Network Service Reconfiguration in Hybrid Optical/Electrical Datacenter Networks

Sicheng Zhao and Zuqing Zhu†
School of Information Science and Technology, University of Science and Technology of China, Hefei, China
†Email: {zqzhu}@ieee.org

***Abstract*—The hybrid optical/electrical datacenter networks (HOE-DCNs), which use both electrical Ethernet switches and optical cross-connects (OXCs) to interconnect top-of-rack (ToR) switches, have been considered as a scalable solution for the ever-increasing datacenter traffic. Meanwhile, the network services in an HOE-DCN generate various virtual network (VNT) requests, and their dynamic nature can degrade the optimality of their virtual network embedding (VNE) schemes as time goes on. Therefore, this paper studies how to realize cost-effective VNT reconfiguration (VNT-Recfg) in an HOE-DCN, such that its IT resource utilization can be balanced well. The problem has its unique difficulty because the topology of the substrate network (*i.e.*, the HOE-DCN) can change in the VNT-Recfg due to the "1-to-1" connectivity of an OXC. We design the procedure of our VNT-Recfg to include two major algorithms, *i.e.*, a selection algorithm to choose virtual machines (VMs) to migrate, and a reconfiguration algorithm to determine the reconfiguration schemes of VMs, virtual links (VLs), and the OXC. The selection algorithm is developed to maintain the IT balance degree, while the reconfiguration algorithm consists of two steps, 1) getting the VM migration schemes to re-balance the IT resource usage, and 2) calculating the reconfiguration schemes of related VLs and the OXC. For the first step, we formulate a mixed integer linear programming (MILP) model and design a time-efficient heuristic. We solve the second step with a novel algorithm that leverages both dynamic programming and a lightweight ILP.**

***Index Terms*—Hybrid optical/electrical datacenter network, Virtual network reconfiguration, VM migration.**

I. Introduction

Recently, the rapid development of cloud computing and related network services has made the traffic in datacenters (DCs) increase with an annual rate of 25% [1]. Therefore, DC networks (DCNs) are facing great challenges to handle enormous traffic cost-efficiently [2, 3]. A promising way to improve the scalability of DCNs is to integrate optical circuit switching (OCS) with electrical packet switching (EPS) and make inter-rack connections hybrid [4]. This is because compared with its electrical counterpart, OCS normally provides more bandwidth capacity with much less power consumption [5–8]. To this end, the hybrid optical/electrical DCNs (HOE-DCNs) that use both electrical Ethernet switches and optical cross-connects (OXCs) to interconnect top-of-rack (ToR) switches have attracted intensive research interests [4, 9].

Fig. 1(a) shows the architecture of an HOE-DCN. Specifically, in addition to the hieratical inter-rack network based on electrical packet switching (EPS), the HOE-DCN also includes an OXC to interconnect all the ToR switches. Hence, inter-rack traffic has one more routing option. Specifically, transient and dynamic flows should still be routed over the EPS-based inter-rack network, while bandwidth-intensive and long-lasting flows can take the advantages of OCS [10, 11].

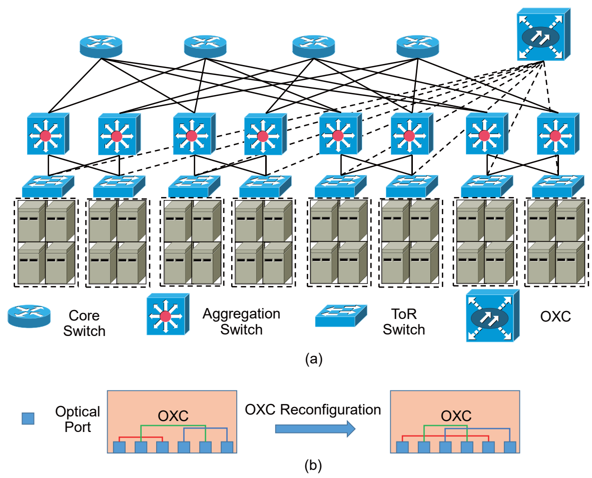

Fig. 1. (a) Architecture of HOE-DCN, and (b) Operation principle of OXC.

Most of the network services running in a DCN involve multiple virtual machines (VMs). For instance, a Hadoop application usually runs its jobs over a cluster that consists of one name-node and multiple data-nodes [10]. As the VMs belong to the same network service and communicate with each other to execute their jobs, they essentially form a virtual network (VNT). Therefore, the management of network services in a DCN can leverage the idea of virtual network embedding (VNE) [12, 13], where each network service corresponds to a VNT and the DCN itself is the substrate network (SNT). Meanwhile, both the resource utilizations of network services and the instances of network services are highly dynamic [14]. This degrades the optimality of VNE results as time goes on, and thus it is necessary to consider VNT reconfiguration [15].

