HERO: Hierarchical Energy Optimization for Data Center Networks

Yan Zhang, *Student Member, IEEE*, and Nirwan Ansari, *Fellow, IEEE*

***Abstract*—The rapid escalating power consumption has become critically important to modern data centers. Existing works on reducing the power consumption of network elements formulate the power optimization problem for a general network topology and require a centralized controller. As the scale of data centers increases, the complexity of solving this optimization problem increases rapidly. Inspired from the hierarchical data center network (DCN) topologies and data center traffic patterns, we design a two-level, pod-level, and core-level power optimization model, namely, Hierarchical EneRgy Optimization (HERO), to reduce the power consumption of network elements by switching off network switches and links while still guaranteeing full connectivity and maximizing link utilization. Given a physical DCN topology and a traffic matrix, we illustrate that two-level power optimizations in HERO fall in the class of capacitated multicommodity minimum cost flow (CMCF) problem, which is NP-hard. Therefore, we design several heuristic algorithms based on different switch elimination criteria to solve the proposed HERO optimization problem. The power-saving performance of the proposed HERO model is evaluated by several experiments with different traffic patterns. Our simulations demonstrate that HERO can reduce power consumptions of network elements effectively with reduced complexity.**

***Index Terms*—Data center networks (DCNs), energy efficiency, green data centers, power optimization.**

I. Introduction

DATA center networks (DCNs) are prevalent and essential to provide a myriad of services and applications. In order to provide a reliable and scalable computing infrastructure, the network capacity of DCNs is especially provisioned for worst case or peak-hour workload, and thus, data centers consume a huge amount of energy. The Report to Congress on Server and Data Center Energy Efficiency [1] indicated that data centers and servers consumed about 61 billion kilowatthours (kWh) in 2006 (1.5% of the total U.S. electricity consumption) for a total electricity cost of about $4.5 billion, and the electricity usage of data centers has been almost doubled from 2000 to 2006. As reported in [2], the total power consumption of data centers in 2010 increases by about 56% from 2005 to 2010 for worldwide data centers and increases by only 36% for U.S. data centers. It has been well established in the research literature that the average server utilization is often below 30% of the maximum utilization in data centers [3]. At low levels of workload, servers are highly energy inefficient because, even in the idle state, the power consumed is over 50% of its peak power for an energy-efficient server [3] and often over 80% for a commodity server [4]. Moreover, switches are not energy efficient currently either, and they consume 70%–80% of their peak power in their idle state [5].

The high operational costs and the mismatch between data center utilization and power consumption have spurred great interest in improving data center energy efficiency. The power consumption of network elements can account for 10%–20% of a data center's total power consumption [6]. Thus, reducing the power cost of network elements without adversely affecting network performance presents a great challenge. Several studies [7]–[10] have investigated the power savings of the network infrastructure in DCNs by switching off some unneeded network devices or by putting them into sleep. In a recent work, a network-wide power manager, ElasticTree [7], was proposed to optimize the energy consumption of DCNs by dynamically optimizing the subset of active network elements, switches, and links to satisfy dynamic traffic loads while still meeting the performance and fault tolerance requirements. Shang *et al.* [8] solved the energy-saving problem in DCNs from a routing perspective. Mann *et al.* [9] presented a network power-aware framework, the VMFLow, for the placement and migration of virtual machines (VMs), taking into account the network topology as well as network traffic demands to optimize the network power consumption while satisfying as many network traffic demands as possible. A more general solution, the VMPlanner [10], has been proposed to reduce the network element power consumption by optimizing both VM placement and traffic flow routing. A survey on power consumption in data centers along with solutions has recently been reported in [11]. All of these prior works formulated the power optimization problem for a general network topology and required a centralized power management controller, which monitors and predicts traffic demands at each network element and controls the power status of all network elements. As the scale of DCNs increases, the complexity of the power optimization problem of DCNs and the required computational time to solve this optimization problem increase rapidly.

Inspired from the hierarchical DCN topologies and data center traffic patterns, we propose a **H**ierarchical **E**ne**R**gy **O**ptimization (HERO) model to reduce the power consumption of data centers. Given a DCN topology and a traffic matrix, we evaluate the possibility of turning off some network elements (i.e., routers, switches, and links) hierarchically, without violating the network connectivity and QoS constraints. In this paper, we extend the previous HERO model [12] by including the switching power penalty. We also include detailed performance

Manuscript received November 12, 2012; revised September 19, 2013; accepted October 1, 2013. Date of publication October 28, 2013; date of current version May 22, 2015.

The authors are with the Advanced Networking Laboratory, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: yz45@njit.edu; nirwan.ansari@njit.edu).

Digital Object Identifier 10.1109/JSYST.2013.2285606

evaluations using the emulated typical data center workloads. The main contributions of this paper are described as follows. First, a hierarchical energy optimization model for DCNs is proposed. One of the major advantages of the proposed hierarchical model is that it reduces the algorithm complexity by transforming the whole network power optimization problem into several subnetwork power optimization problems. Second, several heuristic algorithms to solve the proposed HERO model are verified. One more major advantage of the proposed energy optimization model is that different heuristic algorithms at different levels can be adopted.

The rest of this paper is organized as follows. Section II presents the background and motivation of the hierarchical approach to reduce the power consumption of DCNs. Details of the proposed hierarchical approach, the network connectivity with the proposed HERO model, and the algorithm complexity are described in Section III. Then, the hierarchical heuristic algorithm is presented in Section IV. The performance of the proposed HERO model is evaluated in Section V. Section VI concludes this paper.

II. Background and Motivation

The main idea of the proposed hierarchical network element power optimization model is inspired from the observations and analysis of the DCN architectures and traffic patterns. In this section, we will discuss the architectures and the traffic patterns in DCNs.

A. Data Center Topology

Conventional DCNs [13] typically consist of a two- or three-tier hierarchical switching infrastructure. Two-tier architecture only has the core and the edge switch tiers. In the three-tier architecture, an aggregation switch tier is inserted in the middle between the core and the edge switch tiers. All servers attach to DCNs through edge tier switches. The edge switches and aggregation switches can form several switch groups, and the core switches can form another switch group as shown in Fig. 1(a). As analyzed in previous works, conventional DCNs have some well-known problems, e.g., scalability and resource fragmentation, server to server connectivity, and cost. Several new topologies have been proposed for DCNs recently, and they can be divided into two categories, switch-centric topology, e.g., VL2 [14] and PortLand [15], and server-centric topology, e.g., DCell [16] and BCube [17].

VL2 [14] is a three-tier architecture, which shares many features with the conventional DCN architecture. The main difference is that a Clos topology is formed between the core tier and the aggregation tier switches. VL2 shares almost all of the features with the conventional DCN architecture except the rich connectivity between the aggregation switches and the core switches. PortLand [15] is another three-tier architecture that forms a fat-tree topology. The edge and aggregation switches form a complete bipartite graph, i.e., a Clos graph. A collection of edge and aggregation switches is called a pod in PortLand architecture. Each pod is connected with all core switches and thus forms a second Clos topology. Thus, each pod can form a switch group, and the aggregation switches and the core switches can form another switch group.

