Scalable Data Center Network Architecture With Distributed Placement of Optical Switches and Racks

Jie Xiao, Bin Wu, Xiaohong Jiang, Achille Pattavina, Hong Wen, and Lei Zhang

I. Introduction

The fast evolution of networking technology is pushing networks to be service-oriented. Most of the emerging services are provided by data center networks (DCNs) [1–3], and a network of DCNs is called a cloud [4–6]. Traditionally, a DCN is a factory-scale and massively parallel computing and storage resource. It consists of a huge number of servers organized in racks, which work in parallel with data exchanged by a switch network such as a Clos [7–9] or fat-tree network [10–12]. For example, a web search request may access an inverted index spreading across thousands of servers, among which petabytes of data are interactively processed and switched [13].

The scale of DCNs is expanding rapidly with the quickly increasing number of servers [14]. According to [15], Google had more than 450,000 servers in 30 data centers by 2006, which increased to 36 by 2010. Microsoft and Yahoo! are in similar situations, with the total number of DCN servers nearly doubling every 14 months, which exceeds Moore's law [16].

The rapid expansion of DCNs has led to great concern about system scalability, which is constrained by many environmental factors such as site choice, system cooling, and power supply capacity, as well as energy efficiency and carbon dioxide footprint. Figure 1 shows two examples of Google DCNs. In Fig. 1(a), an area with the size of multiple football fields is needed. In Fig. 1(b), a large lake next to the DCN serves as water-flow system cooling. Just in the United States, DCNs consume more than 2% of the national electricity at present, which is likely to double every five years [17]. The ever-increasing wattage translates to the commonly voiced complaints about power outage and cooling capacities [18].

In addition to the environmental factors, scalability is constrained by the internal architecture of DCNs as well. Existing DCNs adopt electronic switches and routers with multiple stages of O/E and E/O transceivers, leading to bandwidth bottleneck and high power consumption, as well as complex interconnects and bulky systems due to the limited modular capacity and thus a large number of switches and ports. According to a study from IBM [19], power consumption can be greatly reduced simply by replacing copper links using optical interconnects. Moreover, it is shown that over 70% of power can be saved in DCNs if electronic switches are further replaced by optical ones [20]. The high switching capacity of optical switches also leads

Manuscript received August 27, 2013; revised December 3, 2013; accepted December 20, 2013; published February 20, 2014 (Doc. ID 196154).

J. Xiao, B. Wu, and L. Zhang (e-mail: lzhang@tju.edu.cn) are with the School of Computer Science and Technology, Tianjin University, Tianjin 300072, China.

J. Xiao and H. Wen are with the National Key Laboratory on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China.

X. Jiang is with the School of Systems Information Science, Future University Hakodate, Hakodate, Japan.

A. Pattavina is with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy.

(a) (b)

Fig. 1. Examples of Google DCNs. (a) Google DCN in Belgium and (b) lakeside site in Dalles, Oregon.

to a reduced number of switches in a more compact system with much enlarged bandwidth.

To solve the scalability issue, DCN architecture and energy efficiency are widely studied [18–22]. A complete survey on those topics can be found in [22]. In particular, [19] adopts optical interconnects in DCNs, and [21] proposes a hierarchical energy optimization by intelligently shutting off some switches and links. Also, several all-optical or hybrid electrical–optical switches [23–29] are demonstrated for future high-performance DCNs, where state-of-the-art traffic scheduling and routing algorithms [30–33] can be used to enhance the switching performance.

Nevertheless, DCNs cannot be made fully scalable by merely adopting optical switches and improving energy efficiency in the current architecture. To support future sustainable DCN expansion, it is necessary to invent a fully scalable and flexible architecture that can overcome both environmental and internal constraints, as pointed out above. With the fast growth of cloud services, this has become an urgent need.

In this paper, we adopt optical switches in DCNs and cluster server racks and switches into multiple sets called *component sets*. By assuming that the DCN scale is sufficiently large and thus cannot be supported by current architectures, we propose a fully scalable architecture with distributed placement of component sets in a given optical network. This completely removes the environmental constraints, as power supply, system cooling, and warehouse accommodation can be handled in a distributed and less bulky manner. Meanwhile, DCN internal interconnects are provided by wavelengths in the optical network. Together with the use of optical switches, this saves energy and leads to a compact and fully scalable DCN.

Challenges also exist. Distributed placement of component sets introduces additional transmission delay and cost to DCN internal traffic. To this end, we define a *cost scaling factor* θ by translating the additional delay into a cost factor that can be integrated with the transmission cost (see Subsection III.A). θ can be flexibly manipulated to control the extent of component set distribution across the network. In the extreme case of an extra-large θ, the proposed architecture degenerates into the current centralized DCN. Hence, it includes the centralized DCN as a special case, while being fully scalable.

While distributed placement increases the delay and cost of internal traffic, it reduces that of external traffic (for service requests and deliveries) with service interfaces that are closer to clients. Therefore, component set placement is important for balancing internal and external transmission costs of the DCN. Generally, a component set (placed at a single node) should consist of those racks with heavy inter-rack traffic, leaving lightly loaded internal traffic for inter-node transmissions. Thus, component set clustering is also an important issue. Those factors make system cost minimization a very complex yet interesting optimization problem. To support the proposed architecture, we further study this optimization problem under predefined external demands and DCN internal traffic patterns. An integer linear program (ILP) and a heuristic are proposed to minimize the system cost while scaling the DCN.

The rest of the paper is organized as follows. Section II describes the proposed architecture and the placement problem. Section III formulates the ILP for optimally solving the problem. The heuristic is proposed in Section IV, and numerical results are presented in Section V. We conclude the paper in Section VI.

II. Architecture and Problem

A. Proposed Architecture

Our key idea is to spread the DCN component sets across a certain area of the network. Figure 2 illustrates the proposed architecture, where the components inside a dashed ellipse form a component set to be placed at a single node. Basically, the network nodes denoted by the ellipses and the inter-node fiber connections are embedded

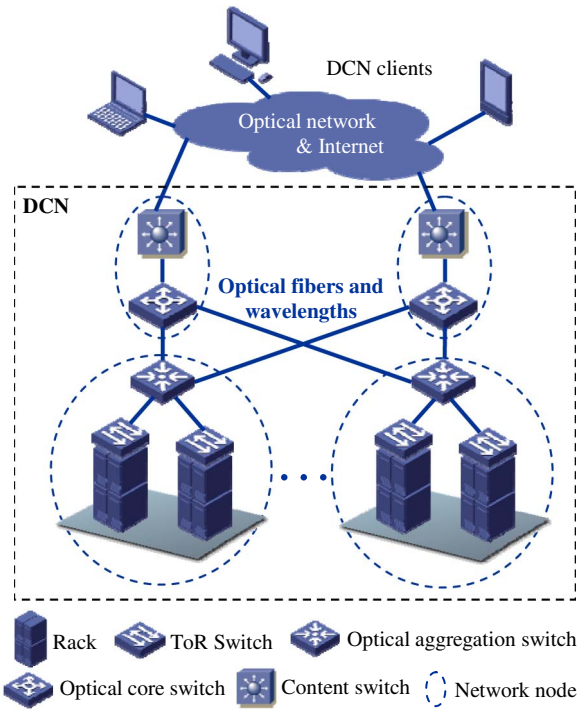


Fig. 2. Distributed placement of optical switches and racks in DCNs.

in the optical network, and the DCN in the rectangle just shows the logical functionality of the components. It is possible that the content and core switches are placed at the same node as the server racks and aggregation switches, though they appear to be separate in Fig. 2. In this distributed architecture, an optical common control channel can be reserved for management and control signaling transmissions among different component sets.

