IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 2018, date of current version xxxx 00, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.DOI

Distributed Power Saving for Large-Scale Software-Defined Data Center Networks

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

[1]Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing, China (e-mail: pat@bupt.edu.cn)
[2]Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19716 USA (e-mail: haos@udel.edu)
[3]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore (e-mail: emdma@ntu.edu.sg)

Corresponding author: Xiaohong Huang (e-mail: huangxh@bupt.edu.cn).

This work has been supported by the Research Fund of Ministry of Education-China Mobile (MCM20160304).

ABSTRACT

Energy Efficiency in Data Center Networks (DCNs) is critical to the operations of modern large-scale Data Centers. One effective way is to make the size of DCNs elastic along with flow demands by centralized routing and scheduling, i.e., turning off idle network components to reduce the power consumption. As such, Software Defined Networking (SDN) is widely used for achieving such elasticity conveniently. Meanwhile, the scale and structure of modern DCNs get much larger and more complex. Central control and global computing become impractical due to the heavy time and space complexity. Therefore, distributed power control is necessary for large-scale Software-Defined DCNs (SDN-DCNs), and yet there are few research achievements in this area. In this paper, we present an extensible energy-efficient mechanism, which (1) leverages distributed flow routing for both intra- and inter-domain elephant flows and (2) extendedly considers distributed energy efficiency for control plane. A local power-saving function is operated within each domain of control plane, and a distributed energy-efficient routing algorithm is computed to optimize the effectiveness for the inter-domain flows. The simulation results demonstrate that this distributed mechanism applies to large-scale DCNs and achieves an effective power saving.

INDEX TERMS

Distributed Inter-Domain Routing, Energy Efficiency, Large-Scale Data Center Networks, Software Defined Networking.

I. INTRODUCTION

Large data centers are equipped with huge amount of electronic equipments, which are typically power-hungry for providing a variety of reliable services [1]. Therefore, energy consumption has become one of the crucial limitations for modern data center operations. Furthermore, as a vital component of data center, the network infrastructure has been observed for consuming a significant portion of total data center power (up to 10-20% [2]), and thus the energy efficiency in Data Center Networks (DCNs) has become a meaningful topic of research. Numerous power-saving schemes have been proposed, typically leveraging Energy-Efficient Routing (EER) strategy, such as ElasticTree [3], EAR [4], and the work presented in [5]. DCNs are typically designed with redundant components to guarantee system stability and reliability, and the utilization of bandwidth is relatively low most of the time. Thus, the key insight of EER is to utilize flow consolidation and bandwidth scheduling to select a subset of links and switches to transmit all the flows, while the idle devices could be put into dormant mode to reduce power consumption.

Software Defined Networking (SDN) is fundamentally functional to control DCNs with manageable and flexible flow strategies. With the decoupled control data planes and unified communication protocols (e.g., the OpenFlow [6]), the network intelligence is logically centralized within the SDN controller, and thus the network gains unprecedented programmability and automation [7], [8]. Also, with the topology and traffic loads obtained dynamically, SDN has flexible routing capability and straightforward controllability for DCN resources (the bandwidth, ports, switches, etc). It is convenient to operate DCN devices (e.g., put them to sleep

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2791630, IEEE Access

or wake them up) by SDN's customized control interfaces. There have been some studies focusing on the EER of Software Defined DCNs (SDN-DCNs), such as the work in [9], EXR [10], [11], Willow [12], GreenDCN [13], and CARPO [14], [15].

EER typically adopts the logically centralized routing and controlling pattern to achieve the global energy optimization. The management component maintains a global view of the whole DCN topology, and the path of each flow can be globally calculated on the premise of satisfying network performance and service quality. One of EER problems is that it is typically mathematically modeled with NP-hard complexity, and heuristic algorithms are generally adopted to meet the requirement of on-line operation. However, as the nodes number of DCN topography increases, such global centralization and optimization still need very strong computing capability.

Today, the data centers have been evolving towards extremely large-scale and complex structure. For example, in 2015, Microsoft's cloud infrastructure delivered more than 200 cloud-based services by over 1 million servers located in more than 100 global data centers [16]. Given a typical Fat-Tree data center hosting over ten thousand servers, there could be more than 1500 switches to support the underlying network. The overwhelming scale of DCNs makes the real-time centric control and global computing impractical due to the heavy time and space complexity in a SDN-enabled environment. The traditional centralized architectures usually cannot fulfill the EER expectation and thus the distributed power control will be required for such large-scale SDN-DCNs. Liu *et al.* proposed a Distributed Flow Scheduling (DFS) model in energy-aware SDN-DCNs [17]. With the DFS, suitable paths are calculated to load *elephant* flows (i.e., the flows transferring significant amount of data) by distributed schedulers, achieving effective bandwidth utilization and efficient flow management. Fernández-Fernández *et al.* proposed DEAR [18], a novel distributed routing algorithm that leverages performance constraints to optimize the power consumption in large-scale SDN with multiple domains.

The routing strategies for inter-domain flows in both DFS and DEAR are simply based on the link utilization, without considering systematic model of optimized energy consumption in routing decisions. The multi-controller multi-domain SDN structure is also simple (one domain contains just one controller), and thus the power saving for control plane cannot be included. In fact, the structure of distributed control plane can be more complex in modern large data centers, such as Orion [19]. The hierarchical control plane may require multiple controllers in one domain with heterogeneous multi-controller structures. As such, to fill this gap and explore energy optimization for large-scale SDN-DCNs, in this paper we propose a distributed energy-efficient mechanism leveraging both intra- and inter-domain flow routing in multi-domain SDN where one domain may contain multiple controllers. In each domain, local power-saving function (E^3MC [20]) is operated distributedly, including the savings for control plane. For inter-domain flows, a partly distributed EER algorithm is computed to optimize the effectiveness.

Specifically, the optimization model is combined with the characteristics of energy consumption of network devices, while the SDN-enabled multi-path routing and bandwidth scheduling are integrated. A set of devices are distributedly selected to be kept alive, while other idle devices can be put into dormant mode via SDN's management interfaces. There are no topology limits to employ this mechanism for modern DCNs. The simulations are conducted based on Poisson process and real traces of data center traffic, and the results show that this distributed energy-saving scheme is efficient for large-scale SDN-DCN with multi-domain and multi-controller.

The main contributions are summarized as follows.

- This is the first study discussing distributed power saving for large-scale software-defined DCN with both data and control planes considered. A concrete mathematical model is built based on energy consumption characteristics with satisfying energy optimization effectiveness.
- In each domain, EER for intra-domain flows is processed distributedly, and control plane power saving is supported with complex and flexible multi-controller structure.
- EER for inter-domain flows is processed for global optimization, and a partly distributed calculation is presented to greatly reduce its processing time, with barely the same power-saving effect than the whole centric approach.
- The concept of "logical link" is proposed for distributed flow routing, with which the large-scale DCN is significantly simplified to accommodate quick and efficient energy optimization.

The remainder of this paper is organized as follows. Section II provides a background on DCN topologies, energy characteristics of network devices, and multi-controller SDN. Section III presents the concept of distributed power optimization of SDN-DCNs, and then describes the multi-domain SDN structure and schematic power-saving architecture. Section IV builds the mathematical model of the distributed mechanism. Section V evaluates the simulated results. Finally, Section VI concludes the paper.

II. BACKGROUND

In this section we provide background on the DCN topologies and the power consumption characteristics of SDN devices. We then introduce the multi-controller SDN paradigm. Note that part of the content has already been presented in our previous work [20], and is briefly reviewed here to make the following sections easier to understand.

A. DATA CENTER NETWORK TOPOLOGIES

Large-scale clusters are increasingly deployed in modern data centers. The computing, analysis, and warehousing applications start to get complicated and diversified. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2791630, IEEE Access

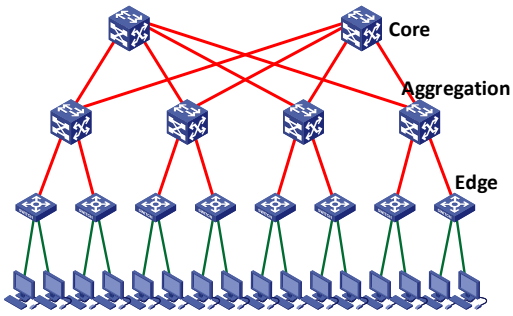

FIGURE 1. The Structure of Traditional 2N-Tree Topology.