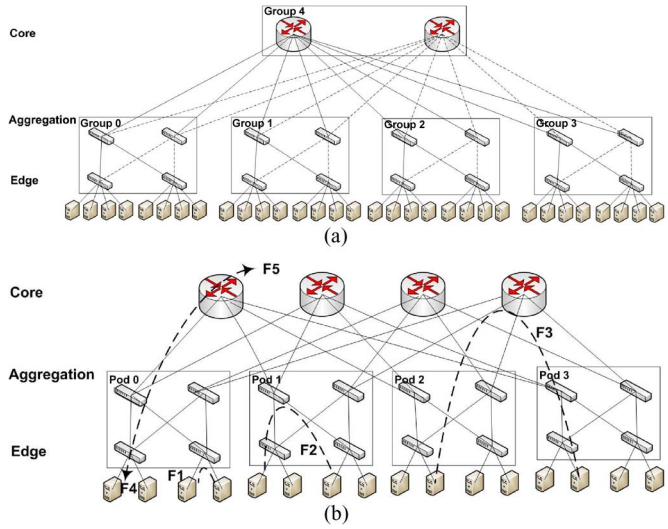


Fig. 1. DCN topologies and traffic categories. (a) Conventional DCN treelike topology. (b) Fat-tree DCN topology and traffic pattern.

DCell [16] and BCube [17] are representative modular DCN architectures, which are constructed with new building blocks of shipping containers instead of server racks. Each container houses up to a few thousands of servers on multiple racks within a standard 40- or 20-ft shipping container. The shipping container-based modular data center simplifies supply management by hooking up power, networking, and cooling infrastructure to commence the services, shortens deployment time, increases system and power density, and reduces cooling and manufacturing cost. Both DCell and BCube are multilevel recursively defined DCN architectures built with miniswitches and servers equipped with multiple network ports. A k-level $DCell_k$ and $BCube_k$ can be constructed recursively with several $(k-1)$-level $DCell_{k-1}$ and $BCube_{k-1}$, respectively. Therefore, each $DCell_{k-1}$ and $BCube_{k-1}$ can form one switch collection, and the k-level switches can form another switch group.

As analyzed earlier, we can observe that almost all the topologies of DCNs typically consist of a two- or three-tier hierarchical switching infrastructure, including the conventional treelike switching infrastructure and newly proposed data center architectures. The networking elements, including switches and links, can be organized into several groups.

B. Data Center Traffic Patterns

Traffic in DCNs can be categorized into five classes: intraedge switch traffic, interedge but intrapod traffic, interpod traffic, incoming traffic, and outgoing traffic. For the intraedge switch traffic shown as F1 in Fig. 1(b), it only requires the connected edge switch and links to be powered on to minimize the power consumption. The interedge but intrapod traffic goes within the same pod but needs to go through different edge switches, shown as F2 in Fig. 1(b). Interpod traffic is the flows that go through different pods, shown as F3 in Fig. 1(b). The incoming (F4) and the outgoing (F5) traffic are traffic flows that go into and out of DCNs, respectively.

The aforementioned observations on hierarchical DCN topologies and data center traffic patterns suggest that a DCN can be divided into several pod-level subnetworks and a

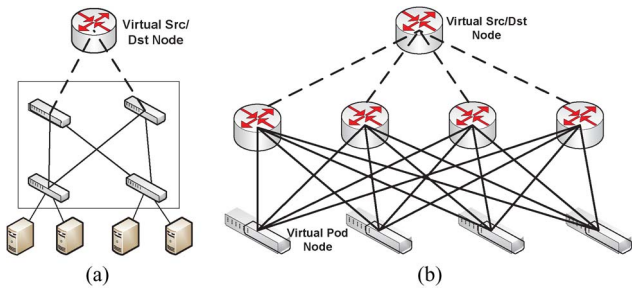


Fig. 2. Subnetwork topologies of a 4-ary fat-tree network. (a) Pod-level subgraph. (b) Core-level subgraph.

core-level subnetwork, and also, the traffic can be reorganized accordingly. As an example, Fig. 2 shows the subnetwork topologies of a 4-ary fat-tree network as shown in Fig. 1(b). There are four pods (Pod0–Pod3), each of which contains edge switches, aggregation switches, and servers. As discussed earlier, the traffic related to one pod can be reorganized as intraedge switch traffic, interedge but intrapod traffic, pod-incoming traffic, and pod-outgoing traffic. The pod-incoming traffic is merged from interpod traffic and incoming traffic from outside of the DCN destination to one of the servers belonging to this pod, while the pod-outgoing traffic is merged from interpod traffic and outgoing traffic to outside of the DCN that originated from one of the servers belonging to this pod. For each pod, a pod-level subnetwork can be formed with the original pod, a virtual node representing the destination of the pod-incoming traffic and the source of the pod-outgoing traffic, and virtual links, each of which connects the virtual node with one of the aggregation switches belonging to this pod as shown in Fig. 2(a). Similarly, by abstracting each pod in Fig. 1(b) to a virtual pod node, a core-level subnetwork can be formulated with these virtual pod nodes, the core switches, a virtual source/destination node representing the source of the incoming traffic and the destination of the outgoing traffic, and some virtual links, each of which connects the virtual source/destination node with one of the core switches as shown in Fig. 2(b). The traffic for this core-level subnetwork can be reorganized from the original traffic for the whole DCN by removing the intraedge switch traffic and interedge but intrapod traffic.

The aforementioned observations and analysis call for hierarchical energy optimization. The power optimization of data centers can be divided into two levels: core level and pod level. The objectives of core-level power optimization are twofold: to determine the core switches that must stay active to send the outgoing traffic and the aggregation switches which serve the out-pod traffic in each pod. The objective of pod-level power optimization is to determine the aggregation switches that must be powered to flow the intrapod traffic. The potential benefit of hierarchical energy optimization is to simplify the network element energy optimization problem by reducing the number of variables greatly.

III. Hierarchical Energy Optimization Algorithm

As analyzed earlier, the networking infrastructure power optimization of DCNs can be divided into the core-level and the pod-level network element power optimization. The optimizer's role in each level is to find the minimum power network subset to meet the performance and fault tolerance goals by powering off the unneeded switches and links. In this section, we describe the hierarchical energy optimization algorithm and also analyze the algorithm complexity.

A. Problem Formulation

Given the network topology, the traffic matrix, and the power consumption of each link and switch, the designed core-level and pod-level power optimizations can be modeled as two capacitated multicommodity minimum cost flow (CMCF) [18] problems. The basic assumptions are given as follows.

