on a ToR switch is set as 1000 units/time-unit, while that of its optical port is set as 10000 units/time-unit. The simulations consider dynamic network environment where VNTs can be set up and torn down on-the-fly. Hence, we use the Poisson model to dynamically generate VNTs with random topologies. The number of VMs in each VNT is randomly selected within $[2, 16]$, and the VMs' connectivity ratio is set as 0.5. The image size of each VM is evenly distributed within $[50, 200]$ units, while the bandwidth usage of each VL is selected within $[15, 60]$ units/time-unit.

Each simulation runs as follows. We first use the global-resource-capacity based VNE algorithm developed in [33] to embed the dynamically-generated VNTs, pause the provisioning of VNTs when unbalanced resource utilizations occur in the HOE-DCN, and then leverage the VNT reconfiguration algorithm that we designed in [25] to select the VMs to migrate and calculate the new VNE schemes for the related VNTs. Next, with the original and new VNE schemes as the

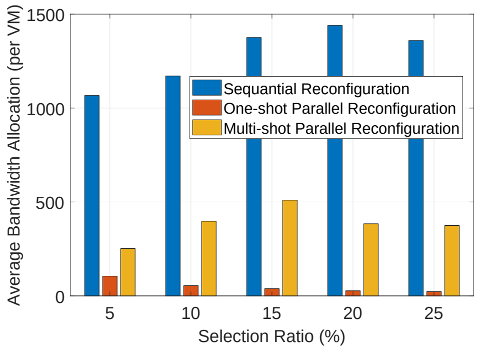


(a) HOE-DCN with 6-ray fat-tree

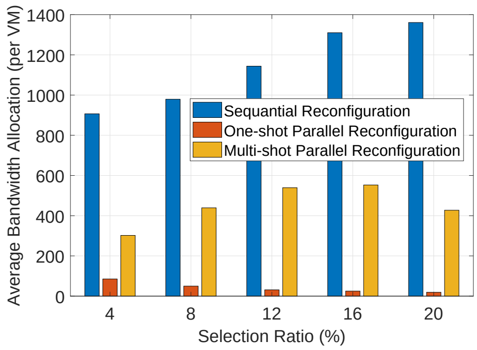


(b) HOE-DCN with 8-ray fat-tree

Fig. 4. Results on overall reconfiguration time.



(a) HOE-DCN with 6-ray fat-tree



(b) HOE-DCN with 8-ray fat-tree

Fig. 5. Average bandwidth allocated to each VM migration.

inputs, we utilize the one-shot and multi-shot approaches to schedule parallel VM migrations in batches and realize the VNT reconfiguration. To evaluate our algorithms with different volumes of VMs to migrate, we define a selection ratio γ, which denotes the ratio of the VMs that are selected to migrate to all the in-service VMs, and changes as $\gamma \in [4\%, 25\%]$ in the simulations. In addition to the proposed algorithms, we also consider a sequential VNT reconfiguration algorithm (*i.e.*, the MUs are migrated one by one) as the benchmark. To guarantee sufficient statistical accuracy, we average the results from 5 independent runs to get each data point.

B. Migration Time

We first obtain the overall reconfiguration time with the three algorithms, and plot the results in Fig. 4. We can see that the reconfiguration time generally increases with the selection ratio γ. This is well expected because a larger γ means that more selected VMs are given to the VNT reconfiguration algorithms for scheduling their migrations. Meanwhile, when comparing the results obtained in the HOE-DCNs with 6-ray and 8-ray fat-tree topologies, we observe that the reconfiguration time in the 8-ray fat-tree is always longer when other parameters are similar. This is because there are more racks in the 8-ray fat-tree and thus more VMs can be accommodated. In other words, with similar γ, more VMs are selected in the HOE-DCN with 8-ray fat-tree for VNT reconfiguration.

The results in Fig. 4 also indicate that the sequential reconfiguration algorithm always uses the longest time for VM migrations, while the multi-shot approach achieves the best performance on overall reconfiguration time. This observation verifies that our multi-shot approach can schedule parallel VM migrations in batches, such that the bandwidth competition in the EPS-based inter-rack network is effectively relieved. Moreover, we can see that the reconfiguration time from the multi-shot approach increases much slower than that from the one-shot approach, when the selection ratio γ increases. This suggests that the multi-shot approach can maintain the overall reconfiguration time well, even when it needs to schedule a larger number of VM migrations. Therefore, the algorithm's effectiveness gets further confirmed.