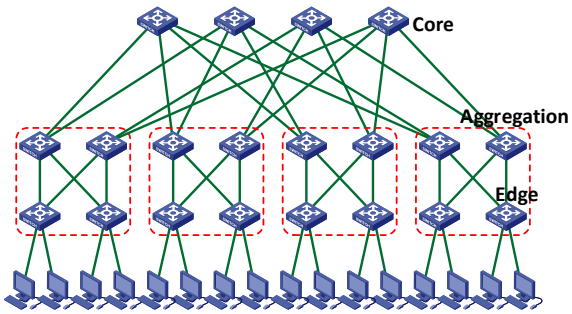

FIGURE 2. The Structure of Fat-Tree Topology with 4 pods.

network services are commensurately getting more important and large-scale. Huge network throughput becomes necessary for significant inter-node communication requirements. Since network demands always scale up to traditional DCN capacity limits, numerous new topologies have been proposed, such as Fat-Tree [21], VL2 [22], DCell [23], BCube [24], etc.

The traditional DCN structures are generally not richly-connected. Fig. 1 shows a typical example of traditional DCN topology, 2N-Tree. From the view point of power saving, making one core switch dormant in 2N-Tree will cut the effective bandwidth in half, and making two switches dormant is not permitted due to the disconnection between servers. As for Fat-Tree and BCube, they have more capabilities in bandwidth and switching paths. Hence more dormant devices can be tolerated, which implies more power can be saved.

Fig. 2 shows an example of Fat-Tree topology, which is tree-like and consists of three layers: edge layer, aggregation layer, and core layer. This 4-pod Fat-Tree is constructed by 1G links and commercial Ethernet GigE switches with 4 linked ports. The scale of Fat-Tree can be easily increased according to its respective structural features. Note that the distributed mechanism in this paper would be suitable for any kind of DCN topologies, including unstructured and irregular topologies. The most common Fat-Tree is simply adopted as the representative large-scale topology in the simulation.

B. DEVICES ENERGY CONSUMPTION CHARACTERISTICS

Here the power consumption of switches and controllers in SDN is profiled, respectively.

The energy consumption of traditional switches is load-insensitive [25], [26]. Such feature is the theoretical basis of the power saving for data plane. Before a switch is put into dormancy, the flows it carries will be rerouted to other alive switches first. The loads of these alive switches will be correspondingly more heavy, but their energy consumption will not be proportionally increased.

The power consumption of switch mainly consists of the costs by chassis, line cards, and ports. The vast majority of the power is consumed to keep the hardware alive. The chassis part and line cards part are fixed and consume the biggest portion (up to 135 Watts together in the preliminary experiment). The alive ports cost a small but notable part. Ports with different line rate configurations may consume different amounts of power (1-2 Watts along with their different line rate configurations in the preliminary experiment). Traffic load in the links going from 0 to the full capacity only increases the power by less than 5%. The influence by the network bandwidth load can be considered negligible.

To make things simple in the following simulation, the power cost by traffic load is ignored, and the number of line cards in each switch is assumed to be 1. According to [25], the power consumption can be summarized by this linear model:

$$Power_{Switch} = Power_{Idle} + \sum_{i=1}^{Configs} Number_i \times Power_i \quad (1)$$

The power consumed by one idle switch with no alive ports is $Power_{Idle}$. The number of line rate configurations of ports is $Configs$. $Number_i$ is the number of ports running at line rate i, with $Power_i$ as the power consumption.

Controller is an application running atop on a specific-purpose server to maintain the network and achieve network functionalities. Therefore, the power profile of controller would follow the power model of data center server. Unlike the switches, energy consumed by controller's workload is a significant part that cannot be ignored [27], [28]. Since CPU is typically the throughput bottleneck in a controller, the energy consumption model is established based on CPU utilization, and can be summarized by a non-linear regression model [28]:

$$Power_{Controller} = Power_{Idle'} + \rho_1 \times Util + \rho_2 \times Util^2 + \rho_3 \times Util^3 \quad (2)$$

$Power_{Idle'}$ is the fixed power consumed by a controller with no workload. $Util$ is the scaling factor of controller's CPU utilization, and ρ_{1-3} are the empirical correction impact factors measured on sample machines. This model is generally a concave-downward function and $Power_{Idle'}$ always consumes more than 50% power in fully loaded case. This

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2791630, IEEE Access

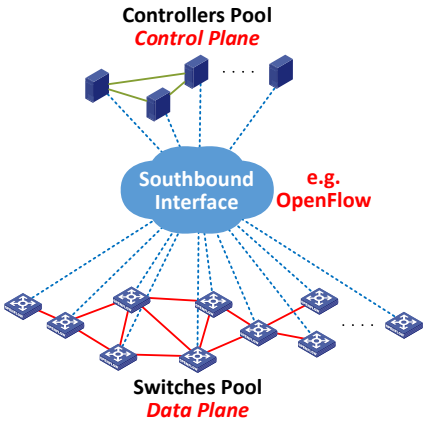

FIGURE 3. Out-of-Band Multi-Controller SDN Structure.

implies that the power can be saved by increasing the utilization and making redundant controllers dormant.

In the preliminary experiment (see Table 2 and 3 in Section V-A), a commercial switch and a data center server running controller application are sampled, which also confirms the credibility of these two energy consumption models.

C. MULTI-CONTROLLER SOFTWARE DEFINED NETWORKS

With the rapidly scaled DCNs, SDN with one single controller cannot support the operation of the entire large-scale network. A more dynamic and scalable control plane with multiple controllers is critical for modern data centers to provide reliable services. A simple out-of-band physical structure of multi-controller SDN is shown in Fig. 3.

There is no standardized architecture of controller pool for multi-controller SDN. The controllers can run on identical mode, like HyperFlow [29]. All the controllers have a global view and run as if they were controlling the whole network. The controllers can run on hierarchical mode, like Kandoo [30]. The underlying local controllers maintain a part view and run local applications, and the top logically centralized root controller maintains a global view and runs non-local control applications. The controllers can run on distributed mode, like WE-bridge [31]. All the controllers are equal in status and have a part view, and there is no root controller running on top of them. The flexible distributed multi-controller structure of this work will be described in Section III-C.

III. CONCEPT DESIGN OF DISTRIBUTED OPTIMIZATION

In this section we review the general heuristic model of the EER in DCN, and then present the idea of the distributed power optimization. Next, we introduce a simple multi-domain SDN-DCN structure with multiple controllers to explain the environment in the simulation, and then describe the modules and process of the distributed mechanism.

A. HEURISTIC EER MODEL WITH SDN

In an OpenFlow-enabled network, traffic between two nodes can be split in the level of flow, which is controlled by controller directly and more fine-grained. In a richly-connected DCN topology, there are generally multiple paths that can be selected to transmit one flow. Dynamic multi-path flow routing with SDN should be more efficient than that with a traditional DCN structure [32]. Moreover, with traffic splitting, the bandwidth utilization can be further improved.[1] For EER, the traffic demands will be loaded on fewer switches to achieve a better energy optimization with still adequate bandwidth.

General EER problem is always modeled as a Multi-Commodity Flow (MCF) Formulation [34], with flow routing matrix and alive switch/port subset as the optimal result. When a flow is loaded on a link, the status of related devices may be changed and there may be a power increment as the cost of the flow. Different from standard MCF model, the power/cost increment is discrete and is irrelevant to the bandwidth demand. It is a NP-Complete mixed-integer linear program (MILP) with heavy computational requirements [35]. According to [3], the solution time is about $O(n^{3.5})$, where n is the number of hosts, and thus the DCN can only scale up to less than 1000 hosts. For availability, most of the previous studies adopt Greedy Heuristic to decrease the computational complexity [3], [4], [9], [12], [15].

The process greedily assigns as many flows as possible to the path with the lowest energy consumption. More specifically, the flows are assigned in an iterative manner. At each iteration, the path bringing minimum energy consumption with adequate capacity is selected to bear one flow. The residual network after this assignment will be regarded as a new network for the next flow in next iteration. According to [3], the solution time of this general greedy heuristic is about $O(n^{2.5})$, and thus the DCN can scale up to less than 7000 hosts. As a result, the common used centralized greedy algorithm becomes impractical in large-scale DCNs with more than 10000 hosts. A distributed mechanism is designed to make the large-scale power saving feasible, including distributed EER for data plane and distributed energy optimization for control plane.

B. DISTRIBUTED POWER OPTIMIZATION

The DCN topology is usually structured and well-organized, and the paths between any two nodes are always fixed and predictable. Thus the set of all the alternative paths between every two hosts can be pre-computed, even in a large-scale topology. In a multi-domain DCN topology, the flows are divided into two groups: intra-domain flows (with source and destination in the same domain) and inter-domain flows (with source and destination in different domains). Objectively speaking, a flow is possible to be forwarded out of a domain and then transfer back to the same domain to save power. This

[1] Splitting flows is typically unpractical because of the harmful impact on TCP packet reordering [33], and is not considered in this paper.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2791630, IEEE Access

may cause a terrible time delay since it may be transmitted by lots of nodes crossing multiple domains. Although the energy is saved a little, a good service quality cannot always be maintained. Therefore, this work assumes that if a flow passed through a domain, it would only pass through this domain once. Here this hypothesis is called the One Time Passing (OTP) restriction. The OTP also makes sure that an intra-domain flow will not go out of its domain. Then the size of the path set of each flow will be further reduced.