In Fig. 2, the top-of-rack (ToR) and content switches are electronic routers or OpenFlow [34] switches, whereas the aggregation and core switches are optical crossbar or packet switching fabrics. An aggregation switch is always collocated with a set of server racks at the same node and is responsible for data switching among them, and the traffic in the same rack is switched by the ToR switch. The content and core switches are always collocated. The former provides service interfaces for external requests and the latter for inter-node data switching of the DCN internal traffic. The core and aggregation switches are interconnected by optical wavelengths and fibers. Logically, they form a typical multistage switch network based on a folded-Clos (or fat-tree) structure [10–12].

Content switches divide traffic into external traffic (for service requests and deliveries) and DCN internal traffic. Each node in the network has a specific amount of service demands, which can be served by a nearby content switch to minimize the external transmission cost. Nevertheless, routing demands to different content switches provides a mechanism to balance the loads over the corresponding core switches, by slightly increasing the transmission cost of the external traffic.

B. Placement Problem

A major task of our work is to make the DCN scalable by distributing the component sets to different nodes. Accordingly, a scalability-related cost is defined as increasing with the size of the component sets. Distributed placement reduces this cost but leads to additional transmission delay and cost of the DCN internal traffic (defined as the *DCN internal overhead*). Such a trade-off is further complicated by the external demands and internal traffic patterns of the DCN, which require the switches and racks to be clustered into proper component sets and be placed at suitable nodes.

More specifically, it is desirable to find a set of nodes in proximity to each other for component set placement, such that the transmission cost and delay of the DCN internal traffic can be minimized while bulky component sets can be avoided. On the other hand, external demands at individual nodes need to be served with small transmission cost, which translates to a more distributed placement of service interfaces and component sets. For a specific DCN service, the worst-case internal traffic pattern among the racks is relatively stable. It can be estimated and planned at the network planning stage. Based on the internal traffic pattern, it is desirable to put those racks with heavy inter-rack traffic into the same component set placed at a single node, such that data switching among them can be locally handled without incurring additional inter-node transmissions. As a result, component sets tend to be large, but this contradicts the scalability. Moreover, component set placement is constrained by the network topology as well. In addition to properly clustering the racks, we need to find a suitable set of nodes in the network and match each of them with one or several component sets for placement. All the above conflicting factors (external and internal traffic, rack clustering and node mapping, and scalability) are combined, leading to a very complex yet interesting cost minimization problem.

Given a set of service demands at individual nodes and the service-oriented internal traffic pattern among the racks of a DCN, the system cost minimization problem involves the following tasks: 1) determine the number of optical switches required, 2) cluster the switches and racks into proper component sets, 3) place the component sets at a suitable set of nodes, 4) find the external service request and delivery routes, 5) figure out the switching and routing scheme for DCN internal traffic, and 6) achieve load balancing among core switches.

III. ILP Formulation

We assume that the *basic cost* of a component set at a warehouse (for system cooling, power supply, warehouse rent, etc.) is characterized by the scalability-related cost function in Fig. 3. If the number of racks in a component set exceeds a specific limit (e.g., N_k, $k \leq 4$ in Fig. 3), the basic cost will step up accordingly. For simplicity, the scalability-related cost also accounts for the costs of ToR and aggregation switches as well as optical interconnects inside the component set, since those costs are generally proportional to the number of racks. In practice, the scalability-related cost function can be properly predefined to control the size of component sets and the extent of DCN distribution, as discussed later.

A. Notation List

Inputs:

$\boldsymbol{V}$: The set of all nodes in a network $\boldsymbol{G}(\boldsymbol{V}, \boldsymbol{E})$.
$\boldsymbol{E}$: The set of all bidirectional links in a network $\boldsymbol{G}(\boldsymbol{V}, \boldsymbol{E})$.
$\boldsymbol{R}$: The set of all server racks in the DCN.

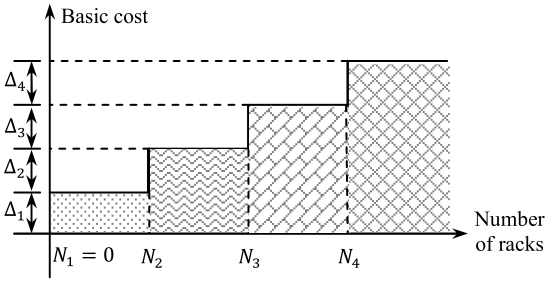


Fig. 3. Scalability-related cost function.

Q: The maximum switching capacity of a core switch.
P_s: The cost of a pair of core and content switches.
P_0: Basic cost savings if a pair of core and content switches is placed at the same node as a set of server racks.
P_{uv}: Transmission cost of per-unit traffic between nodes u and v in the optical network. In this work, it is assumed to be the distance between nodes u and v, and $P_{uv} = P_{vu}$.
θ: Cost scaling factor for taking the inter-node transmission delay of DCN internal traffic into account. $\theta > 1$ and θP_{uv} is the total cost of the DCN internal overhead for per-unit internal traffic transmission between nodes u and v, where the cost of delay accounts for $(\theta - 1)P_{uv}$.
d_v: DCN service demand at a network node $v \in \boldsymbol{V}$.
r_{ij}: The bidirectional DCN internal traffic load between two racks i and j, and $r_{ij} = r_{ji}$.
K: The number of shadowed blocks in Fig. 3.
Δ_k: Step size of the basic cost increase, as shown in Fig. 3.
N_k: Boundary of the number of racks in a component set, as shown in Fig. 3.
α: A predefined constant, and $\alpha \geq \sum_{i\in\boldsymbol{R}} \sum_{j\in\boldsymbol{R}} r_{ij}$.
β: A predefined constant, and $\beta \geq \sum_{v\in\boldsymbol{V}} d_v$.
γ: A predefined constant, and $\gamma \geq |\boldsymbol{R}|$.

Variables:

S_u: Binary variable. It takes 1 if there is a pair of core and content switches at node $u \in \boldsymbol{V}$ and 0 otherwise.
R_m^i: Binary variable. It takes 1 if rack $i \in \boldsymbol{R}$ is placed at node $m \in \boldsymbol{V}$ and 0 otherwise.
Y_u: Binary variable. It takes 1 if a pair of core and content switches is collocated with a set of server racks at node $u \in \boldsymbol{V}$ and 0 otherwise.
J_m^k: Binary variable. It takes 1 if the number of racks at node $m \in \boldsymbol{V}$ exceeds N_k $(k \leq K)$ and 0 otherwise.
X_{mn}^{ij}: Binary variable. It is defined in $\{m, n \in \boldsymbol{V} | m < n\}$ and $\{i, j \in \boldsymbol{R} | i \neq j\}$. It takes 1 if rack i is placed at node m and j at n, and 0 otherwise.
F_u^v: Nonnegative integer variable. It is the demands (counted in wavelengths) at node $v \in \boldsymbol{V}$ that are served by the content and core switches at node $u \in \boldsymbol{V}$.
T_{mn}^u: Nonnegative integer variable in $\{m, n \in \boldsymbol{V} | m < n\}$. It is the total amount of DCN internal traffic between nodes m and n (counted in wavelengths) that are switched by the core switch at node $u \in \boldsymbol{V}$.