Although VNT reconfiguration has already been studied with generic networks [15] and DCNs [16] as SNTs, it, to the best of our knowledge, has not been considered in HOE-DCNs. Note that, the VNT reconfiguration in HOE-DCNs is intrinsically more complex than that in conventional DCNs. This is because the VNT reconfiguration in an HOE-DCN might also change the SNT's topology, which will not happen in a conventional DCN. Specifically, as explained in Fig. 1(b), an OXC only supports "1-to-1" connectivity, and thus if VNT reconfiguration needs to remap a virtual link (VL) to connect

978-3-903176-21-8 © 2020 IFIP

different ToR switches through the OXC, the OXC needs to be reconfigured and so does the SNT's topology.

In this work, we study how to realize cost-effective VNT reconfiguration in an HOE-DCN, such that the IT resource utilization in the HOE-DCN can be balanced to avoid "hot-spots". We first explain the overall procedure of the VNT reconfiguration and how to select the VMs to migrate. Then, we try to get the new VNE schemes of the VNTs (*i.e.*, network services) that have VMs to migrate. This problem can be solved in two steps, 1) determining the VM migration schemes to re-balance the IT resource usage, and 2) calculating the reconfiguration schemes of related VLs and the OXC. For the first step, we formulate a mixed integer linear programming (MILP) model and design a time-efficient heuristic. The sub-problem of the second step is formulated based on the output from the first one, and we solve it by introducing a novel algorithm that leverages both dynamic programming and a lightweight integer linear programming (ILP) model. Finally, we conduct simulations to evaluate our proposal.

The rest of the paper is organized as follows. Section II gives the problem description. Our algorithm design is discussed in Section III. In Section IV, we conduct simulations to evaluate our algorithms. Finally, Section V summarizes the paper.

II. Problem Description

We model the HOE-DCN's topology as $G(V_s, E_s)$, where V_s and E_s are the sets of racks and network links, respectively. Each rack $v_s \in V_s$ consists of a ToR switch that simultaneously connects to both the EPS-based inter-rack network and the OXC, and a pool of servers whose total IT and I/O capacities are C_{v_s} and B_{v_s}, respectively. During operation, the IT and I/O resource usages on rack v_s are c_{v_s} and b_{v_s}, respectively. A link $e_s \in E_s$ can be either an Ethernet link for EPS or an optical one for OCS (*i.e.*, to/from the OXC). Note that, the actual configuration of the OXC determines whether there is an optical link in between two racks, while at any given time, each rack can only talk with one other rack through OXC.

The topology of a network service's VNT is denoted as $G_r(V_r, E_r)$, where V_r is the VM set and E_r is the set of VLs that interconnect the VMs. Here, each VM $v_r \in V_r$ demands for c_{v_r} IT resources, and each VL $(v_r, u_r) \in E_r$ has a bandwidth requirement of $b_{(v_r,u_r)}$. In VNT reconfiguration, one might need to reconfigure both VMs and VLs. A VM can be migrated to any rack whose IT resources are sufficient to carry it, while the remapping of VLs is relatively complex due to the "1-to-1" connectivity of the OXC. Hence, we divide the VLs to reconfigure into two types, *i.e.*, VLs with and without VM migration. "VLs with VM migration" means that they need to be reconfigured because their end VMs have been migrated to different racks, while "VLs without VM migration" refer to the VLs that need to be reconfigured purely due to an upcoming OXC reconfiguration. For example, Fig. 2 gives an example on VLs without VM migration. Before the VNT reconfiguration, *ToR Switches* 1 and 3 can communicate with each other with an optical link through the OXC, and thus the VL to connect *VMs* a and b is mapped on the optical link. The OXC reconfiguration makes *ToR Switches* 1 and 2 be connected by and optical link, and thus *ToR Switches* 1 and 3 cannot talk through the OXC anymore. Therefore, *VL* a-b has to be reconfigured to use the EPS-based inter-rack network.

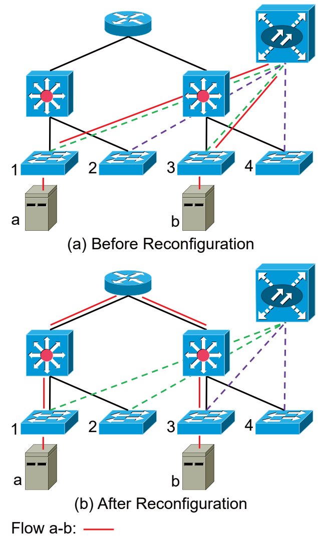

Fig. 2. Example on VLs without VM migration.