The distributed EER assigns these flows also in an iterative manner, with the traffic load and energy consumption together considered as two factors. The difference is that these two kinds of flows will be settled in two steps respectively. First, the intra-domain flows are iteratively assigned within the scope of their domains. The process is like the general centralized EER model, and is distributedly handled in each local optimizer in parallel. Second, the inter-domain flows will then be iteratively routed on the residual network of the first step.

For an inter-domain flow, its candidate paths pass through multiple domains, and are composed of intra-domain links (connecting two nodes in one domain) and inter-domain links (connecting two border nodes in two domains). Therefore, the routing for this flow also has two parts/steps.

First, the intra-domain EER part will be distributedly handled in each related domain in parallel. In one specific domain, there may be multiple candidate routing schemes, since the flow can be transmitted through different ingress border node and egress border node, i.e. border node pairs. Each candidate path between a border node pair corresponds to different energy consumption. In a global view, the optimal intra-domain path between a border node pair can be abstracted as a logical link with one hop, and the energy consumption increment is its cost/weight. All the possible logical links in each domain, as well as their costs/weights, are distributedly calculated by the local optimizer. Then the global large-scale DCN topology can be greatly simplified, since it is only composed of the logical links, inter-domain links, and related border nodes.

An example of the topology simplification for an inter-domain flow is shown in Fig. 4. There are four domains, and a source host in Domain_1 is transmitting a flow to the target host in Domain_4. The topology is simplified with a handful of links and nodes. The solid lines are the physical links used for inter-domain transfer. The dash lines are the expected logical links used for intra-domain transfer.

Second, the inter-domain EER part will be globally handled. For the inter-domain flow, there may be multiple candidate logical links in a domain (e.g. the two logical links in Domain_1 in Fig. 4). The logical links in different domains and the physical inter-domain links can form multiple candidate complete paths. Consequently, the consumed energy may be different when different path is chosen. According to the simplified DCN topology and simplified candidate paths, the global inter-domain EER can be quickly calculated. The complexity of time and space is greatly reduced, and this

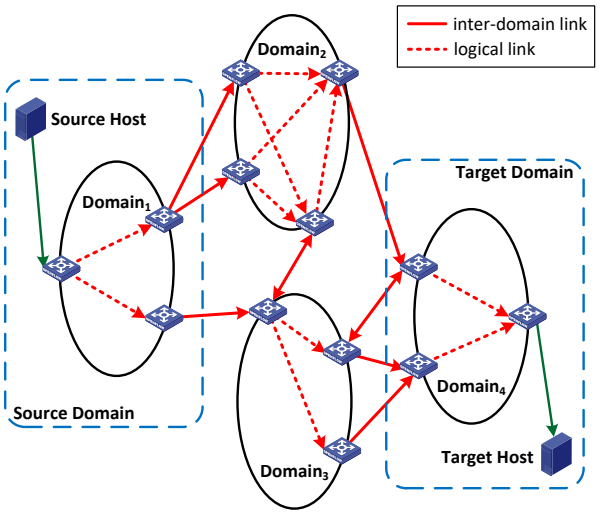

FIGURE 4. An Example of Topology Simplification for an Inter-Domain Flow.

global energy optimization can be achieved with an acceptable level of performance. In summary, during the overall EER for one inter-domain flow, the intra-domain part is firstly handled within each domain in parallel, and then the inter-domain part can also be globally handled with acceptable QoS.

Note that in practice, it is possible that the domains have different numbers of devices. In this work, the data plane energy optimization mainly focuses on the distributed routing mechanism for intra- and inter-domain flows. The total number of switches in each domain is not a computing factor during the routing calculation (see Section IV-A), and these numbers have no effect on this distributed EER. That is, the domain partition strategy is not a main concern. In this paper, Fat-Tree topology is adopted for simulation, which is a classical DCN topology with highly structured connection. Therefore, the domains are equally divided for simplicity (Fig. 8 in Section V-B). However, the distributed EER model can be easily extended to the multi-domain DCN with nonuniform domain scales.

After dealing with data plane, the distributed power optimization for control plane can be executed. This process is also handled in each domain locally and concurrently. The load of controller is estimated by the OpenFlow message arrival rate, and controller throughput can be denoted by its flow processing rate. This gives the controller's utilization as the ratio of current flow processing rate to its maximum capacity, which is the measured rate when its CPU usage went to 100%. Then, the power characteristics of controllers can be employed to build the energy optimization of control plane. In this case, E^3MC [20] can be leveraged to tackle this local multi-controller power optimization, and identical mode of controllers can also be adopted for simplicity of simulation.

To further reduce the distributed EER complexity, only the elephant flows are scheduled in the data plane power saving

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2791630, IEEE Access

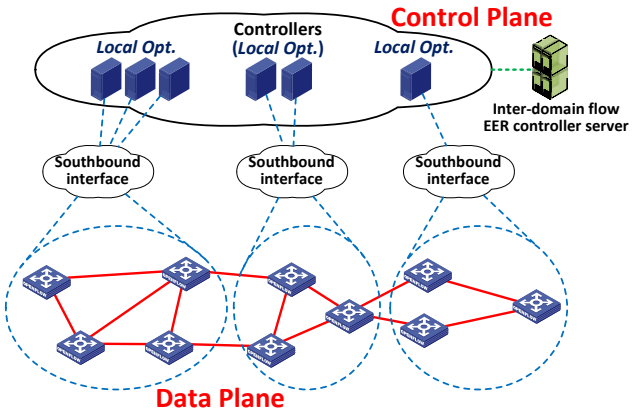

FIGURE 5. Distributed SDN-DCN Structure.

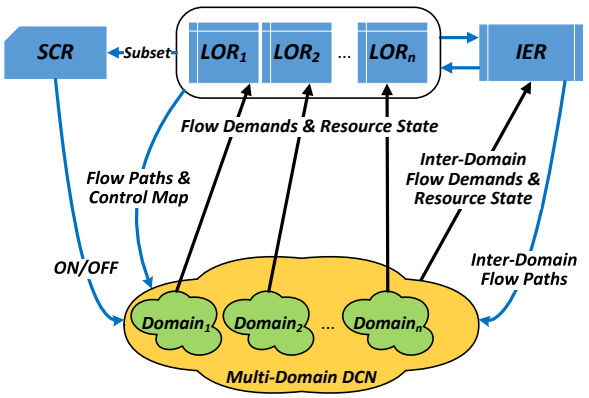

FIGURE 6. Modules Diagram.

case. The elephant flows, which account less than 10% of the total number of flows in DCN, are the main components of the total traffic throughput [36]. As for mice flows, the alive paths assigned to elephant flows can be reused to transfer them. The power saving for control plane is executed distributedly and independently, and all types of flows are considered.

C. DISTRIBUTED SDN-DCN STRUCTURE

In the distributed power saving, the switch pool and controller pool grow and shrink dynamically along with the change of network traffic demands. The inter-domain flows are also energy-efficiently routed, the minimum set of inter-domain border nodes keeps alive, and a global optimization can be achieved. The multi-domain SDN-DCN structure will support such elasticity conveniently.

The south-bound interface channels are established to transmit messages between SDN controller and switches. Here such connection is called the "control network". The connection between controller and switches should be efficient and stable. In an *In-Band* SDN-DCN, network elasticity may cause unexpected disconnection or communication delays between controller and switches. An *Out-of-Band* control structure is easy to deploy but need extra independent control network. To make things simple and easy to understand, the connection approach of the controllers is not discussed, and the *Out-of-Band* structure is adopted in the simulation.

The simplified multi-domain SDN-DCN structure is shown in Fig. 5. The whole large-scale topology is logically or physically divided into several (not many) domains, and each domain can be controlled by multiple controllers (at least one). Different multi-controller structures can be flexibly adopted to achieve local energy optimization functions for the domains with multiple controllers. The local optimizer can be an upper-layer device as the E^{3}MC controller server. If a domain is small enough to be managed by one single controller, the EER service can just be deployed on the controller, and the energy efficiency for control plane can be omitted. The whole large-scale SDN-DCN is surely a multi-controller structure, and the controller(s) in different domain maintains a domain view of the local network.

In Fig. 5 there is a standalone controller server, which runs the inter-domain flow EER functions to calculate the minimum set of inter-domain border nodes. Although this server seems like a root controller, it need not to maintain the detailed global view of the large-scale topography. The connecting relation between the domains is certainly maintained, yet the intra-domain connection is greatly simplified. In this figure the link of this server is dashed, because it can be deployed either on a standalone server or on each local E^{3}MC server that initiates inter-domain flow transfers. The standalone approach is suggested for simplicity.