C. Distribution of Bandwidth Allocations

Next, we investigate the average bandwidth allocated to each VM migration, and Fig. 5 shows the results. We can see that sequential reconfiguration algorithm allocates the most bandwidth to each VM migration, followed by the multi-shot and one-shot approaches in sequence. The phenomenon can be explained as follows. First of all, both the sequential reconfiguration and multi-shot parallel reconfiguration algorithms schedule the VM migrations in batches, which means that they can give all the available bandwidths to a portion of the VM migrations in each batch. Hence, their average bandwidth allocations are larger than that of the one-shot approach, which has to distribute the available bandwidths to all the VM migrations simultaneously. Secondly, different from the sequential reconfiguration, which only migrates one MU at a time, the multi-shot approach still invokes parallel VM migrations, *i.e.*, in the same batch, the available bandwidths can be shared by the migrations of several MUs. Therefore, the average bandwidth allocation from the multi-shot approach is smaller than that of the sequential reconfiguration algorithm. Meanwhile, it can be seen that the average bandwidth allocations from both the one-shot and multi-shot approaches can decrease with γ, while this will not happen for the sequential reconfiguration. This is because when parallel reconfiguration is considered, both the available bandwidth for VM migration and number of VM migrations in a batch can increase with γ.

V. Conclusion

In this paper, we studied the procedure of reconfiguring VNTs in parallel in HOE-DCNs, *i.e.*, given the original and new VNE schemes of several VNTs, how to schedule parallel VM migrations in batches to realize the VNT reconfiguration within the shortest time. We proposed two parallel VM migration algorithms. The first one used the one-shot approach, and we formulated an LP to solve the scheduling problem for it exactly. Then, to overcome the drawbacks due to the bandwidth competition during parallel VM migrations, we proposed the second algorithm by leveraging the multi-shot approach, *i.e.*, invoking multiple batches of parallel VM migrations such that the migration time can be further reduced. The results from extensive simulations verified that both the one-shot and multi-shot approaches provide much shorter migration time than the benchmark using sequential reconfiguration, and the multi-shot approach achieves the fastest VNT reconfiguration.

Acknowledgments

This work was supported in part by the NSFC projects 61871357, 61771445 and 61701472, Zhejiang Lab Research Fund 2019LE0AB01, CAS Key Project (QYZDY-SSW-JSC003), and SPR Program of CAS (XDC02070300).

References

[1] P. Lu *et al.*, "Highly-efficient data migration and backup for Big Data applications in elastic optical inter-datacenter networks," *IEEE Netw.*, vol. 29, pp. 36–42, Sept./Oct. 2015.

[2] Cisco Global Cloud Index: Forecast and Methodology, 2016-2021. [Online]. Available: https://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html

[3] W. Lu *et al.*, "AI-assisted knowledge-defined network orchestration for energy-efficient data center networks," *IEEE Commun. Mag.*, vol. 58, pp. 86–92, Jan. 2020.

[4] N. Farrington *et al.*, "Helios: a hybrid electrical/optical switch architecture for modular data centers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 339–350, Oct. 2010.

[5] Q. Li *et al.*, "Scalable knowledge-defined orchestration for hybrid optical/electrical datacenter networks," *J. Opt. Commun. Netw.*, vol. 12, pp. A113–A122, Feb. 2020.

[6] L. Gong, X. Zhou, W. Lu, and Z. Zhu, "A two-population based evolutionary approach for optimizing routing, modulation and spectrum assignments (RMSA) in O-OFDM networks," *IEEE Commun. Lett.*, vol. 16, pp. 1520–1523, Sept. 2012.

[7] Z. Zhu, W. Lu, L. Zhang, and N. Ansari, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," *J. Lightw. Technol.*, vol. 31, pp. 15–22, Jan. 2013.

[8] W. Lu and Z. Zhu, "Dynamic service provisioning of advance reservation requests in elastic optical networks," *J. Lightw. Technol.*, vol. 31, pp. 1621–1627, May 2013.

[9] L. Gong *et al.*, "Efficient resource allocation for all-optical multicasting over spectrum-sliced elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. 836–847, Aug. 2013.

[10] Y. Yin *et al.*, "Spectral and spatial 2D fragmentation-aware routing and spectrum assignment algorithms in elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. A100–A106, Oct. 2013.

[11] M. Zhang, C. You, H. Jiang, and Z. Zhu, "Dynamic and adaptive bandwidth defragmentation in spectrum-sliced elastic optical networks with time-varying traffic," *J. Lightw. Technol.*, vol. 32, pp. 1014–1023, Mar. 2014.

[12] L. Gong and Z. Zhu, "Virtual optical network embedding (VONE) over elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 450–460, Feb. 2014.

