

Article

Energy Optimization for Software-Defined Data Center Networks Based on Flow Allocation Strategies

Zebin Lu ^{1,2} , Junru Lei ², Yihao He ², Zhengfa Li ², Shuhua Deng ²  and Xieping Gao ^{2,3,*} 

¹ School of Mathematics and Computational Science, Xiangtan University, Xiangtan 411105, China; brightzelool@163.com

² Key Laboratory of Intelligent Computing and Information Processing of Ministry of Education, Xiangtan University, Xiangtan 411105, China; junrulei@foxmail.com (J.L.); heyihaoadolph@163.com (Y.H.); zhengfali@foxmail.com (Z.L.); shuhudeng@163.com (S.D.)

³ College of Software and Communication Engineering, Xiangnan University, Chenzhou 423043, China

* Correspondence: xpgao@xtu.edu.cn

Received: 20 July 2019; Accepted: 7 September 2019; Published: 11 September 2019



Abstract: Nowadays, energy consumption has become an important issue in data center networks. The most promising energy-saving schemes are those that shut down unnecessary network devices and links while meeting the demand of traffic loads. Existing research mainly focuses on the strategies of energy savings in software-defined data center networks (SD-DCN). Few studies have considered both energy savings and the quality of service (QoS) of the traffic load. In this paper, we investigate the energy savings guaranteed by traffic load satisfaction ratio. To ensure the minimum-power consumption in data centers, we formulate the SD-DCN energy consumption optimization problem as an Integer Linear Programming model. To achieve a high success rate for traffic transmission, we propose three flow scheduling strategies. On this foundation, we propose a strategy-based Minimum Energy Consumption (MEC) heuristic algorithm to ensure the QoS satisfaction ratio in the process of energy optimization. The results show that our algorithm can save energy efficiently under the conditions of low traffic load and medium traffic load. Under high traffic load, our algorithm can achieve better network performance than existing solutions in terms of quality of service satisfaction ratio of flow allocation.

Keywords: energy efficiency; software-defined data center networks; QoS; flow allocation

1. Introduction

In the past decade, with the development of big data, cloud computing, virtualization, and other data-intensive applications, data centers are growing rapidly. While data centers significantly simplify the management of the application's data, the traffic transmission in data centers brings great energy consumption. Studies have shown that the electricity consumption of global data centers accounted for 1.1% to 1.5% of the total electricity consumption in 2010, and it can be predicted that the percentage will increase to 8% by 2020 [1,2]. In addition, the energy consumption of data center networks (DCNs) accounts for 20% of the total power consumption when the servers are fully utilized, but it will rise to 50% when servers utilization decreases to 15% [3]. Therefore, energy-saving in data centers has become one of the most important topics.

A data center is a facility composed of a number of technical elements such as network devices, servers, storage systems, and more. To improve the performance and reliability of the network, some novel topological structures have been proposed, such as Fat-Tree [4], VL2 [5], DCell [6], BCube [7], etc. Since a large-scale data center contains a large number of servers, links, and bandwidth-intensive applications, it is not easy to realize flexible scheduling of network resources in traditional distributed

network architectures. Software-defined networking (SDN) is an emerging network architecture [8] that allows network operators to manage and control a network through a centralized controller. Since the control plane is separated from the underlying network device, it makes network management more efficient and convenient. Recently, some studies incorporated SDN into DCNs, named software-defined data center networks (SD-DCNs), to optimize the data center energy consumption by utilizing the scalability and manageability of SDN technology [9,10].

Some studies have proved that software-defined networking is an effective solution to solve energy problems in DCNs [11–14]. The main idea of these works is to design some energy optimization models to save energy under the condition of using as few network resources as possible. While those works can effectively save energy in certain scenarios, they ignore limited network resources that would fail to satisfy the quality of service (QoS) requirements. Typically, the QoS requirements can be roughly classified into bottleneck and additive [15,16].

For a bottleneck requirement, the cost of the path is determined by the value of that constraint at the bottleneck resource, such as bandwidth, CPU, and Ternary Content Addressable Memory (TCAM) [17–20]. For an additive requirement, the cost of the end-to-end path is given by the sum of the values of each link along that path, such as delay and jitter [16,21,22]. Some QoS parameters are multiplicative, such as loss rate [23]. Some researchers classify loss rate as an additive requirement because loss rate can be expressed as an additive requirement in an indirect way [15]. The QoS requirements are summarized in Table 1.

Table 1. Classification of quality of service (QoS) requirements.

| Type | QoS Requirements |
|------------|--|
| bottleneck | bandwidth, CPU, and TCAM [17–20] |
| additive | delay, jitter, and loss rate [15,16,21–23] |

In this paper, we mainly focus on bottleneck requirements, including bandwidth and TCAM. The bandwidth resource is scarce in a data center. Besides, the TCAM size in SDN switches is limited. Using traffic aggregation and other ways to allocate as few network resources as possible will inevitably lead to traffic transmission failure due to insufficient resources [24]. We investigate the problem of how to ensure the efficiency of network traffic transmission while realizing the energy optimization of the data center network. More specifically, we optimize SD-DCN energy consumption with the objective of minimizing the energy cost of switches and links under the constraint of network resources. To improve the QoS satisfaction ratio, we divide the network traffic into elephant flow and mice flow. Then, three different traffic scheduling strategies are used for these two kinds of traffic. Combined with the energy optimization model, a method which can meet both energy-saving and QoS requirements is found.

In general, our contributions are summarized as follows:

- We formulate SD-DCN energy consumption optimization problem as an Integer Linear Programming (ILP) model. Besides, we propose three different flow scheduling strategies to improve the QoS satisfaction ratio;
- We propose a strategy-based Minimum Energy Consumption (MEC) heuristic algorithm to ensure the QoS satisfaction ratio in the process of energy optimization;
- We evaluate and discuss the strategy-based heuristic algorithm in terms of effectiveness.

The rest of the paper is organized as follows: Some related work is discussed in Section 2. The network model and problem statement are introduced in Section 3. The strategy-based Minimum Energy Consumption (MEC) heuristic algorithm is proposed in Section 4. Simulation results and analysis are shown in Section 5. Section 6 concludes the paper.

2. Related Work

Data center network energy consumption is growing increasingly. In order to reduce energy consumption in data center networks, researchers have done a lot of work to achieve high energy efficiency from different levels. From the perspective of energy, Heller et al. [25] proposed ElasticTree, which dynamically controls the number of switches and links used in the network to meet the changing data center traffic loads and turns off unused network elements in the network to save energy. Jiang et al. [26] proposed an energy-aware data center network, which uses as few network devices as possible and turns off unused network devices to reduce energy consumption. Wang et al. [27] proposed a correlation-aware power optimization (CARPO) algorithm that combines link rate adaptation and correlation-aware traffic consolidation to maximize energy savings.

With the development of SDN technology, more and more scholars use SDN to solve data center network energy problems. Tu et al. [11] proposed two energy-saving models for data centers based on SDN. These models can get better energy efficiency in different scenarios. Yoon et al. [12] proposed a power minimization model in a Fat-Tree data center network, and adopted the Simulated Annealing algorithm to obtain the solution. Wei et al. [13] designed an energy-efficient routing algorithm based on the multinomial logit model, and formalized the energy-efficient routing algorithms. Zeng et al. [14] put forward a minimum-activated switches algorithm in SD-DCN to save energy, while the limitations of SDN itself were taken into account, i.e., TCAM-size limitation. Xu et al. [24] proposed a data center network energy-saving algorithm that leveraged SDN technology to minimize energy consumption in data center networks and increase the utilization of switches. Li et al. [28] designed an energy-aware flow scheduling approach called exclusive routing (EXR) which occupies the links in a flow independently based on priority. Existing research efforts mainly applied SDN technology to turn off network devices to save energy. However, several unique features of SDN are often ignored. For example, the TCAM size and link capacity are limited. Using traffic aggregation and other ways to allocate as few network resources as possible will inevitably lead to traffic transmission failure.

