IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received July 2, 2016, accepted August 1, 2016, date of publication October 19, 2016, date of current version November 8, 2016.

Digital Object Identifier 10.1109/ACCESS.2016. 2617871

E^{3}MC: Improving Energy Efficiency via Elastic Multi-Controller SDN in Data Center Networks

KUN XIE[1], XIAOHONG HUANG[1], SHUAI HAO[2], (Student Member, IEEE), MAODE MA[3], (Senior Member, IEEE), PEI ZHANG[1], AND DINGYUAN HU[1]

[1] Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing, China
[2] Department of Computer Science, College of William and Mary, Williamsburg, VA, USA
[3] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore

Corresponding author: X. Huang (huangxh@bupt.edu.cn)

This work was supported in part by the International Science and Technology Cooperation and Exchange Project of China under Grant 2013DFE13130, in part by the Fundamental Research Funds for the Central Universities, and in part by DNSLAB

ABSTRACT Energy consumed by network constitutes a significant portion of the total power budget in modern data centers. Thus, it is critical to understand the energy consumption and improve the power efficiency of data center networks (DCNs). In doing so, one straightforward and effective way is to make the size of DCNs elastic along with traffic demands, i.e., turning off unnecessary network components to reduce the energy consumption. Today, software defined networking (SDN), as one of the most promising solutions for data center management, provides a paradigm to elastically control the resources of DCNs. However, to the best of our knowledge, the features of SDN have not been fully leveraged to improve the power saving, especially for large-scale multi-controller DCNs. To address this problem, we propose E^{3}MC, a mechanism to improve DCN's energy efficiency via the elastic multi-controller SDN. In E^{3}MC, the energy optimizations for both forwarding and control plane are considered by utilizing SDN's fine-grained routing and dynamic control mapping. In particular, the flow network theory and the bin-packing heuristic are used to deal with the forwarding plane and control plane, respectively. Our simulation results show that E^{3}MC can achieve more efficient power management, especially in highly structured topologies such as Fat-Tree and BCube, by saving up to 50% of network energy, at an acceptable level of computation cost.

INDEX TERMS Data center network, energy management, SDN, multi-controller, elastic structure.

I. INTRODUCTION

Originating from Stanford's "clean slate" projects [1], [2], Software Defined Networking (SDN) was proposed as a new network paradigm with centralized control plane decoupled from forwarding plane [3]. Powered by the unified management and communication protocols (e.g., the OpenFlow [4]), network in SDN is directly programmable, and its resource control is more fine-grained and convenient. With these features, SDN becomes an interesting solution for the management of modern data centers. Furthermore, with the increasing demands for both the computing resources and the variety of functionalities, data centers are scaling fast. Therefore, the SDN with multiple controllers is proposed to cope with the scalability of infrastructure [5], [6].

TABLE 1. Energy consumption on global data centers.

unit: TWh	Data Source: ICTresearch [7]		
	2010	**2011**	**2012**
Power Consumption	5563.1	6358.5	7202.9
Growth Proportion	16.8%	14.3%	13.8%

From the perspective of energy, huge and increasing amounts of electricity have been consumed by data centers every year to provide reliable and stable services (shown in Table 1 [7]). High energy cost has become one of the vital concerns for large-scale data centers. The network, as a crucial component of data center infrastructure, would consume a significant part of total electric power (up to 20% [8]). Also, the proportion is even rising due to the rapid development of energy conservation technologies on servers and cooling systems. Therefore, power optimization has become a key challenge in design and operations for Data Center Networks (DCNs).

To improve the network energy efficiency in data centers, many related efforts have been made, and most of them employ flow routing to tackle the problem, such as

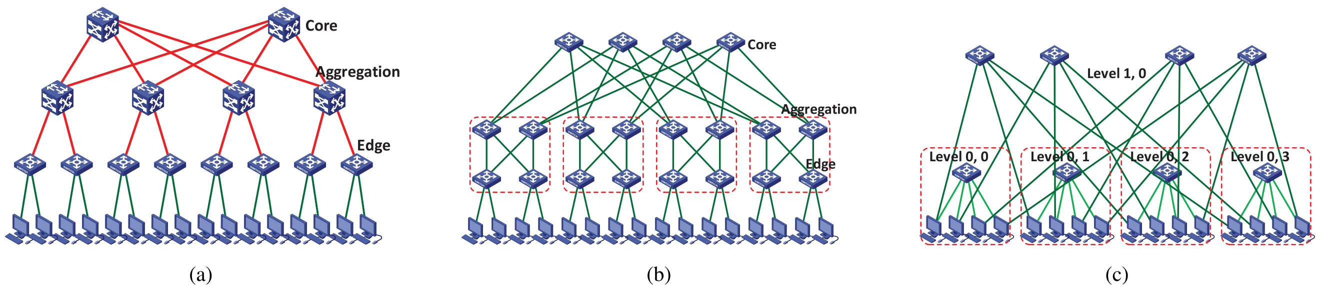


FIGURE 1. **Data center network topologies. (a) 2N-Tree, 1G links and GigE switches in edge, 10G links and 10 GigE switches in aggregation and core. (b) Fat-Tree, all 1G links and commercial Ethernet GigE switches. (c) BCube, all 1G links and commercial Ethernet GigE switches.**

ElasticTree [9] and EAR [10]. The key idea is to converge flows through as few devices as possible and shut down the idle devices, since DCNs are designed to improve the system reliability and performance by redundancy. SDN has been utilized in a few of these solutions, such as CARPO [11] and GreenSDN [12], since it naturally supports such operations to control network components. The structure of centralized control makes it straightforward to manage the resources in DCNs, including bandwidth, ports, and switches. For example, it is convenient to make the devices sleep or wake them up by SDN's control protocols. The network topology and traffic loads can also be dynamically obtained by the controller(s) in real-time.

Therefore, SDN has the innate advantages that can be used to tackle this energy conservation issue. However, its powerful and fine-grained routing functions, such as the dynamic multi-path scheduling, have not been fully leveraged. More importantly, the prior efforts are generally focused on SDN's forwarding plane, and the energy saving of control plane is not conjunctively considered, particularly the multi-controller SDN environment in large-scale data centers. Though there have been some resource-efficient studies on distributed control plane [13], [14], they do not regard the energy as their first concern, and the profile of power consumption for controllers is not combined in their model.

In this paper, we propose the E^{3}MC, a mechanism to improve *E*nergy *E*fficiency via the *E*lastic *M*ulti-*C*ontroller SDN for DCNs. In E^{3}MC, the power optimization for forwarding plane and control plane are jointly considered. In forwarding plane, the multi-path routing with *traffic/flow split* can be employed to improve power efficiency. Also, we study the power saving of multiple controllers in control plane. Like the switch pool, the controller pool can also be elastic along with the traffic demands to reduce energy consumption. In this work, the energy consumption models are used to choose the key devices, and then other idle devices/ports could be shut down or put into dormant mode via SDN's management interfaces to optimize the energy efficiency in DCNs. E^{3}MC can be used in various topologies of modern DCNs. To the best of our knowledge, E^{3}MC is the first to jointly consider forwarding plane and control plane, and the energy consumption characteristics for the DCN energy-efficient work. We conduct the simulations based on Poisson process and real data center traffic dataset, and the results show that the E^{3}MC is an efficient energy saving scheme suitable for various topologies.

The remainder of this paper is organized as follows. We present the background on DCN topologies and energy consumption characteristics for SDN devices in Section II. We describe our multi-controller SDN structure and E^{3}MC modular architecture in Section III. We present the power saving models for forwarding plane and control plane in Section IV and Section V, respectively. We evaluate the simulated results in Section VI. Finally we survey the related work in Section VII and conclude the paper in Section VIII.