The *Out-of-Band* control network can also provide intra- and inter-domain communication functions between all the controllers. Local controller servers run local EER functions and provide necessary information of simplified intra-domain routings (logical links) to the standalone inter-domain controller server. This inter-domain controller server gathers all information of logical links to guide inter-domain flow routing, and then provides border nodes selection information to corresponding local controller servers.

D. MODULES AND PROCESS ARCHITECTURE

The modular architecture of the distributed mechanism is shown in Fig. 6. The system is running on the SDN environment, and its network management for power saving is driven by OpenFlow protocol. The system consists of several logical modules: Local Optimizers (LORs), Inter-domain EER (IER), and State Converter (SCR). When new elephant flow demands arrive at the data plane in each domain, the triggered switches will update the information of the intra- and inter-domain flow demands to each LOR, and the inter-domain flow demands to IER. First, each LOR runs local EER in E^{3}MC server based on the intra-domain demands. Then each related LOR calculates the logical links with corresponding weights for the inter-domain demands, and gives the results to IER as inputs. Based on the inter-domain link status, flow demands, and the logical links, IER

searches an energy-optimized routing scheme for the inter-domain elephant flow, and gives the results to corresponding LOR. Each LOR calculates the subset of the devices in each domain, and gives the results to SCR. The intra- and inter-domain flows will be transmitted according to the energy-efficient paths, then SCR changes the operating status of ports and switches. After the optimization for the data plane, LORs run local E^{3}MC for control plane if there are multiple controllers in their domains, and mice flows are also taken into account. The process for control plane is not shown in Fig. 6 for simplicity.

The frequency of DCN elastic change doesn't need to be too rapid, since the device status changing action also costs power slightly, and the real-time performance is not a high requirement. The sorting order of the status changing actions is important. If a switch/port needs to be put into dormancy, the action will be executed when there is no existing flow passing through the switch/port and its incoming flows have already been rerouted to the new paths. If a switch/port needs to be awakened, the action will be executed beforehand, then the new switching/routing rules can be deployed to process the incoming flows. In such order, the network QoS will not be affected during the status changing of devices.

IV. MATHEMATICAL MODEL FORMULATION

In this section, we build the formal mathematical model to deal with the multi-domain power optimization, and then design a distributed EER heuristic algorithm to search the acceptable optimal solution, including the detailed models for intra- and inter-domain flows, respectively. Next, we briefly present the whole heuristic process.

A. MATHEMATICAL OBJECTIVE

The problem is formulated by such programming: There is bidirectional weighted multi-domain flow network $N(V, L, D)$, which represents the full duplex DCN, having the nodes set V, links set L and domains set D. All the correlative notations are summarized in Table 1, and some of the presentations can be analogized into multi-domain environment. Note that if the device (port or switch) associated with l is sleep, c_l is assumed to be 0. This formal distributed problem can be formulated as a Mixed-Integer Linear Programming (MILP) model. It is still an MCF formulation, with the optimal energy consumption cost by alive switches and ports as the result.

When a flow is loaded on l_{ij}, the related device status may be changed and the power increment after the flow-loading action is $\varphi_{l_{ij}}$. According to (1), the energy increment function is established, where the conditions for α and β describe the original device status:

$$\varphi_{l_{ij}} = \begin{cases} 0 & \alpha_{l_{ij}} = 1, \beta_{v_i} = \beta_{v_j} = 1, \\ 2x & \alpha_{l_{ij}} = 0, \beta_{v_i} = \beta_{v_j} = 1, \\ 2x + y & \alpha_{l_{ij}} = 0, \beta_{v_i} \oplus \beta_{v_j} = 1, \\ 2x + 2y & \alpha_{l_{ij}} = 0, \beta_{v_i} = \beta_{v_j} = 0. \end{cases} \tag{3}$$

TABLE 1. Summary of Notations for Formal Mathematical Model

Notation	Description
V	Set of nodes (switches): $v_i \in V$ $(i = 1, 2, ..., \|V\|)$.
L	Set of physical links (two ports): $l_{ij} \in L$ connects v_i and v_j, $(v_i, v_j \in V, i \neq j)$, $L = L^D \cup E$.
D	Set of domains: $d_i \in D$ $(i = 1, 2, ..., \|D\|)$.
V^d	Nodes set of d: $v_i^d \in V^d$ $(i = 1, 2, ..., \|V^d\|, d \in D)$.
B	Set of border nodes: $b_i \in B$ $(i = 1, 2, ..., \|B\|)$.
L^d	Set of intra-domain links of d: $l_{ij}^d \in L^d$ connects $v_i^d, v_j^d, (v_i^d, v_j^d \in V^d, i \neq j)$.
E	Set of inter-domain links: $e_{ij} \in E$ connects b_i, b_j, $b_i, b_j \in B$ belong to different domains.
W_v	Set of nodes linked to v.
c_l	Non-negative and real-valued bandwidth capacity of l.
F	Set of flows: $f = (s_f, t_f, q_f) \in F$. s_f: source, t_f: sink, q_f: demand.
F^d	Set of intra-domain flows of d: $f^d \in F^d$.
F^N	Set of inter-domain flows: $f^N \in F^N$.
$E_{f^N}^d$	Set of logical links of f^N in d: $e_{f^N}^d \in E_{f^N}^d$ connects $b_i, b_j \in (B \cap V^d)$.
P_f	Set of all the alternative paths for f: $p_f \in P_f$.
γ_{p_f}	Binary decision indicating whether f is loaded on p_f.
$\zeta_{l,f}$	Binary decision indicating whether f is loaded on l.
u_l	Utilization of l: $u_l = (\sum_{f \in F} q_f \zeta_{l,f})/c_l$.
α_l	Binary decision α_l indicating whether l is alive.
β_v	Binary decision β_v indicating whether v is alive.
φ_l	Cost of l: Consumed power when a flow is loaded on l.
φ_p	Cost of p: Consumed power when a flow is loaded on p.
φ_f	Cost of f: Consumed power when f is assigned.
$\varphi_{e_{f^N}^d}$	Cost of $e_{f^N}^d$: Consumed power when $e_{f^N}^d$ is loaded.
ϕ	Energy consumed by the current DCN.
ϕ^d	Energy consumed by the current local domain d.
$p_{f,opt}$	The path of f with lowest power consumption increment.
$\varphi_{f,opt}$	Consumed power when f is loaded on $p_{f,opt}$.
x	Energy consumed by an alive port.
y	Energy consumed by an idle switch (no alive port).

When f is loaded on p_f, the energy increment function of a path can also be established, where l belongs to p_f.

$$\varphi_{p_f} = \sum_{l \in p_f} \varphi_l \tag{4}$$

Then the general objective function can be established to minimize the switches' power consumption based on (3), where α and β describe the status after the transition:

$$\min \phi = \min \left(\sum_{l \in L} \varphi_l\right) = \min \left(2 \sum_{l \in L} \alpha_l x + \sum_{v \in V} \beta_v y\right). \tag{5}$$

Function (5) is a global objective function, and can be rewritten as a multi-domain objective function:

$$\begin{aligned} \min \phi &= \min \left(\sum_{d \in D} \phi^d\right) = \min \left(\sum_{d \in D} \sum_{l^d \in L^d} \varphi_{l^d} + \sum_{e \in E} \varphi_e\right) \\ &= \min \left(\left(\sum_{d \in D} \sum_{l^d \in L^d} \alpha_{l^d} + \sum_{e \in E} \alpha_e\right) 2x + \sum_{d \in D} \sum_{v^d \in V^d} \beta_{v^d} y\right). \end{aligned} \tag{6}$$

The following constraints should be satisfied:

- *Capacity constraint*: The total load of each link must not exceed link capacity (avoid overload and congestion).

$$0 \leq \sum_{f \in F} q_f \zeta_{l,f} \leq c_l, \ i.e., 0 \leq u_l \leq 1 \tag{7}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2791630, IEEE Access

- *Bundling of flow and path*: Each flow is assigned to exactly one path.

$$\forall f \in F, \sum_{p_f \in P_f} \gamma_{p_f} = 1 \tag{8}$$

- *State association of link (port) and switch*: When one port on a switch is alive, the switch is alive. When the ports of a switch are all dormant, the switch will be dormant.

$$\forall v_i \in V, \ \beta_{v_i} = \begin{cases} 1, & \sum_{v_j \in W_{v_i}} \alpha_{l_{ij}} > 0, \\ 0, & \sum_{v_j \in W_{v_i}} \alpha_{l_{ij}} = 0. \end{cases} \tag{9}$$

This multi-domain energy optimization is still an NP-Complete MILP, but the scalability of generally used centralized Greedy algorithm is not good enough. A distributed EER heuristic algorithm is designed to fit large-scale using, with two steps of iterative optimization for intra- and inter-domain flows, respectively.

