Joint Optimization on Switch Activation and Flow Routing towards Energy Efficient Software Defined Data Center Networks

Deze Zeng*, Guang Yang*, Lin Gu†, Song Guo‡ and Hong Yao*
*School of Computer Science, China University of Geosciences, Wuhan, China
†School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China
‡School of Computer Science and Engineering, University of Aizu, Aizuwakamatsu, Japan

***Abstract*—The rapid development of cloud computing has raised big concerns over the high energy consumption of modern data centers. To satisfy the ever increasing data traffic needs, the energy consumption of data center network (DCN) also takes a significant proportion. The newly emerging technology, Software Defined Networking (SDN), which allows flexible control of network devices, brings a new opportunity towards DCN energy optimization. In this paper, we investigate how to design an energy-efficient network management strategy with guaranteed satisfaction of network traffic demands in Software Defined Data Center Networks (SD-DCNs). To this end, three issues will be tackled: 1) the subset of switches that shall be activated, i.e., switch activation, 2) multi-path routing scheduling for all flows and 3) forwarding rule placement in SDN switches. They are jointly considered and formulated as an integer linear programming (ILP) problem. A heuristic algorithm to deal with its high computational complexity is proposed. Extensive simulation-based evaluations are conducted to validate the high efficiency of our algorithm.**

I. Introduction

The explosive demands on cloud services impose a heavy burden on modern data centers, especially on the energy consumption. Due to the huge energy consumption, a recent study [1] shows that electricity cost has become the dominant operational expenditure (OPEX), even surpassing the capital expenditure (CAPEX), e.g., the hardware cost, to data center providers. Natural Resources Defense Council pointed out that in 2013 US data centers consumed 91 billion kilowatt-hours, which are enough to power all the households in New York City over twice and are on-track to reach 140 billion kilowatt-hours by 2020[1]. Data center energy consumption optimization has become an emergent issue for researchers from both academia and industry.

In a typical data center, energy consumption mainly consists of two parts: servers and network devices (e.g., switches). Many efforts have been devoted to server energy optimization and many effective strategies have been proposed, e.g., [2]–[4]. On the other hand, it is shown that the network energy consumption accounts for as high as 20% of the total energy and shall not be ignored [5]. Modern DCNs are usually deployed in a hierarchical multi-rooted network topology, such as fat tree, DCell, VL2, Port-Land, and BCube. Fig. 1 shows a 4-pod fat tree with four hierarchical layers, including 4 core switches, 8 aggregation switches, 8 edge switches and 16 servers. It can be seen that many network devices are provisioned to support the data transferring between servers. Selectively deactivating part of these devices without violating the quality-of-service (QoS) has been proved as a promising solution for energy saving [6]–[9]. On the other hand, the hierarchical multi-rooted topology makes multiple paths possible between any two servers in the data center. To take the advantage of multiple paths in DCNs, multi-path routing [10] is highly recommended.

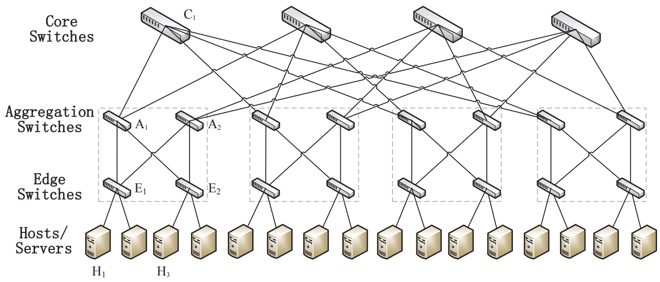


Fig. 1. Fat tree topology with 4 pods

Recently, Software Defined Networking (SDN) emerges as a new networking paradigm [11] and has been attracting much attention in the networking research community. Thanks to the decoupling of control plane and data plane, SDN introduces more flexible maintenance and controlling to DCN, e.g., flow routing, switch activation/deactivation, etc. As a result, a growing trend is to build software defined data center networks (SD-DCN) by incorporating SDN into DCN. Recent work [12], [13] has proved that SDN is indeed a feasible and promising solution for DCN energy optimization. Yet, we notice that existing studies only consider how to apply SDN, e.g., flow routing, for DCN energy optimization. Some inherent characteristics of SDN itself are usually overlooked. For example, it is well known that the ternary content addressable memory (TCAM), which stores forwarding rules,

[1]http://www.nrdc.org/energy/data-center-efficiency-assessment.asp

is considerably expensive and size-limited [14]. However, to enable multi-path routing for a flow in SD-DCN, each switch in the selected paths shall allocate one forwarding rule. Unlike traditional DCN, multi-path routing in SD-DCN is additionally constrained by the limited TCAM size.