1) A physical network topology is given. The DCN infrastructure can be modeled by a graph $G = (V, E)$, where V and E are the set of vertices and edges, respectively. The vertices represent network nodes while the edges represent network links. We use N and L to denote the cardinality of V and E, namely, $N = |V|$ and $L = |E|$, respectively. C_{ij} denotes the capacity of link from node i to node j, and $\alpha \in [0, 1]$ represents the maximum link utilization. The core-level and pod-level power optimizations can be solved based on the core-level subgraph $G^c = (V^c, E^c)$ and a series of pod-level subgraphs $G_i^p = (V_i^p, E_i^p)$ $(i \in [1, N_p])$, respectively, where N_p is the total number of pods. The core-level subgraph $G^c = (V^c, E^c)$ as shown in Fig. 2(b) is formulated in terms of the core switches, the links connecting the aggregation switches and core switches, and the virtual nodes, each of which represents a pod. V^c and E^c denote the set of core-level vertices and edges, and $N^c = |V^c|$ and $L^c = |E^c|$ are the total number of nodes and links in the core-level subgraph G^c. A pod-level subgraph $G_i^p = (V_i^p, E_i^p)$ $(i \in [1, N_p])$ as shown in Fig. 2(a) is formulated in terms of a pod and a virtual node representing the destination of the out-pod traffic. V_i^p and E_i^p denote the set of pod-level vertices and edges in the pod-level subgraph G_i^p for the pod p_i $(i \in [1, N_p])$, and $N_i^p = |V_i^p|$ and $L_i^p = |E_i^p|$ are the total number of nodes and links in the pod-level subgraph G_i^p for the pod p_i.
2) The average amount of traffic exchanged by any source/destination node pair is also given. Denote t_{sd} as the average amount of traffic for the source–target pair $(s, d) \in V \times V$, meaning the traffic that goes from source s to destination d. $T = \{t_{sd}\}$ $(\forall s, d \in V)$ represents the traffic demand matrix which specifies the amount of traffic t_{sd} to be transmitted for each source–destination pair $(s, d) \in V \times V$. By transforming the traffic demand matrix $T = \{t_{sd}\}$ $(\forall s, d \in V)$, we can get the core-level traffic matrix $T^c = \{t_{sd}^c\}$ $(\forall s, d \in V^c)$ and a series of pod-level traffic matrices $T_i^p = \{t_{sd}^{p_i}\}$ $(i \in [1, N_p])$, where p_i $(i \in [1, N_p])$ denotes the ith pod and N_p is the total number of pods. $t_{p_m p_n}^c$ is the average amount of traffic going from source pod p_m to destination pod p_n.
3) The power consumption of each node and link is given. Denote P_i^N and P_{ij}^L as the power consumption of node i and that of the link between node i and node j, respectively.

Based on the aforementioned definitions, the core-level CMCF problem can be formulated as follows.

The objective function is to minimize

$$P^c_{total} = \sum_{i=1}^{N^c}\sum_{j=1}^{N^c} x_{ij}P^L_{ij} + \sum_{i=1}^{N^c} y_i P^N_i \tag{1}$$

where P^c_{total} denotes the total power consumption of network switches and links in the core-level subgraph G^c. $x_{ij} \in \{0,1\}$ and $y_i \in \{0,1\}$ represent the power status of link $(i,j) \in E$ and node $i \in V$, respectively. $x_{ij} \in \{0,1\}$ is a binary variable that is equal to 1 if the link between node i and node j is powered on; otherwise, it is equal to 0. Similarly, $y_i \in \{0,1\}$ is a binary variable that takes the value of 1 if node i is powered on.

It is subject to the following.

1) Flow conservation: Commodities are neither created nor destroyed at intermediate nodes

$$\sum_{i=1}^{N^c}\left(f^{sd}_{ij}\right)^c - \sum_{i=1}^{N^c}\left(f^{sd}_{ji}\right)^c = 0, \quad (\forall s,d,i,j \in V^c, j \neq s,d) \tag{2}$$

where $(f^{sd}_{ij})^c$ denotes the amount of traffic flow from node s to node d routing through the arc from node i to node j in the core-level subgraph G^c.

2) Demand satisfaction: Each source and sink sends and receives an amount of flow equal to its demand, respectively

$$\sum_{i=1}^{N^c}\left(f^{sd}_{ij}\right)^c - \sum_{i=1}^{N^c}\left(f^{sd}_{ji}\right)^c = \begin{cases} t^c_{sd} & \forall s,d \in V^c, j = s \\ -t^c_{sd} & \forall s,d \in V^c, j = d. \end{cases} \tag{3}$$

3) Capacity and utilization constraint: The total flow along each link f^c_{ij} $(\forall i,j \in V^c)$ must be smaller than the link capacity weighed by the link utilization requirement factor α

$$f^c_{ij} = \sum_{s=1}^{N^c}\sum_{d=1}^{N^c}\left(f^{sd}_{ij}\right)^c \leq \alpha C_{ij}x_{ij}, \forall i,j \in V^c \tag{4}$$

where f^c_{ij} represents the total amount of traffic flowing on the link $(i,j) \in E^c$ in the core-level subgraph G^c.

4) Switch turnoff rule: A node can be turned off only if all incoming and outgoing links are actually turned off

$$\sum_{j=1}^{N^c} x_{ij} + \sum_{j=1}^{N^c} x_{ji} \leq M y_i, \forall i \in V^c \tag{5}$$

where M is twice the total number of links connected to node i in the subgraph G^c.

Similarly, the pod-level CMCF power optimization can also be formulated as follows.

The objective function for pod p_m power optimization is to minimize

$$P^{p_m}_{total} = \sum_{i=1}^{N^p_m}\sum_{j=1}^{N^p_m} x_{ij}P^L_{ij} + \sum_{i=1}^{N^p_m} y_i P^N_i \tag{6}$$

where $P^{p_m}_{total}$ denotes the total power consumption of network switches and links in the pod-level subgraph G^p_m.

It is subject to the following.

1) Flow conservation

$$\sum_{i=1}^{N^p_m}\left(f^{sd}_{ij}\right)^{p_m} - \sum_{i=1}^{N^p_m}\left(f^{sd}_{ji}\right)^{p_m} = 0, (\forall s,d,i,j \in V^p_m, j \neq s,d) \tag{7}$$

where $(f^{sd}_{ij})^{p_m}$ denotes the amount of flow from node s to node d routing through the arc from node i to node j in the core-level subgraph G^p_m.

2) Demand satisfaction

$$\sum_{i=1}^{N^p_m}\left(f^{sd}_{ij}\right)^{p_m} - \sum_{i=1}^{N^p_m}\left(f^{sd}_{ji}\right)^{p_m} = \begin{cases} t^{p_m}_{sd} & \forall s,d \in V^p_m, j = s \\ -t^{p_m}_{sd} & \forall s,d \in V^p_m, j = d \end{cases} \tag{8}$$

where f^{sd}_{ij} denotes the amount of traffic flow from source s to destination d that is routed through the arc between node i and node j in pod p_m.