B. DISTRIBUTED MODEL FOR INTRA-DOMAIN FLOWS

The intra-domain flows are first to be dealt with, and the process is taken in each related local optimizer in parallel. In the domain view, it is a standard local EER approach tackled by Greedy heuristic in an iteration manner.

In domain d, all the $f^d \in F^d$ are greedily assigned to $p_{f^d,opt}$ one after another. $p_{f^d,opt}$ is the path of f^d with lowest energy consumption increment $\varphi_{f^d,opt}$ based on the intra-domain residual network of the last flow. The process can be expressed by the following formula:

$$\begin{cases} \phi_0^d = 0, \\ \phi_n^d = \phi_{n-1}^d + \varphi_{f_n^d,opt}, \ n = (1, 2, ..., |F^d|). \end{cases} \tag{10}$$

After $n-1$ intra-domain flows are routed by $n-1$ iterations, f_n^d is assigned to the residual network of f_{n-1}^d, and minimal energy $\varphi_{f_n^d,opt}$ is consumed. ϕ_n^d is the current optimal solution satisfying n intra-domain flow demands, and $\phi_{|F^d|}^d$ is the optimized energy consumption of d with F^d all assigned.

The path $p_{f^d,opt}$ for f^d that cost minimal energy $\varphi_{f^d,opt}$ is selected by the following function:

$$\varphi_{f^d,opt} = \min_{p_{f^d} \in P_{f^d}} \varphi_{p_{f^d}} = \min_{p_{f^d} \in P_{f^d}} \Big(\sum_{l^d \in p_{f^d}} \varphi_{l^d} \Big). \tag{11}$$

φ_{l^d} can be calculated by (3). The current energy consumption increment of the whole DCN is $\sum_{d \in D} \phi_{|F^d|}^d$.

C. DISTRIBUTED MODEL FOR INTER-DOMAIN FLOWS

The inter-domain flows are second to be dealt with, and also routed in a greedy and iterative manner. In the whole DCN view, it is a global EER approach but with a distributed topology simplification.

All the $f^N \in F^N$ are greedily assigned to $p_{f^N,opt}$ one after another. $p_{f^N,opt}$ is the path of f^N with lowest energy consumption increment $\varphi_{f^N,opt}$ based on the inter-domain residual network of the last flow. The process can be expressed by the following formula:

$$\begin{cases} \phi_0^N = 0, \\ \phi_n^N = \phi_{n-1}^N + \varphi_{f_n^N,opt}, \ n = (1, 2, ..., |F^N|). \end{cases} \tag{12}$$

After $n-1$ inter-domain flows are routed by $n-1$ iterations, f_n^N is assigned to the residual network of f_{n-1}^N, and minimal energy $\varphi_{f_n^N,opt}$ is consumed. ϕ_n^N is the current optimal solution satisfying n inter-domain flow demands, and $\phi_{|F^N|}^N$ is the optimized energy consumption with F^N all assigned.

The path $p_{f^N,opt}$ for f^N that cost minimal energy $\varphi_{f^N,opt}$ is selected by the following function:

$$\varphi_{f^N,opt} = \min_{p_{f^N} \in P_{f^N}} \varphi_{p_{f^N}}. \tag{13}$$

Every path $p_{f^N} \in P_{f^N}$ is comprised of related inter-domain links e and intra-domain logical links $e_{f^N}^d$, where d indicates all the related domains. φ_e can be calculated by (3). The intra-domain routing scheme of each $e_{f^N}^d$ in p_{f^N} with power weight $\varphi_{e_{f^N}^d}$ needs to be distributedly calculated in each related domain by the local optimizer concurrently. When every e and $e_{f^N}^d$ in p_{f^N} gets a cost/weight, $\varphi_{p_{f^N}}$, as the cost of p_{f^N}, can be achieved.

In one domain d passed by p_{f^N}, the optimal intra-domain path, corresponding to the $e_{f^N}^d$ in p_{f^N}, is selected with related border node pair as the source and sink. f^N is routed as an intra-domain flow according to (11), with $\varphi_{e_{f^N}^d}$ as the minimal energy consumption. The residual network is the domain topology with all former flows assigned. Different $p_{f^N} \in P_{f^N}$ through d may correspond to different $e_{f^N}^d \in E_{f^N}^d$, and the $\varphi_{e_{f^N}^d}$ for every $e_{f^N}^d \in E_{f^N}^d$ is calculated.

When energy consumption $\varphi_{p_{f^N}}$ of every $p_{f^N} \in P_{f^N}$ is achieved, according to the simplified DCN with all related e and $e_{f^N}^d$, (13) can be rewritten as:

$$\begin{aligned} \varphi_{f^N,opt} &= \min_{p_{f^N} \in P_{f^N}} \varphi_{p_{f^N}} \\ &= \min_{p_{f^N} \in P_{f^N}} \Big(\sum_{e \in p_{f^N}} \varphi_e + \sum_{d \in D, e_{f^N}^d \in p_{f^N}} \varphi_{e_{f^N}^d} \Big). \end{aligned} \tag{14}$$

The computing for one f^N consists of two sections: the distributed computing for $E_{f^N}^d$ in each related domain d and the global computing for $\varphi_{f^N,opt}$ based on the simplified DCN. After the assignments of all the $f^N \in F^N$ one after another, the whole energy consumption increment of the large-scale data plane can be calculated as:

$$\sum_{d \in D} \phi_{|F^d|}^d + \phi_{|F^N|}^N. \tag{15}$$

D. HEURISTIC PROCESS OF DISTRIBUTED EER

The heuristic process of the distributed EER model is presented in Algorithm 1. There are two main steps respectively dealing with intra- and inter-domain flows. The pseudo codes in each $d \in D$ are processed in a distributed way. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2791630, IEEE Access

calculating pressure is mostly spread across a number of local optimizers, and the computation time can be greatly reduced.

Algorithm 1: Distributed Energy-Efficient Routing

Input:
Resident Network: N, Domains Set: D,
Flow Demands Sets: F^d, F^N, Set of Possible Paths for Each f: P_f.
Output:
Alive Devices Subset with Optimal Power Consumption.
• **1. EER for Intra-domain Flows**;
foreach $d \in D$ **do**
  $\phi^d \leftarrow 0$;
  foreach $f^d \in F^d$ **do**
    $\varphi_{f^d,opt} \leftarrow +\infty$;
    foreach $p_{f^d} \in P_{f^d}$ **do**
      Calculating $\varphi_{p_{f^d}}$;
      if $\varphi_{p_{f^d}} < \varphi_{f^d,opt}$ **then**
        $\varphi_{f^d,opt} \leftarrow \varphi_{p_{f^d}}$;
      end
    end
    Route f^d along the located optimal path $p_{f^d,opt}$;
    $\phi^d \leftarrow \phi^d + \varphi_{f^d,opt}$;
  end
end
$\phi \leftarrow \sum_{d \in D} \phi^d$;
ϕ is the current energy consumption after all intra-domain flows are assigned;
• **2. EER for Inter-domain Flows**;
foreach $f^N \in F^N$ **do**
  $\varphi_{f^N,opt} \leftarrow +\infty$;
  foreach $p_{f^N} \in P_{f^N}$ **do**
    foreach $d \in D$ **do**
      if *there is* $e^d_{f^N} \in p_{f^N}$ **then**
        Calculating $\varphi_{e^d_{f^N}}$;
      end
    end
    Calculating $\varphi_{p_{f^N}}$;
    if $\varphi_{p_{f^N}} < \varphi_{f^N,opt}$ **then**
      $\varphi_{f^N,opt} \leftarrow \varphi_{p_{f^N}}$;
    end
  end
  Route f^N along the located optimal path $p_{f^N,opt}$;
  $\phi \leftarrow \phi + \varphi_{f^N,opt}$;
end
ϕ is the final energy consumption after all types of flows are assigned;
The alive devices along all the selected paths form the alive devices subset.

In a rough estimating, the computation complexity of the general centralized greedy approach is about $O(|F| \times |P_f| \times L_{p_f})$ in Fat-Tree, where $|F|$ is the number of elephant flows to schedule, $|P_f|$ is the number of the alternative paths for flow f, and L_{p_f} is the length of path p_f (the computation cost to find the energy cost of path p_f). In the distributed EER, $|F|$ is divided into intra-domain part and inter-domain part, and the computation cost of intra-domain part is distributed to each domain. The computation to find the energy cost of path p_{f^N} of inter-domain flow f^N is also divided into intra-domain part and inter-domain part, and the computation cost for intra-domain part is also distributed in each domain.