II. BACKGROUND

In this section we first present the DCN topologies used in our study, and then provide a background on the power consumption characteristics of SDN's switches and controllers, respectively.

A. DATA CENTER NETWORK TOPOLOGIES

Figure 1 shows the typical topologies of three modern DCNs. First, the 2N-Tree is a traditional DCN structure consisting of three layers of switches. As the service demands scale up to the capacity limits, numerous new DCN topologies have been proposed [15]–[18]. These new designs mainly fall into two categories: the *switch-oriented* and *server-centric* approach. In switch-oriented topologies, such as Fat-Tree [15] and VL2 [16], switches are still the core devices to provide high capability for routing. The server-centric designs, such as DCell [17] and BCube [18], organize the data centers more like a mesh structure. Interconnection and routing functions are integrated with the servers, while switches only provide simple crossbar forwarding.

In a traditional 2N-Tree, shutting one core switch down will cut the effective bandwidth in half. Shutting two switches down will cause the disconnection between servers and thus is not permitted. Those new topologies, such as Fat-Tree and BCube in Figure 1, have more capabilities in bandwidth and switching paths, resulting in more devices being dormant and more energy to be saved.

B. ENERGY CONSUMPTION CHARACTERISTICS

Here we profile the power consumption of SDN's switches and controllers, respectively.

1) POWER CHARACTERISTICS OF SWITCHES

The energy consumption performance of switches has been studied by Mahadevan *et al.* based on the traditional switching devices [19]. They observed that the switches' power consumption consists of the costs by the chassis, line-cards, and ports. The first two parts are fixed and consume the major proportion of power (up to 150 Watts together). The ports themselves cost a small but notable part since turning one on idly would consume 1-2 Watts along with their different line rates set. In addition, traffic going from zero to full capacity will increase the power by less than 5%, which implies that the influence by the utilization of port capacity (i.e., the bandwidth load) is negligible. Therefore, in our work, we assume there is only one line-card, and the power cost by traffic is zero. Finally, the power consumption is given by a linear model:

$$\begin{aligned} Power_{switch} &= Power_{idle} \\ &\quad + \sum_{i=1}^{configs} PortsOnNum_i \times Power_i \end{aligned} \tag{1}$$

$Power_{idle}$ is the power consumed by the switch with no port being turned on. The *configs* is the number of port configurations for different line rates. The $PortsOnNum_i$ is the number of ports running at line rate i, and $Power_i$ is the corresponding power consumed by the ports at rate i.

2) POWER CHARACTERISTICS OF CONTROLLERS

In SDN's paradigm, the controller is an application running atop on a general-purpose server to maintain the network and achieve network functionalities such as routing and security. Therefore, the power profile of controllers would follow the power model of servers in data centers [20], [21]. Unlike the switches, energy consumed by controller's workload is a significant part which cannot be ignored. We employ the model based on CPU utilization [21], since CPU is typically the throughput bottleneck in a controller. This gives us a nonlinear regression model:

$$\begin{aligned} Power_{controller} &= Power_{idle'} + \rho_1 \times util \\ &\quad + \rho_2 \times util^2 + \rho_3 \times util^3 \end{aligned} \tag{2}$$

$Power_{idle'}$ is fixed and denotes the power consumed by a controller with zero load. The *util* is the scaling factor of controller's CPU utilization, and ρ_{1-3} are the empirical correction impact factors measured on sample machines. In our preliminary experiment, this model is generally a concave-downward function and $Power_{idle'}$ always consumes more than 50% power in fully loaded case. This implies that we can save the power by increasing the utilization and shutting down redundant controllers.

Note that the energy consumption model used in E^{3}MC is not exclusive. Different models could be utilized to generate corresponding energy consuming functions in our algorithm, which may produce different energy-saving effects.

III. NETWORK STRUCTURE AND SCHEMATIC ARCHITECTURE

In this section, we introduce the multi-controller SDN and describe the design and modules of E^{3}MC.

A. MULTI-CONTROLLER SDN STRUCTURE

Deploying multiple controllers improves the scalability of SDN network in data centers. For the power saving in E^{3}MC, both the controller pool and switch pool will be able to grow and shrink dynamically, and the multi-controller SDN structure will support such elasticity.

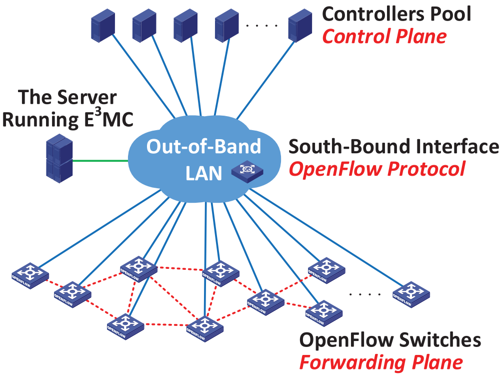


FIGURE 2. Multi-controller SDN structure.

Our network structure of E^{3}MC is shown in Figure 2.[1] The network used for SDN south-bound interface channels is separate and dedicated (out-of-band controllers connection). Each switch could communicate with every controller directly with no OpenFlow switches passed through, so the loads/switches can be shifted across controllers (not a physical complete bipartite graph). An Ethernet LAN is deployed to link north and south, so from the view of south-bound connection, the scalability will be improved by eliminating the single-point failure. Finally, a server running E^{3}MC is connected to the LAN.

The structure in [14] is a general centralized approach, where all controllers in controller cluster adopt Equal Mode (i.e., all as one). Although integrated consistency is guaranteed, distributed control function and scalability are weak, since each controller maintains the whole global view of DCN. Hence, in our multi-controller SDN, each controller maintains a part view of the forwarding network, and each switch has only one master controller. Moreover, the mapping between switches and controllers is dynamic. When a master controller is down, the switches under its control would be migrated to another one. When aggregated load decreases, controller pool can shrink to save energy (Section V). In our current architecture, a standalone E^{3}MC server maintains the

[1] Note that the DCN topologies in Section II describe the connections of the forwarding plane.

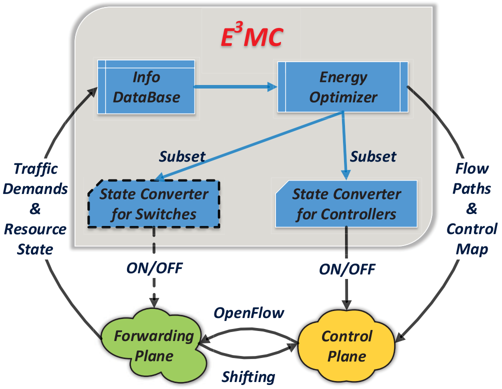


FIGURE 3. Modules diagram.

global view, and even if this energy optimization server is down, the functions of network can still work.

B. E^{3}MC SCHEMATIC ARCHITECTURE

E^{3}MC aims to manage and improve the energy consumption of multi-controller DCN, and its modular architecture, as shown in Figure 3, is integrated with SDN. Both the clusters of controllers and OpenFlow switches dynamically grow or shrink according to the traffic conditions, and the mapping between the switches and controllers would be dynamic. The system consists of four logical modules: Information Database (IDB), Energy Optimizer (EO), State Converter for Switches (SCS), and State Converter for Controllers (SCC). The optimizer, running the energy models on E^{3}MC server, is the kernel of the system and manages the global DCN states to accomplish our elastic energy-efficient mechanism.