Therefore, in this paper, we are motivated to investigate how to minimize the switch energy consumption by jointly considering switch activation, multi-path flow routing scheduling and forwarding rule placement, while guaranteeing the data center traffic demand satisfaction. Our main contributions are summarized as follows:

- To our best knowledge, we are the first to study minimum-energy switch activation for multi-path routing in SD-DCN with the consideration of the limitations of SDN itself. Specially, we first formulate the energy optimization problem as an integer linear programming (ILP) problem.
- To deal with the high computational complexity of solving ILP, we propose an efficient heuristic algorithm. Through extensive simulation based studies, we show the high efficiency of the proposed ILP-based algorithm by the fact that it produces near-optimal solutions with much less scheduling time.

The rest of the paper is organized as follows. Section II summarizes some recent related work. In Section III, we give the problem statement and system model. The ILP formulation and the heuristic algorithm are proposed in Sections IV and V, respectively. Section VI shows our simulation results and analysis. Finally, Section VII concludes our work.

II. Related Work

The huge energy consumption of DCN has attracted much attention recently in the community. Pioneering researchers have proposed many different methods to reduce the DCN energy consumption. Shang et al. [6] advocate using as few network devices as possible to provide the routing service and propose energy-aware routing to reduce the energy consumption in DCNs. Abts et al. [7] dynamically adjust the link rate according to the predicted bandwidth requirement to save energy consumption. Later, Want et al. [9] invent a correlation-aware power optimization (CARPO) algorithm that dynamically consolidates correlated data flows onto a subset of links and switches in a DCN and shuts down under-utilized network devices for energy savings. With the emergence of SDN, its flexibility makes it attractive to data center energy optimization. Heller et al. [8] design and implement ElasticTree using OpenFlow switches to dynamically activate network elements according to the data center traffic demands to reduce the energy consumption. Recently, Li et al. [13] leverage SDN technique to realize exclusive routing that enables a flow exclusively occupying the links along its routing path according to its priority, so as to save energy consumption compared with traditional fair-sharing routing. Most studies, as discussed above, mainly focus on exploring the SDN technology to manipulate the network devices in DCNs to reduce energy consumption. However, the inherent characteristics of SDN itself are usually overlooked. Although a recent study by Giroire et al. [15] considers the TCAM size limitation, the multi-path routing in DCNs is not explored. We are motivated to jointly address the two issues.

III. System Model and Problem Statement

In this paper, we consider a hierarchical DCN topology as an undirected graph $G = (N, E)$, where N includes both SDN-enabled (e.g., OpenFlow) switch set V and the host (i.e., servers) set H, i.e., $N = V \cup H$ and edge set E represents the links between the nodes in N. The switches and the hosts are initially interconnected according to the adopted topology (e.g., fat tree shown in Fig. 1). The data center links are capacity constrained. We consider an undirected link capacity model the same as [16], in which the flow in both directions share the capacity of a link. We use $C_{uv}, u, v \in N$ to denote the link capacity between nodes u and v. Generally, the link capacity at the core layer is larger than the ones at the aggregation layer and edge layer. In the SD-DCN, the switches are programmable and the forwarding rules are stored in TCAM, where each flow going through the switch must have one corresponding rule. It describes the actions (e.g., forwarding, header modifying, discarding, etc.) to deal with the flow packets. However, due to the expensive price of TCAM, the TCAMs of SDN switches are usually size-limited. The number of rules that can be put in switch $v \in V$ is constrained by its TCAM size S_v.

There are a lot of applications, e.g., data analysis applications using MapReduce, running in a data center. These applications may frequently require data transferring between servers. These data are generally called as intra-data-center traffic [17], imposing a huge burden on the underlying DCN. As one server may host multiple applications at the same time, many flows with different traffic demands may coexist between servers. We denote the flow set from host s to d as L^{sd}. Real trace based empirical studies [8], [17] show that the flow traffic demands vary over time. For example, the demand in the daytime is usually higher than the one at night. Let $F^{sd,k}, s, d \in H$ denote the demand of traffic $k \in L^{sd}$ from s to d in a period. For example, $F^{h_1 h_2,1}$ = 100Mbps indicates that the demand of flow 1 from host h_1 to h_2 is 100Mbps.