In this work, we assume that the EPS-based inter-rack network uses the fat-tree topology as shown in Fig. 1(a). It is known that a fat-tree topology is non-blocking, which means that the bandwidth capacity between any two racks is sufficient to carry all the flows between the servers in the two racks, as long as the links from the servers to their ToR switches are not congested. We define the I/O resource requirement of a VM v_r as b_{v_r}, which is the summation of the bandwidth demands of all the VLs that connect to it. Meanwhile, we assume that each VL can be either optical-preferred or do-not-care, and this parameter is pre-determined by its service provider based on the traffic condition on it. Therefore, we define the primary objective of the VNT reconfiguration as to balance the IT resource utilization in the HOE-DCN, while there is also a secondary objective, *i.e.*, to maximize the number of optical-preferred VLs that are mapped onto optical links.

III. Algorithm Design

A. Overall Procedure

Algorithm 1 provides the overall procedure of VNT Reconfiguration (VNT-Recfg) in an HOE-DCN. We first choose the VMs that should be migrated to re-balance the IT resource utilization in the HOE-DCN with a selection algorithm (*Lines* 2-3). Then, we leverage a reconfiguration algorithm to get the reconfiguration schemes of selected VMs, related VLs and the OXC (*Line* 4). In *Line* 5, we migrate the selected VMs to their new racks and remaps all the affected VLs accordingly.

B. Selection Algorithm

The average IT resource utilization in the HOE-DCN can be defined as

$$\bar{c} = \frac{1}{|V_s|} \cdot \sum_{v_s \in V_s} \frac{c_{v_s}}{C_{v_s}}, \tag{1}$$

Algorithm 1: Overall Procedure of VNT-Recfg

while *the SNT is operational* **do**
  choose VMs to migrate with a selection algorithm;
  store the selected VMs in set V_R^s;
  obtain new mapping schemes of VMs in V_R^s and OXC reconfiguration with a reconfiguration algorithm;
  migrate VMs in V_R^s accordingly and remap all the affected VLs;
  wait for the next reconfiguration time;
end

where $|V_s|$ refers to the total number of racks in the HOE-DCN, and c_{v_s} denotes the IT resource usage on rack $v_s \in V_s$. We assume that all the racks have the same IT capacity.

Algorithm 2 describes our two-step proposal to select most "critical" VMs to migrate for re-balancing IT resource utilization in the HOE-DCN. *Lines* 1-2 are for the initialization. The first for-loop is the initial phase to select VMs to migrate (*Lines* 3-16). In *Line* 4, we check all the racks whose IT resource usages are lager than the average utilization $\bar{c}$. *Line* 5 initializes several temporary variables, where $V_r^{v_s}$ will store the VMs that can be moved away from rack v_s, and *Line* 6 stores the VMs embedded on rack v_s in set V_r^c. Each iteration of the while-loop covering *Lines* 7-14 tries to select a VM to migrate such that the resulting IT resource usage on rack v_s becomes closer to the average utilization $\bar{c}$. Note that, n is the counter to record the number of VMs selected in the initial phase (*Line* 12). The second for-loop is the final phase for VM selection (*Lines* 17-22). Here, we introduce a selection ratio $\gamma \in (0, 1]$ to determine the number of VMs that will be chosen for migration, to control the complexity of VNT-Recfg. In *Line* 18, we find the rack v_s^* whose IT resource usage is the maximum. *Lines* 19-21 select the first VM in $V_r^{v_s^*}$ to insert in V_R^s, and update the related temporary variables accordingly. The time complexity of *Algorithm* 2 is $O(|V_s| \cdot |V_R|^2)$, where V_R is the set of VMs in all the active VNTs.

C. Reconfiguration Algorithm

After VM selection, we use a reconfiguration algorithm to get new mapping schemes of the selected VMs and reconfiguration scheme of the OXC. Hence, the reconfiguration algorithm operates in two steps, 1) obtaining the VM migration schemes to re-balance IT resource usage, and 2) getting the reconfiguration schemes of related VLs and the OXC.

1) Determining VM Migration Schemes: This problem can be solved exactly with an MILP model, and Table I lists its parameters and variables.