For control plane, there is no inter-domain part, since all the power savings are respectively processed in each domain in parallel. The detailed process can be found in our previous work [20], and is omitted in this paper due to space constraints. The computation time for control plane can be much less than the time for data plane. In the distributed approach, the time cost of control plane is also distributed among the domains.

TABLE 2. Energy Proportionality of Sampled Switch

Different Scenarios	Energy Proportion (Unit: *Watts*)
One alive port: x	0.90
Idle Switch (No Port On): y	135
Ports All On (No Traffic)	178
Ports All On (Fully Loaded)	183

V. SIMULATION AND ANALYSIS

In this section, we simulate the distributed policy to validate the theoretical analysis. Compared with the centric approach, the distributed approach achieves barely the same power-saving effect, with processing time significantly shortened.

A. SIMULATION DESIGNS

Typical Fat-Tree is employed as the representative DCN topology being tested in different sizes. The real datasets of DCN traffic from IMC 2010 Data Center Measurement [37] are adopted as the flow model to generate the experimental traffic with suitable enhancement, aggregation, and composition. The inter-domain flows are also suitably enhanced to highlight the advantages of the distributed energy-aware inter-domain flow routing. Both mice flows and elephant flows are produced. The elephant flows are mainly routed to guide the power saving for data plane, and both the two types of flows are counted for arrival rates guiding the power saving for control plane.

The traffic demands between hosts are assumed to obey commonly used Poisson distribution with parameter $\lambda = 50$, the duration obeys exponential distribution with parameter $\lambda' = 1/5$. One controller is assumed to control 4 edge switches with peak flow arrival rate at most. For the experimental reliability, sample values of energy consumption function parameters, such as x, y in (3), need to be representative in the simulation. The OpenFlow switches are considered as standard network devices following 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 2 summarizes the energy proportionality (x and y) for the sampled commercial Ethernet GigE switches.

Controllers run on the servers with 2.4 GHz Intel Xeon CPU, 48 GB RAM and Ubuntu 14.04. The energy model is profiled based on (2). Table 3 summarizes the sample values of the parameters ($Power_{Idle'}$, ρ_1, ρ_2, and ρ_3) used in the energy consuming function of the controllers.

The simulation is processed using MATLAB 2015b on server with 3.1GHz Intel Core CPU, 16 GB RAM and Windows 8.1. The calculating works in different domains are all processed in this server one after another, and are assumed to be proceeded in parallel. For simplicity, the *Out-of-Band*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2791630, IEEE Access

TABLE 3. Energy Characteristics of Sampled Controller

Parameters in (2)	Sampled Values
$Power_{Idle'}$	155
ρ_1	345
ρ_2	-359
ρ_3	144

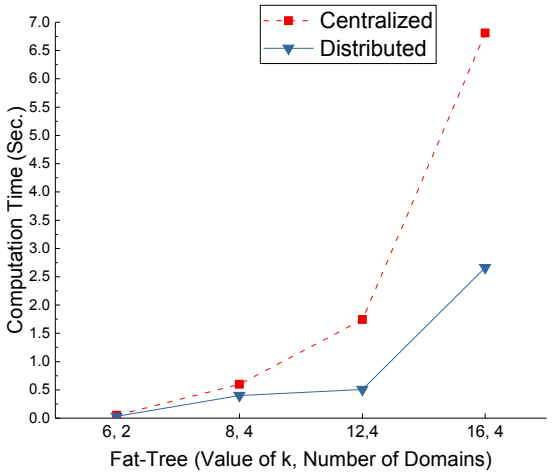

FIGURE 7. Delay of Computation Time via Centralization and Distribution.

control approach is adopted in the hypothetical OpenFlow-enabled SDN-DCN structure with a standalone EER server for inter-domain flows, as shown in Fig. 5.

B. PERFORMANCE ANALYSIS

The primary metric in the simulation is the *Energy Consumption Level*, which can be calculated as:

$$\frac{Power\ consumed\ with\ the\ optimization}{Power\ consumed\ without\ the\ optimization}. \quad (16)$$

To emphasize the advantages of the distributed power optimization, the scalabilities of the centralized and distributed optimizers are firstly discussed, based on computation time vs topology size. The centralized optimization adopts the leftmost greedy way in [3]. The time cost of control plane is not considered (ignored). The Fat-Tree scenario is set up with $k = 6$, 8, 12, and 16, and k is the pods number. The corresponding numbers of servers are 54, 128, 432, and 1024. The topologies are all equally partitioned into 2 domains (16 and 128 servers) and 4 domains (432 and 1024 servers). For the sake of fairness, the centralized and distributed optimizations are both programmed and run on the same programming platform. The results are shown in Fig. 7, and the improvement in system scalability using the distributed approach is obvious. The amount of saved computation time is related to the number of topology domains, the proportion of inter-domain flows, and the method of domain partition.

Note that the "large-scale" DCN in this paper refers to a DCN with a huge amount of network devices but deployed in one location (e.g. one local clustered container data center). The control network in the simulation is *Out-of-Band*, and can be deployed as a simple Ethernet LAN, which can also provide communication functions between multiple controllers (including distributed intra-domain EER servers and the standalone inter-domain EER server). The link length of such control network is short, the number of its middle forwarding nodes is low, and the propagation delay of messages can be ignored. On the other hand, the data sizes of the messages in control network are always quite small, and thus the transmission delay can also be ignored. Therefore, in this paper the processing delay (computation time) is the main concern, and the delay caused by the information transmitting between intra- and inter-domain EER servers is not discussed.

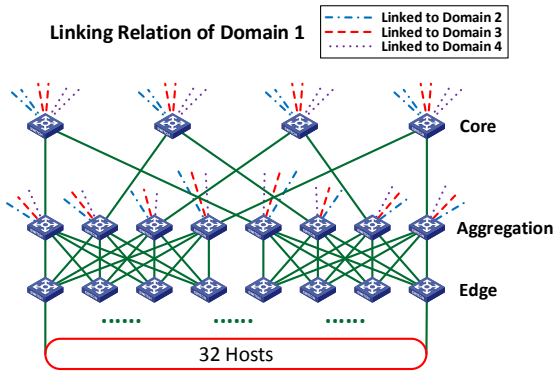

FIGURE 8. Topology of One Domain of Fat-Tree (8 Pods, 4 Equal Domains).

To evaluate the efficiency of the distributed power optimization, the energy-saving levels of the centralized and distributed optimizers are compared, based on different amounts of traffic demands. The centralized energy optimization for data plane is the common used greedy way in [9] (similarly hereinafter). The energy optimization for control plane is the Bin-Packing way in [20]. The Fat-Tree scenario is set up with 8-port switches with 10 Mbps links, and the number of servers in the data center is 128. The topology is equally partitioned into 4 domains, and each domain has 32 servers, 4 core switches, 8 aggregation switches, 8 edge switches, and 2 pods. The topology of one domain is shown in Fig. 8, with the connections between domains described. The maximum number of controllers in one domain is 3. The offered flows (mice and elephant) are then generated with different flow parameters, and the energy-saving effects of centralized and distributed optimizers are compared. The hosts of flows are uniformly distributed in the topology.

The energy consumption level of elephant flow loads is firstly tested with different average bandwidth sizes, and the redundancy is suitably considered during the computation to ensure network reliability and QoS. The total number of elephant flows is settled around 80, and the flow arriving rate is settled to keep one controllers alive in each domain. The energy consumption results are shown in Fig. 9. When the average demand of elephant flows increases, the energy consumption level will increase gradually.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2791630, IEEE Access

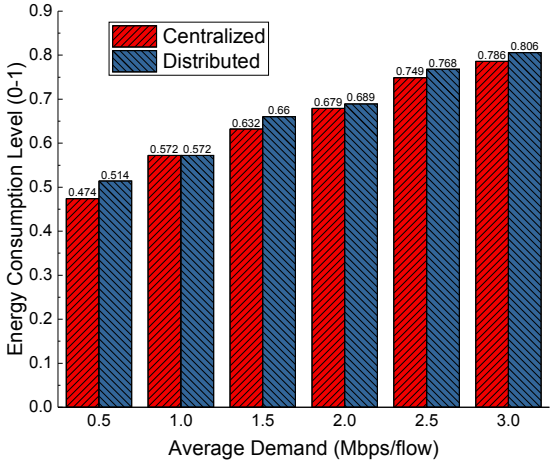

FIGURE 9. Energy Consumption Level of Different Demands of Flows.