3) Capacity and utilization constraint

$$f^{p_m}_{ij} = \sum_{s=1}^{N^p_m}\sum_{d=1}^{N^p_m}\left(f^{sd}_{ij}\right)^{p_m} \leq \alpha C_{ij}x_{ij}, \quad \forall i,j \in V^p_m \tag{9}$$

where $f^{p_m}_{ij}$ represents the total amount of traffic flowing on the link $(i,j) \in E^p_m$ in the pod-level subgraph G^p_m.

4) Switch turnoff rule

$$\sum_{j=1}^{N^p_m} x_{ij} + \sum_{j=1}^{N^p_m} x_{ji} \leq M y_i, \quad \forall i \in V^p_m. \tag{10}$$

In order to avoid switching frequently between ON/OFF power states, we introduce the switching power penalty of each switch in our optimization model: The core-level and pod-level objective functions can be extended with (11) and (12), respectively, while the constraints in the core-level and pod-level power optimizations can be kept the same as described earlier

$$P'^c_{total} = \sum_{i=1}^{N^c}\sum_{j=1}^{N^c} x_{ij}P^L_{ij} + \sum_{i=1}^{N^c} y_i P^N_i + \sum_{i=1}^{N^c} P^s\left[y_i(1-y^-_i) + y^-_i(1-y_i)\right] \tag{11}$$

$$P'^{p_m}_{total} = \sum_{i=1}^{N^p_m}\sum_{i=1}^{N^p_m} x_{ij}P^L_{ij} + \sum_{i=1}^{N^p_m} y_i P^N_i + \sum_{i=1}^{N^p_m} P^s\left[y_i(1-y^-_i) + y^-_i(1-y_i)\right] \tag{12}$$

where P^s represents the switching penalty for turning a node off if it was on and for turning a node on if it was off and y^-_i denotes the power status of node i determined at the last power optimization.

B. Algorithm Description

A formal description of the HERO algorithm is shown in Algorithm 1. The HERO algorithm can be performed in four steps. In the first step, the power status of the edge switches and edge links connecting the end hosts and edge switches is determined according to traffic matrix T. All edge switches, connecting to any source server or destination server in the traffic matrix T, must be powered on, and others can be powered off. The power status of the core switches and core-level links connecting the core switches and aggregation switches

is determined by solving the core-level CMCF optimization problem with the core-level subgraph G^c, the traffic demand matrix among pods T^c, and the power consumptions of each core switch P_i^N and the core-level links P_{ij}^L. The aggregation switches serve the pod-incoming and the pod-outgoing traffic, which are also the traffic to be considered in the core-level power optimization, so the aggregation switches with the lowest power consumptions should be selected in the core-level power optimization. However, the aggregation switches are not involved directly in the core-level power optimization because they are contained in the virtual pod nodes. The link connecting an aggregation switch and a core switch is unique, and HERO performs power optimization from the core-level power optimization to the pod-level power optimization; therefore, the power status of the link connecting one core switch and one aggregation switch in the core-level power optimization also determines the power status of the aggregation switch that this link connects to since, if the link is powered up, the switch that it connects to must be powered on anyway. Therefore, in order to guarantee that the aggregation switches with the lowest power consumptions are selected for flowing the pod-incoming and the pod-outgoing traffic in the core-level power optimization, the power consumption parameter of the core-level link uses the power consumption of the aggregation switch that it uniquely connects to instead of its own power consumption in the core-level power optimization problem. This is based on the observation that the power consumed by one link is much smaller than one switch. The power status of these selected aggregation switches will be used as the input to the pod-level optimization problem in Step 3. The power consumptions of the virtual pod nodes, the virtual source/destination node, and the virtual links connecting the core switches with the virtual source/destination node in the core-level power optimization are assumed to be zero.

Algorithm 1 Hierarchical Energy Optimization Algorithm

Step 1: Determine the power status of edge switches and edge links according to traffic demand matrix T.
Step 2: Solve the core-level CMCF optimization problem.
Step 2.1: The power status of core switches and core-level links connecting the aggregation switches and the core switches is decided by solving the core-level CMCF optimization problem.
Step 2.2: The aggregation switches serving the out-pod traffic in each pod are selected with the power status of the core-level links, and the selected aggregation switches are powered on.
Step 3: Solve the pod-level CMCF optimization problem.
for $i = 1$ to N^p **do**
Determine the power status of the aggregation switches and the pod-level links connecting the edge switches and the aggregation switches by solving the pod-level optimization problem.
end for
Step 4: In order to provision the whole network connectivity and to meet QoS goals, a merging process is performed.

Then, in each pod, the power status of the aggregation switches serving intrapod traffic and that of the pod-level links connecting the edge switches and aggregation switches are determined by solving the pod-level optimization problem with the pod-level subgraph G_i^p, the traffic demand matrix T_i^p in pod p_i, the power consumption of each link P_{ij}^L and node p_i^N in pod p_i, and the power status of the aggregation switches that serve the out-pod traffic determined in Step 2. The aggregation switches selected to be powered on in the second step and the link connecting the selected aggregation switch to the virtual node in each pod are switched on.

Finally, in order to maintain the whole network connectivity, some fault tolerance, and QoS guarantees, a complementary process is performed. The basic network connectivity to route traffic defined in the given traffic matrix is ensured by the HERO algorithm, which will be illustrated in the next section. However, in some special cases, the whole network connectivity could be broken. For example, if all the traffic flows in a traffic matrix can be classified into intraedge traffic or interedge but intrapod traffic, the optimal solution obtained by the optimization problem in HERO will put all core switches into the idle state, and thus, the connection between the data center and the Internet outside the data center is broken. Similarly, the connection between a pod and other pods in a data center or the connection between a pod and the Internet outside the data center may be broken. In this paper, two basic rules are implemented in the complementary process. One is that at least one core switch is powered on. If none of the core switches is turned on, one core switch is randomly selected to be powered on. Another rule is that at least one aggregation switch that can connect to one active core switch must be turned on in each pod. If no such aggregation switch is turned on in one pod, the one which connects to a powered-on core switch will be switched on.

C. Connectivity Certification

In this section, we illustrate how the network connectivity can be ensured to route the traffic defined in the given traffic matrix by HERO.

