Dynamic Flow Consolidation for Energy Savings in Green DCNs

Chao Zhu*†, Yu Xiao†, Yong Cui*, Zhenjie Yang*, ShiHan Xiao*, Antti Ylä-Jääski†
*Tsinghua University
†Aalto University

***Abstract*—Energy consumption of data center has become an important challenge due to high electric cost and carbon dioxide emissions. Previous work has mainly focused on saving energy cost of servers, though the energy consumption of data center networks (DCNs), consisting of networking equipments like switches, also takes a significant part of the overall energy consumption. In this paper, we propose *ProCons*, an energy saving mechanism that dynamically consolidates traffic flows onto a small set of networking equipments in order to shut down idle ones for energy saving. Different from previous works that assume the traffic demands to be stable, *ProCons* takes into account the variance of traffic demand over time, and predicts future demand based on historical statistics. The traffic flows are then scheduled based on the predicted future demands and the capacity of each link. We evaluate *ProCons* with real life traces collected from data centers using a flow-level simulator. Our experimental results show that using *ProCons*, 40% of energy savings for DCNs can be gained while maintaining the good performance of flow transmission.**

I. Introduction

Energy consumption has become an essential challenge for large-scale data centers. According to the statistics [1], the annual energy consumption of data centers accounted for about 1.3% of all worldwide electricity use in 2010. There are three major energy consumers in data centers, including cooling systems, servers, and networking equipments [2]. A lot of effort has been invested on reducing the energy consumption of servers. However, little effort has been put on the energy cost of the networking equipments that constitute the data center networks (DCNs), though it accounts for approximately 25% of the overall energy cost of the data centers [3]. As the energy efficiency of the cooling systems keeps increasing, the energy consumption of the DCNs is estimated to take up to 50% of the overall cost in near future [4]. Therefore, it is essential and urgent to invest more effort on reducing the energy consumption of DCNs.

Regarding the provisioning of DCNs, it is expected to have enough networking equipments that provide sufficient bandwidths for interconnecting all the servers in the data center [5]. The bandwidth resources are usually planned according to the maximum workload of communications. In practice, as reported in [6], the average link utilization of aggregation layer links is 8% during 95% of the time, while the utilization of core layer links can rarely exceed over 40%. In other words, the traffic in data centers can rarely fully utilize the link capacities. The idle resources waste energy and also cause unnecessary operating cost. To address this issue, traffic consolidation [2] was proposed for reducing the waste by consolidating traffic flows onto a subset of the networking equipments and shutting down or hibernating idle ones.

So far several traffic consolidation strategies [2], [3], [7] have been proposed. Most of them focused on heuristically consolidating the traffic in DCNs with the assumption that the demand of flows is always fixed. However, according to the measurements [8], the traffic in DCNs varies regularly from time to time, which requires the decisions of traffic consolidation to be adjusted frequently on a regular basis with the change in traffic demand [9]. Otherwise, consolidating traffic greedily while ignoring the variance may cause network congestion on certain links while wasting more energy on others.

According to the research [8], flow traffic demands have a correlation with their historical traffic, and the uncertain future flow traffic demand can be better represented using probabilistic characterization. We tracked several flows from the real data center trace that lasts more than two hours provided by [10] and analysed the variance in their traffic sizes. We observed that the flows' traffic sizes maintain within a certain range during a short period of time. Based on this key observation, we propose in this paper a flow consolidation framework *ProbCons* that uses probabilistic variables calculated by historical traffic matrix to predict the future DCNs' flow traffic demands.

We analyse the details of energy consumption of network equipments and formulate the flow assignment problem. Finding the optimal flow assignment for integer flows alone is known to be a NP-complete problem. To address this issue, we propose a heuristic algorithm *PCA* with good computational efficiency to consolidate the appropriate flows onto a subset of network equipments based on the probabilistic prediction of their traffic demands. We simulate our solution with a large scale real Google data center traces provided by [10]. Evaluation results show that ProCons can provide a more precise bandwidth reservation compared with previous works. In addition, it addresses an efficient balance between energy consumption and transmission performance. The main contributions of our work are summarized as follows:

- We figure out the correlation between traffic demands in DCNs and historical traffic sizes by analyzing real life traces collected from data centers.