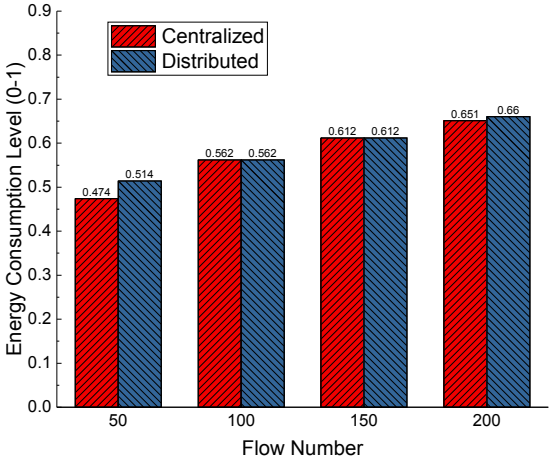

FIGURE 10. Energy Consumption Level of Different Numbers of Flows.

The energy consumption level is then tested with different flow numbers, with redundancy suitably considered. The average demand of elephant flows is settled around 0.8 Mbps, and the flow arriving rate is also settled to keep one controllers alive in each domain. The energy consumption results are shown in Fig. 10, and the trends are almost the same as those in Fig. 10. When the flow number rises, the results in Fig. 10 are even better. This is because more elephant flows with relatively low bandwidth demands (same total throughput) can be much more flexibly routed in the network and the utilization of the bandwidth is relatively higher.

Because of the wholly global optimization, the centralized approach should logically achieve better results than usual distributed approach. However, seeing from Fig. 7, 9, and 10, the effectiveness of the distributed approach is almost the same as the centralized approach (little high), and the computation time is greatly shortened. When the whole traffic demand is relatively low (0.5 Mbps in Fig. 9 and 50 flows in Fig. 10), the energy consumption level of the distributed approach is not as good as the result of centralized approach.

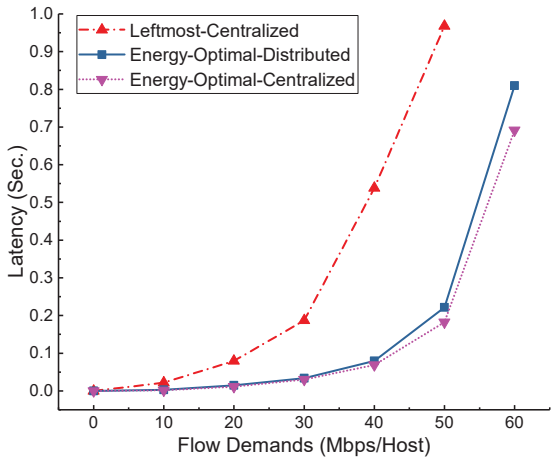

FIGURE 11. Queuing Latency via Centralization and Distribution.

This is because in the centralized approach only the minimum subset of the nodes in Fat-Tree (just one core switch) keeps alive. In the distributed approach, in order to keep intra-domain connectivity, there has to be at least one alive core switch in each domain, since the intra-domain flows cannot utilize the nodes in other domains. At the other points, there may also be a little more energy consumption in distributed way. This is because some inter-domain paths that repeatedly pass through one domain and intra-domain paths that pass out of its domain are not permitted.

C. LATENCY AND REAL SCENARIO

Since the propagation and transmission delay can be ignored and the processing delay (i.e., computation time) is already discussed in Fig. 7, the latency analyzed here is mainly the queuing delay in the ports of switches. The Fat-Tree scenario is set up with $k = 8$ with 128 servers, and the link bandwidth is set to be 100 Mbps. Like DFS [17], credit-based flow control is employed to avoid packet loss. The input ports of switches are all buffer-enabled, and the buffer size is 1 MB. Possion distribution is used to generate the uniform flow series like the previous simulations. The destinations of these flows are uniformly distributed among different pods, which means all the flows are set to pass through the core layer. Then the proportion of intra-domain flows is about 25%, and the baseline latency is set to be around 180 μs (5 times forwarding by switches) [3]. The leftmost-centralized[2] [3], energy-optimal-centralized[3] [9] and the proposed energy-optimal-distributed routing approaches are implemented for comparison.

The flow demands of the hosts are increased, and the latency is shown in Fig. 11. When saturation point is reached, the latency significantly rises because of flow collisions and network congestion. The latency performance of the

[2]The leftmost path with sufficient capacity is assigned to forward the flow.
[3]With sufficient capacity, the path that consumes minimal energy is assigned to forward the flow.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2791630, IEEE Access

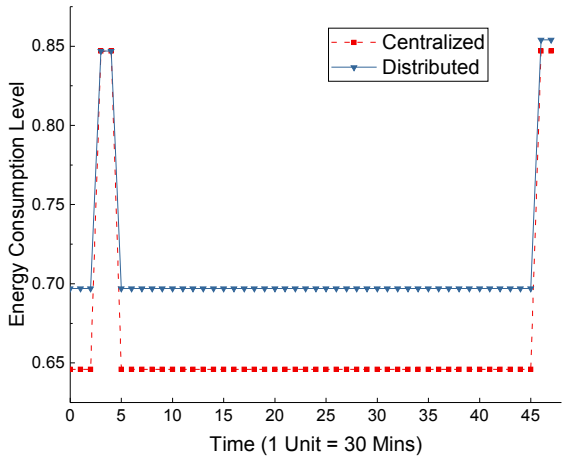

FIGURE 12. Energy Consumption Level of Real Trace in One Day.

energy-optimal-distributed routing is almost the same as the performance of the energy-optimal-centralized routing. The latter performs a little better because the distributed routing scheme has OTP restriction, which is more likely to cause flow congestions. However, the latency performance of the distributed routing is much better than the performance of leftmost-centralized routing. This proves that the distributed routing can also achieve effective network forwarding.

The distributed energy optimization is also tested leveraging a real flow trace in one day, and the trace is selected from Dataset UNV1 (11/01/2009) in [37]. Based on this dataset, the traffic data was recorded every 30 minutes, and thus the energy optimization will be executed every 30 minutes for 48 times. The topology is also a Fat-Tree structure with 4-pods, and is divided into 2 domains. Each domain has 8 servers, 2 core switches, 4 aggregation switches, 4 edge switches, 2 pods, and one controller. Due to the distributed control structure, both of the two controllers need keeping alive. Then the power saving just for data plane is examined. The centralized optimization is also calculated as a criterion.

The optimized results are shown in Fig. 12, and the intra-day trends are obvious. The values at some time units are much higher than before (the x-axis points of 3, 4, 46, and 47). This is because at these time units some switches need to be awakened due to a few specific flows, and the energy consumption is observably increased. Since diurnal pattern of network traffic exists in all data centers, the curves would accurately reflect such patterns (two energy consumption peaks in the morning and night, respectively).

VI. CONCLUSION

In this paper, we present a distributed energy-saving mechanism via multi-controller SDN in large-scale multi-domain data center networks, which dynamically consolidates workloads onto a small set of devices and puts the redundant ones into dormancy to save power. SDN's fine-grained routing policy is leveraged for scheduling flows to further improve the efficiency. The distributed approach can improve the scalability of the energy optimization and maintain the power-saving effectiveness.

In particular, we propose a distributed energy-efficient routing scheme to search the global optimal routing solution, where both the intra- and inter-domain elephant flows are routed distributedly to save the power for data plane. Meanwhile, the optimal subset of controllers is also distributedly selected to save the power for control plane. The evaluation of real DCN data demonstrates that this scheme can achieve an effective power saving. With reduced computation load by distributed greedy heuristic, the proposed mechanism can be leveraged in a large-scale topology with much better time performance.

REFERENCES

[1] A. Shehabi, S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, E. Masanet, N. Horner, I. Azevado, and W. Lintner, "United States Data Center Energy Usage Report," Lawrence Berkeley National Laboratory, Tech. Rep. LBNL-1005775, Jun. 2016. [Online]. Available: https://datacenters.lbl.gov/resources/united-states-data-center-energy-usage

[2] A. Greenberg, J. Hamilton, D. Maltz, and P. Patel, "The Cost of a Cloud: Research Problems in Data Center Networks," ACM SIGCOMM Computer Communication Review, vol. 39, no. 1, pp. 68–73, 2009.

[3] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "ElasticTree: Saving Energy in Data Center Networks," in Proc. NSDI'10, 2010, pp. 249–264.

[4] Y. Shang, D. Li, and M. Xu, "Energy-Aware Routing in Data Center Network," in Proc. The 1st ACM SIGCOMM Workshop on Green Networking (Green Networking'10), 2010, pp. 1–8.

[5] ——, "Greening Data Center Networks with Flow Preemption and Energy-aware Routing," in Proc. The 19th IEEE Workshop on Local Metropolitan Area Networks (LANMAN'13), 2013, pp. 1–6.