1) The connectivity between the servers and edge switches is maintained since the power statuses of edge switches and edge links are determined according to the traffic matrix directly.
2) The connectivity between the edge and aggregation switches in each pod and the connectivity between the aggregation and core switches are guaranteed by the pod-level optimization and the core-level optimization, respectively.
3) The optimal solutions of the pod-level optimization and the core-level optimization may activate different aggregation switches. Thus, the connectivity for out-pod traffic might be broken. In order to ensure the connectivity for out-pod traffic, the links connecting the aggregation switches activated in the core-level optimization and the edge switches in each pod are required to be activated in the pod-level optimization. To realize this, the power

TABLE I
COMPARISON OF ALGORITHM COMPLEXITY

Parameters	Non-Hierarchial	HERO-core	HERO-pod
N_{nodes}	$5K^2/4$	$K^2/4+K+1$	$K+1$
N_{links}	$3K^3/4$	$K^3/4+K^2/4$	$K^2/2+K/2$
N_{var}	$3K^3/2*N_D+3K^3/4+5K^2/4$	$(K^3+K^2)*N_D^c/2+K^3/4+K^2/2+K+1$	$(K^2+K)*N_D^{p_i}+K^2/2+3K/2+1$
N_{con}	$3K^3/2+5K^2/4*(N_D+1)$	$(K^3+K^2)/2+(K^2/4+K+1)*(N_D^c+1)$	$K^2+K+(K+1)*(N_D^{p_i}+1)$

K denotes the degree of the fat-tree structure, and N_D, N_D^c and $N_D^{p_i}$ denote the total number of traffic demands for the entire network, the total number of core-level traffic demands, and the total number of pod-level traffic demands in pod p_i, respectively.

status of the aggregation switches determined in the core-level optimization is input to the pod-level optimization. If the leftover capacity of the active aggregation switches is not enough to accommodate the intrapod interedge traffic, more aggregation switches will be activated in the pod-level optimization. Therefore, the connectivity of out-pod traffic flows is ensured by introducing the power status of aggregation switches obtained in the core-level optimization as the initial condition for the pod-level optimization.

D. Algorithm Complexity

The algorithm complexity of the CMCF optimization problems increases with the increase of the total number of variables and the total number of constraints. As described in Section III-A, the total number of variables in the CMCF power optimization problem can be expressed as the sum of the product of the total number of traffic demands and twice the total number of links considering two directions of communication, the total number of nodes, and the total number of links as follows:

$$N_{var} = N_{links} \times 2 \times N_{demands} + N_{links} + N_{nodes} \quad (13)$$

where N_{var}, N_{nodes}, N_{links}, and $N_{demands}$ denote the total number of optimization variables, nodes, links, and traffic demands, respectively.

The total number of constraints in the CMCF power optimization problem can be expressed as the sum of twice the total number of links, the product of the total number of nodes and the total number of demands, and the total number of nodes by the following equation:

$$N_{con} = N_{links} \times 2 + N_{nodes} \times (N_{demands} + 1) \quad (14)$$

where N_{con} represents the total number of constraints.

Table I compares some major parameters related to the algorithm complexity of the CMCF optimization with a K-ary fat-tree topology. As shown in Fig. 1(b), a K-ary fat-tree DCN is built up with $5K^2/4K$-port switches, and the edges switches and aggregation switches are constructed to form K pods, each of which consists of $K/2$ edge switches and $K/2$ aggregation switches. The edge and aggregation switches form a complete bipartite graph in each pod. Therefore, the total numbers of nodes and links in the pod-level optimization are $K+1$ and $K^2/2+K/2$, respectively. There are $(K/2)^2K$-port core switches, and each core switch has one port connected to each of k pods, while each pod is connected to all core switches, and thus, if each pod is virtualized as a node, the core switches and these virtual nodes form a second bipartite graph. Therefore, the total numbers of nodes and links in the core-level optimization are $K^2/4+K+1$ and $K^3/4+K^2/4$, respectively. The total numbers of switches and links of the whole K-ary fat-tree network are $5K^2/4$ and $3K^3/4$, respectively.

The algorithm complexities of hierarchical and nonhierarchical energy optimization problems can be compared in terms of the ratio of the total number of variables and the ratio of the total number of constraints. Since the total numbers of links and nodes in each pod are much smaller than those in the core level and the core-level and pod-level optimization problems can be solved serially, the algorithm complexity of the core-level power optimization problem dominates the algorithm complexity in the hierarchical power optimization model. Hence, the ratio of the total number of variables and the ratio of the total number of constraints between the hierarchical and the nonhierarchical energy optimization algorithm in a K-ary fat-tree DCN can be expressed as

$$\rho_{var} = \frac{K^3/2 \times N_D^c + K^3/4 + (K^2/4+K)}{3/2K^3 * N_D + 3/4K^3 + 5/4K^2} \quad (15)$$

$$\rho_{con} = \frac{K^3/2 + (K^2/4+K)*(N_D^c+1)}{3/2K^3 + 5/4K^2*(N_D+1)} \quad (16)$$

where ρ_{var} and ρ_{con} denote the ratio of the total number of variables and the ratio of the total number of constraints, respectively. N_D and N_D^c denote the total number of the traffic demands of the entire network and the core-level traffic demands, respectively.

Fig. 3 presents the ratios of the total number of variables and the total number of constraints between hierarchical and nonhierarchical energy optimization algorithms with different values of K and the total number of flows. Under different K and N_D values, it is obvious that the ratios of the total number of variables and the ratios of the total number of constraints between the hierarchical and the nonhierarchical model are smaller than 35% and 40%, respectively, implying that at least 65% and 60% of variables and constraints in the CMCF optimization problem can be reduced with HERO as compared to the nonhierarchical power optimization model, respectively. The ratio of the total number of constraints decreases with the increase of parameter K with the same number of flows.

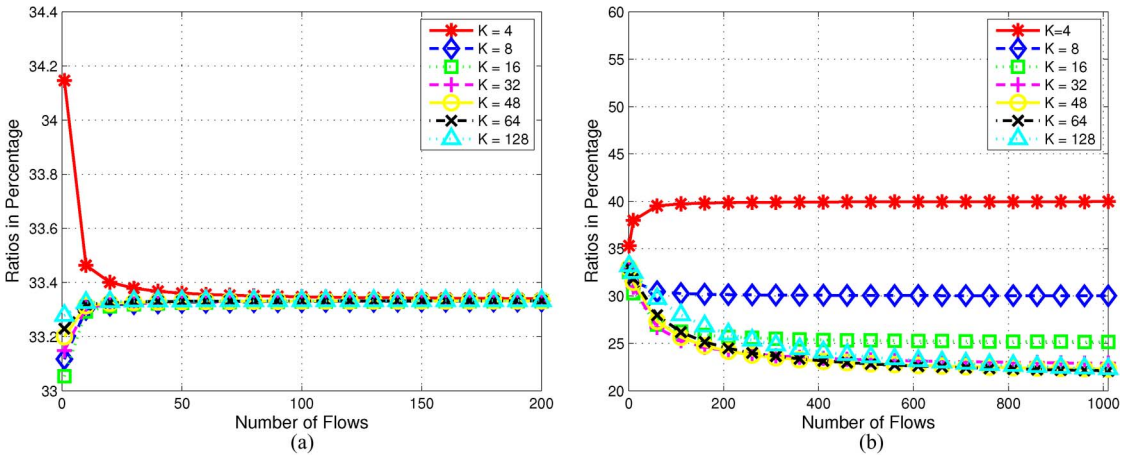


Fig. 3. Ratios of the total number of (a) variables and (b) constraints between hierarchical and nonhierarchical energy optimization algorithms for the fat-tree network topology. (a) Ratio of the total number of variables. (b) Ratio of the total number of constraints.