In terms of QoS guarantee, Huang et al. [29] studied the rules allocation optimization problem in multipath routing, aiming to reduce rule space occupation and guarantee high QoS satisfaction ratio. Wang et al. [30] proposed a bandwidth allocation scheme named Blocking Island (BI). For one thing, Blocking Island scheme can reduce the searching space; for another, the success rate of bandwidth allocation can be improved. As far as we know, existing works on QoS satisfaction ratio have not been applied in the SD-DCN.

The closest work to this paper is [14]. In that paper, the authors were the first to study the minimum-energy switch activation for multipath routing in SD-DCN, taking into account the limitations of TCAM size in SDN. Their main contributions include formulating the energy optimization problem as an ILP problem and proposing a heuristic algorithm to deal with the high computational complexity of solving ILP. Inspired by [14], our motivation is to ensure the efficiency of network traffic transmission while optimizing data center network energy consumption. More precisely, different from the previous work of [14], which considers only the energy consumption of nodes; we extend the work of [14], considering both nodes and links as energy consumption units. In addition, we divide the network traffic into mice flow and elephant flow and propose three different traffic scheduling strategies to improve the QoS satisfaction ratio.

3. Network Model and Problem Formulation

In this section, we describe the network model and formulate the SD-DCN energy consumption optimization problem in detail. The major notations used in this paper are summarized in Table 2.

Table 2. Summary of notations.

| Notation | Description |
|--------------|---|
| $G = (N, E)$ | Undirected graph where N is the set of switches and hosts (i.e., servers) and E is the set of links |
| S | Set of switches |
| H | Set of hosts |
| C_u | TCAM size of switch u |
| $B_{(u,v)}$ | Capacity of link (u, v) |
| $p(u, v)$ | Power consumption of link (u, v) |
| $p(u)$ | Power consumption of switch u |
| K | Set of all traffic flows, each flow $k \in K$ |
| r_k | Traffic demand by flow k from host s_k to d_k |
| $X(u, v)$ | Binary variable indicating whether link (u, v) is powered on |
| $Y(u)$ | Binary variable indicating whether switch u is powered on |
| f_{uv}^k | The amount of flow k that is routed through link (u, v) |
| x_{uv}^k | Binary variable indicating whether flow k goes through switch u on outgoing link (u, v) |

3.1. Network Model

Typical traditional DCN architectures consist of two- or three-tier trees of switches or routers. 2N-Tree is a typical traditional DCN topology. In order to improve the reliability and scalability of the network, many new DCN architectures have been proposed [4–7]. These new architectures can be classified into two categories: Switch-centric topology, e.g., Fat-Tree [4] and VL2 [5]; and server-centric topology, e.g., DCell [6] and BCube [7].

In this paper, our solution is designed to support any switch-centric network topology, such as Fat-Tree and VL2. Fat-Tree is the most popular topology used in data centers. In the remaining paper, we use the Fat-Tree DCN topology for analysis and experimental simulation. The Fat-Tree topology consists of three switch layers, namely, core layer, aggregation layer, and edge layer. An n -ray Fat-Tree network topology is shown in Figure 1. At the top, the core layer consists of $(n/2)^2$ n -port core switches. In the medium, there are n pods, each consisting of two layers of $n/2$ switches. The aggregation switches and the core switches are connected to each other, while the edge switches are connected to the servers. Each edge switch can contain at most $n/2$ hosts. More precisely, Figure 1 provides an example for $n = 4$, which is the 4-pod data center network topology.

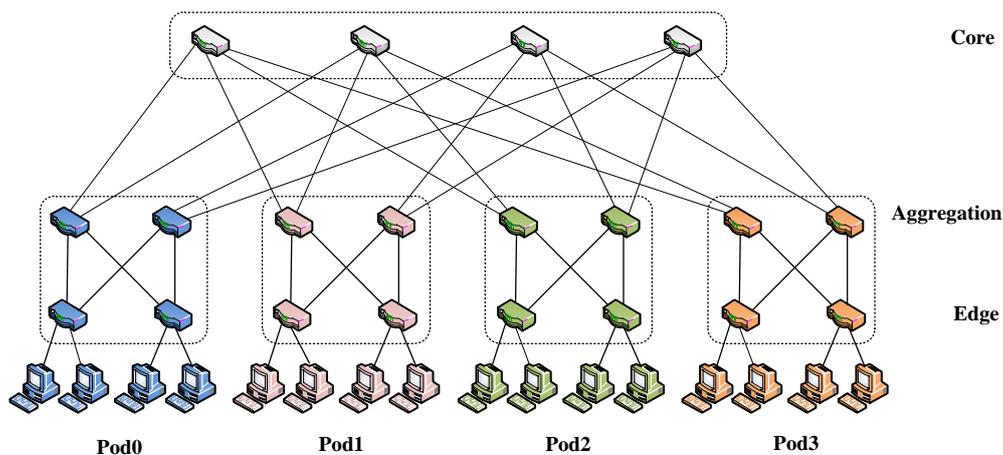


Figure 1. An example of Fat-Tree DCN topology with 4-pod.

We model an undirected graph $G = (N, E)$ for the data center network. N consists of both the hosts (i.e., servers) set, H ; and the SDN-enabled switches (i.e., OpenFlow switches) set, S , i.e.,

$N = S \cup H$ and E represent a set of links between nodes in N . According to the SDN communication process, each flow goes through a switch, and a forwarding rule must be deployed in the TCAM. TCAM is both expensive and power-hungry. Usually, the SDN switch can be equipped with limited TCAM, so the forwarding rules that can be deployed in SDN switch are limited. We let C_u denote the TCAM size of switch u . In addition, data center link capacity is limited, and we let $B_{(u,v)}$ be the link capacity on link $(u, v) \in E$. For each link $(u, v) \in E$, $p(u, v)$ represents the power consumption of link (u, v) and $p(u)$ represents the power consumption of switch u .

There are many applications in the data center, such as applications that use MapReduce for data analysis. Applications may transfer data between servers frequently, and the data is referred to as intra-data-center traffic [14]. Data center network flows are represented as the set K , each flow $k \in K$. We use source–destination pairs to represent traffic demands, which are denoted as $K = \{(s_1, d_1, r_1), (s_2, d_2, r_2), \dots, (s_k, d_k, r_k)\}$. For each demand $(s_k, d_k, r_k) \in K$ — s_k, d_k, r_k stand for source, destination, and demand of flow k , respectively. In addition, we define the variable f_{uv}^k to represent the amount of flow k that is routed through link (u, v) . Usually, DCN infrastructures use abundant network devices to provide routing and forwarding services to meet data center peak traffic demands. Data center traffic is much lower than peak traffic load most of the time, which leaves a great many of switches and links to keep idle or under-utilized. Typically, the average utilization rate of a data center is only about 20% to 30% [31]. These idle and under-utilized network devices still consume considerable energy. SDN technology can help us to choose the routing paths in real-time to meet the traffic demands and shut down redundant network devices, such as switches and links. Therefore, data center network energy optimization aims at minimizing activated switches and links to meet traffic demands. At the same time, we plan to use multipath routing to improve the QoS satisfaction ratio of flow allocation, especially in the case of high traffic load.