IDB gathers the traffic demands in a proper time granularity and maintains topological resources states, then sends them to EO as input. EO first runs the power model for forwarding plane, and calculates the flow paths to obtain the minimum subset of ports and switches. Based on such results, EO then runs the power saving model for control plane and calculates the minimum subset of controllers. With EO's outputs, SCS and SCC change the power states of ports, switches, and controllers. In Figure 3, SCS and its control logic for forwarding plane is dashed, because it can be deployed on either the E^{3}MC server or the controllers. We suggest deploying it in controllers to leverage SDN's structure and also improve the system scalability.

Note that the sorting order of the device state changing actions is important. If a switch/port needs to be shut down (or put into dormant mode), the action would be executed when there is no existing flow passing through the switch/port and its incoming flows had already been rerouted to the new paths. If a switch/port needs to be turned on (or wake up), the action would be executed beforehand, then the new switching/routing rules could be deployed to process the incoming flows. If a controller needs to be shut down, the action would be executed when all switches under its control have been assigned to new master controller. If a controller needs to be turned on, the action would be executed beforehand, and then its new assigned switches could be shifted onto it. On the other hand, the network QoS will be affected during the state change of controllers, and this affection is mostly caused by the switch shifting actions (i.e., the switch migration time). Since the duration of switch migration only takes few tens of milliseconds [13], we believe that the impact of network QoS would remain at an acceptable level.

IV. FORWARDING PLANE POWER OPTIMIZATION MODEL

In this section we formalize the power optimization model with a greedy heuristic for forwarding plane, which is precalculated on EO as a requisite for the model of control plane.

TABLE 2. Summary of notations for forwarding plane model.

V	Set of the nodes: $v_i \in V$ $(i = 1, 2, \ldots, \lvert V\rvert)$.
H, G	Set of hosts, switches: $H \cup G = V$.
A	Set of links/edges: $a_{ij} \in A$ connects nodes from v_i to v_j $(v_i, v_j \in V, i \neq j)$.
W_v	Set of nodes linked to v.
c_{ij}	Bandwidth capacity of edge a_{ij}.
Q	Set of traffic demands: $q = (s, t, d) \in Q$. s: source, t: sink, d: demand, $\lvert Q\rvert = \lvert H\rvert \times \lvert H\rvert$
$f_q(a_{ij})$	Flow of demand q along edge a_{ij}.
$f_q^+(v_i)$	Flow of demand q from node v_i. $f_q^+(v_i) = \sum_{v_j \in W_{v_i}} f_q(a_{ij})$
$f_q^-(v_j)$	Flow of demand q into node v_j. $f_q^-(v_j) = \sum_{v_i \in W_{v_j}} f_q(a_{ij})$.
u_{ij}	Utilization of a_{ij}: $u_{ij} = \sum_{q \in Q} f_q(a_{ij})/c_{ij}$.
$R_{\lvert Q\rvert \times \lvert A\rvert}$	Routing matrix with element $r_{q,a} = f_q(a)$.
$P_{\lvert V\rvert \times \lvert V\rvert}$	Port state matrix: binary decision p_{ij} indicating the port on v_i linked to v_j is ON or OFF.
$M_{1 \times \lvert V\rvert}$	Switch state vector: binary decision m_i indicating the node v_i is ON or OFF.
φ_{ij}	Weight of a_{ij}: The consumed power when flow is loaded along a_{ij}.
ϕ	Energy consumed by the whole forwarding plane: $\phi = \sum_{a_{ij} \in A} \varphi_{ij}$.
x	Energy consumed by an ON port.
y	Energy consumed by an idle switch (no ON port).

A. MODEL DESCRIPTION

The notations used in this section are summarized in Table 2. A bidirectional weighted flow network $N(V, A)$, which represents the forwarding plane, consists of node set V and link/edge set A. Suppose there would be only one link between each pair of nodes (full duplex). The edge $a_{ij} \in A$ connects nodes from v_i to v_j, $v_i, v_j \in V$ $(i, j = 1, 2, \ldots, \lvert V\rvert,\ i \neq j)$, and thus a_{ij} has a non-negative and real-valued capacity c_{ij}. If $a_{ij} \notin A$ or the device (port or switch) associated to a_{ij} has been turned down, we have $c_{ij} = 0$.

There is a set of traffic demands, Q, with element $q = (s, t, d)$, where s is the source host, t is the sink host, and d is the demand value. The flow of demand q refers to f_q.

The main idea of our model is to assign flows for optimal power consumption, leveraging the fine-grained routing function of SDN. When a flow is loaded along edge a_{ij}, the corresponding device states might be changed and the power increment after the flow-loading action is φ_{ij}. Based on the power characteristics of switches (Section II), we can arrive at the energy increment function, where the conditions for p_{ij}/p_{ji} and m_i/m_j describe the original device states:

$$\varphi_{ij} = \begin{cases} 0 & p_{ij} = p_{ji} = 1,\ m_i = m_j = 1, \\ 2x & p_{ij} = p_{ji} = 0,\ m_i = m_j = 1, \\ 2x + y & p_{ij} = p_{ji} = 0,\ m_i \oplus m_j = 1, \\ 2x + 2y & p_{ij} = p_{ji} = 0,\ m_i = m_j = 0. \end{cases} \tag{3}$$

Then we can get the objective function of the forwarding plane to minimize the switches' power consumption, where p_{ij} and m_i describe the states after the transition:

$$MIN(\sum_{a\in A} \varphi_a) = MIN(\sum_{a\in A} x \times p_a + \sum_{v\in V} y \times m_v) \tag{4}$$

The following constraints should be satisfied:

- *Capacity Constraints*: The total load of each link must not exceed their capacity.

$$0 \le \sum_{q\in Q} f_q(a_{ij}) \le c_{ij}, \quad i.e.,\ 0 \le u_{ij} \le 1 \tag{5}$$

- *Flow Conservation*: The flow going out of a switch must be equal to the flow coming into it, unless it is a host.

$$f_q^+(g) = f_q^-(g), \quad \forall g \in G \tag{6}$$

- *Demand Satisfaction*: The flow produced by a source is completely consumed by the corresponding sink(s).

$$f_q^+(s) = f_q^-(t) = d, \quad f_q^-(s) = f_q^+(t) = 0\ (s, t \in H) \tag{7}$$

- *Port Power Symmetrical Constraints*: The state of the port on v_i linked to v_j, must be same as the state of the port on v_j linked to v_i. If there is flow in either direction, both should be powered on.

$$p_{ij} = p_{ji} = \begin{cases} 1, & \sum_{q\in Q} f_q(a_{ij}) + \sum_{q\in Q} f_q(a_{ji}) > 0, \\ 0, & \sum_{q\in Q} f_q(a_{ij}) + \sum_{q\in Q} f_q(a_{ji}) = 0. \end{cases} \tag{8}$$

- *Port and Switch State Association*: When one port on a switch is powered on, the switch should be powered on. When the ports of a switch are all turned off, the switch should be turned off.

$$\forall v_i \in G,\ m_i = \begin{cases} 1, & \sum_{v_j \in W_{v_i}} p_{ij} \ge 1, \\ 0, & \sum_{v_j \in W_{v_i}} p_{ij} = 0. \end{cases} \tag{9}$$

This formal model is a standard Multi-Commodity Flow (MCF) Formulation [22], with flow routing matrix and switches/ports subset as optimal results. It is a NP-Complete mixed-integer linear program with heavy computational requirements [23]. According to [9], the solution time is about $O(|H|^{3.5})$, and thus can only scale up to networks with less than 1000 nodes. For availability, a heuristic algorithm should be carefully chosen to decrease the computational complexity, as discussed in the following.