A DCN is usually over-provisioned to satisfy the peak traffic demands. However, with the consideration of the temporal characteristics discussed above, there is no need to always activate all the switches, especially when the traffic demands are extremely low at night. This provides an opportunity to save energy consumption by deactivating redundant switches. Accordingly, the flow routing shall be carefully scheduled on the sub-topology, which consists of only the activated switches, while at the same time the network traffic demands should be satisfied. Thanks to SDN technology, we are able to adjust the DCN topology and orchestrate the flow routing paths at runtime. A natural question arises as which switches shall be activated and how to route the flows in the sub-topology such that minimum number of switches are used.

IV. Problem Formulation

A. Traffic Demand Satisfaction Constraints

To fully explore the available transmission potential in a DCN, multi-path routing is widely applied. A flow therefore may be split into multiple sub-flows, which go through different paths from the source node to the intended destination. Let $f_{uv}^{sd,k}$ denote the flow over the link $(u,v) \in E$ carrying data of flow $k \in L^{sd}$ destined to host $d \in H$ from $s \in H$. We have the following constraints for flow conservation:

$$\sum_{v \in N} f_{sv}^{sd,k} - \sum_{v \in N} f_{vs}^{sd,k} = F^{sd,k}, \forall s, d \in H, k \in L^{sd}, \quad (1)$$

$$\sum_{w \in N} f_{vw}^{sd,k} - \sum_{u \in N} f_{uv}^{sd,k} = 0, \forall s, d \in H, k \in L^{sd}, v \in N, \quad (2)$$

$$\sum_{u \in N} f_{ud}^{sd,k} - \sum_{u \in N} f_{du}^{sd,k} = F^{sd,k}, \forall s, d \in H, \forall k \in L^{sd}, \quad (3)$$

referring to the constraints for flow $k \in L^{sd}$ on the source node $s \in H$, intermediate switch $v \in V$ and the destination $d \in H$, respectively.

B. Link Capacity Constraints

In multi-path routing, no matter how a flow is split or merged, the total data that can flow along a link is limited by its capacity. As we consider an undirected link capacity model, the overall flows, either from u to v or from v to u, on link (u,v) shall not exceed its capacity C_{uv}, i.e.,

$$\sum_{s \in H} \sum_{d \in H} \sum_{k \in L^{sd}} (f_{uv}^{sd,k} + f_{vu}^{sd,k}) \leq C_{uv}, \forall e_{uv} \in E. \quad (4)$$

C. Forwarding Rule Placement Constraints

According to the SDN rule policy, for any given flow going through a switch, there must be one forwarding rule in the TCAM to describe the flow handling action. Note that no matter how a flow is split or merged at a switch, we only need one rule to describe the multi-path routing action. We therefore define a binary variable $x_v^{sd,k}$ to denote whether flow k from s to d passing v or not. Obviously, we have $x_v^{sd,k} = 1$ if and only if there exists $f_{uv}^{sd,k} > 0$, i.e.,

$$x_v^{sd,k} = \begin{cases} 1, \text{ if } \exists f_{uv}^{sd,k} > 0, \\ 0, \text{ if } \forall e_{uv} \in E, f_{uv}^{sd,k} = 0. \end{cases} \quad (5)$$

The relationship described above can be represented in a linear form as

$$\frac{\sum_{e_{uv} \in E} f_{uv}^{sd,k}}{A} \leq x_v^{sd,k} \leq A \sum_{e_{uv} \in E} f_{uv}^{sd,k}, \quad (6)$$
$$\forall s, d \in H, k \in L^{sd}, v \in V,$$

where A is an arbitrary large number.

The number of rules that can be stored in switch $v \in V$ is constrained by its TCAM size S_v, provided the switch is activated. Otherwise, no rule can be stored in the switch. Therefore, we first define a binary variable y_v to denote whether switch v is on or not as

$$y_v = \begin{cases} 1, \text{ if switch } v \in V \text{ is on}, \\ 0, \text{ otherwise}. \end{cases} \quad (7)$$

and we further have

$$\sum_{s \in H} \sum_{d \in H} \sum_{k \in L^{sd}} x_v^{sd,k} \leq S_v \cdot y_v, \forall s, d \in H, v \in N. \quad (8)$$

Constraint (8) is explained as follows. On one hand, when switch v is powered on, i.e., $y_v = 1$, it describes the TCAM-size constraint. On the other hand, when $y_v = 0$, equivalently the available TCAM size can be regarded as 0, we have $x_v^{sd,k} \equiv 0, \forall s, d \in H, k \in L^{sd}$ from (8) and further derive $f_v^{sd,k} \equiv 0, \forall s, d \in H, k \in L^{sd}$ if $y_v = 0$ from (6), implying that no flow can pass through a deactivated switch.