3.2. Problem Formulation

Next, we formulate the SD-DCN energy consumption optimization problem as an Integer Linear Programming (ILP) model. To better describe the problem, we define some binary variables for every switch, link, and forwarding rule deployment. Let $X(u, v)$ be a binary variable to indicate if link (u, v) is powered on, i.e., $X(u, v) = 1$ if (u, v) is powered on, otherwise $X(u, v) = 0$. That is,

$$X(u, v) = \begin{cases} 1, & \text{if } (u, v) \text{ is powered on,} \\ 0, & \text{otherwise.} \end{cases}$$

Similarly, $Y(u)$ is used to indicate if switch u is powered on. For each network flow that goes through a switch, a forwarding rule must be stored in the TCAM of the switch to determine the next hop for data forwarding. Note that if a flow is split at a switch, we need to deploy a rule for each subflow to describe multipath routing action in this switch. Let x_{uv}^k be a binary variable to indicate whether flow k goes through link (u, v) or not. If flow k goes through link (u, v) , a rule needs to be installed at switch u for flow k on outgoing link (u, v) . In the case of multipath routing, for each flow k , $\sum_{v \in N} x_{uv}^k$ represents all rules that are installed in switch u due to the flow k which is split at switch u . Therefore, we can use variable x_{uv}^k to represent the usage of rules in switch u . Obviously, we have $x_{uv}^k = 1$ if $f_{uv}^k > 0$ on link (u, v) . That is,

$$x_{uv}^k = \begin{cases} 1, & \text{if } f_{uv}^k > 0, u \in S, \\ 0, & \text{otherwise.} \end{cases}$$

The above expression is a nonlinear one, and cannot be solved directly by the optimization solver. We can better represent the constraint relationship above in a linear form as follows:

$$\frac{f_{uv}^k}{M} \leq x_{uv}^k \leq M f_{uv}^k,$$

where M is a sufficiently large number, e.g., 10^5 .

Based on the definition above, we describe the SD-DCN energy optimization problem as follows:

$$\min : \sum_{(u,v) \in E} p(u,v)X(u,v) + \sum_{u \in S} p(u)Y(u).$$

Subject to:

$$\frac{f_{uv}^k}{M} \leq x_{uv}^k \leq Mf_{uv}^k, \tag{1}$$

$$\sum_{v \in N} f_{uv}^k - \sum_{v \in N} f_{vu}^k = \begin{cases} r_k, & \text{if } u = s_k, \\ -r_k, & \text{if } u = d_k, \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

$$\sum_{k \in K} (f_{uv}^k + f_{vu}^k) \leq B_{(u,v)}X(u,v), \forall (u,v) \in E, \tag{3}$$

$$\sum_{k \in K} \sum_{(u,v) \in E} x_{uv}^k \leq C_u Y(u), \forall u \in S, \tag{4}$$

$$X(u,v) \in \{0,1\}, Y(u) \in \{0,1\}, x_{uv}^k \in \{0,1\},$$

$$f_{uv}^k \geq 0, \forall (u,v) \in E, k \in K.$$

Our objective is to minimize the energy consumption of data center network devices, including both switches and links. In the objective function, $p(u,v)$ and $p(u)$ are constants. To maximize the use of DCN data transmission potential, the model adopts a multipath routing strategy. Each flow may be reaching the destination node through different paths from the source node. The flow conservation constraints are given by (2). The link capacity limits the total data that a link can accommodate. Constraint (3) states that the network traffic on link (u,v) must be less than or equal to the link capacity B_{uv} . Because the network model is an undirected graph, we need to consider both the link (u,v) and the link (v,u) . As for the constraint (4), for one thing, if the switch u is powered on, that is, $Y(u) = 1$, the constraint can be described as the TCAM size constraint at this time. For another, if the switch u is powered off, that is, $Y(u) = 0$, we can easily find that the available size of TCAM is 0. We have $x_{uv}^k \equiv 0$, and further derive $f_{uv}^k \equiv 0$ from constraint (1). At this time, there is no flow going through the switch u .

4. Algorithm Design

The optimization problem described in the previous section can be solved by optimization software such as CPLEX [32] and Gurobi [33], but the computation time increases exponentially with the increase of network size. In a real network scenario with tens to hundreds of nodes, it is difficult to calculate an optimal solution in a limited time, especially with a large number of traffic demands. We propose a strategy-based Minimum Energy Consumption (MEC) heuristic algorithm, which can be effectively applied to large-scale networks. The procedure is shown in Algorithm 1. In this paper, we use three different strategies: Random-order Demand (RD); Biggest Demand First (BDF); and Smallest Demand First (SDF). When we choose the Smallest Demand First (SDF) strategy, the mice flows will be scheduled first. The Biggest Demand First (BDF) strategy is the opposite.

Essentially, our proposed strategy-based MEC algorithm allocates traffic demands in a specific order. For each flow, our algorithm selects paths from the source node to the destination node under the link capacity and the switch TCAM size constraints, and each path selection is aimed at minimizing energy consumption. When our algorithm schedules a flow, it first selects the path in the switches and links that have been turned on. To ensure the successful transmission of traffic, it inevitably needs to turn on additional switches and links to support the path selection. Therefore, the quality of the algorithm will be affected by the ordering of traffic demands.

We find that it takes a lot of time to select the optimal path for network flows, especially in large-scale DCNs. Fortunately, the topologies of the data center network have their characteristics, such as the Fat-Tree architecture, where any pair of servers are connected in a certain way. We can calculate the set of candidate paths in advance and call them directly when some paths are needed. With the method, we can effectively reduce the computational time during algorithm execution. When calculating candidate paths, for small-scale networks, such as 4-pod Fat-Tree, we can calculate all possible paths (without loops). Let us assume that the network contains V switches. If we calculate all the paths in the network, then the time complexity of the algorithm is $O(V!)$ [34]. It is not advisable to compute all paths in medium and large networks, such as 8-pod Fat-Tree. Therefore, we set up the search $depth = 6$ for the search candidate path, that is, the shortest path set between source and destination.

Algorithm 1 Strategy-based Minimum Energy Consumption Heuristic Algorithm

Input: graph $G = (N, E)$, TCAM size C , link capacity B , flow demand F

Output: activated link set $aLink$, activated switch set $aSwitch$, flow path set $fPath$

```

1: Find candidate path information to candidatePath
2: Initialize:  $fPath = \emptyset$ ,  $aSwitch = \emptyset$ , and  $aLink = \emptyset$ 
3:  $demand = sortFlowDemand(strategy, F)$ 
4: for all  $(s_k, d_k, r_k)$  in demand do
5:   search the candidate path set pathX for  $(s_k, d_k, r_k)$  from candidatePath
6:   while  $r_k \neq 0$  do
7:     for  $p$  in pathX do
8:        $i = 0$ 
9:        $activePower[i] = getNewlyActivePower(G, p)$ 
10:       $i = i + 1$ 
11:     end for
12:      $index = getIndex(argmin(activePower))$ 
13:      $path = pathX[index]$ 
14:     obtain path  $pTCAM$  and path  $pCapacity$ 
15:     if  $pTCAM == 0$  or  $pCapacity == 0$  then
16:        $candidatePath = candidatePath \setminus \{path\}$ 
17:       continue
18:     end if
19:     if  $pCapacity \geq r_k$  then
20:        $r_k = 0$ 
21:     else
22:        $r_k = r_k - pCapacity$ 
23:     end if
24:     Let  $fPath = fPath \cup \{path\}$ ,  $aSwitch = aSwitch \cup \{getSwitch(path)\}$ ,  $aLink =$ 
        $aLink \cup \{getLink(path)\}$  and update link capacity and TCAM size)
25:   end while
26: end for