B. MINIMUM-COST FLOW IN GREEDY HEURISTIC

We employ a greedy heuristic to reduce the computation time, so that our model could be suitable for large networks and online usage. The DCN topology is usually structured and well-organized, and the paths between any two nodes in DCNs are always fixed and predictable. Based on (3), we can greedily assign as many traffic flows as possible to the lowest energy consuming path.

The process assigns traffic demands set Q in an iterative manner. At each iteration, the path (may be more than one for traffic/flow split) bringing minimum energy consumption is selected to bear one q. The residual network of q will be regarded as a new network for the next step. The process can be expressed by the following formula:

$$\begin{cases} \phi^0 = 0, \\ \phi^n = \phi^{n-1} + \phi_{q_n}^{opt}, \quad n = (1, 2, \ldots, |Q|). \end{cases} \tag{10}$$

Demand q_n is assigned to the residual network of q_{n-1}, and minimal energy is consumed with value of $\phi_{q_n}^{opt}$. The ϕ^n is the current greedy optimal solution satisfying n demands, and $\phi^{|Q|}$ is the final result.

For one demand q in current iteration, ϕ_q is the total power consumed by q, and $\varphi_q(a_{ij})$ is the consumed power when the flow of q is loaded along a_{ij} (by turning on associated switches and ports). The path for q that cost minimal energy ϕ_q^{opt} need to be found and the objective function can be summarized as follows:

$$\phi_q^{opt} = MIN(\phi_q) = MIN(\sum_{a_{ij}\in A} \varphi_q(a_{ij})). \tag{11}$$

The energy increment function $\varphi_q(a_{ij})$ and the constraints are same as those listed in last subsection. It is a special case of Minimum-Cost Flow Problem (MCFP) from flow network graph theory, with $\varphi_q(a_{ij})$ as the cost function and one more constraint:

- *Antisymmetry constraints*: The flow from v_i to v_j must be opposite of the flow from v_j to v_i with a negative opposite cost for feedback.

$$f(a_{ij}) = -f(a_{ji}). \tag{12}$$

The following classical theorem [24] characterizes minimum-cost flow:

Theorem 1: A flow is minimum-cost if and only if there are no negative augmenting cycles.

Since our cost function (3) is discrete, *Negative Cost Cycle Cancelling* (NCCC) algorithm [25] is used to solve our problem, which is a successive approximation algorithm with constraints satisfied.[2] The pseudo code of NCCC algorithm is presented in Algorithm 1.

Algorithm 1 NCCC: Find a f_q Without Negative Augmenting Cycle

```
Input:
    Demand $q = q_n = (s, t, d)$,
    resident network $N = N_{q_{n-1}}(V, A)$.
Output:
    Routing solution $R_{1\times|A|}$.
  Step 1: Find a feasible flow $f_q^0$ with value $d$ and cost $\phi_q^0$.[3]
      $f_q^k = f_q^0, \phi_q^k = \phi_q^0.$
  Step 2: Get the residual network $N^k$ base on $f_q^k$. If there is
  no negative cost cycle in $N^k$, process is over and $f_q^k$ is our
  minimum-cost flow $f_q$. If not, find the negative augmenting
  cycle $Z^k$ with residual capacity $\theta^k$ and negative cost $\varphi^k$,
  and go to next step.
  Step 3: Augment $f_q^k$ along $Z^k$ with value $\theta^k$, and get a new
  flow $f_q^{k+1}$ which has same value $d$ but lesser cost $\phi_q^{k+1}$. Go
  to Step 2.
      $f_q^{k+1} = f_q^k Z^k \theta^k, \phi_q^{k+1} = \phi_q^k + \varphi^k, k = k + 1.$
```

Obviously, with the traffic/flow split, the capacity utilization could be further improved and the traffic demands would be loaded on fewer switches. In SDN, traffic between two nodes can be split at the level of flows to use multiple paths. Because of the inherent advantages of SDN, dynamic multipath scheduling should be more efficient than that in a traditional DCN structure [26]. In our model, flow-split can also be supported to utilize more bandwidth (automatically supported in NCCC algorithm). This function could be achieved with appropriate protocols and applications deployed on the controller(s) and switches. However, splitting flow in the granularity of packet may cause a potential problem of TCP performance for the reordering [27], and we can optionally choose whether or not to add appropriate constraints to prevent flows from being split.

Since the paths between two nodes in DCNs are generally fixed, we can just select the path in a determined order to further reduce the computational complexity ($O(|H|^2)$ with flow-split and $O(|H|^{2.5})$ without flow-split [9]). Thus, the calculation for finding the lowest energy consuming path can be omitted, while the power saving effect may reduce a little.

Due to the inherent problem of all greedy algorithms, the real optimal solution is not guaranteed, regardless of splitting the flow is supported or not. Nevertheless, in practice, we can still get a reasonable result to meet our power saving needs.

V. CONTROL PLANE POWER OPTIMIZATION MODEL

With the computed results of forwarding plane, in this section we formalize the power optimization for control plane with a bin-packing based model, which is also calculated on EO module.

A. THE WORKLOAD OF CONTROLLERS

Like [13], we simply refer to controller's CPU cost as the total messages it processed. By this way, the load of controller can be estimated by the OpenFlow message arrival rate, and the throughput would be denoted by the flow processing rate. This gives the controller's CPU utilization as the ratio of current flow processing rate to the maximum capacity, which is the measured rate when CPU usage went to 100%. Then, we can employ the power characteristics of controllers (see (2) in Section II) to build the energy optimization model of control plane.

Note that according to the measured flow characteristics in a data center, the minimum flows arrival interval is less than 10 μs while the maximum value is up to 1 ms [28]. Assuming a data center with 100 edge switches, peak flow arrival rate can be up to 10M per second with the minimum rate to 100K. If the flow throughput rate is 100K for one commodity controller's capacity, such as OpenDaylight or Beacon, it requires only 1 controller to process the minimum load, but 100 for peak load. Furthermore, in a data center with larger scale, there is reasonable need to have multiple controllers. Therefore, the energy saving for controller pool is meaningful, since the number of active controllers vary widely between peak and median loads (1-2 orders of magnitude).

B. MODEL DESCRIPTION

Different from other studies on the improvement of resource efficiency in multiple controllers, we consider the power characteristics as the key parameters in our mechanism. We assume that there is only one machine type of controller servers, resulting in unique energy profile. The roles of these controllers are routing/forwarding control and switches management. According to the power characteristics of controller, the workload of a controller in our model is mainly the sum of flow arrival rates at all switches under its control. Thus, the general idea of the model is to assign alive switches to the controllers for optimal power consumption. The output results from the optimization in Section IV, such as the set of alive switches and the matrix of flow routing, are part of the input of this model. Note that for a switch with only one controller, the load generated by the switch cannot be split or shared onto other controllers. In the following we present the mathematical model of our scheme. The notations used in our formulation are summarized in Table 3.

We have controller set $E = \{e_1, e_2, ..., e_{|E|}\}$ representing control plane. Each controller $e_i \in E$ $(i = 1, 2, ..., |E|)$ has non-negative and real-valued flow processing capacity c_i. If the controller e_i is turned down, we assume that $c_i = 0$.

[2] The common *Minimum Cost Augmented Path* algorithm cannot be used here because of the discreteness of function (3).

[3] The algorithm for solving the maximum flow problem can be used here to find f_q^0, such as Ford-Fulkerson, Edmonds-Karp, etc.

TABLE 3. Summary of notations for control plane model.