The results shown in Fig. 3 are the worst case because the ratio values are calculated under the condition that the total number of the core-level flows is the same as that of the entire network, while in fact, the total number of the core-level flows is usually much smaller than that of the whole network.

In general, there are multiple independent pods in a data center. Since the pod-level optimization problems can be solved in parallel, the total computational time to obtain the optimal solution of the HERO model is roughly the sum of the computational time to solve the core-level optimization problem and one pod-level optimization problem. As compared to the nonhierarchical optimization model, the computational time to obtain the optimal solution of the HERO model can also be cut down because both the total number of variables and the total number of constraints are reduced in the optimization problems of the HERO model.

IV. Heuristic Algorithms

The proposed HERO model for DCNs falls in the class of the CMCF problem. CMCF is NP-hard, so exact methods can only be used to solve trivial cases. Simple greedy heuristics of node optimization and link optimization were proposed [19] by trying to switch off an additional network node or link. An improved version of this heuristic algorithm was reported in [20] by explicitly considering the power consumption amount of the devices. The basic idea is to iterate through the node set or link set by sorting the nodes or links according to their power consumption in the node optimization or in the link optimization stage. An energy-aware greedy heuristic routing algorithm for DCNs was presented in [8]. The basic idea of this heuristic routing is to gradually eliminate the lightest loaded switch from those involved in the routing, based on the total throughput of flows carried by each active switch. The problem of this heuristic is that it does not take the switch power consumption model into consideration while different switch models may coexist in a data center.

In this paper, several simple greedy heuristic algorithms based on different switch elimination criteria are designed to solve the hierarchical optimization problem for large DCNs. The proposed energy-efficient heuristic algorithm is based on the observation that the power consumed by one link is much smaller than that of one switch. The switch elimination criterion could be switch throughput, switch power consumption, and switch power efficiency that gradually eliminates the lightest loaded switch based on the total throughput of flows carried by each active switch, the switch with the highest maximum power consumption per port, and the lowest power-efficient switches, respectively. The power efficiency of a switch is defined as the total throughput of flows carried by the switch divided by its power consumption. The basic idea of the proposed heuristic algorithms is to try to turn off the core switches and core-level links iteratively as well as the aggregation switches and pod-level links iteratively in the core-level and the pod-level power optimization, respectively, so that as few switches and links as possible are powered on in DCNs to meet the traffic demands and QoS goals.

We assume that all switches and links are powered on initially and the traffic is routed as normal, and then, we try to selectively power off the switches and links hierarchically in the core-level and pod-level energy optimizations. In the core-level heuristic, we sort the core switches based on different criteria and then try to switch off the selected core switches by checking if the traffic can be flowed through other active core switches; if so, we turn them off. Based on the throughput-based criterion and switch-power-efficiency criterion, the core switches are sorted in the ascending order, while the core switches are sorted in the descending order if the switch power consumption criterion is adopted. Therefore, the core switch with the lowest throughput, the lowest power efficiency, or the highest power consumption model is eliminated first. A similar heuristic is applied to the pod-level optimization except that it tries to turn off aggregation switches in each pod. As an improvement to the switch power consumption-based criterion, another switch power consumption-based criterion is proposed by sorting the core switches in the descending order of the total power consumption of each core switch and its connecting aggregation switches in the core-level power optimization.

V. Performance Evaluation

In this section, we first compare the optimal power consumptions of network elements of the HERO model and those of the nonhierarchical optimization model. Then, we investigate the power-saving performance of different heuristic algorithms. The relationship of the power consumptions and the network utilization is further studied with a diurnal traffic variation over a day. Moreover, the effects of the power penalty when powering on or off network elements in the power optimization model is examined.

Three different switch power consumption models investigated in [7] are referenced to build a DCN in each evaluation experiment. Each switch selects its switch model randomly (uniform distribution) from these three types. In [7], the power consumptions of three different 48-port switch models with all ports staying at the idle state are measured as 151, 133, and 76 W, respectively. Also, a 1/3 extra power consumption required to turn on all ports is observed in [7]. A positive linear relationship between the switch power consumption with all ports staying at the idle state and the number of switch ports is assumed in our evaluations, meaning that a switch consumes more power with the increase of the number of switch ports.

A. Traffic Patterns

In the absence of commercial DCN traffic traces, we generate several traffic patterns to verify the power-saving performance of HERO.

1) Random elephant: An end host sends large elephant flows to any other end host in DCN with uniform probability.
2) All-to-all data shuffle: Every end host transfers a large amount of data to every other hosts.
3) Staggered (P_{out}, P_{edge}, and P_{pod}): A source host sends traffic out of a DCN with probability P_{out} to an end host which is connected by the same edge switch as the source host with probability P_{edge}, to an end host which is within the same pod as the source host but connected by different edge switches with probability P_{pod}, and to the rest of the end hosts with probability $1 - P_{edge} - P_{pod} - P_{out}$.
4) Diurnal traffic variation: The data center traffic variation over one day exhibits a clear diurnal pattern, in which traffic peaks during the day and falls at night.

B. Simulation Results

1) HERO Versus Nonhierarchical Model: The traffic patterns "random elephant" and "all-to-all data shuffle" are two extreme cases with large elephant traffic flows only and with small traffic flows only, respectively. In order to compare the optimal power savings of the proposed HERO model and those of the nonhierarchical optimization model, CPLEX [21] is used to obtain the solutions of the CMCF problems in the evaluations with these two traffic patterns.

Since current commodity switches are not power proportional to work loads (even more than 80% of its peak power consumption at idle state) and the elephant flows dominate the traffic volume in DCNs, we investigate the performance of the HERO model with elephant flows only first, the traffic demand of which for each source–destination pair will consume the whole edge link capacity. Fig. 4 shows the average power consumption of a 4-ary fat-tree network with different numbers of traffic flows. For each number of traffic flows, 1000 simulations are performed with randomly generated traffic flows, and the power consumption is normalized with respect to the maximum power consumption of the whole network. The power consumptions of the nonhierarchical power-saving model and traffic-load proportional power consumption model are also plotted for comparison. The nonhierarchical power-saving model takes the whole DCN topology as an input to the optimization problem and does not consider the hierarchical property of the DCN topology. ElasticTree [7] is a typical example of the nonhierarchical power optimization model. The traffic-load proportional power consumption model is a primary design goal for power optimization schemes, and thus, this model provides the lower bound of power consumptions under different traffic loads. Such an energy-proportional model ideally consumes no power when idle, nearly no power when very little traffic is transmitted through the network, and gradually more power as the traffic increases. As shown in Fig. 4, smaller than 5% more of the maximum power consumption of the whole network is consumed in HERO than that of the nonhierarchical model. Owing to the QoS and fault tolerance protection rules implemented in the complementary procedure, more than about 20% of the maximum power consumption of the whole network is needed as compared to the traffic-load power proportional model under the worst case. The lower bound and the upper bound power consumption under different numbers of elephant flows are also shown in Fig. 4, which reflects the influence of source and destination locations on the power consumption. Since no aggregation and core switches are required to be powered on for intraedge switch traffic flows, the power consumption under some specific number of traffic flows is minimized if all the flows are intraedge switch traffic. Similarly, the power consumption is maximized if all the flows are interpod traffic.