```

In Algorithm 1, we first use the data center network topology information to calculate the candidate path set *candidatePath* (line 1). The *candidatePath* saves the candidate paths for all source and destination pairs in the data center. In line 3, we can use different sorting strategies for the traffic demands set F , namely, RD, BDF, and SDF. For each traffic demand, e.g., from s_k to d_k , we can find out the candidate path set *pathX* from *candidatePath*, as shown in line 5. To schedule traffic demand successfully, the appropriate path is selected from *pathX* until the flow is scheduled to be completed (line 6). In continuous traffic scheduling, the algorithm will inevitably turn on new switches and links to meet traffic demands. To consume as little energy as possible, we traverse the path set *pathx* and

calculate the newly activated energy consumption of each path. Next, we find the path corresponding to the minimum value of the newly activated energy consumption set *activePower* (lines 7–13). Since the switch TCAM size and link capacity on different paths have different limits, we need to take these limitations into account. *pTCAM* represents the minimum of the remaining TCAM size in all switches in the path and *pCapacity* represents the minimum of all link capacity in the path, which are shown in line 14. If the TCAM size of a switch or the link capacity is used up in the path, delete the candidate path from the *candidatePath* and go to the next loop (lines 15–18). Otherwise, the path can then be allocated for traffic demand (s_k, d_k, r_k) , and if traffic demand can be accommodated by link capacity, all r_k shall be routed by the path (lines 19–20). If the available capacity of the path is less than the traffic demand r_k , the algorithm greedily uses up the current path capacity and splits the remaining flow to the next candidate path for routing (line 22). Finally, we will update the flow path set *fPath*, the active switch set *aSwitch* and the active link set *aLink*. Simultaneously, we shall update the statuses of the entire data center network, as shown in line 24.

5. Performance Evaluation

We design simulation experiments to evaluate the performance of our proposed algorithm. Fat-Tree is the most popular switch-centric data center network architecture. We use 4-pod and 8-pod Fat-Tree data center network topologies in our simulation experiments. There are 36 nodes (composed of 20 switches and 16 hosts) and 24 links in 4-pod Fat-Tree; and 208 nodes (composed of 80 switches and 128 hosts) and 192 links in 8-pod Fat-Tree. Unless otherwise specified, the link capacity of the data center network is set as 1 Gbps. TCAM sizes of switches are generated in [250, 1250] uniformly at random [14].

In [35], the authors first give the definition of elephant flow and mice flow—most (e.g., 80%) of the traffic is actually carried by only a small number of connections (elephants), while the remaining large amount of connections are very small in size or lifetime (mice). Some related work [36,37] detects the elephant flows or mice flows based on transferred bytes, e.g., a flow transferring more than 1 MB of data is regarded as elephant flow. The other is to use the flow rate to detect elephant flows and mice flows [38,39], e.g., the rate of the elephant flow is within 1–10 Mbps.

Similar to previous works [38,39], we use the flow rate to distinguish between elephant flow and mice flow. The traffic demands of mice flows and elephant flows are generated uniformly at random in [100 Kbps, 1 Mbps] and [1 Mbps, 100 Mbps], respectively. Since the number of elephant flows and mice flows is different, and the mice flows make up the vast majority, the proportion of the elephant flows and the mice flows was chosen as 1:9 and 2:8 in our simulation experiment. The traffic demands are generated in an all-to-all manner. The algorithm was implemented and run in Python. All experiments were performed on a machine with a 3.0 GHz Intel i5-7400 CPU and 4 GB of RAM.

5.1. Energy Savings

Under low traffic load and medium traffic load, we ensure that all traffic demands are allocated successfully or close to 100% to evaluate the energy savings of the algorithm. Energy-saving is achieved by turning off inactive switches and links through the SDN controller. To illustrate the energy savings of the algorithm, we define the energy-saving percentage as an indicator of energy savings:

$$ES = \left(1 - \frac{|X| \cdot p(u, v) + |Y| \cdot p(u)}{m \cdot p(u, v) + n \cdot p(u)} \right) \times 100\%,$$

where $|X|$ and $|Y|$ represent the used links and nodes, respectively; and m and n represent the total number of links and switches in the network, respectively. We assume each link (u, v) has the same power consumption $p(u, v) = 0.6$ kW and each switch consumes the same power $p(u) = 3$ kW [40].

To evaluate the efficiency of our proposed algorithm, we compare the minimum switch activation multipath routing algorithm in [14] and calculate the energy-saving percentage based on the above energy-saving evaluation indicator. Since the order of traffic demands in [14] is random, we name this

algorithm as the Minimum Switch Activation multipath routing algorithm based on the strategy of the Random-order Demand (RD-MSA), which is compared with the Minimum Energy Consumption algorithm under the strategy of Random-order Demand (RD-MEC) presented in this paper. In our experiment, the percentage of energy savings is a function of the number of flows. Based on the characteristics of data center traffic, we choose the proportion of elephant flows and mice flows as 1:9 and 2:8 to generate different flow numbers. For making the evaluation of the algorithm in energy savings more effective, the number of flows is selected based on the fact that the traffic demands are allocated successfully or close to 100%. Therefore, we choose the number of flows varying from 100 to 600 and 1000 to 3500 for the 4-pod Fat-Tree and the 8-pod Fat-Tree topology, respectively. We use the optimization solver to solve the optimal solution of energy savings in 4-pod Fat-Tree. Due to the increase in network size and the number of flows in 8-pod Fat-Tree, it takes several weeks or more to solve the optimal solution in our experimental platform. We only solve the optimal solution of energy savings in 4-pod Fat-Tree. Our experiments are conducted 20 times to obtain the average energy savings, as shown in Figures 2 and 3.

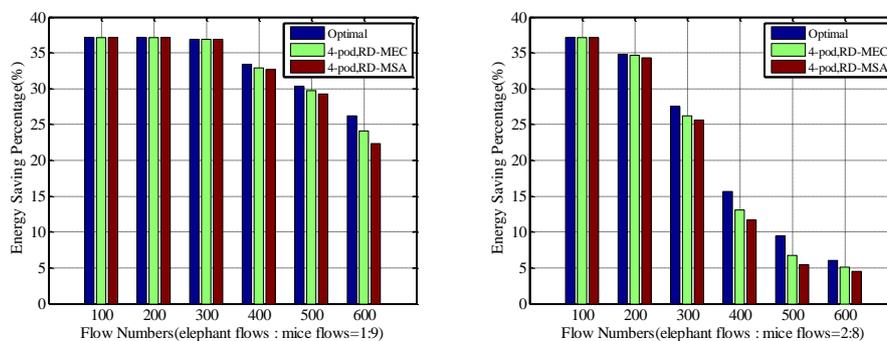


Figure 2. 4-pod Fat-Tree energy savings percentage.