[6] "OpenFlow Switch Specification Ver 1.5.1," Open Networking Foundation, Tech. Rep. TS-025, Mar. 2015. [Online]. Available: https://www.opennetworking.org/sdn-resources/technical-library

[7] "Software-Defined Networking: The New Norm for Networks," Open Networking Foundation, Tech. Rep., Apr. 2012. [Online]. Available: https://www.opennetworking.org/sdn-resources/technical-library

[8] E. Haleplidis, K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology," IRTF RFC 7426, Jan. 2015. [Online]. Available: https://www.rfc-editor.org/info/rfc7426

[9] R. Tu, X. Wang, and Y. Yang, "Energy-Saving Model for SDN Data Centers," Journal of Supercomputing, vol. 70, no. 3, pp. 1477–1495, 2014.

[10] D. Li, Y. Shang, and C. Chen, "Software Defined Green Data Center Network with Exclusive Routing," in Proc. IEEE INFOCOM'14, 2014, pp. 1743–1751.

[11] D. Li, Y. Shang, W. He, and C. Chen, "EXR: Greening Data Center Network with Software Defined Exclusive Routing," IEEE Trans. Comput., vol. 64, no. 9, pp. 2534–2544, 2015.

[12] D. Li, Y. Yu, W. He, K. Zheng, and B. He, "Willow: Saving Data Center Network Energy for Network-Limited Flows," IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 9, pp. 2610–2619, 2015.

[13] L. Wang, F. Zhang, J. A. Aroca, A. V. Vasilakos, K. Zheng, C. Hou, D. Li, and Z. Liu, "GreenDCN: A General Framework for Achieving Energy Efficiency in Data Center Networks," IEEE J. Sel. Areas Commun., vol. 32, no. 1, pp. 4–15, 2014.

[14] X. Wang, Y. Yao, X. Wang, K. Lu, and Q. Cao, "CARPO: Correlation-Aware Power Optimization in Data Center Networks," in Proc. IEEE INFOCOM'12, 2012, pp. 1125–1133.

[15] 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, 2016.

[16] "Microsoft's Cloud Infrastructure: Datacenters and Network Fact Sheet," Microsoft Corporation, Tech. Rep., Jun. 2015. [Online]. Available: http://download.microsoft.com/download/8/2/9/8297F7C7-AE81-4E99-B1DB-D65A01F7A8EF/Microsoft_Cloud_Infrastructure_Datacenter_and_Network_Fact_Sheet.pdf

[17] R. Liu, H. Gu, X. Yu, and X. Nian, "Distributed Flow Scheduling in Energy-Aware Data Center Networks," IEEE Commun. Lett., vol. 17, no. 4, pp. 801–804, 2013.
[18] A. Fernández-Fernández, C. Cervelló-Pastor, and L. Ochoa-Aday, "Energy-Aware Routing in Multiple Domains Software-Defined Networks," Advances in Distributed Computing and Artificial Intelligence Journal, vol. 5, no. 3, pp. 13–19, 2016.
[19] Y. Fu, J. Bi, K. Gao, Z. Chen, J. Wu, and B. Hao, "Orion: A Hybrid Hierarchical Control Plane of Software-Defined Networking for Large-Scale Networks," in Proc. The 22nd IEEE International Conference on Network Protocols (ICNP'14), 2014, pp. 569–576.
[20] K. Xie, X. Huang, S. Hao, M. Ma, P. Zhang, and D. Hu, "E^{3}MC: Improving Energy Efficiency via Elastic Multi-Controller SDN in Data Center Networks," IEEE Access, vol. 4, pp. 6780–6791, 2016.
[21] M. Al-Fares, A. Loukissas, and A. Vahdat, "A Scalable, Commodity Data Center Network Architecture," in Proc. ACM SIGCOMM'08, 2008, pp. 63–74.
[22] A. Greenberg, J. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. Maltz, P. Patel, and S. Sengupta, "VL2: A Scalable and Flexible Data Center Network," in Proc. ACM SIGCOMM'09, 2009, pp. 51–62.
[23] 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'08, 2008, pp. 75–86.
[24] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, and C. Tian, "BCube: A High Performance, Server-Centric Network Architecture for Modular Data Centers," in Proc. ACM SIGCOMM'09, 2009, pp. 63–74.
[25] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A Power Benchmarking Framework for Network Devices," in Proc. Networking'09, 2009, pp. 795–808.
[26] H. Hlavacs, G. D. Costa, and J. M. Pierson, "Energy Consumption of Residential and Professional Switches," in Proc. The 2009 International Conference on Computational Science and Engineering (CSE'09), 2009, pp. 240–246.
[27] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing," Future Generation Computer Systems, vol. 28, no. 5, pp. 755–768, 2012.
[28] L. Luo, W. Wu, and F. Zhang, "Energy Modeling based on Cloud Data Center," Ruan Jian Xue Bao/Journal of Software, vol. 25, no. 7, pp. 1371–1387, 2014, abstract in English available.
[29] A. Tootoonchian and Y. Ganjali, "HyperFlow: A Distributed Control Plane for OpenFlow," in Proc. The 2010 Internet Network Management Conference on Research on Enterprise Networking (INM/WREN'10), 2010, pp. 3–3.
[30] S. H. Yeganeh and Y. Ganjali, "Kandoo: A Framework for Efficient and Scalable Offloading of Control Applications," in Proc. The First Workshop on Hot Topics in Software Defined Networks (HotSDN'12), 2012, pp. 19–24.
[31] P. Lin, J. Bi, Z. Chen, Y. Wang, H. Hu, and A. Xu, "WE-Bridge: West-East Bridge for SDN Inter-domain Network Peering," in Proc. The 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS'14), 2014, pp. 111–112.
[32] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic Flow Scheduling for Data Center Networks," in Proc. NSDI'10, 2010, pp. 281–296.
[33] S. Kandula, D. Katabi, S. Sinha, and A. Berger, "Dynamic Load Balancing Without Packet Reordering," ACM SIGCOMM Computer Communication Review, vol. 37, no. 2, pp. 51–62, 2007.
[34] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, Network flows: Theory, algorithms, and applications. Englewood Cliffs, NJ, USA: Prentice Hall, 1993.
[35] F. Giroire, D. Mazauric, J. Moulierac, and B. Onfroy, "Minimizing Routing Energy Consumption: from Theoretical to Practical Results," in Proc. IEEE/ACM Green Computing and Communications (GreenCom'10), 2010, pp. 252–259.
[36] T. Benson, A. Akella, and D. A. Maltz, "Network Traffic Characteristics of Data Centers in the Wild," in Proc. The 10th ACM SIGCOMM Conference on Internet Measurement (IMC'10), 2010, pp. 267–280.
[37] "Data set for imc 2010 data center measurement," 2010. [Online]. Available: http://pages.cs.wisc.edu/~tbenson/IMC10_Data.html

KUN XIE is a Ph.D. candidate at the Institute of Network Technology, Beijing University of Posts and Telecommunications (BUPT), Beijing, China. He received his Master degree in Computer Applications Technology from North China Electric Power University (NCEPU), in 2010. He received his B.E. degree in Network Engineering also from NCEPU, in 2007. His research interests lie on the computer networking system and next-generation network, including the Software Defined Networking, network performance analysis, power saving of data center networks and so on.

DR. XIAOHONG HUANG received her B.E. degree from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2000 and Ph.D. degree from the school of Electrical and Electronic Engineering (EEE), Nanyang Technological University, Singapore in 2005. Since 2005, Dr. Huang has joined BUPT and now she is an associate professor and director of Network and Information Center in Institute of Network Technology of BUPT. Dr. Huang has published more than 50 academic papers in the area of WDM optical networks, IP networks and other related fields. Her current interests are performance analysis of computer networks, service classification and so on.

SHUAI HAO received his Ph.D. degree in Computer Science from the College of William and Mary, Williamsburg, VA, in 2017. He is a visiting researcher in Department of Electrical & Computer Engineering at the University of Delaware, Newark, DE. His research interests lie on the networking and security, including Internet Topology, Internet Infrastructure, Network Attack and Defense, and Web Security and Privacy.

DR. MAODE MA received his Ph.D. degree in computer science from Hong Kong University of Science and Technology in 1999. Now, Dr. Ma is an Associate Professor in the School of Electrical and Electronic Engineering at Nanyang Technological University in Singapore. He has extensive research interests including network security and wireless networking. Dr. Ma has more than 300 international academic publications including over 140 journal papers and more than 160 conference papers. He currently serves as the Editor-in-Chief of *International Journal of Computer and Communication Engineering* and *International Journal of Electronic Transport*. He also serves as a Senior Editor or an Associate Editor for other 5 international academic journals. Dr. Ma is a Fellow of *IET*, a Senior Member of *IEEE Communication Society and IEEE Education Society*, and a Member of *ACM*. He is the Chair of the *IEEE Education Society*, Singapore Chapter and the Chair of the *ACM*, Singapore Chapter. He is serving as an *IEEE Communication Society Distinguished Lecturer*.