B. ILP Formulation

$$\text{minimize} \left\{ \sum_{u\in\boldsymbol{V}} P_s S_u + \sum_{m\in\boldsymbol{V}} \sum_{k\leq K} \Delta_k J_m^k + \sum_{u\in\boldsymbol{V}} \sum_{v\in\boldsymbol{V}} P_{uv} F_u^v + \sum_{m\in\boldsymbol{V}} \sum_{n\in\boldsymbol{V}, n>m} \sum_{u\in\boldsymbol{V}} \theta (P_{mu} + P_{nu}) T_{mn}^u - \sum_{u\in\boldsymbol{V}} P_0 Y_u \right\}. \quad (1)$$

Subject to

$$\sum_{m\in\boldsymbol{V}} R_m^i = 1, \quad \forall\, i \in \boldsymbol{R}; \quad (2)$$

$$X_{mn}^{ij} \geq R_m^i + R_n^j - 1, \quad \forall\, m, n \in \boldsymbol{V}{:}m < n, \quad \forall\, i, j \in \boldsymbol{R}{:}i \neq j; \quad (3)$$

$$S_u \geq \frac{1}{\alpha} \sum_{m\in\boldsymbol{V}} \sum_{n\in\boldsymbol{V}, n>m} T_{mn}^u, \quad \forall\, u \in \boldsymbol{V}; \quad (4)$$

$$S_u \geq \frac{1}{\beta} \sum_{v\in\boldsymbol{V}} F_u^v, \quad \forall\, u \in \boldsymbol{V}; \quad (5)$$

$$J_m^k \geq \frac{1}{\gamma} \left(\sum_{i\in\boldsymbol{R}} R_m^i - N_k \right), \quad \forall\, m \in \boldsymbol{V}, \quad \forall k; \quad (6)$$

$$\sum_{u\in\boldsymbol{V}} T_{mn}^u = \sum_{i\in\boldsymbol{R}} \sum_{j\in\boldsymbol{R}, j\neq i} r_{ij} X_{mn}^{ij}, \quad \forall\, m, n \in \boldsymbol{V}{:}m < n; \quad (7)$$

$$\sum_{v\in\boldsymbol{V}} F_u^v + \sum_{m\in\boldsymbol{V}} \sum_{n\in\boldsymbol{V}, n>m} T_{mn}^u \leq Q, \quad \forall\, u \in \boldsymbol{V}; \quad (8)$$

$$\sum_{u\in\boldsymbol{V}} F_u^v = d_v, \quad \forall\, v \in \boldsymbol{V}; \quad (9)$$

$$Y_u \leq S_u, \quad \forall\, u \in \boldsymbol{V}; \quad (10)$$

$$Y_u \leq \sum_{i\in\boldsymbol{R}} R_u^i, \quad \forall\, u \in \boldsymbol{V}. \quad (11)$$

Objective (1) minimizes the system cost. The first term is the cost of all content and core switches. The second term is the sum of basic costs at those nodes where a component set is placed, where a bulky component set is punished more according to the scalability-related cost function, and $J_m^k = 0$ if no rack is placed at node m. The third term formulates the total transmission cost of all external traffic. The DCN internal overhead due to inter-node transmissions is accounted for in the fourth term, where both delay and transmission costs of all internal traffic are accounted for by using θ. Finally, cost can be saved in the fifth term if a pair of content and core switches is collocated with a set of server racks for resource sharing. In addition, the total cost of all racks is a constant and thus is ignored in objective (1).

By constraint (2), each rack can be placed at one node. If rack i is placed at node m and rack j at node n, X_{mn}^{ij} will take 1 as formulated in constraint (3). Constraints (4) and (5) check whether a pair of content and core switches is placed at node u. If any inter-node DCN internal traffic is switched at node u as formulated in constraint (4), or any external demand is served by the DCN service interface at u as formulated in constraint (5), then a pair of content and core switches is placed at this node. Constraint (6) identifies the zone (i.e., the shadowed block in Fig. 3) that the number of racks at node m falls into, such that the basic cost can be properly calculated in the second term of objective (1).

Constraint (7) says that the sum of all inter-node internal traffic equals that switched by all core switches, where X_{mn}^{ij} is formulated in constraint (3). Constraint (8) limits the sum load of external and internal traffic at a core switch to its maximum switching capacity. Constraint (9) formulates the flow conservation of external demands. Finally, constraints (10) and (11) check whether a pair of content and core switches is collocated with any racks at a particular node u, which provides Y_u to count the last term in objective (1).

IV. Heuristic

In this section, we propose a heuristic, distributed DCN placement (DDP) (see Fig. 4). It consists of two stages detailed in Subsections IV.A and IV.B. Stage I in Fig. 4(a) is for component set clustering. Stage II in Fig. 4(b) is for component set placement and traffic routing. For simplicity, we use "core switch" to denote a pair of content and core switches, and assume that the system parameter settings will not result in more component sets than $|\boldsymbol{V}|$.

A. Component Set Clustering

To reduce the DCN internal overhead (due to inter-node transmissions of the internal traffic), it is desirable to put those racks with heavy inter-rack traffic into the same component set placed at a single node, so that data switching among them can be handled locally. Based on this idea, we first assume that a component set $\boldsymbol{C}_i$ consists of a core switch and several server racks (but later on either one could be missing). The server racks are sequentially added to $\boldsymbol{C}_i$ to minimize an average cost $\mathcal{A}_i$ as defined in

$$\mathcal{A}_i = \frac{P_s + P_{C_i} - P_0}{Q_{C_i}}. \tag{12}$$

The cost of a core switch P_s and the basic cost savings P_0 due to collocation of switches and racks are defined in Subsection III.A. P_{C_i} is the basic cost of $\boldsymbol{C}_i$. As more racks are sequentially added to $\boldsymbol{C}_i$, P_{C_i} takes the scalability-related cost function in Fig. 3 into account. Also, Q_{C_i} accounts for the total amount of inter-rack internal traffic in $\boldsymbol{C}_i$. If there is only a single rack in $\boldsymbol{C}_i$, Q_{C_i} can be defined as a small positive value.

By minimizing $\mathcal{A}_i$ in Eq. (12), server racks can be clustered into component sets, and the number of required core switches can be determined accordingly. Figure 4(a) gives the pseudo-code of this process in `Clustering` $(\boldsymbol{R}, \{\boldsymbol{C}_i\})$, with inputs (global variables) defined in Subsection III.A. It includes two subroutines. In particular, `Find_a_component_set` $(\boldsymbol{\mathcal{R}}, \boldsymbol{C}_i)$ is meant to find a single component set $\boldsymbol{C}_i$ consisting of several server racks, and a core switch is added to each $\boldsymbol{C}_i$ later on. After the set of all server racks $\boldsymbol{R}$ have been properly clustered into a set of component sets $\{\boldsymbol{C}_i\}$, the total amount of inter-node DCN internal traffic L (i.e., traffic across the component sets) can be calculated based on $\{\boldsymbol{C}_i\}$ and the predefined DCN internal traffic r_{ij} among the server racks. Then, by taking all external demands $\Sigma_{v\in V} d_v$ and the maximum switching capacity Q of a core switch into account, `Core_switch_adjustment` $(\{\boldsymbol{C}_i\})$ adjusts the number of core switches by checking whether some switches should be removed from or added to the component sets in $\{\boldsymbol{C}_i\}$. With the number of core switches N_C defined in Eq. (13), sufficient switching capacity is ensured for handling all external demands and DCN internal traffic, because traffic routing in the optical network (which is not necessarily shortest-path based) provides an additional mechanism to achieve load balancing among the core switches.

$$N_C = \left\lceil \frac{1}{Q}\left(L + \sum_{v\in V} d_v\right)\right\rceil. \tag{13}$$

B. Component Set Placement and Traffic Routing

Stage II of DDP [see Fig. 4(b)] adopts iterative processes to place the component sets one by one. Note that the switch and rack clustering process (i.e., Stage I of DDP) may result in three types of component sets, some with both a core switch and several server racks and others with only one or the other. In Step 1 of Fig. 4(b), component sets with a core switch in each (which can provide service interfaces to external demands using the content switch) are classified into a set $\boldsymbol{X}$ and others (consisting of only server racks) into another set $\boldsymbol{Y}$. The former are placed first in Steps 2–4 and then the latter in Step 5.