E	Set of controllers: $e_i \in E$ $(i = 1, 2, ..., \|E\|)$.
G	Set of alive switches: $g_j \in G$ $(j = 1, 2, ..., \|G\|)$.
W_i	Set of alive switches controlled by e_i.
c_i	Flow processing capacity of e_i: $c_i \geq 0$.
$l(g_j)$	Flow arrival rate through g_j.
u_i	Utilization of e_i: $u_i = \sum_{g \in W_i} l(g)/c_i$.
$R_{\|G\| \times \|E\|}$	Mapping matrix: binary decision $r_{g,h}$ indicating whether g is controlled by h.
$N_{1 \times \|E\|}$	Controller state vector: binary decision n_i indicating the controller e_i is ON or OFF.
φ_i	Energy consumed by e_i.
ϕ	Energy consumed by all the controllers: $\phi = \sum_{e_i \in E} \varphi_i$.
a, b, c, d	Parameters in energy consuming function (13).
k	Number of alive controllers: $k = \sum_{i=1}^{\|E\|} n_i$.

The number of alive controllers is k. There is alive switches set G with element $g_j \in G$ $(j = 1, 2, ..., |G|)$, while the flow arrival rate through g_j is $l(g_j)$. The $l(g_j)$ can be calculated based on the forwarding matrix (Section IV). The power consumed by e_i is φ_i. When a switch is assigned to controller e_i, based on the empirical correction of controller's non-linear power consumption model (Section II), we can arrive at the energy consuming function of one controller:

$$\varphi_i = a + b \cdot u_i + c \cdot u_i^2 + d \cdot u_i^3 \tag{13}$$

Then we can minimize the power consumption of control plane as:

$$\begin{aligned} MIN(\phi) &= MIN(\sum_{e_i \in E} \varphi_i) \\ &= MIN(a \sum_{i=1}^{|E|} n_i + b \sum_{i=1}^{|E|} n_i u_i \\ &\quad + c \sum_{i=1}^{|E|} n_i u_i^2 + d \sum_{i=1}^{|E|} n_i u_i^3) \end{aligned} \tag{14}$$

Meanwhile, the following constraints should be satisfied:

- *Capacity Constraints*: The total workload of each controller must not exceed their capacity.

$$0 \leq \sum_{g \in W_i} l(g) \leq c_i, \quad i.e., \ 0 \leq u_i \leq 1 \tag{15}$$

- *Exclusiveness Constraints*: One switch is controlled by only one controller in any time slot.

$$\sum_{i=1}^{|E|} r_{g_j, e_i} = 1, \ \forall j = 1, 2, ..., |G|. \tag{16}$$

- *Switch and Controller State Association*: When a switch is controlled by a controller, the controller should be powered on. When there is no switch assigned to the controller, it should be turned off.

$$\forall e_i \in E, \ n_i = \begin{cases} 1, & \sum_{j=1}^{|G|} r_{g_j, e_i} \geq 1, \\ 0, & \sum_{j=1}^{|G|} r_{g_j, e_i} = 0. \end{cases} \tag{17}$$

C. OPTIMIZATION BIN-PACKING

In general, the energy consuming function (13) is concave-downward, and a is more than 50% of φ_i. Therefore, the smaller k is, the better. This is a typical one-dimensional Bin-Packing problem with no split, so we can summarize the objective function as an integer linear programming model:

$$\begin{aligned} MIN(k) &= MIN(\sum_{i=1}^{|E|} n_i) \\ s.t. \ n_i &= 0 \ or \ 1, \\ r_{g_j, e_i} &= 0 \ or \ 1, \\ \sum_{j=1}^{|G|} l(g_j) r_{g_j, e_i} &\leq c_i n_i, \ i = 1, 2, ..., |E|, \\ \sum_{i=1}^{|E|} r_{g_j, e_i} &= 1, \ j = 1, 2, \ldots, |G|. \end{aligned} \tag{18}$$

In computational complexity theory, Bin-Packing problem is known to be combinatorial NP-hard [29]. Similar to the treatment for forwarding plan, we employ an approximation algorithm, the Best-Fit Decreasing (BFD) heuristic, to compute a feasible solution. In BFD, switches are first sorted in a non-increasing order by their weights (i.e., aggregated flow arrival rate $l(g_j)$), and then assigned to the selected controllers by this order. In each step of Best-Fit (BF) selection strategy, the partially filled controller with the smallest but sufficient residual capacity is selected to carry the current switch. If there is no alive controllers having enough capacity to handle the switch, then a new controller should be turned on. BFD heuristic is also a greedy-based algorithm, with the time complexity of $O(n \log n)$. The BF procedure is presented in Algorithm 2.

Since the numbers of switches (i.e., the objects packed into controllers/bins) is much smaller than the number of flows, the computational cost for control plane could be much less than the cost for forwarding plane. Therefore, an exact algorithm could still be used to solve the problem, such as MTP [30] or the one proposed in [31].

VI. SIMULATION AND ANALYSIS

In this section, we simulate our energy saving scheme using MATLAB simulator to validate the theoretical analysis.

A. DATA PREPARATION

To validate our models, we test the scheme in three typical DCN topologies: 2N-Tree, Fat-Tree, and BCube. We choose 16 as the number of hosts in our experiments, since it is suitable for all of three topologies' architectures. Given one

Algorithm 2 BF Procedure: BF Selection Strategy for Switch g_j

```
Relevant Note:
    The current number of alive controller is $K$.
    The original capacity of each controller is $C$.
    The residual capacity of controller $e_i$ is $RE_i$.
    For other notations refer to Table 3.

Step 1: Set $RE_i = C - \sum_{g \in W_i} l(g),\ i = 1, 2, ..., K$.
Step 2: If $RE_i - l(g_j) < 0,\ \forall i = 1, 2, ..., K$, there is no alive
    controller capable to carry $g_j$. Then turn on a new controller
    and pack $g_j$ into it, and the procedure is over. If not, go to
    next step.
Step 3: Set $I \leftarrow arg\ min_i(RE_i - l(g_j))$ (for $\{i | RE_i - l(g_j) \geq 0\}$). Pack $g_j$ into controller $e_I$, and the procedure is over.
```

controller controlling 2 core or 4 edge switches with peak flow arrival rate at least, the number of controllers is 5 in 2N-Tree and Fat-Tree and 4 in BCube. The redundancy should also be considered properly during the computation to ensure the reliability and QoS.

Poisson process has been commonly used for describing flow arrivals. We assume that traffic demand between each host obeys Poisson distribution with parameter $\lambda = 50$, the duration obeys exponential distribution with parameter $\lambda' = 1/5$, and the quantity for each demand obeys uniform distribution relative to link capacity.

In addition, a real DCN traffic dataset [32] is also analyzed in our simulation. We choose 16 nodes from Dataset UNV1 as our 16 hosts. According to [28], diurnal patterns exist in all data centers, and based on the dataset, the traffic data was recorded every 30 minutes. Thus, one day trace (11/01/2009) is used to test and validate the effectiveness of our models, and the time interval between two optimization is set to 30 minutes.

For the OpenFlow switches, we consider them as standard network devices and thus they would follow the model in (1) in Section II. Virtual switch such as OVS is not considered in this paper since they are software running on general servers. Table 4 summarizes the energy proportionality (x and y in the energy increment function (3)) for two types of switches (edge and core) we sampled.

TABLE 4. Energy proportionality of switches (48-port).

Energy Proportion	*Unit: Watts*	
	Edge Switch	Core Switch
One alive port: x	0.90	1.19
Idle Switch No Port On: y	135	176
Ports All On No Traffic	178	233
Ports All On Fully Loaded	183	239

For the controllers, we run them on servers with 2.4 GHz Intel Xeon CPU, 48 GB RAM and Ubuntu Server 14.04, then profile their energy model based on (2) in Section II. Table 5 summarizes the sample values of parameters (a, b, c and d) in the energy consuming function (13) of the controllers.