Objective:

The objective of VM migration is to re-balance the IT resource usage in the HOE-DCN, where the balance degree of IT resource can be represented as

$$\tilde{c} = c_{\max} - c_{\min}. \tag{2}$$

Therefore, the MILP's optimization objective should be

$$\text{Minimize} \quad \tilde{c}. \tag{3}$$

Algorithm 2: Select VMs to Migrate

1 $n = 0$, $V_R^s = \emptyset$;
2 calculate average IT resource usage $\bar{c}$ with Eq. (1);
3 **for** *each rack* $v_s \in V_s$ **do**
4   **if** $\frac{c_{v_s}}{C_{v_s}} > \bar{c}$ **then**
5     $tc_{v_s} = c_{v_s}$, $c = c_{v_s}$, $V_r^{v_s} = \emptyset$, $V_r^c = \emptyset$;
6   store the VMs embedded on v_s in set V_r^c;
7   **while** $c > \bar{c}$ **do**
8       $v_r^s = \underset{v_r \in V_r^c}{\text{argmin}} \left[\left| \frac{(c - c_{v_r})}{C_{v_s}} - \bar{c} \right| \right]$;
9       $c = c - c_{v_r^s}$;
10   **if** $(c \geq \bar{c})$ *OR* $\left(\left|\frac{(c + c_{v_r})}{C_{v_s}} - \bar{c}\right| > \left|\frac{c}{C_{v_s}} - \bar{c}\right|\right)$ **then**
11         $V_r^{v_s} = V_r^{v_s} \cup \{v_r^s\}$, $V_r^c = V_r^c \setminus v_r^s$;
12         $n = n + 1$;
13   **end**
14   **end**
15   **end**
16 **end**
17 **for** $i = 1$ *to* $\lceil \gamma \cdot n \rceil$ **do**
18   $v_s^* = \underset{v_s \in V_s}{\text{argmax}}(tc_{v_s})$;
19   select the first VM v_r in $V_r^{v_s^*}$;
20   $V_R^s = V_R^s \cup \{v_r\}$, $V_r^{v_s^*} = V_r^{v_s^*} \setminus v_r$;
21   $tc_{v_s^*} = tc_{v_s^*} - c_{v_r}$;
22 **end**
23 **return** V_R^s;

TABLE I
NOTATIONS OF THE MILP FOR DETERMINING VM MIGRATION SCHEMES

Notations	Definitions
Parameters:	
C_{v_s}	total IT capacity of servers in rack $v_s \in V_s$.
B_{v_s}	total I/O capacity of servers in rack $v_s \in V_s$.
c_{v_s}	IT usage in rack $v_s \in V_s$ before VNT-Recfg.
b_{v_s}	I/O usage in rack $v_s \in V_s$ before VNT-Recfg.
V_R^s	set of VMs that are selected for migration.
c_{v_r}	IT usage of VM $v_r \in V_R^s$.
b_{v_r}	I/O usage of VM $v_r \in V_R^s$.
$\delta_{v_s}^{v_r}$	boolean parameter that equals 1 if VM v_r is embedded on rack v_s before VNT-Recfg, and 0 otherwise.
Variables :	
$\widetilde{c}_{v_s}$	IT usage in rack $v_s \in V_s$ after VNT-Recfg.
$\widetilde{\delta}_{v_s}^{v_r}$	boolean variable that equals 1 if VM v_r gets embedded on rack v_s after VNT-Recfg, and 0 otherwise.
$c_{\max}$	the maximum IT usage on a rack after VNT-Recfg.
$c_{\min}$	the minimum IT usage on a rack after VNT-Recfg.

Constraints:

$$\sum_{v_s \in V_s} \widetilde{\delta}_{v_s}^{v_r} = 1, \quad \forall v_r \in V_R^s. \tag{4}$$

Eq. (4) ensures that each VM gets mapped onto one and only one rack after migration.

$$c_{v_s} + \sum_{v_r \in V_R^s} c_{v_r} \cdot (\widetilde{\delta}_{v_s}^{v_r} - \delta_{v_s}^{v_r}) \leq C_{v_s}, \ \forall v_s \in V_s. \tag{5}$$

Eq. (5) ensures that the IT resource usage on each rack does not exceed its IT capacity after VM migration.

$$b_{v_s} + \sum_{v_r \in V_R^s} b_{v_r} \cdot (\widetilde{\delta}_{v_s}^{v_r} - \delta_{v_s}^{v_r}) \leq B_{v_s}, \ \forall v_s \in V_s. \tag{6}$$

Eq. (6) ensures that the I/O resource usage on each rack does not exceed its I/O capacity after VM migration. Note that, the I/O usage of a VM is the summation of the bandwidth demands of all the VLs that connect to it.