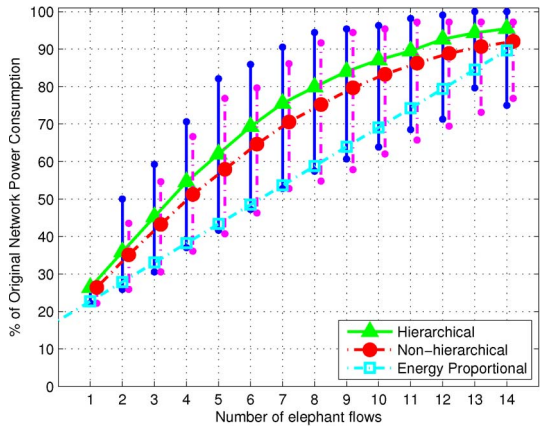


Fig. 4. Power consumptions of a 4-ary fat-tree DCN with different number of elephant traffic flows.

A data shuffle is an expensive but necessary operation for some data center operations. Fig. 5 shows the normalized power consumption of a 4-ary fat-tree network with all-to-all traffic under different traffic loads. With all-to-all traffic,

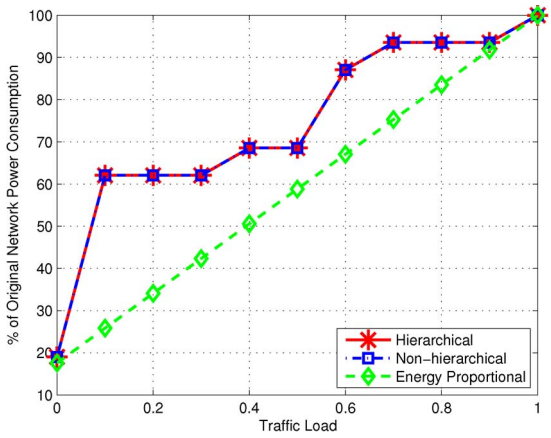


Fig. 5. Power consumptions of a 4-ary fat-tree DCN with all-to-all data shuffle under different traffic loads.

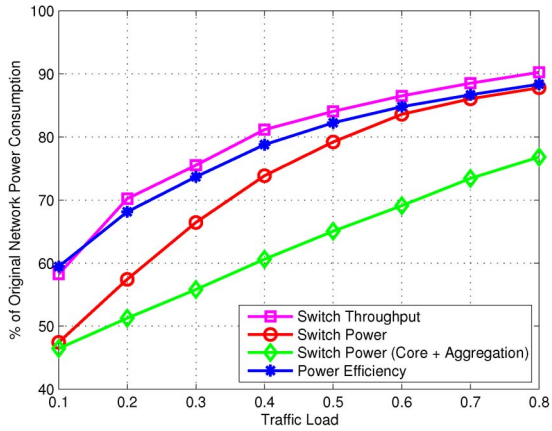


Fig. 6. Power consumptions of different heuristic algorithms in a 16-ary fat-tree DCN with staggered (0.2, 0.4, and 0.24) traffic.

any host will have a traffic flow to any other end host. The traffic load generated at any host will be distributed to other hosts uniformly. As shown in Fig. 5, the power consumptions of HERO and the nonhierarchical model are almost the same under different traffic loads. The power consumption at low traffic load is much higher than that of the traffic-load power proportional model because every edge switch is active with all-to-all traffic.

2) Performance of Heuristic Algorithms: The staggered traffic pattern is a mix of intraedge traffic, interedge but intrapod traffic, interpod traffic, and outgoing traffic as discussed in Section II-B, and it is generated to emulate typical data center traffic patterns based on the published work [14], [22], [23]. With the staggered traffic pattern, experiments are conducted to study the DCN power consumptions with different heuristic algorithms. The power consumptions of four heuristic algorithms under different traffic loads with staggered traffic pattern (0.2, 0.4, and 0.24) in a 16-ary fat-tree DCN are shown in Fig. 6. Traffic demands are generated randomly. Around 20% of the traffic leaves the data center, and 50%, 30%, and 20% of the remaining traffic are intraedge switch traffic, interedge but intrapod traffic, and interpod traffic within a DCN, respectively. As shown in this figure, throughput-based and power-efficiency-based switch elimination criteria consume almost the same power. The switch power consumption method based on the total power consumption of the core and aggregation switches in the core-level power optimization, labeled with "Switch Power (Core + Aggregation)," achieves the minimum power consumption among these four heuristics. At low utilization, about 37% and 53% of the original DCN power consumption can be saved by using this heuristic algorithm in a 16-ary fat-tree DCN, respectively, which are quite close to the corresponding 38% and 60% optimal power savings analyzed in [7]. Also, a linear relationship between the power consumption and the traffic load in the range of 20%–70% of the maximum traffic load can be observed using this heuristic algorithm. Therefore, the power consumption achieved by this heuristic algorithm is very close to that of the optimal solutions of HERO and the nonhierarchical model.

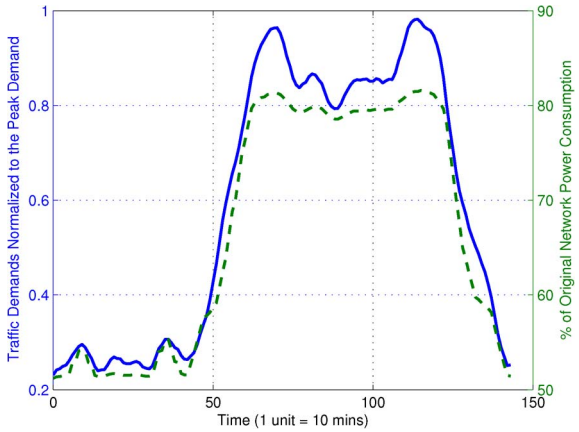


Fig. 7. Power consumptions over a day with a diurnal traffic variation pattern in a 16-ary fat-tree DCN with staggered (0.2, 0.4, and 0.24) traffic.

3) Diurnal Traffic Variation and Switching Power Penalty: The performance of HERO with a diurnal traffic variation over a day is further studied. The power consumptions with the proposed HERO model over one day in 10-min intervals with a diurnal traffic variation pattern with staggered traffic pattern (0.2, 0.4, and 0.24) in a 16-ary fat-tree DCN are depicted in Fig. 7. In this evaluation, the heuristic algorithm based on the total power consumption of the core and aggregation switches in the core-level power optimization is used. From this figure, we can see that the power consumptions also show a clear diurnal pattern and follow the network utilization curve.