D. An ILP Formulation

According to the definition in (7), our objective to minimize the number of switch for energy saving can be represented as $\min : \sum_{v \in Vs} y_v$. By summarizing all constraints discussed above, we can formulate the problem as an ILP problem as:

ILP:

$$\min : \sum_{v \in V} y_v,$$
$$\text{s.t.} : (1), (2), (3), (4), (6) \text{ and } (8).$$

V. Algorithm Design

It is computationally prohibitive to solve the ILP problem and get the optimal solution for large-scale cases. We therefore propose a computation-efficient heuristic algorithm in this section. The main idea of our algorithm is to select paths for each flow demand under the link capacity and TCAM size constraints, aiming at minimizing the current number of activated switches.

Our algorithm first requires calculating all the shortest paths between any pair of servers in the base topology. We notice that it is time consuming to search all these paths at runtime, especially for large-scale DCNs. Fortunately, the DCN topologies are usually regular and therefore it is possible to find out all shortest paths between any pair of servers in advance. Actually, the number of shortest paths is proportional to the number of core switches. We can save the obtained paths in a file or database for future reuse.

In our algorithm, we first retrieve the path information to T (see line 3). For each server pair, e.g., s and d, we can then find out the candidate path set $allPath$ from T, as shown in line 6. To ensure the traffic demand satisfaction, we then start to schedule the routing paths for flow $F^{sd,k}$ from $allPath$ until $F^{sd,k}$ gets fully satisfied, i.e., $F^{sd,k} == 0$ (line 7). To schedule a path for a sub-flow, we may inevitably need to activate some switches to support the selected path. In order to activate switches as less as possible, we always choose a path $path$ with the minimum number of newly activated switches for each sub-flow, as shown in line 8.

Algorithm 1 Minimum Switch Activation Multi-Path Routing

1: **INPUT**: DCN graph G, TCAM size S, flow demand matrix F
2: **OUTPUT**: Flow path set $FPath$, activated switch set SW
3: Retrieve path information to T
4: Initialize: $SW = \emptyset$, and $FPath = \emptyset$
5: **for all** $F^{sd,k}$ in F **do**
6: search the path candidate set $allPath$ for s and d from T
7: **while** $F^{sd,k} \neq 0$ **do**
8: select $path$ as the one with minimum number of newly activated switches from $allPath$
9: derive path TCAM size $pathTCAM$ and path capacity $pathCapacity$
10: **if** $pathTCAM == 0$ or $pathCapacity == 0$ **then**
11: $allPath = allPath \setminus \{path\}$
12: continue
13: **end if**
14: **if** $pathCapacity >= F^{sd,k}$ **then**
15: $F^{sd,k} = 0$
16: **else**
17: $F^{sd,k} = F^{sd,k} - pathCapacity$
18: **end if**
19: Let $SW = SW \bigcup \{getSwitch(path)\}$, $FPath = FPath \bigcup \{path\}$ and update the switch status (e.g., link and TCAM usage)
20: $allPath = allPath \setminus \{path\}$
21: **end while**
22: **end for**

Unfortunately, not all the chosen paths can be explored directly with the consideration of TCAM size and link capacity. We define the smallest TCAM size and the link capacity in the path as $pathTCAM$ and $pathCpacity$, respectively, which are first derived in line 9. If there exists one switch whose TCAM or one link whose capacity has been used up, the whole path cannot be chosen and we shall consider the next path candidate (lines 10-13). Otherwise, $path$ can be allocated for sub-flow $F^{sd,k}$. If the path capacity is sufficient to satisfy current flow demand $F^{sd,k}$, all $F^{sd,k}$ shall be routed through $path$ (line 15). If the flow demand exceeds the available path capacity, we greedily use up the capacity of current path first and let the next path to handle the residual flow (line 17). In either case, we shall update the path candidates $allPath$, flow path set $FPath$, activated switch set SW, and the switch statuses (e.g., available link capacity, TCAM size and forwarding rules), as shown in lines 19 and 20.