TABLE 5. Energy characteristics of controllers.

Parameters in Function (13)	Sample Values
a	155
b	345
c	-359
d	144

B. SIMULATION RESULTS

We run the simulation using MATLAB 2015b on a server with 3.1GHz Intel Core CPU, 16 GB RAM and Windows 8.1. The primary metric in the simulation is the *Energy Consumption Level*, which can be calculated by:

$$\frac{\textit{Power consumed with the optimization}}{\textit{Power consumed without the optimization}}. \tag{19}$$

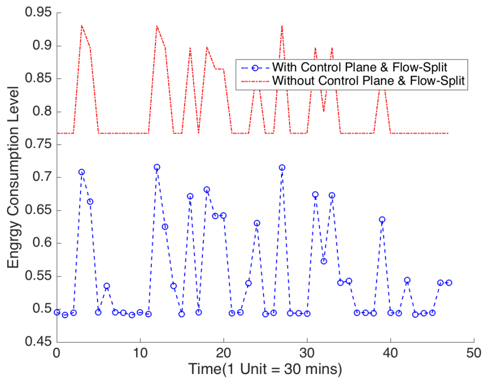
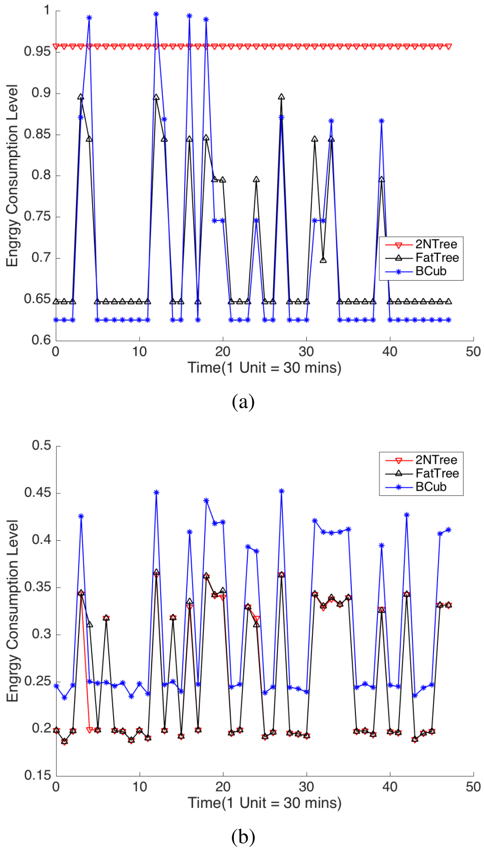


FIGURE 4. Optimization improvement (Poisson, Fat-Tree, 16 Hosts 48 Times).

1) EFFICIENCY OF POISSON PROCESS

The energy conservation effects of Poisson process in Fat-Tree DCN are shown in Figure 4. The top line represents the results of applying the model to forwarding plane, without considering the flow-split, where about 20% power saving can be observed. It is calculated based on the greedy routing model from [9], which is also utilized in [10] and [11]. The below line is the results of our model with flow-split, where the power saving can be further improved to 30-50%. The simulation results of Poisson process in different topologies are shown in Figure 5. We first test the energy saving ratio of forwarding plane (Figure 5(a)). We are able to identify that the energy cost by switches could be reduced by around 30%-40% in Fat-Tree and BCUbe. However, only about 5% powering saving can be observed in 2N-Tree. This is because there is only one core switch could be shut down in our simulation topology. Sometimes in BCube there's nearly no energy saving, because its forwarding function is mainly achieved by servers which cannot be dormant. This is also the



(a)

(b)

FIGURE 5. Results for poisson process in different topologies. (a) Forwarding Plane 16 Hosts 48 Times. (b) Control Plane 16 Hosts 48 Times.

reason why BCube may get pretty better results occasionally. The energy saving ratios of control plane are both acceptable (Figure 5(b)). The results of 2N-Tree and Fat-Tree have basically the same changing trend and range (also seen in Figure 7(b)). This is because they are both tree-like structures of three layers having the same Minimum Spanning Tree, and their workloads are mainly the flow arrival rate of the switches in our simulation. The results of BCube have different changing trend due to its server-centric network structure and different number of controllers.

2) EFFICIENCY OF REAL DCN DATA

The simulation results of real DCN data are shown in Figure 6 and 7. The energy conservation effects have the same characteristics as those for Poisson process, which demonstrates our model can be effective in a real multi-controller SDN-DCN environment. The energy saving ratio changes much more gently and the results are even better than Poisson process, because the DCN has been with high redundancy and its traffic is stable and highly aggregated.[4] In Figure 6, the energy saving ratio of our model is mainly 50%, and the other one is about 23%. In Figure 7(a), the values of Fat-Tree and BCube at the last two time slots are much higher than before, but this does not happen in Figure 7(b). This is because at these two time slots some switches need to be wakened due to a few specific flows, and the energy cost in forwarding plane is observably increased. However, the amount of these flows is relatively small, and thus they have little impact on the energy cost in control plane.

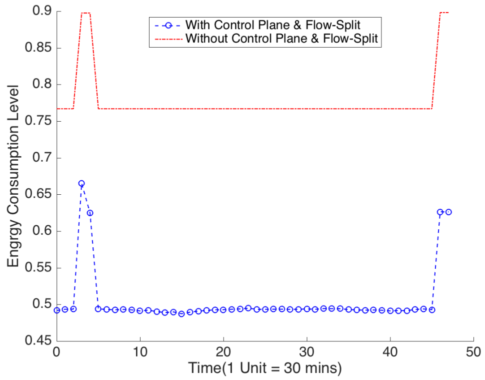
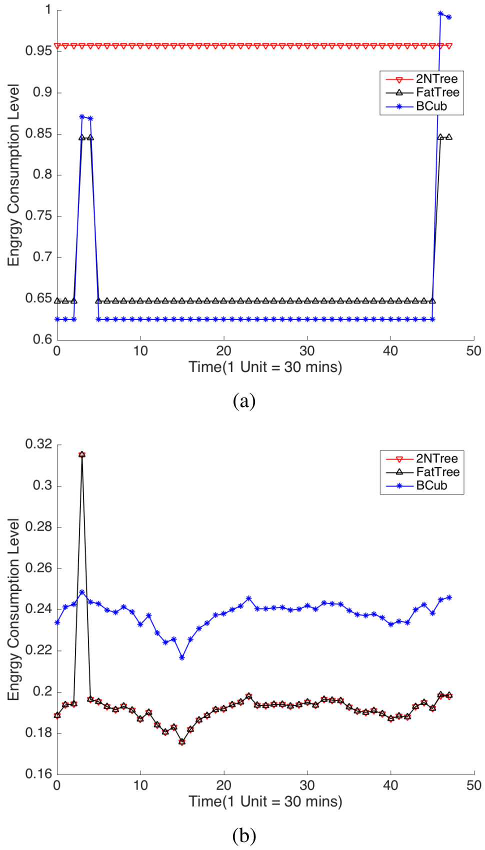


FIGURE 6. Optimization improvement (Real Data, Fat-Tree, 16 Hosts 48 Times).



(a)

(b)

FIGURE 7. Results for real DCN data in different topologies. (a) Forwarding plane 16 hosts 48 times. (b) Control plane 16 hosts 48 times.