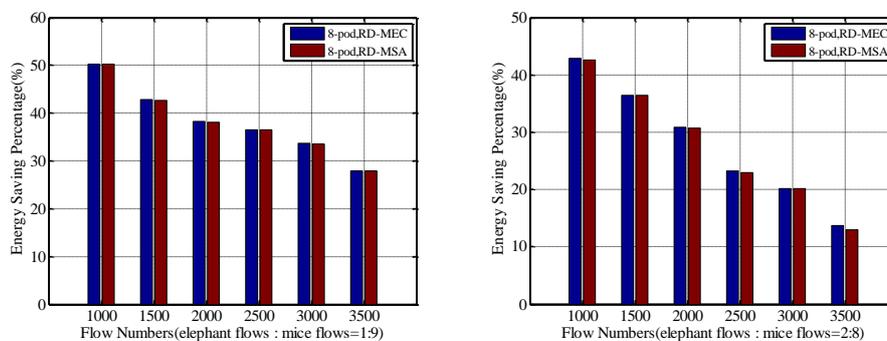


Figure 3. 8-pod Fat-Tree energy savings percentage.

In 4-pod, our algorithm RD-MEC saves about 37% energy when the proportion of elephant flows and mice flows is 1:9 and the flow number is less than 300; and saves about the same percentage when the proportion of elephant flows and mice flows is 2:8 and the flow number is 100. It can be seen from Figure 2 that our algorithm RD-MEC is better than the algorithm RD-MSA and closer to the optimal solution. In 8-pod, our algorithm RD-MEC saves about 50% of energy when the proportion of elephant flows and mice flows is 1:9 and the flow number is 1000; and about 42% of the energy percentage saved when the proportion of elephant flows and mice flows is 2:8 and the flow number is 1000. Our algorithm can reduce energy consumption, as it consolidates traffic flows onto a small network subset. In addition, the energy-saving percentage shows a decreasing trend with the increase of flow number. The reason for this is that more switches and links need to be activated to meet the increased traffic demands.

It can also be found that more energy can be saved when the proportion of elephant flows and mice flows is 1:9 at the same flow number. For example, in 4-pod, our algorithm RD-MEC can save about 24% energy when the proportion of elephant flows and mice flows is 1:9 and the flow number is 600; and only saves about 5% energy in the proportion of 2:8. It can be predicted that all network devices have to turn on to guarantee the QoS satisfaction. We can infer that 8-pod can save more energy than 4-pod under the same proportion and flow number. As a result, relatively large networks can gain more energy savings under the same flow number settings.

In summary, our algorithm RD-MEC is not worse than the algorithm RD-MSA in both 4-pod and 8-pod. In most cases, the algorithm RD-MEC is slightly better than RD-MSA and close to the optimal solution. This is attributed to the fact that our algorithm RD-MEC considers the energy consumption of links additionally, while the algorithm RD-MSA only considers the energy consumption of nodes. For each flow, our algorithm finds out all possible paths and selects the path with minimum newly-activated energy consumption.

5.2. QoS Satisfaction Ratio

We define the QoS satisfaction ratio as the ratio of the number of flows that are successfully allocated to the total number of flows. In many scenarios, the QoS satisfaction ratio is a very important performance indicator. For example, in real-time voice and video transmission, network congestion data retransmission will lead to a bad user experience. Under high traffic load, the result of our test indicates that some traffic demands cannot be allocated to a valid path due to the limitation of network bandwidth resources and switch TCAM size, so traffic transmission fails and QoS satisfaction ratio decreases. In this case, we can hardly get energy savings because traffic demands of the entire network are already full, and the switches and links have to turn on to guarantee the QoS satisfaction. Therefore, we compare the performance of the algorithm on the QoS satisfaction ratio in the case of high traffic load. Firstly, we compare the QoS satisfaction ratio of the MEC algorithm under the three strategies of Random-order Demand (RD), Biggest Demand First (BDF), and Smallest Demand First (SDF), named RD-MEC, BDF-MEC, and SDF-MEC, respectively.

5.2.1. QoS Satisfaction Ratio of Total Flows

To evaluate the QoS satisfaction ratio of total flows (By default, we call it QoS satisfaction ratio in this section.), we chose the high traffic load. Accordingly, we chose the number of flows varying from 600 to 1500 and 3000 to 12,000 for the 4-pod Fat-Tree and the 8-pod Fat-Tree topology, respectively. In each setting, 20 instances are run to obtain the QoS satisfaction ratio of three strategies in Figures 4 and 5. It is easy to observe that the SDF-MEC is the best in QoS satisfaction ratio, followed by RD-MEC under three different strategies. The two algorithms have a relatively stable trend of decreasing QoS satisfaction ratio with the increase of flow number. The worst is BDF-MEC. The reason is that the relatively small number of flows will saturate the network load when the maximum traffic demand is scheduled first, and the capacity of some links has been exhausted, which makes the subsequent large number of mice flows unable to be successfully routed. We can notice that the QoS satisfaction ratio of MEC algorithm is gradually decreasing under the three different strategies in 4-pod or 8-pod with the increase of flow number. This is because some switches and links are fully loaded in the network, which leads to subsequent flows being unable to be allocated successfully. In addition, the trend of decreasing QoS satisfaction ratio is gentler than the proportion 2:8 when the proportion of elephant flows and mice flows is 1:9. The reason for this is that the throughput increases in the network when the proportion of elephant flows and mice flows is 2:8. As a result, the network load is preferentially saturated. Thus, the subsequent flows cannot be successfully allocated, which reduces the QoS satisfaction ratio.

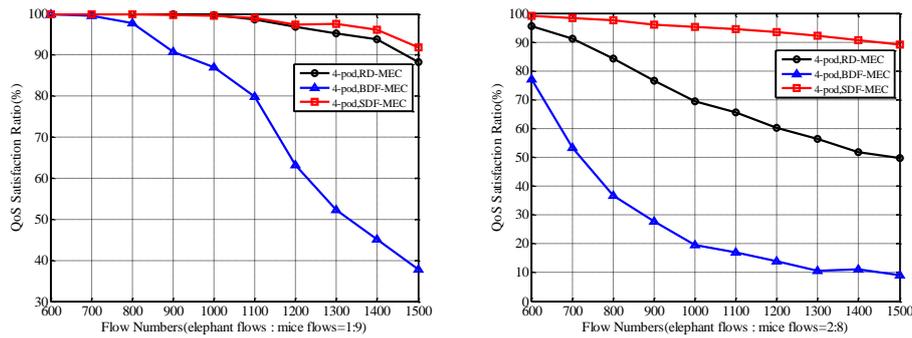


Figure 4. QoS satisfaction ratio of Random-order Demand Minimum Energy Consumption (RD-MEC), Biggest Demand First Minimum Energy Consumption (BDF-MEC), and Smallest Demand First Minimum Energy Consumption (SDF-MEC) in 4-pod Fat-Tree.

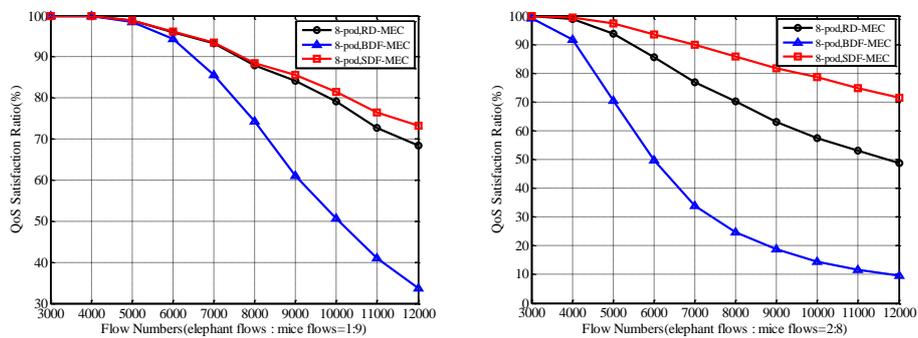


Figure 5. QoS satisfaction ratio of RD-MEC, BDF-MEC, and SDF-MEC in 8-pod Fat-Tree.