$$\widetilde{c}_{v_s} = c_{v_s} + \sum_{v_r \in V_R^s} c_{v_r} \cdot (\widetilde{\delta}_{v_s}^{v_r} - \delta_{v_s}^{v_r}),\ \forall v_s \in V_s. \tag{7}$$

Eq. (7) calculates the IT resource usage on each rack after VM migration.

$$\begin{aligned} c_{\max} &\geq \frac{\widetilde{c}_{v_s}}{C_{v_s}},\ \forall v_s \in V_s, \\ c_{\min} &\leq \frac{\widetilde{c}_{v_s}}{C_{v_s}},\ \forall v_s \in V_s. \end{aligned} \tag{8}$$

Eq. (8) ensures that we get correct values of $c_{\max}$ and $c_{\min}$.

As the MILP model above can become intractable for large-scale problems, we also design a heuristic, namely, minimum-first VM migration (MF-VMM), to determine the VM migration schemes. *Algorithm* 3 shows the detailed procedure of MF-VMM. *Lines* 1-4 are for the initialization. The while-loop covering *Lines* 5-19 tries to determine the remapping scheme of a selected VM in each iteration. Specifically, we first initialize $v_r^{\min}$ and lb, and find the rack v_s^* with the minimum IT resource usage (*Line* 6). Then, we select the VM that provides the minimum value of $|\frac{\tilde{c}_{v_s^*}+c_{v_r}}{C_{v_s}} - \bar{c}|$, and update $v_r^{\min}$ and lb accordingly (*Lines* 7-11). Next, we check whether $v_r^{\min} \neq 0$. If yes, we determine the remapping scheme of $v_r^{\min}$ and update the related IT and I/O resource usages accordingly (*Lines* 13-15). Otherwise, we remove v_s from $\tilde{V}_s$ since this rack cannot carry more VMs. The time complexity of *Algorithm* 3 is $O(|V_R^s| \cdot (|V_s| + |V_R^s|^2))$.

Algorithm 3: Minimum-First VM Migration

1 $\tilde{V}_s = V_s$;
2 **for** *each* $v_s \in \tilde{V}_s$ **do**
3   calculate IT usage $\tilde{c}_{v_s}$ and I/O usage $\tilde{b}_{v_s}$ on rack v_s when the VMs in V_R^s have been migrated;
4 **end**
5 **while** $V_R^s \neq \emptyset$ **do**
6     $v_r^{\min} = 0$, $lb = +\infty$, $v_s^* = \underset{v_s \in \tilde{V}_s}{\operatorname{argmin}}(\frac{\tilde{c}_{v_s}}{C_{v_s}})$;
7   **for** *each VM* $v_r \in V_R^s$ **do**
8   **if** *(*$|\frac{\tilde{c}_{v_s^*}+c_{v_r}}{C_{v_s}} - \bar{c}| < lb$*) AND (*$\tilde{b}_{v_s^*} + b_{v_r} \leq B_{v_s^*}$*)* **then**
9             $v_r^{\min} = v_r$, $lb = |\frac{\tilde{c}_{v_s^*}+c_{v_r}}{C_{v_s}} - \bar{c}|$;
10   **end**
11   **end**
12   **if** $v_r^{\min} \neq 0$ **then**
13   decide to remap VM $v_r^{\min}$ onto rack v_s;
14         $\tilde{c}_{v_s^*} = \tilde{c}_{v_s^*} + c_{v_r^{\min}}$, $\tilde{b}_{v_s^*} = \tilde{b}_{v_s^*} + b_{v_r^{\min}}$;
15   remove $v_r^{\min}$ from V_R^s;
16   **else**
17   remove v_s from $\tilde{V}_s$;
18   **end**
19 **end**

2) Determining Reconfiguration Schemes of OXC and VLs: The secondary objective of VNT-Recfg is to maximize the number of optical-preferred VLs that are mapped onto optical links, and this objective is fulfilled in this step. The key issue that needs to be resolved in this step is to determine the reconfiguration scheme of the OXC. Note that, reconfiguring too many ports in the OXC would induce excessive operation complexity, and thus we introduce a preset threshold α to limit the number of reconfigured ports in the OXC.