In Step 2 of Fig. 4(b), a set $\boldsymbol{CS} \subseteq \boldsymbol{X}$ is defined to denote the set of component sets in $\boldsymbol{X}$ that have already been placed so far, and $\overline{\boldsymbol{CS}}$ is its complementary set in $\boldsymbol{X}$. A not-yet-placed component set $CS_i \in \overline{\boldsymbol{CS}}$ is chosen to be placed in Step 3. CS_i is chosen as the one that has the largest sum load of internal traffic with all component sets placed in previous rounds. If it is the first one to be placed, the one that has the largest internal traffic load with another component set is chosen as CS_i. By giving priority to CS_i with a large traffic load, CS_i will have the privilege of being placed prior to others and thus can take a better position (i.e., node) to reduce the DCN internal overhead.

Based on $CS_i \in \overline{\boldsymbol{CS}}$ chosen in Step 2, Step 3 finds a suitable node to place CS_i. This is achieved by trying each and all nodes $u \in \boldsymbol{V}$. A particular node u is tried by assuming that CS_i has been (temporarily) placed at the node, and then a cost metric $\mathcal{B}_u$ is calculated under this assumption. After all nodes $u \in \boldsymbol{V}$ have been tried, the one that results in the minimum value of $\mathcal{B}_u$ is chosen to place CS_i. Next, the above process (Steps 2 and 3) is repeated to place another $CS_i \in \boldsymbol{X}$, as shown in Step 4. Accordingly, the component sets are placed one by one until all $CS_i \in \boldsymbol{X}$ have been placed in the network.

The focus is then turned to how to define and calculate $\mathcal{B}_u$. In fact, we define $\mathcal{B}_u$ to take both external demands and DCN internal traffic into account, under the assumption that CS_i has been (temporarily) placed at node u:

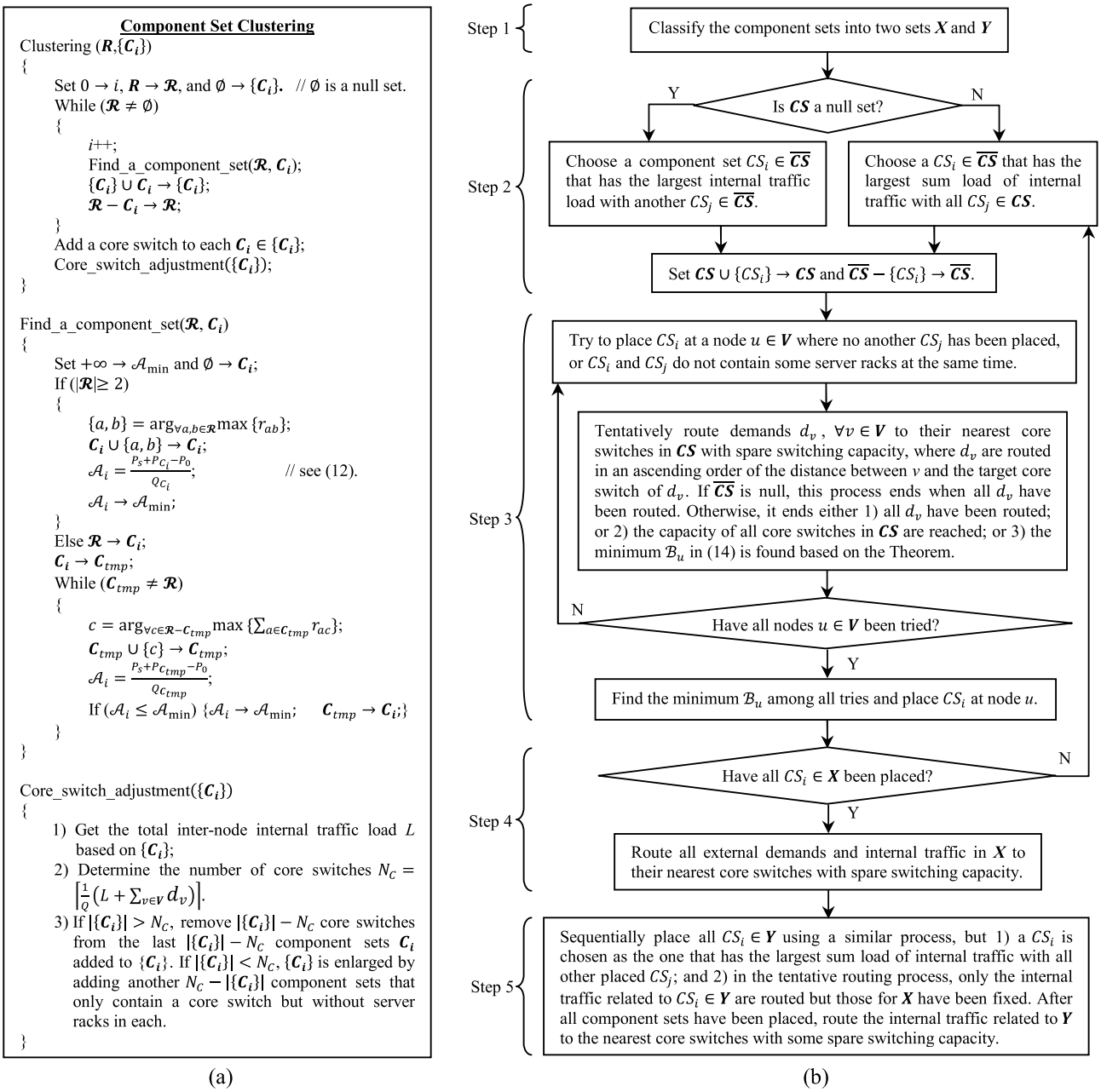


Fig. 4. Heuristic algorithm DDP for distributed DCN placement. (a) Component set clustering (Stage I of DDP) and (b) component set placement and traffic routing (Stage II of DDP).

$$\mathcal{B}_u = \frac{P_C + P_I + P_D}{T_I + T_D}. \tag{14}$$

In particular, P_C is the total cost of all core switches placed so far, and P_I is the cost of the corresponding internal overhead (including both transmission and delay costs scaled by θ). T_I is the sum load of all inter-node internal traffic among the placed component sets. The remaining parameters P_D and T_D are related to the external demand routing as explained below.