Moreover, we compare the QoS satisfaction ratio of the SDF-MEC and the RD-MSA proposed in [14]. The number of flows is set the same as in the previous experiment. Our experiments are conducted 20 times to obtain the average QoS satisfaction ratio of the SDF-MEC and the RD-MSA, as shown in Figures 6 and 7. It is easy to observe that the proposed algorithm SDF-MEC is significantly better than RD-MSA regardless of 4-pod or 8-pod. For example, the algorithm SDF-MEC in QoS satisfaction ratio is at least 25% higher than the algorithm RD-MSA when the proportion of elephant flows and mice flows is 2:8 and the flow number is 1000 in the 4-pod Fat-Tree topology. In addition, the QoS satisfaction ratio of the two algorithms decreases gradually with the increase of flow number. The algorithm SDF-MEC has a more smoothly decreasing trend, which guarantees the performance in terms of QoS satisfaction ratio effectively.

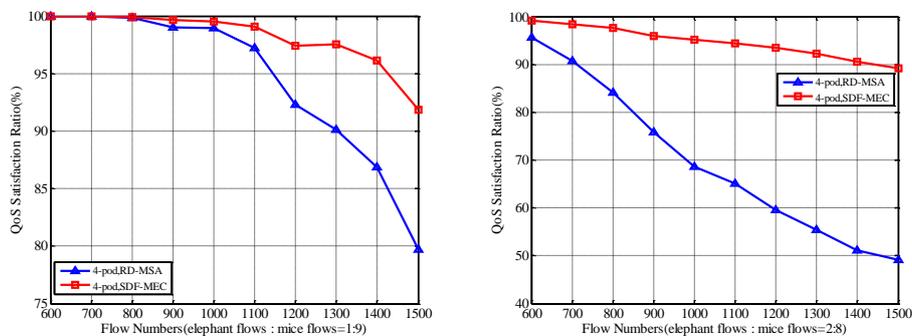


Figure 6. QoS satisfaction ratio of Random-order Demand minimum switch activation (RD-MSA) and SDF-MEC in 4-pod Fat-Tree.

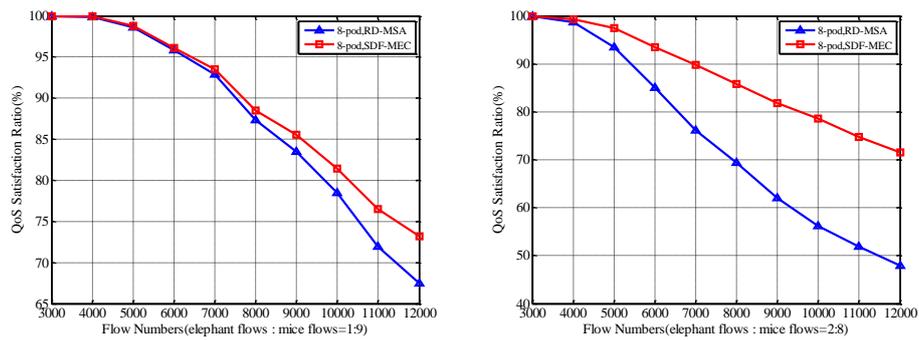


Figure 7. QoS satisfaction ratio of RD-MSA and SDF-MEC in 8-pod Fat-Tree.

5.2.2. QoS Satisfaction Ratio of Mice Flows and Elephant Flows

Then, we discuss the QoS satisfaction ratio of mice flows and elephant flows, respectively. We first describe the QoS satisfaction ratio of mice flows. As shown in Figures 8 and 9, the SDF-MEC can achieve the highest QoS satisfaction ratio in both 4-pod and 8-pod when the proportion of elephant flows and mice flows are in both 1:9 and 2:8. The results of RD-MSA and RD-MEC are similar, but RD-MEC is slightly better than RD-MSA. The BDF-MEC is worst in QoS satisfaction ratio. In other words, the QoS satisfaction ratio of mice flows is consistent with the QoS satisfaction ratio of total number of flows.

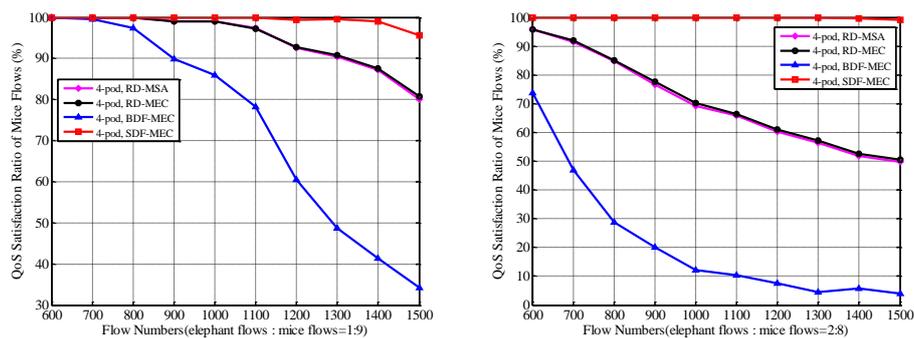


Figure 8. QoS satisfaction ratio of mice flows in 4-pod Fat-Tree.

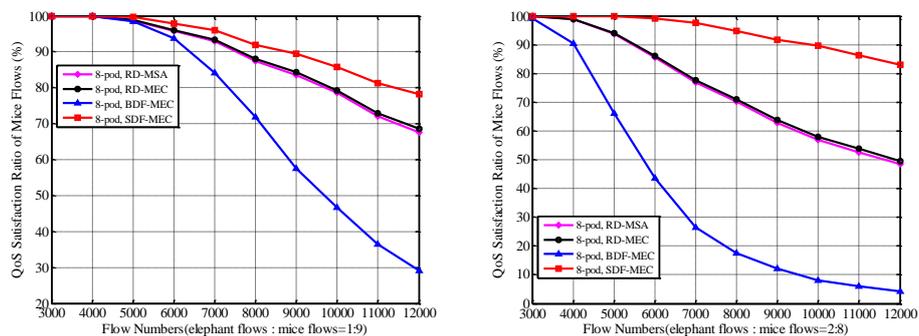


Figure 9. QoS satisfaction ratio of mice flows in 8-pod Fat-Tree.

The experimental results of the QoS satisfaction ratio of elephant flows are shown in Figures 10 and 11. In 4-pod, our algorithm RD-MEC is slightly better than RD-MSA and achieves the best QoS satisfaction ratio of elephant flows when the proportion of elephant flows and mice flows is 1:9. When the proportion is 1:9, SDF-MEC is the worst, but the result is the opposite when the proportion is 2:8. This is because the number of mice flows is more when the proportion is 1:9. When the SDF strategy is selected, the mice flows are scheduled first and occupy most of the network

resources, which results in the lowest QoS satisfaction ratio of the subsequent arriving elephant flows. When the proportion is 2:8, the number of mice flows is less than the proportion of 1:9. At the same time, the QoS satisfaction ratio of mice flows is 100% or close to 100% (Figure 8) when we use the strategy of SDF. It indicates that there are more available resources in the network to schedule the subsequent arriving elephant flows. In this case, the SDF-MEC can achieve the highest QoS satisfaction ratio of elephant flows.