VI. Performance Evaluation

In this section, we present our simulation-based experimental results to investigate the efficiency of our proposed algorithm. Without loss of generality, we use fat tree as the base data center topology throughout this experiment. Unless other specified, the default link capacity is set as 1Gbps. The switches' TCAM sizes and the flow traffic demands are generated in [250, 1250] and [100Kbps, 100Mbps] uniformly at random, respectively. The source and the destination of each flow is randomly chosen from the hosts in the network.

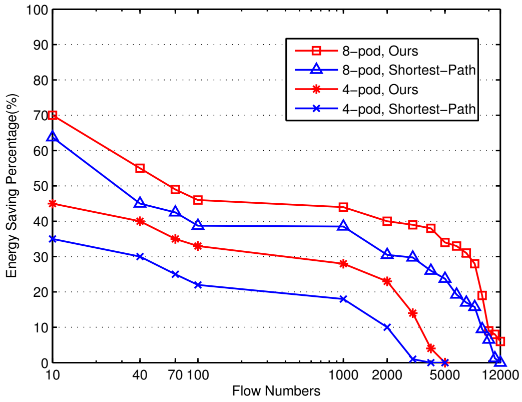


Fig. 2. Energy savings percentage under different flow number

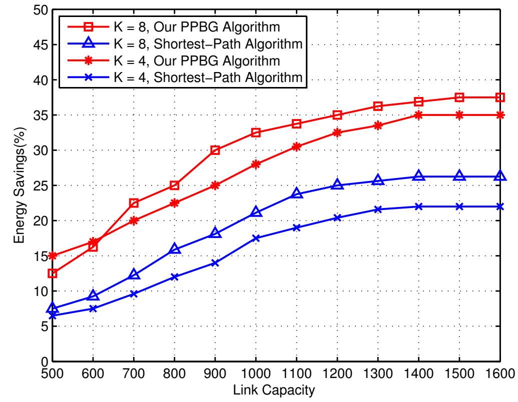


Fig. 3. Energy savings percentage under different link capacities

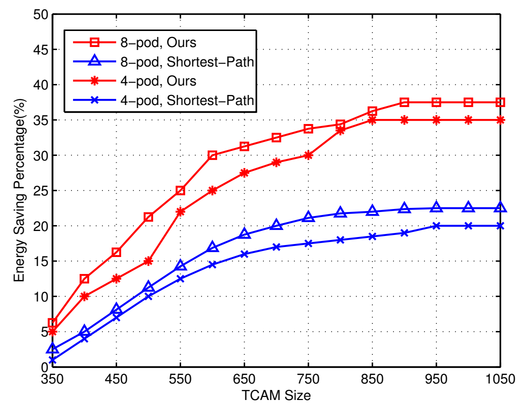


Fig. 4. Energy savings percentage under different TCAM sizes

TABLE I. PERFORMANCE COMPARISON ON 6-POD FAT TREE (54 HOSTS)

Traffic Demand = 1	Gurobi Optimizer		Our Algorithm	
Flow Number	Activated Switch Number	Scheduling Time	Activated Switch Number	Scheduling Time
54	45	23s	45	5ms
540	45	4168s	45	20ms
1080	45	24284s	45	50ms
2700	45	15 hour	45	200ms
5400	45	$>$15 hour	45	1s

TABLE II. PERFORMANCE COMPARISON ON 10-POD FAT TREE (250 HOSTS)

Traffic Demand = 1	Gurobi Optimizer		Our Algorithm	
Flow Number	Activated Switch Number	Scheduling Time	Activated Switch Number	Scheduling Time
250	105	12 hour	105	8s
2500	105	$\gg$24 hour	105	8s
5000	105	$\gg$24 hour	106	9s
12500	105	$\gg$24 hour	108	11s
25000	105	$\gg$24 hour	108	14s

A. On the Scheduling Time

We first investigate the scheduling time of both our proposed algorithm and the optimal solution using commercial Gurobi optimizer [18] on Lenovo ThinkPad T440 equipped with 2.6Ghz Intel Core i7 and 8GB RAM. Two different data center scales, i.e., 6-pod with 54 hosts and 10-pod with 250 hosts, are considered, respectively. The average number of flows emitted from a host is set as 1, 10, 20, 50 and 100 in different simulation instances. The performance evaluation results on both scheduling time and number of activated switches are shown in Tables I and II. We can first notice that the number of activated switches obtained by our algorithm is almost the same as the optimal solution, especially in the small-scale case with 6 pods. The computation time of our algorithm is much lower than the one using Gurobi and the advantage becomes more significant with larger network size or number of flows. For example, it takes several days to get the optimal solution in the case with 10 pods and 12,500 flows while only 11 seconds to obtain our solution, which only requires three more switches than the optimal one.