When a node $u \in \boldsymbol{V}$ is tried, each demand in the network tries to find a serving core switch among those placed so far. Since $CS_i \in \boldsymbol{X}$ (with a core switch) are placed one by one, the total switching capacity of the already placed core switches is limited if the placement has not been completed yet. As a result, only a subset of demands can be served during the course, and each demand tries to find the nearest core switch with some spare switching capacity as its serving core switch. To this end, the demands are considered (i.e., served or routed) one by one in ascending order of the distance between the demanding node and its target core switch. In other words, those demands with a shorter request and service transmission distance will be considered earlier. In Eq. (14), P_D is the total transmission cost of all demands that are currently served and T_D is the

sum load of the served demands. Therefore, $\mathcal{B}_u$ in Eq. (14) gauges the average cost per unit traffic during the placement process by taking both external and internal traffic into account. It is then easy to understand that CS_i chosen in Step 2 of Fig. 4(b) should be placed at the node that can lead to the minimum value of $\mathcal{B}_u$. On the other hand, the following theorem (proved in Appendix A) shows that $\mathcal{B}_u$ for a particular $u \in \boldsymbol{V}$ will be minimized at a certain stage during the process of (tentative) external demand routing, which ends the process according to the last condition (i.e., condition 3) specified in Step 3 of Fig. 4(b).

Theorem: Assume that the demand routing process in Stage II of DDP is not constrained by any conditions. As the number of routed demands k increases, there exists a $k_{\min}$ such that $\mathcal{B}_u$ for $\forall\ u \in \boldsymbol{V}$ keeps decreasing for $k \le k_{\min}$ and increasing for $k > k_{\min}$.

Note that when a node is tried for possible placement of $CS_i \in \boldsymbol{X}$, the external demand routing is tentative and is just for calculating $\mathcal{B}_u$ in Eq. (14). Accordingly, each time when a new node is tried, or a new CS_i (other than the last one in $\boldsymbol{X}$) is placed, demand routing will be carried out again and the results will be renewed. As specified in Step 4, the final routing scheme of external demands and inter-node internal traffic in $\boldsymbol{X}$ will not be fixed until the last $CS_i \in \boldsymbol{X}$ has been placed.

Finally, Step 5 in Fig. 4(b) adopts a similar process to place the component sets in $\boldsymbol{Y}$. The difference is that only the internal traffic of those component sets in $\boldsymbol{Y}$ are routed, whereas the routing of external demands and internal traffic in $\boldsymbol{X}$ has been fixed at this point.

C. Time Complexity of DDP

The time complexity of DDP takes both Stages I and II into account. In Stage I, `Find_a_component_set` $(\boldsymbol{\mathcal{R}}, \boldsymbol{C}_i)$ needs at most $O(|\boldsymbol{R}|^2)$ operations to add a rack to a specific $\boldsymbol{C}_i$. Since at most $|\boldsymbol{R}|$ racks can be added, running the process at most $|\boldsymbol{R}|$ times in `Clustering` $(\boldsymbol{R}, \{\boldsymbol{C}_i\})$ results in a complexity of $O(|\boldsymbol{R}|^3)$. This dominates $O(|\boldsymbol{R}|^2)$ of `Core_switch_adjustment` $(\{\boldsymbol{C}_i\})$ for calculating L and leads to a total complexity of $O(|\boldsymbol{R}|^3)$ for Stage I. On the other hand, the complexity of Stage II in DDP is dominated by Steps 2–4, where $O(|\boldsymbol{V}|^2)$ operations are needed to choose a CS_i in Step 2. After that, Step 3 tries $|\boldsymbol{V}|$ nodes for possible placement of CS_i, in which at most $|\boldsymbol{V}|$ demands will be sequentially routed. Since Step 3 is carried out after Step 2 is completed, the complexity of Steps 2 and 3 still remains $O(|\boldsymbol{V}|^2)$. Then Step 4 controls the process, to be repeated for at most $|\boldsymbol{V}|$ times. Consequently, the total complexity of Steps 2–4 is $O(|\boldsymbol{V}|^3)$. As a result, the complexity of DDP is $O(|\boldsymbol{R}|^3 + |\boldsymbol{V}|^3)$.

V. Numerical Results

Simulations are carried out based on the pan-European COST 239 network, with 11 nodes and 26 links, as shown in Fig. 5(a). We consider 10 server racks in the DCN, with an internal traffic load between an arbitrary pair of server racks defined in Fig. 5(b). External demands on the DCN service and per-unit-traffic transmission cost at each link are defined in Figs. 5(c) and 5(d), respectively. Figure 5(e) gives simulation parameters, including those for the scalability-related cost function.

A. ILP-Based Optimal Solution

The ILP-based optimal solution is shown in Fig. 5(a). Racks are divided into four component sets and are placed at nodes 2, 4, 5, and 8, respectively. In particular, the one placed at node 5 does not include a core switch, and thus three core switches are required (placed at nodes 2, 4, and 8). Note that the number of required core switches is determined by the optimization process based on the given DCN internal traffic pattern and external demands, instead of being predefined.

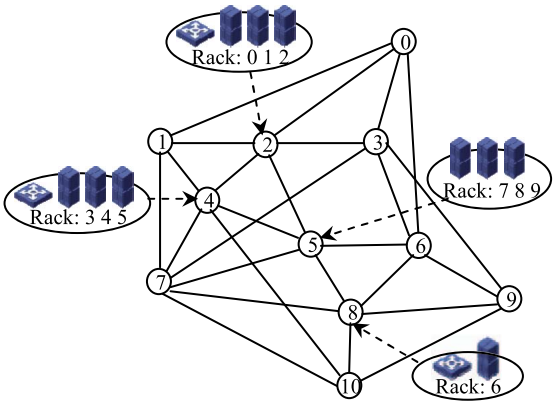
Figure 5(c) lists the routes of external demands in the optimal ILP solution, where the number above each arrow indicates the traffic load on the corresponding link or switched locally at the same node (e.g., $2 \overset{6}{\leftrightarrow} 2$). We can see that most of the demands are served by their closest core switches. Nevertheless, the three units of demands at node 5 are split into $2 + 1$ and are served by core switches at nodes 4 and 8, respectively. This is because the core switch at node 4 runs out of its switching capacity, and thus one unit of demand at node 5 must take a slightly longer route to be served by the core switch placed at node 8. In fact, this shows that the proposed architecture provides a mechanism for balancing traffic loads among the core switches via external demand routing in the optical network.

Figure 5(f) shows how the inter-node internal traffic is routed and switched. Internal traffic between two component sets is generally switched by one or both of the two core switches collocated with the component sets. In particular, the internal traffic between nodes 2 and 4 is switched by both core switches, where each takes almost half of the load. Since there is no core switch at node 5, inter-node internal traffic at node 5 is switched by nearby core switches placed at other nodes.

B. Heuristic DDP Solution

The DDP solution is shown in Fig. 6. In particular, Fig. 6(a) shows how the core switches and server racks are clustered into component sets and placed at different nodes. The number of core switches required by DDP is four, which is one more than that in the optimal solution [see Figs. 5(a) and 6(a)].

Figure 6(b) lists the routes for external demands. Similar to Fig. 5(c), external demands are generally routed based on shortest paths. Nevertheless, since the core switch at node 5 can only serve three units of demands at node 9, the remaining two units are served by the next closest core switch (at node 4) that can offer some spare switching capacity.