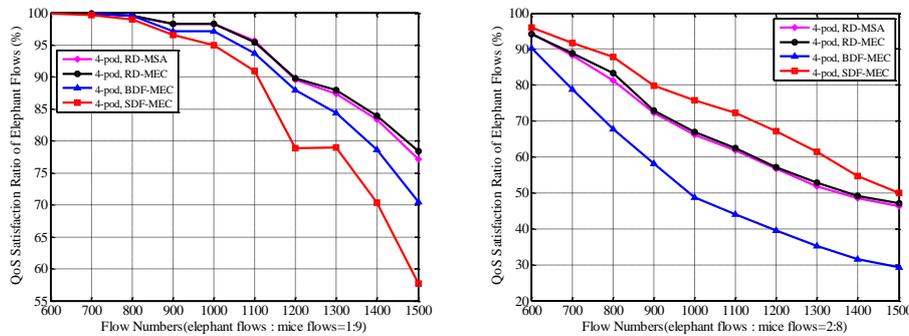


Figure 10. QoS satisfaction ratio of elephant flows in 4-pod Fat-Tree.

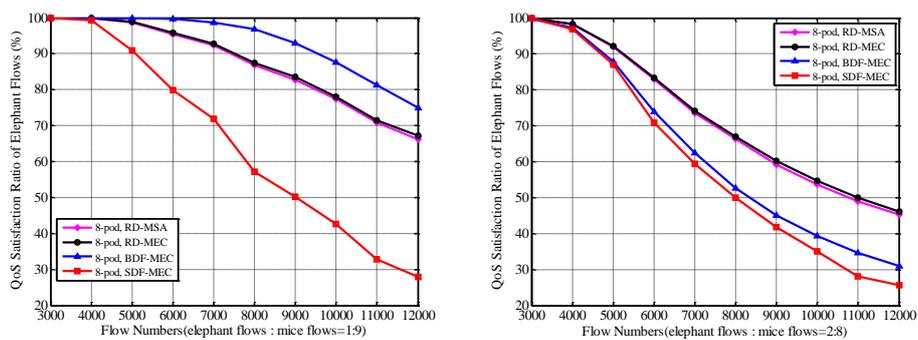


Figure 11. QoS satisfaction ratio of elephant flows in 8-pod Fat-Tree.

In 8-pod, the BDF-MEC’s QoS satisfaction ratio of the elephant flows is best when the proportion of elephant flows and mice flows is 1:9, and the experimental results of the RD-MEC and the RD-MSA are similar, but the RD-MEC is slightly better than the RD-MSA when the proportion of elephant flows and mice flows is 2:8. The experimental results of the SDF-MEC are the worst, because a large number of mice flows occupy most of the network resources and the subsequent arrival of the elephant flows cannot be successfully routed. Furthermore, the BDF-MEC is better than the SDF-MEC. This is because the elephant flows being scheduled first can guarantee that most of elephant flows achieve successful transmission.

In summary, the SDF-MEC can achieve the highest QoS satisfaction ratio of mice flows in both 4-pod and 8-pod. In terms of QoS satisfaction ratio of elephant flows, the BDF-MEC is better than the SDF-MEC in most cases, and one of the strategy-based algorithms can achieve the best QoS satisfaction ratio of elephant flows.

6. Conclusions

In this paper, we study the QoS-guaranteed energy optimization problem in data center networks, which benefits from the flexible dynamic control of network devices by Software Defined Networking. We can selectively shut down some switches and links to achieve energy optimization. In this paper, we present three different strategies based on the order of flow scheduling, namely, Random-order Demand (RD), Biggest Demand First (BDF), and Smallest Demand First (SDF). Then, we propose a strategy-based Minimum Energy Consumption (MEC) heuristic algorithm based on different strategies.

We also take the unique features of SDN, such as the limited size of TCAM, flow conservation constraints, and link capacity constraints into consideration, and formulate the SD-DCN energy consumption optimization problem as an Integer Linear Programming (ILP) model.

Simulation results show that our strategy-based heuristic algorithm can achieve better network performance in terms of QoS satisfaction ratio under the high traffic load and bring efficient energy savings under the low traffic load and medium traffic load. More specifically, in terms of energy savings, the RD-MEC can save energy effectively under the conditions of low traffic load and medium traffic load. In terms of QoS satisfaction ratio of total flows and mice flows, the SDF-MEC is superior to RD-MEC, BDF-MEC, and RD-MSA under high traffic load. As for QoS satisfaction ratio of elephant flows, the BDF-MEC is better than the SDF-MEC in most cases, and one of the strategy-based algorithms can achieve the best QoS satisfaction ratio of elephant flows. In our future work, we will consider other QoS parameters, such as end-to-end delay. In addition, we will investigate the performance of our strategy-based heuristic in the backbone network and analyze the effect of TCAM size.

Author Contributions: Conceptualization, X.G., Z.L. (Zebin Lu), J.L., Y.H., and Z.L. (Zhengfa Li); Methodology, X.G., Z.L. (Zebin Lu) and Z.L. (Zhengfa Li); Resources, J.L. and Y.H.; Supervision, X.G.; Writing—original draft, Z.L. (Zebin Lu); Writing—review and editing, X.G. and S.D.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No.: 61802328) and Hunan Provincial Innovation Foundation For Postgraduate (Grant No.: CX2015B210).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Koomey, J. *Growth in Data Center Electricity Use 2005 to 2010*; Technical Report; Analytics Press: El Dorado Hills, CA, USA, 2011.
2. Gao, P.X.; Curtis, A.R.; Wong, B.; Keshav, S. It's not easy being green. *ACM SIGCOMM Comput. Commun. Rev.* **2012**, *42*, 211–222. [[CrossRef](#)]
3. Abts, D.; Marty, M.R.; Wells, P.M.; Klausler, P.; Liu, H. Energy proportional datacenter networks. *ACM SIGARCH Comput. Archit. News* **2010**, *38*, 338–347. [[CrossRef](#)]
4. Al-Fares, M.; Loukissas, A.; Vahdat, A. A scalable, commodity data center network architecture. *ACM SIGCOMM Comput. Commun. Rev.* **2008**, *38*, 63–74. [[CrossRef](#)]
5. Greenberg, A.; Hamilton, J.R.; Jain, N.; Kandula, S.; Kim, C.; Lahiri, P.; Maltz, D.A.; Patel, P.; Sengupta, S. VL2: A scalable and flexible data center network. *ACM SIGCOMM Comput. Commun. Rev.* **2009**, *39*, 51–62. [[CrossRef](#)]
6. Guo, C.; Wu, H.; Tan, K.; Shi, L.; Zhang, Y.; Lu, S. Dcell: A scalable and fault-tolerant network structure for data centers. *ACM SIGCOMM Comput. Commun. Rev.* **2008**, *38*, 75–86. [[CrossRef](#)]
7. Guo, C.; Lu, G.; Li, D.; Wu, H.; Zhang, X.; Shi, Y.; Tian, C.; Zhang, Y.; Lu, S. BCube: A high performance, server-centric network architecture for modular data centers. *ACM SIGCOMM Comput. Commun. Rev.* **2009**, *39*, 63–74. [[CrossRef](#)]
8. McKeown, N.; Anderson, T.; Balakrishnan, H.; Parulkar, G.; Peterson, L.; Rexford, J.; Shenker, S.; Turner, J. OpenFlow: Enabling innovation in campus networks. *ACM SIGCOMM Comput. Commun. Rev.* **2008**, *38*, 69–74. [[CrossRef](#)]
9. Hu, C.; Yang, J.; Gong, Z.; Deng, S.; Zhao, H. DesktopDC: Setting all programmable data center networking testbed on desk. *ACM SIGCOMM Comput. Commun. Rev.* **2014**, *44*, 593–594. [[CrossRef](#)]
10. Gao, X.; Xu, Z.; Wang, H.; Li, L.; Wang, X. *Reduced Cooling Redundancy: A New Security Vulnerability in a Hot Data Center*; NDSS: New York, NY, USA, 2018.
11. Tu, R.; Wang, X.; Yang, Y. Energy-saving model for SDN data centers. *J. Supercomput.* **2014**, *70*, 1477–1495. [[CrossRef](#)]
12. Yoon, M.S.; Kamal, A.E. Power minimization in fat-tree SDN datacenter operation. In Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, USA, 6–10 December 2015; pp. 1–7.