C. COMPUTATION TIME AND PERFORMANCE OF GREEDY HEURISTIC

Poisson process is used here to simulate the network flows. E^{3}MC's computation time of an optimization is basically the

[4] Note that the spikes in Fig. 6 are caused by traffic burst from DCN trace.

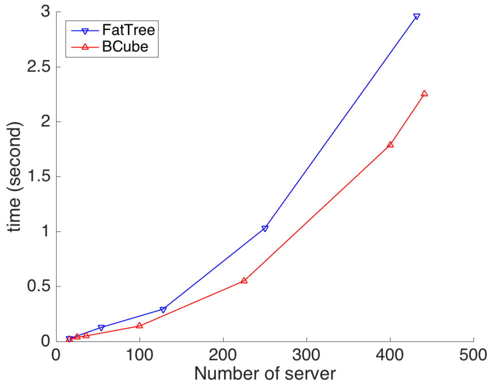


FIGURE 8. **Computation Time (Poisson, Fat-Tree and BCube).**

time cost for forwarding plane. The analysis of E^{3}MC's time performance for Fat-Tree and BCube in different network size is shown in Figure 8, plotted with servers number on x-axis and computation time on y-axis. As we explained in Section IV, in spite of its optimal solution (nearly), formal model's computation time is fairly high, especially for the large-scale DCNs. However, by leveraging our greedy heuristics, the cost has been significantly improved. Running in a server with 3.1GHz Intel Core CPU, 16 GB RAM, the optimization for the scale with 400 hosts only costs less than 2 seconds, which implies the feasibility of E^{3}MC's deployment to some extent in real DCN environment.

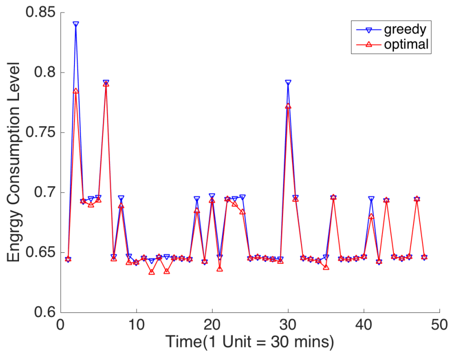


FIGURE 9. **Greedy VS optimal (Poisson, Fat-Tree).**

Figure 9 gives a comparison of the performances using greedy heuristic and formal model (MTP for control plane). We still use Poisson process to generate the flows and calculate 48 times as one day. Compare with formal model's optimal solution, the result of greedy heuristic is also reasonable. The energy efficiency using greedy heuristic is basically the same as the performance of the formal model. The two curves have almost the same changing trend and nearly overlap.

Note that in our simulation we do not refer to the latency and packet loss as primary performance metrics for analysis. Our scheme mainly focuses on the power saving and assumes that the network resources, e.g., the bandwidth, are sufficient to deal with the traffic. First, we assume that there are adequate controllers in control plane. More controllers can be turned on if more flows need to be processed. Thus, we do not consider the latency caused by overloaded controllers. On the other hand, in forwarding plane, we assume that the switches would not introduce the latency and packet loss due to the limit of capability of computation/forwarding. Finally, the non-shortest paths may be used due to the dormant switches and thus the additional latency may occur. However, this could be mitigated by selecting the path in a determined scope to avoid excessive hops, since the paths between two nodes in DCNs are generally fixed. Moreover, the computation latency and transmission delay of a package within one node are related to its length, which has no practical sense in our MATLAB simulation environment in the granularity of network flow.

VII. RELATED WORK

Our work aims to improve the energy efficiency in SDN-enable DCNs. Several prior efforts on power saving problems in data center environment have been carried out in the last few years. Most of them tackled the problems by flow routing policies, and SDN is also mentioned in a few of these solutions.

Heller *et al.* proposed the ElasticTree [9], which dynamically adjusts the set of active network elements to change the traffic in a data center of Fat-Tree topology. Shang *et al.* proposed Energy-Aware Routing [10], which also uses as few network devices as possible with no/little sacrifice on the network performance, and Fat-Tree and BCube are both simulated. Wang *et al.* proposed the CARPO [11], a power optimization algorithm which applies correlation analysis among flows and integrats traffic consolidation with link rate adaptation for maximized energy-saving. Rodrigues *et al.* presented an energy-efficient SDN emulation environment [12], and three power saving protocols can be emulated by operated at corresponding layers of the network. However, those proposed models, including [33]–[36], mainly play a role in forwarding plane, without the flow-split being considered. Energy-saving for SDN's control plane, specially in the multi-controller scenario, is not well-examined.

In fact, there have already been some resource-efficient efforts focused on control plane. ElastiCon by Dixit *et al.* [13] is a distributed controller architecture in which the controller pool is elastic according to traffic conditions. ElastiCon is more concerned with how to shift the load dynamically across controllers. Fu *et al.* [14] proposed a dormant multi-controller model using quantitative analysis to evaluate the performance of the multi-controller system. They also have not regarded energy consumption as the first consideration, and the power profile for controller is not combined in their model.

VIII. CONCLUSION

In this paper, we present E^{3}MC, an elastic multi-controller SDN energy-aware model which dynamically consolidates workloads onto a small set of devices and shut the redundant ones down to save power. In forwarding plane, routing policies are used to aggregate traffic flows and traffic/flow split is supported to further improve the efficiency. In control plane, E^{3}MC considered the energy saving for multi-controller SDN in large-scale DCN. E^{3}MC can be used in various topologies, but the results may vary in different network structures. We test our model both on Poisson process and real DCN data, and the results show the effectiveness of our scheme, especially in relatively structured topologies such as Fat-Tree and BCube, for saving energy by 40-50%.

With the optimizations of our model, the characteristics and patterns of flow demands could be analyzed for the foresighted elasticity. The redundancy rate could be added in our model thoroughly to ensure the availability and stability. For practical usability, the customized control protocols for managing the state of devices and the momentary energy consumption produced by the state changing action should also be considered.

In our current architecture, a stand-alone E^{3}MC server maintains the global view of the whole SDN-DCN, which may introduce single-point failure. The server's heavy computing load is also a limitation for the scalability. Thus, the E^{3}MC server could also be extended to work in a distributed way (e.g., deploying E^{3}MC daemon in each controller). We leave such improvement for future work.

ACKNOWLEDGMENTS

This paper is supported by International Science and Technology Cooperation and Exchange Project of China (2013DFE13130), ''the Fundamental Research Funds for the Central Universities'' and DNSLAB. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] *Clean Slate Design for the Internet*. (May 1, 2016). [Online]. Available: http://cleanslate.stanford.edu/

[2] *Ethane: A Protection Architecture for Enterprise Networks*. (May 1, 2016). [Online]. Available: http://yuba.stanford.edu/ethane/

[3] *Software-Defined Networking: The New Norm for Networks, ONF White Paper*. (May 1, 2016). [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf

[4] *OpenFlow Switch Specification, Version 1.5.1*. (May 1, 2016). [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.1.pdf

[5] A. Tootoonchian and Y. Ganjali, ''HyperFlow: A distributed control plane for OpenFlow,'' in *Proc. Internet Netw. Manage. Conf. Res. Enterprise Netw.*, 2010, p. 3.

[6] S. H. Yeganeh and Y. Ganjali, ''Kandoo: A framework for efficient and scalable offloading of control applications,'' in *Proc. 1st ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2012, pp. 19–24.

[7] *The ICTresearch*, (in Chinese). (Nov. 15, 2014). [Online]. Available: http://www.ictresearch.com.cn

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

[9] B. Heller *et al.*, ''ElasticTree: Saving energy in data center networks,'' in *Proc. USENIX NSDI*, 2010, p. 17.