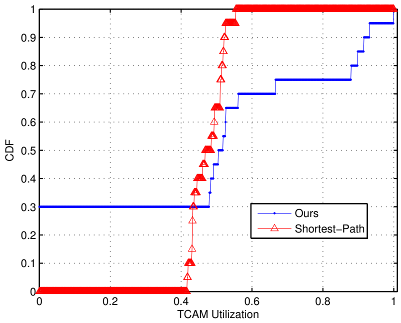
B. On the Energy Saving

Next, we investigate how our algorithm can save the energy consumption by deactivating the switches and routing the flows through the activated switches. To clearly show the advantage of our algorithm, we compare it against shortest-path algorithm which always activates the switches on one shortest-path for each flow. We first vary the number of flows from 10 to 12,000. For each settings, 20 instances are run to obtain the average energy saving percentage in Fig. 2. We first notice that our algorithm shows obvious advantage over shortest-path algorithm. To both algorithms, we further observe that the energy saving percentage declines with the increase of flow number. This is due to the fact that more switches must be activated to accommodate the huge traffic demands incurred by larger number of flows. Specially, when the number of flows exceeds a certain value, e.g., 5000 for 4-pod fat tree, we even cannot obtain any energy saving any more because the network traffic demands already saturate the base topology and all the switches must be activated to ensure the QoS at this point, leaving no optimization space. On the other hand, one may also notice that, under the same flow number setting, higher energy saving percentage can be obtained in a comparatively larger data center scale. This is attributed to that more switches can be deactivated without violating the QoS requirements.

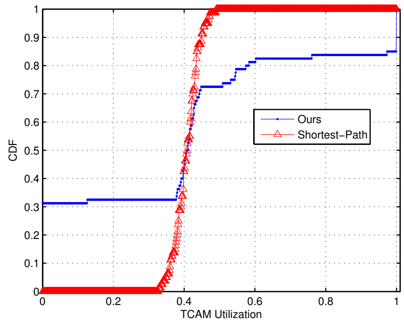
We further study the effect of link capacity and TCAM size on the energy savings percentage on 4-pod and 8-pod fat tree, respectively. We first fix the TCAM size as 750 and vary the link capacity from 500 to 1,600. The evaluation results are shown in Fig. 3. Once again, we observe the advantage on energy saving of our algorithm over shortest-path algorithm. Furthermore, one can obviously notice that higher link capacity indicates higher energy saving percentage. This is because more flow can go through a switch without violating the link capacity constraints and thus less switches need to be activated. In Fig. 4, we fix the link capacity as 1Gbps and vary the TCAM size from 350 to 1050. Similar phenomenon can be discovered as higher TCAM size allows more flows to go through a switch, resulting in less switches to be activated.

C. On the TCAM Utilization

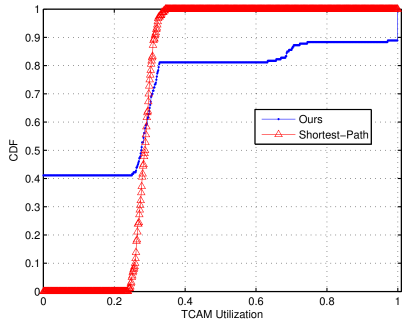
To get an insightful understanding on the reason leading to the energy saving by our algorithm, we further conduct a group of experiments to investigate the TCAM utilization in the two different scheduling strategies. To study the TCAM utilization, we set the link capacity as 1Gbps and TCAM size as 750 to obtain the cumulative distribution function (CDF) of the TCAM utilization. The results on 4-pod, 8-pod and 12-pod fat tree are presented in Fig. 5(a), Fig. 5(b) and Fig. 5(c), respectively. From all three figures, we notice that our algorithm always exhibits a higher utilization ratio than shortest-path algorithm thanks to the design philosophy that



(a) 4-pod Fat Tree



(b) 8-pod Fat Tree



(c) 12-pod Fat Tree

Fig. 5. CDF of TCAM Utilization Raio

the TCAMs on activated switches are always fully explored. Another interesting thing that can be observed from Fig. 5(a) is that around 30% switches with TCAM utilization of 0 are actually deactivated. Furthermore, once a switch is activated, its TCAM utilization is at least 48%. This also explains why we can achieve higher energy efficiency.