[13] H. Jiang, Y. Wang, L. Gong, and Z. Zhu, "Availability-aware survivable virtual network embedding (A-SVNE) in optical datacenter networks," *J. Opt. Commun. Netw.*, vol. 7, pp. 1160–1171, Dec. 2015.

[14] L. Gong, H. Jiang, Y. Wang, and Z. Zhu, "Novel location-constrained virtual network embedding (LC-VNE) algorithms towards integrated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 24, pp. 3648–3661, Dec. 2016.

[15] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN: an intellectual history of programmable networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 87–98, Apr. 2014.

[16] C. Chen *et al.*, "Demonstrations of efficient online spectrum defragmentation in software-defined elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 4701–4711, Dec. 2014.

[17] S. Li *et al.*, "Protocol oblivious forwarding (POF): Software-defined networking with enhanced programmability," *IEEE Netw.*, vol. 31, pp. 12–20, Mar./Apr. 2017.

[18] Z. Zhu *et al.*, "Demonstration of cooperative resource allocation in an OpenFlow-controlled multidomain and multinational SD-EON testbed," *J. Lightw. Technol.*, vol. 33, pp. 1508–1514, Apr. 2015.

[19] S. Li *et al.*, "Improving SDN scalability with protocol-oblivious source routing: A system-level study," *IEEE Trans. Netw. Serv. Manag.*, vol. 15, pp. 275–288, Mar. 2018.

[20] A. Mestres *et al.*, "Knowledge-defined networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 47, pp. 2–10, Jul. 2017.

[21] J. Guo and Z. Zhu, "When deep learning meets inter-datacenter optical network management: Advantages and vulnerabilities," *J. Lightw. Technol.*, vol. 36, pp. 4761–4773, Oct. 2018.

[22] H. Fang *et al.*, "Predictive analytics based knowledge-defined orchestration in a hybrid optical/electrical datacenter network testbed," *J. Lightw. Technol.*, vol. 37, pp. 4921–4934, Oct. 2019.

[23] B. Li, W. Lu, and Z. Zhu, "Deep-NFVOrch: Leveraging deep reinforcement learning to achieve adaptive vNF service chaining in EON-DCIs," *J. Opt. Commun. Netw.*, vol. 12, pp. A18–A27, Jan. 2020.

[24] W. Lu *et al.*, "Leveraging predictive analytics to achieve knowledge-defined orchestration in a hybrid optical/electrical DC network: Collaborative forecasting and decision making," in *Proc. of OFC 2018*, pp. 1–3, Mar. 2018.

[25] S. Zhao and Z. Zhu, "Network service reconfiguration in hybrid optical/electrical datacenter networks," in *Proc. of ONDM 2020*, pp. 1–6, May 2020.

[26] J. Liu *et al.*, "On dynamic service function chain deployment and readjustment," *IEEE Trans. Netw. Serv. Manag.*, vol. 14, pp. 543–553, Sept. 2017.

[27] S. Zhao, D. Li, K. Han, and Z. Zhu, "Proactive and hitless vSDN reconfiguration to balance substrate TCAM utilization: From algorithm design to system prototype," *IEEE Trans. Netw. Serv. Manag.*, vol. 16, pp. 647–660, Jun. 2019.

[28] B. Kong *et al.*, "Demonstration of application-driven network slicing and orchestration in optical/packet domains: On-demand vDC expansion for Hadoop MapReduce optimization," *Opt. Express*, vol. 26, pp. 14 066–14 085, 2018.

[29] K. Chen, C. Chen, and P. Wang, "Network aware load-balancing via parallel VM migration for data centers," in *Proc. of ICCCN 2014*, pp. 1–8, Aug. 2014.

[30] S. Xiao *et al.*, "Traffic-aware virtual machine migration in topology-adaptive DCN," in *Proc. of ICNP 2016*, pp. 1–10, Oct. 2016.

[31] Y. Zhang and N. Ansari, "On architecture design, congestion notification, TCP incast and power consumption in data centers," *IEEE Commun. Surveys Tuts.*, vol. 15, pp. 39–64, First Quarter 2012.

[32] G. Strang, "Karmarkar's algorithm and its place in applied mathematics," *Math. Intell.*, vol. 9, pp. 4–10, Jan. 1987.

[33] L. Gong, Y. Wen, Z. Zhu, and T. Lee, "Toward profit-seeking virtual network embedding algorithm via global resource capacity," in *Proc. of INFOCOM 2014*, pp. 1–9, Apr. 2014.