[10] Y. Shang, D. Li, and M. Xu, ''Energy-aware routing in data center network,'' in *Proc. 1st ACM SIGCOMM Workshop Green Netw.*, 2010, pp. 1–7.

[11] X. Wang, X. Wang, K. Zheng, Y. Yao, and Q. Cao, ''Correlation-aware traffic consolidation for power optimization of data center networks,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 4, pp. 992–1006, Apr. 2015.

[12] B. B. Rodrigues, A. C. Riekstin, G. C. Januario, V. T. Nascimento, T. C. M. B. Carvalho, and C. Meirosu, ''GreenSDN: Bringing energy efficiency to an SDN emulation environment,'' in *Proc. 14th IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM))*, May 2015, pp. 948–953.

[13] A. Dixit, F. Hao, S. Mukherjee, T. V. Lakshman, and R. Kompella, ''ElastiCon: An elastic distributed SDN controller,'' in *Proc. 10th ACM/IEEE Symp. Archit. Netw. Commun. Syst. (ANCS)*, Oct. 2014, pp. 17–28.

[14] Y. Fu, J. Bi, J. Wu, Z. Chen, K. Wang, and M. Luo, ''A dormant multi-controller model for software defined networking,'' *China Commun.*, vol. 11, no. 3, pp. 45–55, 2014.

[15] M. Al-Fares, A. Loukissas, and A. Vahdat, ''A scalable, commodity data center network architecture,'' in *Proc. ACM SIGCOMM Conf.*, 2008, pp. 63–74.

[16] A. Greenberg *et al.*, ''VL2: A scalable and flexible data center network,'' in *Proc. ACM SIGCOMM Conf.*, 2009, pp. 51–62.

[17] 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. ACM SIGCOMM Conf.*, 2008, pp. 75–86.

[18] C. Guo *et al.*, ''BCube: A high performance, server-centric network architecture for modular data centers,'' in *Proc. ACM SIGCOMM Conf.*, 2009, pp. 63–74.

[19] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, ''A power benchmarking framework for network devices,'' in *Proc. IFIP Netw.*, 2009, pp. 795–808.

[20] A. Beloglazov, J. Abawajy, and R. Buyya, ''Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing,'' *Future Generat. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012.

[21] L. Luo, W. Wu, and F. Zhang, ''Energy modeling based on cloud data center,'' *J. Softw.*, vol. 25, no. 7, pp. 1371–1387, 2014.

[22] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.

[23] F. Giroire, D. Mazauric, J. Moulierac, and B. Onfroy, ''Minimizing routing energy consumption: From theoretical to practical results,'' in *Proc. IEEE/ACM Green Comput. Commun. (GreenCom)*, Dec. 2010, pp. 252–259.

[24] R. G. Busacker and T. L. Saaty, *Finite Graphs and Networks: An Introduction With Applications*. New York, NY, USA: McGraw-Hill, 1965.

[25] M. Klein, ''A primal method for minimal cost flows with applications to the assignment and transportation problems,'' *Manage. Sci.*, vol. 14, no. 3, pp. 205–220, 1967.

[26] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, ''Hedera: Dynamic flow scheduling for data center networks,'' in *Proc. NSDI*, 2010, p. 19.

[27] S. Kandula, D. Katabi, S. Sinha, and A. Berger, ''Dynamic load balancing without packet reordering,'' *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 2, pp. 51–62, 2007.

[28] T. Benson, A. Akella, and D. A. Maltz, ''Network traffic characteristics of data centers in the wild,'' in *Proc. ACM SIGCOMM Conf. Internet Meas. (IMC)*, 2010, pp. 267–280.

[29] S. A. Cook, ''The complexity of theorem-proving procedures,'' in *Proc. 3rd Annu. ACM Symp. Theory Comput. (STOC)*, 1971, pp. 151–158.

[30] S. Martello and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations*. New York, NY, USA: Wiley, 1990.

[31] R. E. Korf, ''An improved algorithm for optimal bin packing,'' in *Proc. 18th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2003, pp. 1252–1258.

[32] *Data Set for IMC 2010 Data Center Measurement*. (Mar. 20, 2015). [Online]. Available: http://pages.cs.wisc.edu/~/IMC10_Data.html

[33] R. Bolla, R. Bruschi, F. Davoli, and C. Lombardo, ''Fine-grained energy-efficient consolidation in SDN networks and devices,'' *IEEE Trans. Netw. Service Manage.*, vol. 12, no. 2, pp. 132–145, Jun. 2015.

[34] F. Giroire, J. Moulierac, and T. K. Phan, ''Optimizing rule placement in software-defined networks for energy-aware routing,'' in *Proc. IEEE GLOBECOM*, Dec. 2014, pp. 2523–2529.

[35] T. M. Nam, N. H. Thanh, N. Q. Thu, H. T. Hieu, and S. Covaci, ''Energy-aware routing based on power profile of devices in data center networks using SDN,'' in *Proc. 12th Int. Conf. Elect. Eng./Electron., Comput., Telecommun. Inf. Technol. (ECTI-CON)*, 2015, pp. 1–6.

[36] L. Prete, F. Farina, M. Campanella, and A. Biancini, ''Energy efficient minimum spanning tree in OpenFlow networks,'' in *Proc. Eur. Workshop Softw. Defined Netw. (EWSDN)*, 2012, pp. 36–41.

KUN XIE received the B.E. degree in network engineering and the master's degree in computer applications technology from North China Electric Power University, in 2007 and 2010, respectively. He is currently pursuing the Ph.D. degree with the Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing, China. His research interests lie on the computer networking system and next-generation network, including the software defined networking, network performance analysis, and power-saving of data center network.

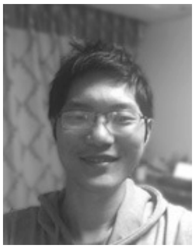
XIAOHONG HUANG received the B.E. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2000, and the Ph.D. degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2005. Since 2005, he has been with BUPT, where she is currently an Associate Professor and the Director of the Network and Information Center with the Institute of Network Technology, BUPT. She has authored over 50 academic papers in WDM optical networks, IP networks, and other related fields. Her current interests are performance analysis of computer networks and service classification.

SHUAI HAO received the B.S. degree in computer science from North China Electric Power University and the master's degree in computer science from the Beijing University of Posts and Telecommunications. He is currently pursuing the Ph.D. degree with the College of William and Mary, Williamsburg, USA. His research interests lie in the networking system and security, including the domain name system, internet measurement, and cloud security.

MAODE MA received the Ph.D. degree in computer science from The Hong Kong University of Science and Technology in 1999. He is an Associate Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He has extensive research interests, including wireless networking and network security, and has over 250 international academic publications, including over 110 journal papers and over 140 conference papers.

Dr. Ma is a fellow of the IET, and a Senior Member of the IEEE Communication Society and the IEEE Education Society. He is the Chair of the IEEE Education Society, Singapore Chapter. He is also an IEEE Communication Society Distinguished Lecturer. He currently serves as the Editor-in-Chief of the *International Journal of Computer and Communication Engineering* and the *International Journal of Electronic Transport*. He also serves as a Senior Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS and an Associate Editor for other four international academic journals.

PEI ZHANG received the Ph.D. degree from the Beijing University of Posts and Telecommunications in 2012. He holds a post-doctoral position with the Tsinghua National Laboratory for Information Science and Technology. His research interests include network measurement, management, and security.



DINGYUAN HU received the B.S. degree from the University of Science and Technology Beijing in 2015. He is currently pursuing the master's degree with the Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing, China. His research interests lie on the next-generation network and software-defined networking.