Algorithm 4 shows the overall procedure to determine the reconfiguration schemes of OXC and VLs. Here, we introduce a set R_s to store all the rack pairs in the HOE-DCN. After determining the VM migration schemes in the previous subsection, we know all the rack pairs that will have inter-rack VLs in between after the VNT-Recfg. In *Lines* 1-3, for each rack pair u_s-v_s, we assume that the two racks have an optical link through the OXC, calculate the maximum number of optical-preferred VLs (n_{u_s,v_s}) that the optical link connecting u_s and v_s can carry, and get the mapping schemes for the VLs. This problem can be solved with dynamic programming, and we will explain the details in *Algorithm* 5. Next, based on the results of the dynamic programming, we use an ILP model to get the OXC reconfiguration scheme that maximizes the number of optical-preferred VLs that are mapped onto optical links (*Line* 4). The ILP model will also be elaborated on in the following. Finally, we determine the mapping schemes of all the inter-rack VLs, based on the ILP's results.

Algorithm 4: Obtain Reconfiguration Schemes of OXC and VLs

1 **for** *each* $(u_s, v_s) \in R_s$ **do**
2   use dynamic programming to get the maximum number of optical-preferred VLs (n_{u_s,v_s}) that the optical link for $u_s \leftrightarrow v_s$ can carry, and obtain the mapping schemes;
3 **end**
4 use a lightweight ILP to get the OXC reconfiguration scheme that maximizes the number of optical-preferred VLs that are mapped onto optical links;
5 determine the mapping schemes of all inter-rack VLs;

Algorithm 5 explains the dynamic programming for solving the problem described in *Line* 2 of *Algorithm* 4, which is essentially a one-dimensional knapsack problem. We first define several parameters. $E_{(u_s,v_s)}$ is the set of optical-preferred VLs that are currently embedded on the rack pair u_s-v_s. $B_{(u_s,v_s)}$ is the bandwidth capacity of the optical link between racks u_s and v_s. $b_{(u_r,v_r)}$ is the bandwidth demand of a VL $(u_r, v_r) \in E_{(u_s,v_s)}$. The dynamic programming gets n_{u_s,v_s} and the corresponding VL mapping schemes $\{x_{(u_r,v_r)},\ (u_r, v_r) \in E_{(u_s,v_s)}\}$. Here, $x_{(u_r,v_r)} = 1$ means that the optical-preferred VL (u_r, v_r) will be embedded on an optical link through the OXC. The time complexity of *Algorithm* 5 is $O\left(|E_{(u_s,v_s)}| \cdot B_{(u_s,v_s)}\right)$.

Next, we introduce the ILP model for solving the problem in *Line* 4 of *Algorithm* 4. Table II lists the notations.

Objective:

Algorithm 5: Dynamic Programming

Input: $E_{(u_s,v_s)}$, $B_{(u_s,v_s)}$, $b_{(u_r,v_r)}$.
Output: $x_{(u_r,v_r)}$, n_{u_s,v_s}.
temporarily assign index of each VL $(u_r, v_r) \in E_{(u_s,v_s)}$ as $i \in [1, |E_{(u_s,v_s)}|]$;
set $\mathbf{M} = \{m_{i,j},\ i \in [1, |E_{(u_s,v_s)}|],\ j \in [1, B_{(u_s,v_s)}]\}$ as a zero matrix;
for *each* $i \in [1, |E_{(u_s,v_s)}|]$ **do**
  for *each* $j \in [1, B_{(u_s,v_s)}]$ **do**
    if $j < b_i$ **then**
      $m_{i,j} = m_{i-1,j}$;
    else if $j = b_i$ **then**
      $m_{i,j} = \max(m_{i-1,j},\ 1)$;
    else
      $m_{i,j} = \max(m_{i-1,j},\ m_{i-1,j-b_i} + 1)$;
    end
  end
end
for *each* $k \in [0, |E_{(u_s,v_s)}| - 2]$ **do**
  $i = |E_{(u_s,v_s)}| - k$;
  if $m_{i,B_{(u_s,v_s)}} > m_{i-1,B_{(u_s,v_s)}}$ **then**
    $x_i = 1$, $B_{(u_s,v_s)} = B_{(u_s,v_s)} - b_i$;
    if $B_{(u_s,v_s)} = 0$ **then**
      break;
    end
  end
end
$n_{u_s,v_s} = \sum_{i=1}^{|E_{(u_s,v_s)}|} x_i$;
restore indices of VLs in $E_{(u_s,v_s)}$ and get $\{x_{(u_r,v_r)}\}$ based on $\{x_i,\ i \in [1, |E_{(u_s,v_s)}|]\}$ accordingly;

TABLE II
NOTATIONS OF THE ILP FOR DETERMINING OXC RECONFIGURATION SCHEME

Notations	Definitions
Parameters:	
R_s	set of rack pairs in the HOE-DCN.
$f_{(u_s,v_s)}$	boolean parameter that equals 1 if racks u_s and v_s have an optical link before VNT-Recfg, and 0 otherwise.
n_{u_s,v_s}	the number of optical-preferred VLs on the optical link in between racks u_s and v_s.
α	preset threshold on number of reconfigured ports in the OXC.
Variables:	
$\widetilde{f}_{(u_s,v_s)}$	boolean variable that equals 1 if racks u_s and v_s have an optical link after VNT-Recfg, and 0 otherwise.