VII. Conclusion

In this paper, we investigate how to minimize the energy consumption in SD-DCNs by selectively activating the switches and carefully scheduling the multi-path routing, according to the data center traffic demands. Specially, we take the inherent feature of SDN, i.e., TCAM size limitation, into consideration and formulate the problem into an ILP problem. To address the computation complexity on solving the ILP, we further propose a heuristic minimum switch activation multi-path routing algorithm. Through extensive simulations, the high energy efficiency of our algorithm is proved by the fact that it can find near-optimal solution while requiring substantially less scheduling time.

Acknowledgement

This research was partially supported by Strategic International Collaborative Research Program (SICORP) Japanese (JST) - US (NSF) Joint Research "Big Data and Disaster Research" (BDD), the NSF of China (Grant No. 61272470, 61305087, 61402425, 61502439, 61501412), the Fundamental Research Funds for National University, China University of Geosciences, Wuhan (Grant No. CUG14065, CUGL150829), and the Provincial Natural Science Foundation of Hubei (Grant No.2015CFA065).

References

[1] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No "Power" Struggles: Coordinated Multi-level Power Management for the Data Center," in *Proceedings of the 13th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pp. 48–59, ACM, 2008.

[2] D. Grunwald, C. B. Morrey III, P. Levis, M. Neufeld, and K. I. Farkas, "Policies for dynamic clock scheduling," in *Proceedings of the 4th Conference on Symposium on Operating System Design & Implementation-Volume 4*, pp. 6–6, USENIX Association, 2000.

[3] L. Rao, X. Liu, L. Xie, and W. Liu, "Minimizing Electricity Cost: Optimization of Distributed Internet Data Centers in a Multi-Electricity-Market Environment," in *Proceedings of the 29th International Conference on Computer Communications (INFOCOM)*, pp. 1–9, IEEE, 2010.

[4] B. L. Hong Xu, Chen Feng, "Temperature Aware Workload Management in Geo-distributed Datacenters," in *Proceedings of International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, pp. 33–36, ACM, 2013.

[5] A. Greenberg, J. Hamilton, D. A. 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, 2008.

[6] Y. Shang, D. Li, and M. Xu, "Energy-aware routing in data center network," in *Proceedings of the first ACM SIGCOMM workshop on Green networking*, pp. 1–8, ACM, 2010.

[7] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu, "Energy proportional datacenter networks," in *ACM SIGARCH Computer Architecture News*, no. 3, pp. 338–347, ACM, 2010.

[8] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "Elastictree: Saving energy in data center networks.," in *NSDI*, vol. 10, pp. 249–264, 2010.

[9] X. Wang, Y. Yao, X. Wang, K. Lu, and Q. Cao, "Carpo: Correlation-aware power optimization in data center networks," in *INFOCOM, 2012 Proceedings IEEE*, pp. 1125–1133, IEEE, 2012.

[10] C. E. Hopps, "Analysis of an equal-cost multi-path algorithm," 2000.

[11] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[12] N. Yamada, H. Takeshita, S. Okamoto, and T. Sato, "Using optical-approaches to raise energy efficiency of future central and/or linked distributed data center network services," *International Journal of Networking and Computing*, vol. 4, no. 2, pp. 209–222, 2014.

[13] D. Li, Y. Shang, and C. Chen, "Software defined green data center network with exclusive routing," in *INFOCOM, 2014 Proceedings IEEE*, pp. 1743–1751, IEEE, 2014.

[14] K. Kannan and S. Banerjee, "Compact tcam: Flow entry compaction in tcam for power aware sdn," in *Distributed Computing and Networking*, pp. 439–444, Springer, 2013.

[15] F. Giroire, J. Moulierac, and T. K. Phan, "Optimizing rule placement in software-defined networks for energy-aware routing," 2014.

[16] C. Raack, A. M. Koster, S. Orlowski, and R. Wessäly, "On cut-based inequalities for capacitated network design polyhedra," *Networks*, vol. 57, no. 2, pp. 141–156, 2011.

[17] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, (New York, NY, USA), pp. 267–280, ACM, 2010.

[18] G. Optimization *et al.*, "Gurobi optimizer reference manual," *URL: http://www. gurobi. com*, 2012.