In previous evaluations, the power penalty when powering on or off network elements is ignored. We investigate such total power penalty in a 16-ary fat-tree DCN over a one-day diurnal traffic variation with the HERO model and its extended optimization model, which considers the switching power penalty of each switch, respectively. In this evaluation, we conduct several experiments with different settings to the switching power penalty parameter P^S in (11) and (12) in the range of [10%, 50%] of the switch's power consumption. The results show that the total switching power penalty in a day with HERO is more than five times larger than that of its extended optimization model which takes the switching power penalty into account. We also found that, by taking the switching power cost into the power optimization model, the unnecessary switching power penalty can be minimized.

VI. Conclusion

With the increase of scale of DCNs, existing centralized power management schemes to reduce the power consumptions of network elements in DCNs suffer from the scalability problem. Inspired from the hierarchical architectures and traffic patterns of data centers, we have proposed and evaluated the HERO model to reduce the power consumption of network elements without violating the connectivity and QoS constraints. In particular, the power optimization of networking elements by switching off the idle switches or links can be divided into the core-level and the pod-level network element power optimization. Given a physical DCN topology, the traffic matrix, and the power consumption of each link and switch, the designed core-level and pod-level power optimization problems in the proposed HERO model fall in the class of CMCF problem, which is NP-hard. We have designed several heuristic algorithms based on different switch elimination criteria to solve the hierarchical optimization problem for large data centers. We have evaluated the performance of the proposed HERO model by generating several traffic patterns, including random elephant, all-to-all data shuffle, staggered (P_{out}, P_{edge}, and P_{pod}), and diurnal traffic variation. The simulation results showed that the proposed HERO model can achieve optimized power consumptions of a DCN similar to that of the nonhierarchical model but with much less computational efforts.

References

[1] "Environmental Protection Agency, Report to Congress on Server and Data Center Energy Efficiency Public Law 109-431," U.S. Environmental Protection Agency, Washington, DC, USA, Tech. Rep. ENERGY STAR Program, Aug. 2007.

[2] J. G. Koomey, Growth in Data Center Electricity Use 2005 to 2010, Aug. 4, 2011.

[3] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007.

[4] S. Dawson-Haggerty, A. Krioukov, and D. Culler, "Power Optimization—A Reality Check," EECS Dept., Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2009-140, Oct. 2009.

[5] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A power benchmarking framework for network devices," in *Proc. 8th Int. IFIP-TC 6 Netw. Conf.*, Aachen, Germany, 2009, pp. 795–808.

[6] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," *Sigcomm Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, Jan. 2009.

[7] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "ElasticTree: Saving energy in data center networks," in *Proc. 7th Symp. Netw. Syst. Design Implementation*, San Jose, CA, USA, Apr. 2010, pp. 249–264.

[8] Y. Shang, D. Li, and M. Xu, "Energy-aware routing in data center network," in *Proc. SIGCOMM*, New Delhi, India, Aug. 30–Sep. 3, 2010, pp. 1–8.

[9] V. Mann, A. Kumar, P. Dutta, and S. Kalyanaraman, "VMFlow: Leveraging VM mobility to reduce network power costs in data centers," in *Proc. 10th Int. IFIP TC 6 Conf. Netw.*, Valencia, Spain, May 9–13, 2011, pp. 198–211.

[10] W. Fang, X. Liang, S. Li, L. Chiaraviglio, and N. Xiong, "VMPlanner: Optimizing virtual machine placement and traffic flow routing to reduce network power," *Comput. Netw.*, vol. 57, no. 1, pp. 179–196, Jan. 2013.

[11] Y. Zhang and N. Ansari, "On architecture design, congestion notification, TCP incast and power consumption in data centers," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 36–64, First Quarter, 2013.

[12] Y. Zhang and N. Ansari, "HERO: Hierarchical energy optimization for data center networks," in *Proc. IEEE ICC*, Ottawa, ON, Canada, Jun. 10–15, 2012, pp. 2924–2928.

[13] Cisco Data Center Infrastructure 2.5 Design Guide. [Online]. Available: https://www.cisco.com/application/pdf/en/us/guest/netsol/ns107/c649/ccmigration_09186a008073377d.pdf

[14] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: A scalable and flexible data center network," in *Proc. SIGCOMM*, Barcelona, Spain, Aug. 17–21, 2009, pp. 51–62.

[15] R. Niranjan 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," in *Proc. SIGCOMM*, Barcelona, Spain, Aug. 17–21, 2009, pp. 39–50.

[16] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "DCell: A scalable and fault-tolerant network structure for data centers," in *Proc. SIGCOMM*, Seattle, WA, USA, Aug. 17–22, 2008, pp. 75–86.

[17] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "BCube: A high performance, server-centric network architecture for modular data centers," in *Proc. SIGCOMM*, Barcelona, Spain, Aug. 17–21, 2009, pp. 63–74.

[18] I. Ghamlouche, T. G. Crainic, and M. Gendreau, "Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design," *Operations Res.*, vol. 51, no. 4, pp. 655–667, Jul. 2003.

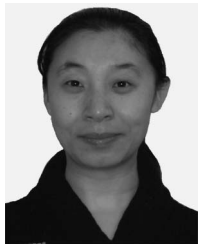
[19] L. Chiaraviglio, M. Mellia, and F. Neri, "Reducing power consumption in backbone networks," in *Proc. IEEE ICC*, Dresden, Germany, Jun. 14–18, 2009, pp. 1–6.

[20] L. Chiaraviglio, M. Mellia, and F. Neri, "Energy-aware backbone networks: A case study," in *Proc. IEEE Int. Conf. Commun.*, Dresden, Germany, Jun. 14–18, 2009, pp. 1–5.

[21] IBM ILOG CPLEX Optimizer. [Online]. Available: http://www-01.ibm.com/software/integration/optimization/cplex-optimizer

[22] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of data center traffic: Measurements and analysis," in *Proc. 9th ACM SIGCOMM Conf. IMC*, Chicago, IL, USA, 2009, pp. 202–208.

[23] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," in *Proc. ACM Workshop Res. Enterprise Netw.*, Barcelona, Spain, Aug. 2009, pp. 65–72.



Yan Zhang (S'10) received the B.E. and M.E. degrees in electrical engineering from Shandong University, Jinan, China, in 2001 and 2004, respectively. She is currently working toward the Ph.D. degree in computer engineering in the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA.

Her research interests include congestion control and energy optimization in data center networks, content delivery acceleration over wide area networks, and energy-efficient networking.

Nirwan Ansari (S'78–M'83–SM'94–F'09) received the Ph.D. degree from Purdue University, West Lafayette, IN, USA.

He is a Professor of electrical and computer engineering with the New Jersey Institute of Technology, Newark, NJ, USA. He has contributed over 450 technical papers, over one-third published in widely cited refereed journals/magazines. He is the holder of over 20 U.S. patents. He has guest edited a number of special issues, covering various emerging topics in communications and networking. His current research focuses on various aspects of broad-band networks and multimedia communications.

Prof. Ansari has assumed various leadership roles in the IEEE Communications Society, and some of his recognitions include the Thomas Edison Patent Award (2010), New Jersey Inventors Hall of Fame Inventor of the Year Award (2012), and a couple of best paper awards.