(a)

(i,j)	r_{ij}	(i,j)	r_{ij}	(i,j)	r_{ij}	(i,j)	r_{ij}	(i,j)	r_{ij}
(0, 1)	13	(1, 2)	11	(2, 4)	7	(3, 7)	1	(5, 7)	5
(0, 2)	11	(1, 3)	9	(2, 5)	5	(3, 8)	1	(5, 8)	5
(0, 3)	9	(1, 4)	7	(2, 6)	3	(3, 9)	3	(5, 9)	5
(0, 4)	7	(1, 5)	5	(2, 7)	1	(4, 5)	5	(6, 7)	7
(0, 5)	5	(1, 6)	3	(2, 8)	1	(4, 6)	3	(6, 8)	7
(0, 6)	5	(1, 7)	3	(2, 9)	3	(4, 7)	3	(6, 9)	7
(0, 7)	5	(1, 8)	3	(3, 4)	7	(4, 8)	3	(7, 8)	9
(0, 8)	5	(1, 9)	3	(3, 5)	5	(4, 9)	3	(7, 9)	9
(0, 9)	5	(2, 3)	9	(3, 6)	3	(5, 6)	5	(8, 9)	11

(b)

Node (v)	d_v	Demand routing
0: Copenhagen	5	$2 \overset{5}{\leftrightarrow} 0$
1: London	4	$4 \overset{4}{\leftrightarrow} 1$
2: Amsterdam	6	$2 \overset{6}{\leftrightarrow} 2$
3: Berlin	5	$2 \overset{5}{\leftrightarrow} 3$
4: Brussels	7	$4 \overset{7}{\leftrightarrow} 4$
5: Luxembourg	3	$4 \overset{2}{\leftrightarrow} 5$ $8 \overset{1}{\leftrightarrow} 5$
6: Prague	5	$8 \overset{5}{\leftrightarrow} 6$
7: Paris	6	$4 \overset{6}{\leftrightarrow} 7$
8: Zürich	8	$8 \overset{8}{\leftrightarrow} 8$
9: Vienna	5	$8 \overset{5}{\leftrightarrow} 9$
10: Milan	7	$8 \overset{7}{\leftrightarrow} 10$

(c)

Simulation Parameters									
P_s=30000	P_0=5000	Q=78	θ=1.2	K=4	α=1000	β=1000	γ=100	N_1=0	
N_2=3	N_3=5	N_3=8	Δ_1=20000	Δ_2=100000	Δ_3=180000	Δ_4=250000			

(e)

Link	Cost	Link	Cost	Link	Cost
(0, 1)	1310	(2, 5)	390	(5, 8)	350
(0, 2)	760	(3, 6)	340	(6, 8)	565
(0, 3)	390	(3, 7)	1090	(6, 9)	320
(0, 6)	740	(3, 9)	660	(7, 8)	600
(1, 2)	550	(4, 5)	220	(7, 10)	820
(1, 4)	390	(4, 7)	300	(8, 9)	730
(1, 7)	450	(4, 10)	930	(8, 10)	320
(2, 3)	660	(5, 6)	730	(9, 10)	820
(2, 4)	210	(5, 7)	400		

(d)

(m,n)	u	T^u_{mn}
(2, 4)	2	33
(2, 4)	4	30
(2, 5)	2	29
(2, 8)	8	11
(4, 5)	4	29
(4, 8)	8	11
(5, 8)	8	21

(f)

Fig. 5. ILP-based optimal solution with a system cost of 238,181 for distributed placement of the DCN. (a) Pan-European COST 239 network, (b) internal traffic load between two DCN server racks, (c) demand at each node and demand routing, (d) link cost in distance (kilometers), (e) simulation parameters, and (f) inter-node DCN internal traffic.

Figure 6(c) shows how the inter-node DCN internal traffic is switched in DDP. For each pair of component sets, the inter-node internal traffic is almost equally handled by the two collocated core switches.

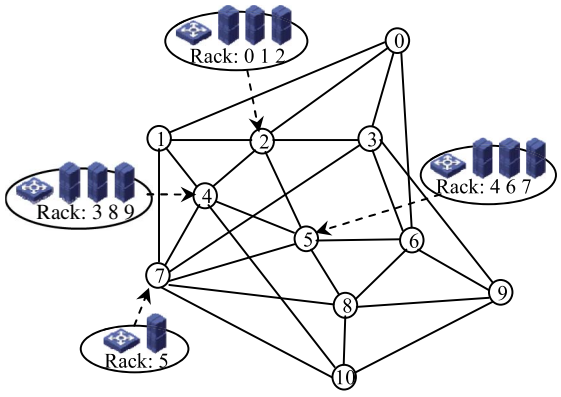
It is shown that the system cost achieved by DDP in Fig. 6 is only 13.88% above that by the optimal ILP. This confirms the superior performance of the proposed heuristic.

C. Impact of System Parameters

In this part, we study the impact of system parameters on the solutions. To ensure the accuracy of the analysis, solutions are obtained from the optimal ILP rather than DDP.

Figure 7 shows how the cost scaling factor θ affects the solution in our experiment. θ can be manipulated to leverage not only between delay and transmission costs in the DCN internal overhead as defined in Subsection III.A but also between the costs of external and internal traffic. As θ increases from 1.0 to 2.4, the number of component sets stays the same, but the nodes populated by the component sets become closer to each other (from {2, 4, 8, 5} to {2, 4, 5, 7}). This is because the relative importance (i.e., weight) of the DCN internal overhead increases with θ, and it should be reduced by putting the component sets closer to each other. Generally, the number of component sets may also change with θ, though it is not observed in this particular example.

Figure 8 shows the impact of the scalability-related cost function (see Fig. 3). As the cost for bulky component sets increases, the number of clustered component sets increases to reduce the number of racks in each. In the extreme case where the scalability-related cost is very high, 10 component sets will be generated and be placed at 10 different nodes among all 11 in COST 239 (see the last pair of columns in Fig. 8). Also, the number of required core switches increases with the scalability-related cost as well, due to more inter-node DCN internal traffic resulting from the more distributed placement. However, it increases slower than the number of component sets, leading to some component sets with only server racks.



(a)

Node (v)	d_v	Routes
0	5	$2 \overset{5}{\leftrightarrow} 0$
1	4	$4 \overset{4}{\leftrightarrow} 1$
2	6	$2 \overset{6}{\leftrightarrow} 2$
3	5	$2 \overset{5}{\leftrightarrow} 3$
4	7	$4 \overset{7}{\leftrightarrow} 4$
5	3	$5 \overset{3}{\leftrightarrow} 5$
6	5	$5 \overset{5}{\leftrightarrow} 6$
7	6	$7 \overset{6}{\leftrightarrow} 7$
8	8	$5 \overset{8}{\leftrightarrow} 8$
9	5	$5 \overset{3}{\leftrightarrow} 9 \quad 4 \overset{2}{\leftrightarrow} 5 \overset{2}{\leftrightarrow} 6 \overset{2}{\leftrightarrow} 9$
10	7	$5 \overset{7}{\leftrightarrow} 10$

(b)

(m, n)	u	T^u_{mn}
(2, 4)	2	24
(2, 4)	4	23
(2, 5)	2	21
(2, 5)	5	20
(2, 7)	2	7
(2, 7)	7	8
(4, 5)	4	24
(4, 5)	5	25
(4, 7)	4	7
(4, 7)	7	8
(5, 7)	5	7
(5, 7)	7	8

(c)

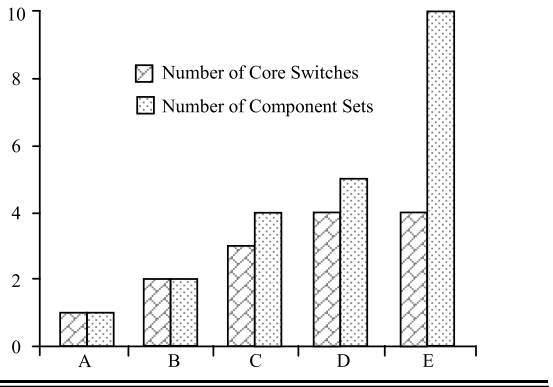
Fig. 6. Heuristic DDP solution with an overall system cost of 271,238 (13.88% above the optimal solution in Fig. 5). (a) Heuristic DDP solution, (b) demand at each node and demand routing, and (c) inter-node DCN internal traffic.