The objective is to maximize the number of optical-preferred VLs that are mapped onto optical links.

$$\text{Maximize} \quad \frac{1}{2} \sum_{(u_s,v_s)\in R_s} n_{u_s,v_s} \cdot \widetilde{f}_{(u_s,v_s)}. \tag{9}$$

Constraints:

$$\sum_{\{v_s:(u_s,v_s)\in R_s\}} \widetilde{f}_{(u_s,v_s)} = 1,\ \forall u_s \in V_s, \tag{10}$$

$$\widetilde{f}_{(u_s,v_s)} = \widetilde{f}_{(v_s,u_s)},\ \forall (u_s,v_s) \in R_s, \tag{11}$$

Eqs. (10) and (11) ensure that a rack can only talk with one other rack through the OXC.

$$|V_s| - \sum_{(u_s,v_s)\in R_s} \widetilde{f}_{(u_s,v_s)} \cdot f_{(u_s,v_s)} \leq \alpha. \tag{12}$$

Eq. (12) ensures that the number of reconfigured ports in the OXC cannot exceed the preset threshold α.

It can be see that this ILP model is lightweight because the number of variables and constraints are very limited. We will verify its time-efficiency further in the simulations.

IV. PERFORMANCE EVALUATION

In this section, we perform simulations to evaluate the performance of our algorithms. Note that, we only design a heuristic for VM migration, while for getting the reconfiguration schemes of OXC and VLs, we get exact solutions directly.

As mentioned before, the EPS-based inter-rack network in the HOE-DCN uses the k-ary fat-tree topology. Specifically, the aggregation and ToR switches are grouped into k pods, each of which consists of $\frac{k}{2}$ ToR switches. Hence, there are $\frac{k^2}{2}$ ToR switches in the HOE-DCN, and each ToR switch has $\frac{k}{2}$ Ethernet ports connecting to aggregation switches and one optical port connecting to the OXC. The simulations consider 4-ray and 6-ray fat-tree topologies, *i.e.*, having 8 and 18 racks, respectively. We set the capacity of each Ethernet port on a ToR switch as 1000 units, while the capacity of its optical port is $10,000$ units. The total IT resource capacity of each rack in 4-ray and 6-ray fat-trees are 2000 and 3000 units, respectively.

For the network services, their VNTs are dynamically generated with random topologies and follow the Poisson model. For each VNT, the number of VMs distributes within $[2, 8]$ and $[2, 16]$ in 4-ray and 6-ray fat-trees, respectively, and the connectivity ratio of the VMs is 0.5. The IT resource demand of each VM is randomly selected within $[50, 200]$ units, and the bandwidth requirement of each VL distributes within $[10, 40]$ units. The ratio of optical-preferred VLs is chosen within $\{0.25, 0.50, 0.75\}$. The simulations first use an existing VNE algorithm (*e.g.*, the one in [17]) to embed the dynamic VNTs in the SNT, and they will pause when unbalanced IT utilization happens. Here, we define the unbalanced IT utilization as that $\tilde{c}$ from Eq. (2) is larger than 0.5. Then, we use our algorithms to calculate VNT reconfiguration schemes. For the selection algorithm, we change the selection ratio γ from 0.5 to 1.0 to study the tradeoff between IT balance degree and operation complexity. Due to its time complexity, we only solve the MILP for the 4-ray fat-tree.

Fig. 3 shows the results for the 4-ray fat-tree. For the primary objective of VNT-Recfg, Fig. 3(a) indicates that the performance of the MILP- and MF-VMM-based approaches does not have noticeable difference when the selection ratio is less than 0.7. However, when γ keeps increasing, the MILP-based approach outperforms the MF-VMM-based one in terms of IT balance degree (*i.e.*, $\tilde{c}$). The results about the secondary objective, *i.e.*, the number of optical-preferred VLs that are mapped onto optical links, are plotted in Fig. 3(b). Here, a "successful embedding" means that one optical-preferred VL gets embedded on an optical link after VNT-Recfg. The simulations set the threshold on the number of reconfigured ports in the OXC as $\alpha \in \{0, 4, 8\}$. We can see that the MILP- and MF-VMM-based approaches also perform similarly in terms of the secondary objective. Moreover, since the MILP does not consider the secondary objective explicitly, the MF-VMM-based approach can perform better in certain cases.