13. Wei, M.; Zhou, J.; Gao, Y. Energy efficient routing algorithm of software defined data center network. In Proceedings of the 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), Guangzhou, China, 6–8 May 2017; pp. 171–176.
14. Zeng, D.; Yang, G.; Gu, L.; Guo, S.; Yao, H. Joint optimization on switch activation and flow routing towards energy efficient software defined data center networks. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 23–27 May 2016; pp. 1–6.
15. Lorenz, D.H.; Orda, A. Optimal partition of QoS requirements on unicast paths and multicast trees. *IEEE/ACM Trans. Netw.* **2002**, *10*, 102–114. [[CrossRef](#)]
16. Korkmaz, T.; Krunz, M. A randomized algorithm for finding a path subject to multiple QoS requirements. *Comput. Netw.* **2001**, *36*, 251–268. [[CrossRef](#)]
17. Chen, S.; Nahrstedt, K. Distributed quality-of-service routing in ad hoc networks. *IEEE J. Sel. Areas Commun.* **1999**, *17*, 1488–1505. [[CrossRef](#)]
18. Gong, S.; Chen, J.; Kang, Q.; Meng, Q.; Zhu, Q.; Zhao, S. An efficient and coordinated mapping algorithm in virtualized SDN networks. *Front. Inf. Technol. Electron. Eng.* **2016**, *17*, 701–716. [[CrossRef](#)]
19. Orda, A.; Sprintson, A. Precomputation schemes for QoS routing. *IEEE/ACM Trans. Netw.* **2003**, *11*, 578–591. [[CrossRef](#)]
20. Lorenz, D.H.; Orda, A.; Raz, D. Optimal partition of QoS requirements for many-to-many connections. In Proceedings of the IEEE INFOCOM 2003 Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428), San Francisco, CA, USA, 30 March–3 April 2003; Volume 3, pp. 1670–1679.
21. Lorenz, D.H.; Orda, A. QoS routing in networks with uncertain parameters. *IEEE/ACM Trans. Netw.* **1998**, *6*, 768–778. [[CrossRef](#)]
22. Orda, A.; Sprintson, A. A scalable approach to the partition of QoS requirements in unicast and multicast. *IEEE/ACM Trans. Netw.* **2005**, *13*, 1146–1159. [[CrossRef](#)]
23. Wang, Z.; Crowcroft, J. Quality-of-service routing for supporting multimedia applications. *IEEE J. Sel. Areas Commun.* **1996**, *14*, 1228–1234. [[CrossRef](#)]
24. Xu, G.; Dai, B.; Huang, B.; Yang, J.; Wen, S. Bandwidth-aware energy efficient flow scheduling with SDN in data center networks. *Future Gener. Comput. Syst.* **2017**, *68*, 163–174. [[CrossRef](#)]
25. Heller, B.; Seetharaman, S.; Mahadevan, P.; Yiakoumis, Y.; Sharma, P.; Banerjee, S.; McKeown, N. Elastictree: Saving energy in data center networks. In Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation (NSDI), San Jose, CA, USA, 28–30 April 2010; Volume 10, pp. 249–264.
26. Jiang, H.P.; Chuck, D.; Chen, W.M. Energy-aware data center networks. *J. Netw. Comput. Appl.* **2016**, *68*, 80–89. [[CrossRef](#)]
27. Wang, X.; Wang, X.; Zheng, K.; Yao, Y.; Cao, Q. Correlation-aware traffic consolidation for power optimization of data center networks. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *27*, 992–1006. [[CrossRef](#)]
28. Li, D.; Shang, Y.; He, W.; Chen, C. EXR: Greening data center network with software defined exclusive routing. *IEEE Trans. Comput.* **2015**, *64*, 2534–2544. [[CrossRef](#)]
29. Huang, H.; Li, P.; Guo, S.; Ye, B. The joint optimization of rules allocation and traffic engineering in software defined network. In Proceedings of the 2014 IEEE 22nd International Symposium of Quality of Service (IWQoS), Hong Kong, China, 26–27 May 2014; pp. 141–146.
30. Wang, T.; Qin, B.; Su, Z.; Xia, Y.; Hamdi, M.; Fofou, S.; Hamila, R. Towards bandwidth guaranteed energy efficient data center networking. *J. Cloud Comput.* **2015**, *4*, 9. [[CrossRef](#)]
31. Gao, X.; Gu, Z.; Kayaalp, M.; Pendarakis, D.; Wang, H. ContainerLeaks: Emerging security threats of information leakages in container clouds. In Proceedings of the 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Denver, CO, USA, 26–29 June 2017; pp. 237–248.
32. IBM ILOG CPLEX Optimization Studio. Available online: <https://www.ibm.com/products/ilog-cplex-optimization-studio> (accessed on 10 September 2019).
33. Gurobi Optimizer Inc. Gurobi Optimizer Reference Manual, 2015. Available online: <http://www.gurobi.com> (accessed on 10 September 2019).
34. Zhu, H.; Liao, X.; de Laat, C.; Grosso, P. Joint flow routing-scheduling for energy efficient software defined data center networks: A prototype of energy-aware network management platform. *J. Netw. Comput. Appl.* **2016**, *63*, 110–124. [[CrossRef](#)]

35. Guo, L.; Matta, I. The war between mice and elephants. In Proceedings of the Ninth International Conference on Network Protocols, Riverside, CA, USA, 11–14 November 2001; pp. 180–188.
36. Curtis, A.R.; Mogul, J.C.; Tourrilhes, J.; Yalagandula, P.; Sharma, P.; Banerjee, S. DevoFlow: Scaling flow management for high-performance networks. *ACM SIGCOMM Comput. Commun. Rev.* **2011**, *41*, 254–265. [[CrossRef](#)]
37. Curtis, A.R.; Kim, W.; Yalagandula, P. Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection. *Infocom* **2011**, *11*, 1629–1637.
38. Liu, R.; Gu, H.; Yu, X.; Nian, X. Distributed flow scheduling in energy-aware data center networks. *IEEE Commun. Lett.* **2013**, *17*, 801–804. [[CrossRef](#)]
39. Jiang, J.W.; Lan, T.; Ha, S.; Chen, M.; Chiang, M. Joint VM placement and routing for data center traffic engineering. In Proceedings of the 2012 Proceedings IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012; pp. 2876–2880.
40. Wang, R.; Gao, S.; Yang, W.; Jiang, Z. Energy aware routing with link disjoint backup paths. *Comput. Netw.* **2017**, *115*, 42–53. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).