θ	Nodes populated	
1.0	2 4 8	5
1.2	2 4 8	5
1.4	2 4 5	7
1.6	2 4 5	7
1.8	2 4 5	7
2.0	2 4 5	7
2.2	2 4 5	7
2.4	2 4 5	7

Nodes with pure servers

Nodes with a core switch

Fig. 7. Component set placement changes with θ.



A: $N_1 = 0$ $N_2 = 3$ N_3=5 N_4=8 Δ_1=5000 Δ_2=8000 Δ_3=15000 Δ_4=20000
B: $N_1 = 0$ $N_2 = 3$ N_3=5 N_4=8 Δ_1=10000 Δ_2=15000 Δ_3=30000 Δ_4=40000
C: $N_1 = 0$ $N_2 = 3$ N_3=5 N_4=8 Δ_1=20000 Δ_2=50000 Δ_3=100000 Δ_4=150000
D: $N_1 = 0$ $N_2 = 2$ N_3=4 N_4=7 Δ_1=20000 Δ_2=200000 Δ_3=320000 Δ_4=400000
E: $N_1 = 0$ $N_2 = 1$ N_3=3 N_4=6 Δ_1=20000 Δ_2=100000 Δ_3=300000 Δ_4=450000

Fig. 8. Solution changes with the scalability-related cost function in Fig. 3.

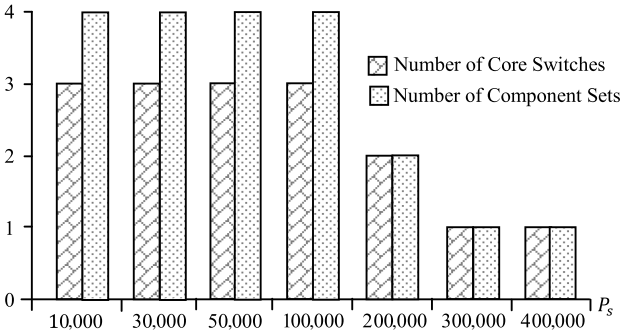


Fig. 9. Solution changes with P_s.

Figure 9 shows the impact of P_s (the cost of a pair of content and core switches). As P_s increases, the number of core switches decreases, and finally it could be reduced to 1 in the extreme case (see the last two pairs of columns in Fig. 9). Meanwhile, the number of component sets becomes smaller, and more inter-rack traffic is handled by the aggregation switches at the more bulky component sets.

VI. Conclusion

We proposed a scalable DCN architecture using optical switches and interconnects, where switches and server racks are distributed across a given optical network. This solves the critical concern about DCN scalability but leads to additional internal overhead due to delay and transmission costs of the DCN internal traffic. By leveraging among multiple conflicting factors and taking sufficient care of the DCN internal overhead, we minimized the system cost of deploying a DCN in a distributed manner. An ILP and a heuristic DDP were proposed to place the DCN component sets under a given set of external demands and internal traffic patterns. Our work addresses both scalability and cost minimization issues from a network point of view for practical deployment of distributed DCNs in an optical

network. Future work may consider the survivability aspects of the proposed architecture.

Appendix A: Proof of the Theorem

Theorem: Assume that the demand routing process in Stage II of DDP is not constrained by any conditions. As the number of routed demands k increases, there exists a $k_{\min}$ such that $\mathcal{B}_u$ for $\forall u \in V$ keeps decreasing for $k \le k_{\min}$ and increasing for $k_{\min}$.

Proof: Let $\mathcal{D}_k$ be the kth routed demand and $\mathcal{C}_k$ be the length of its route from the demanding node to the target core switch. Since the demands are routed one by one in an ascending order of $\mathcal{C}_k$, we have

$$\mathcal{C}_1 < \mathcal{C}_2 < \mathcal{C}_3 < \cdots < \mathcal{C}_{k-1} < \mathcal{C}_k < \cdots. \tag{A1}$$

For simplicity, we define

$$\mathcal{P}_k = P_C + P_I + \sum_k \mathcal{C}_k \mathcal{D}_k, \tag{A2}$$

$$\mathcal{T}_k = T_I + \sum_k \mathcal{D}_k, \tag{A3}$$

where $\{P_C, P_I, T_I\}$ stay the same as in Eq. (14) and are unchanged as the demands are sequentially routed, whereas P_D and T_D in Eq. (14) grows as $P_D = \sum_k \mathcal{C}_k \mathcal{D}_k$ and $T_D = \sum_k \mathcal{D}_k$. By using $\mathcal{B}^k$ to denote $\mathcal{B}_u$ in Eq. (14), we have

$$\mathcal{B}^k = \frac{\mathcal{P}_k}{\mathcal{T}_k}, \tag{A4}$$

and

$$\mathcal{P}_k = \mathcal{P}_{k-1} + \mathcal{C}_k \mathcal{D}_k, \tag{A5}$$

$$\mathcal{T}_k = \mathcal{T}_{k-1} + \mathcal{D}_k. \tag{A6}$$

We now assume that $\mathcal{B}^k < \mathcal{B}^{k-1}$ for a specific value of k. According to Eqs. (A4)–(A6), we get

$$\frac{\mathcal{P}_{k-1} + \mathcal{C}_k \mathcal{D}_k}{\mathcal{T}_{k-1} + \mathcal{D}_k} < \frac{\mathcal{P}_{k-1}}{\mathcal{T}_{k-1}}. \tag{A7}$$

From Eqs. (A1) and (A7), we have

$$\mathcal{C}_{k-1} < \mathcal{C}_k < \frac{\mathcal{P}_{k-1}}{\mathcal{T}_{k-1}} = \frac{\mathcal{P}_{k-2} + \mathcal{C}_{k-1}\mathcal{D}_{k-1}}{\mathcal{T}_{k-2} + \mathcal{D}_{k-1}}, \tag{A8}$$

and thus

$$\mathcal{C}_{k-1} < \frac{\mathcal{P}_{k-2}}{\mathcal{T}_{k-2}}. \tag{A9}$$

Based on Eqs. (A4)–(A6), Eq. (A9) is equivalent to

$$\mathcal{B}^{k-1} < \mathcal{B}^{k-2}. \tag{A10}$$

Therefore, we have the induction that $\mathcal{B}^k < \mathcal{B}^{k-1}$ entails $\mathcal{B}^{k-1} < \mathcal{B}^{k-2}$, and so on. In other words, if $\mathcal{B}^k < \mathcal{B}^{k-1}$ for a particular k, then the series of $\mathcal{B}^k$ must keep decreasing until k.

Similarly, we can prove that $\mathcal{B}^{k+1} > \mathcal{B}^k$ for a particular k entails the series of $\mathcal{B}^k$ to keep increasing beyond $k+1$. Note that $\mathcal{B}^k$ indeed denotes $\mathcal{B}_u$ for $\forall u \in V$. Then the theorem is proved by combining the two parts with the branch point k defined as $k_{\min}$.