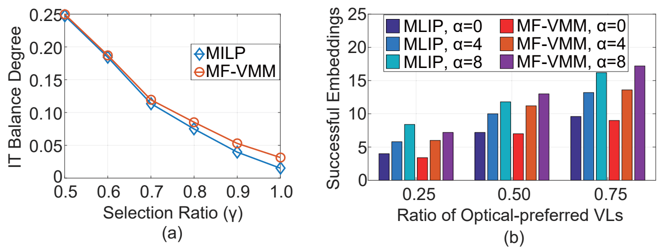

Fig. 3. Simulation results for 4-ray fat-tree, (a) IT Balance degree, and (b) Number of optical-preferred VLs mapped onto optical links.

The average running time of the algorithms is listed in Table III. Both of them take more time as γ increases, because a lager γ means that more VMs will be migrated. Meanwhile, the MF-VMM-based approach is much more time-efficient.

TABLE III
RUNNING TIME OF ALGORITHMS (SECONDS)

γ	0.5	0.7	0.8	0.9	1.0
MILP	0.19	0.64	1.77	16.38	437.44
MF-VMM	1.09e-4	1.44e-4	1.60e-4	1.86e-4	2.22e-4

Fig. 4 illustrates the results for the 6-ray fat-tree. Due to the time complexity of MILP, we only the MF-VMM-based approach. In Fig. 4(a), we can still see that VNT-Recfg maintains the IT balance degree well. The trends for the secondary objective in Fig. 4(b) also stay the same.

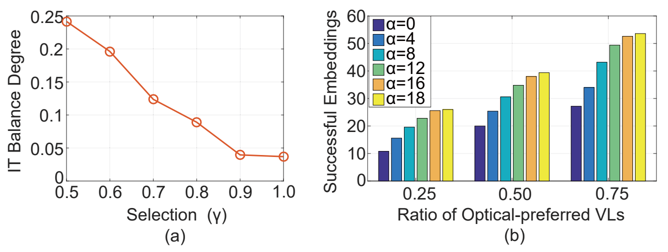

Fig. 4. Simulation results for 6-ray fat-tree, (a) IT Balance degree, and (b) Number of optical-preferred VLs mapped onto optical links.

V. CONCLUSION

In this work, we comprehensively studied how to realize cost-effective VNT-Recfg in an HOE-DCN. The procedure of our VNT-Recfg includes two major algorithms, *i.e.*, a selection algorithm to choose VMs to migrate, and a reconfiguration algorithm to determine the reconfiguration schemes of VMs, VLs, and the OXC. The selection algorithm was designed to maintain the IT balance degree, while the reconfiguration algorithm consists of two steps, 1) getting the VM migration schemes, and 2) calculating the reconfiguration schemes of related VLs and the OXC. For the first step, we formulated an MILP model and designed a time-efficient heuristic, namely, MF-VMM. We tackled the second step with a novel algorithm that includes dynamic programming and a lightweight ILP. Simulation results showed that both MILP- and MF-VMM-based approaches maintain IT balance degree well and embed optical-preferred VLs on optical links through the OXC, and MF-VMM-based approach is much more time-efficient.

ACKNOWLEDGMENTS

This work was supported in part by the NSFC projects 61871357, CAS Key Project (QYZDY-SSW-JSC003), and SPR Program of CAS (XDC02070300).

REFERENCES

[1] 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

[2] 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.

[3] W. Lu *et al.*, "AI-assisted knowledge-defined network orchestration for energy-efficient datacenter networks," *IEEE Commun. Mag., in Press*, 2019.

[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] Z. Zhu *et al.*, "RF photonics signal processing in subcarrier multiplexed optical-label switching communication systems," *J. Lightw. Technol.*, vol. 21, pp. 3155–3166, Dec. 2003.

[6] 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.

[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] 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.

[9] Z. Zhu, W. Lu, L. Liang, and B. Kong, "Predictive analytics in hybrid optical/electrical DC networks," in *Proc. of OFC 2019*, pp. 1–3, Mar. 2019.

[10] 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.

[11] 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.

[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] 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.

[14] 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.

[15] 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.

[16] J. Duan and Y. Yang, "A load balancing and multi-tenancy oriented data center virtualization framework," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, pp. 2131–2144, Aug. 2017.

[17] 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.