Acknowledgments

This work is supported by the Major State Basic Research Program of China (973 Project Nos. 2013CB329301 and 2010CB327806), the National Natural Science Fund of China (NSFC Project Nos. 61372085, 61032003, 61271165, and 61202379), and the Research Fund for the Doctoral Program of Higher Education of China (RFDP Project Nos. 20120185110025, 20120185110030, and 20120032120041). It is also supported by Tianjin Key Laboratory of Cognitive Computing and Application, School of Computer Science and Technology, Tianjin University, Tianjin, China.

References

[1] K. Chen, C. Guo, H. Wu, J. Yuan, Z. Feng, Y. Chen, S. Lu, and W. Wu, "DAC: Generic and automatic address configuration for data center networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 84–99, 2012.

[2] M. Bari, R. Boutaba, R. Esteves, L. Granville, M. Podlesny, M. Rabbani, Q. Zhang, and M. Zhani, "Data center network virtualization: A survey," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 2, pp. 909–928, 2013.

[3] C. Lam, H. Liu, B. Koley, X. Zhao, V. Kamalov, and V. Gill, "Fiber optic communication technologies: What's needed for datacenter network operations," *IEEE Commun. Mag.*, vol. 48, no. 7, pp. 32–39, July 2010.

[4] Z. Zheng, T. Zhou, M. Lyu, and I. King, "Component ranking for fault-tolerant cloud applications," *IEEE Trans. Serv. Comput.*, vol. 5, no. 4, pp. 540–550, 2012.

[5] L. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: Towards a cloud definition," *Comput. Commun. Rev.*, vol. 39, no. 1, pp. 50–55, Jan. 2009.

[6] P. Wright, T. Harmer, J. Hawkins, and Y. L. Sun, "A commodity-focused multi-cloud marketplace exemplar application," in *2011 IEEE Int. Conf. on Cloud Computing (CLOUD)*, 2011, pp. 590–597.

[7] H. J. Chao, Z. Jing, and S. Y. Liew, "Matching algorithms for three-stage bufferless Clos network switches," *IEEE Commun. Mag.*, vol. 41, no. 10, pp. 46–54, 2003.

[8] S. Jiang, G. Hu, S. Y. Liew, and H. J. Chao, "Scheduling algorithms for shared fiber-delay-line optical packet switches—Part II: The three-stage Clos-network case," *J. Lightwave Technol.*, vol. 23, no. 4, pp. 1601–1609, 2005.

[9] F. Wang and M. Hamdi, "Strictly non-blocking conditions for the central-stage buffered Clos-network," *IEEE Commun. Lett.*, vol. 12, no. 3, pp. 206–208, 2008.

[10] X. Yuan, W. Nienaber, Z. Duan, and R. Melhem, "Oblivious routing in fat-tree based system area networks with uncertain traffic demands," *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, pp. 1439–1452, 2009.

[11] S. Coll, F. J. Mora, J. Duato, and F. Petrini, "Efficient and scalable hardware-based multicast in fat-tree networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 9, pp. 1285–1298, 2009.

[12] F. O. Sem-Jacobsen, T. Skeie, O. Lysne, and J. Duato, "Dynamic fault tolerance in fat trees," *IEEE Trans. Comput.*, vol. 60, no. 4, pp. 508–525, 2011.

[13] R. N. Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, "PortLand: A scalable fault-tolerant layer 2 data center network fabric," *Comput. Commun. Rev.*, vol. 39, no. 4, pp. 39–50, Oct. 2009.

[14] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "DCell: A scalable and fault-tolerant network structure for data centers," *Comput. Commun. Rev.*, vol. 38, no. 4, pp. 75–86, Oct. 2008.

[15] T. Hoff, "Google architecture," July 2007 [Online]. Available: http://highscalability.com/google-architecture.

[16] J. Snyder, "Microsoft: Datacenter growth defies Moore's law," 2007 [Online]. Available: http://www.pcworld.com/article/id,130921/article.html.

[17] U.S. Environmental Protection Agency, "Report to congress on server and data center efficiency (public law 109-431)," ENERGY STAR Program, Aug. 2007.

[18] D. D. Kandlur and T. W. Keller, "Green data centers and hot chips," in *Proc. 46th Annu. Design Automation Conf.*, 2009, pp. 888–890.

[19] A. Benner, "Optical interconnect opportunities in supercomputers and high end computing," in *Optical Fiber Communication Conf. and Expo. and the Nat. Fiber Optic Engineers Conf. (OFC/NFOEC)*, 2012, paper OTu2B.4.

[20] "Vision and roadmap: Routing telecom and data centers toward efficient energy use," poster presented at Workshop on Routing Telecom and Data Centers, 2009.

[21] Y. Zhang and N. Ansari, "HERO: Hierarchical energy optimization for data center networks," *IEEE Syst. J.*, to be published.

[22] Y. Zhang and N. Ansari, "On architecture design, congestion notification, TCP incast and power consumption in data centers," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 1, pp. 39–64, 2013.

[23] G. Wang, D. G. Andersen, M. Kaminsky, K. Papagiannaki, T. E. Ng, M. Kozuch, and M. Ryan, "c-Through: Part-time optics in data centers," in *Proc. ACM SIGCOMM*, 2010, pp. 327–338.

[24] 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 *Proc. ACM SIGCOMM*, 2010, pp. 339–350.

[25] X. Ye, Y. Yin, S. J. B. Yoo, P. Mejia, R. Proietti, and V. Akella, "DOS: A scalable optical switch for datacenters," in *Proc. 6th ACM/IEEE Symp. on Architectures for Networking and Communications Systems (ANCS)*, 2010, p. 24.

[26] A. Singla, A. Singh, K. Ramachandran, L. Xu, and Y. Zhang, "Proteus: A topology malleable data center network," in *Proc. Ninth ACM SIGCOMM Workshop on Hot Topics in Networks (Hotnets)*, 2010, p. 8.

[27] K. Xi, Y.-H. Kao, M. Yang, and H. J. Chao, "Petabit optical switch for data center networks," Tech. Rep., Polytechnic Institute of New York University, 2010.

[28] J. Gripp, J. E. Simsarian, J. D. LeGrange, P. Bernasconi, and D. T. Neilson, "Photonic terabit routers: The IRIS project," in *Optical Fiber Communication Conf.*, 2010, paper OThP3.

[29] C. Kachris and I. Tomkos, "A survey on optical interconnects for data centers," *IEEE Commun. Surv. Tutorials*, vol. 14, no. 4, pp. 1021–1036, 2012.

[30] Y. Zhang and N. Ansari, "On mitigating TCP incast in data center networks," *IEEE INFOCOM*, Shanghai, 2011, pp. 51–55.

[31] B. Towles and W. J. Dally, "Guaranteed scheduling for switches with configuration overhead," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 835–847, Oct. 2003.

[32] B. Wu, K. L. Yeung, P.-H. Ho, and X. H. Jiang, "Minimum delay scheduling for performance guaranteed switches with optical fabrics," *J. Lightwave Technol.*, vol. 27, no. 16, pp. 3453–3465, Aug. 2009.

[33] B. Wu, K. L. Yeung, M. Hamdi, and X. Li, "Minimizing internal speedup for performance guaranteed switches with optical fabrics," *IEEE/ACM Trans. Netw.*, vol. 17, no. 2, pp. 632–645, Apr. 2009.

[34] L. Liu, D. Zhang, T. Tsuritani, R. Vilalta, R. Casellas, L. Hong, I. Morita, H. Guo, J. Wu, R. Martinez, and R. Munoz, "Field trial of an OpenFlow-based unified control plane for multilayer multi-granularity optical switching networks," *J. Lightwave Technol.*, vol. 31, no. 4, pp. 506–514, 2013.