This work is supported by the NSFC (no.61422206, 61120106008), National 863 project (no.2013AA010401), Academy of Finland (no.278207, 268096)

978-1-4799-7575-4/14/$31.00 ©2015 IEEE

Fig. 1. Variations in iPlayers regional activity levels across the UK, user activity varies from extremely heavy (olive green) to extremely low (pink). [11]

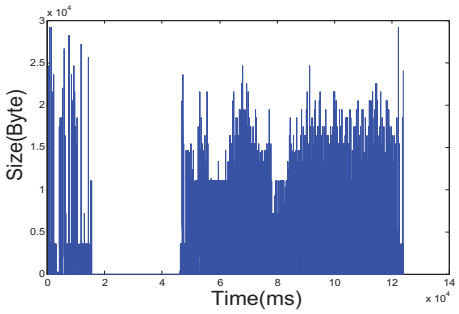


Fig. 2. Real Data Center Flow Variation

- We propose a novel framework, ProCons, which predicts future traffic demand based on historical traffic matrix and schedules traffic consolidation based on a lightweight heuristic algorithm.
- We evaluate our solution with real traces of Google data centers using a flow-level simulator. The results demonstrate significant energy savings without degrading network performance.

The reminder of this paper is organised as follows. In section II, we track the flows from a real data center trace and analyse the characteristic of flows traffic demands. Section III presents the formulation of the flow assignment problem. We propose the design of ProCons in Section IV and evaluate the performance of our work in Section V. Section VI introduces the related works about energy saving in DCNs, before we conclude our work in Section VII.

II. Flow Variation Characteristic

In this section, we analyse the long-standing characteristic of the flows in DCNs and give an analysis of the flows traffic variation during a short period by tracking flows from a real data center trace. We find flows in DCNs may follow certain variation routine both in the long-standing or short-standing period. This key characteristic provide the basis for predicting flows' incoming traffic demand according to their historical record.

A. Flow Variation in Long-standing Period

Taking the data centers supporting the BBC iplayer for example, the traffic demands in different data centers that are located in different regions have various features. As shown in Figure 1 [11], the flows in the data center networks that are located in the area colored with olive green, which means extremely heavy work loads may usually larger traffic demand compared to those flows in the DCNs that are located in the area with lower activity (colored with pink in the picture). And this feature would usually last for a very long time period and would not change easily. Based on this analysis, the controller of networks could aggregate more flows in a greater degree and shut down more equipments for energy saving if the DCNs are located in a lower activity region, while the controller will provision more bandwidth for flows in the DCNs that locate in the region with heavier activity. Of course, the controller of the DCNs can not find out which region the data centers are located in, but the controller can learn from the historical traffic pattern to figure out the order of the magnitude of the flow traffic. The flow usually with heavy demand will have a high probability to achieve a higher load in the future. Thus probabilistic prediction of future traffic demand based on the historical pattern would achieve a precise accuracy rate in a certain degree.

B. Flow Variation in Short-standing Period

Besides the long-standing traffic size characteristic, the traffic demand created by different type of devices also have different variation features in short-standing period. The traffic demands created by the fixed-line devices (e.g., Internet-enabled TV, desktop computers) peaks during evening hours. In contrast, the traffic demand of the cellular network access peaks during commute times. Furthermore, according to the observations [11], the mobile device traffic demand has an even complicated characteristic, it usually peaks during commutes and evening hours. Some flows, such like those created by the cellular network, have an extremely short term variation and it is unpracticable to predict them in an efficient way. But for some other flows as vary during evening hours, such like created by Internet enabled TV, we can predict their incoming demands in a longer term during a certain time period. For instance, the flow traffic of a stream media may require a high demand last for one or two hours when a family watch a movie in the prime time in the evening. Figure 2 shows the variation of a stream media flow that random chose from real DCNs trace last more than two hours. We can observe that there are two main period that the flow maintain a high demand and require almost no traffic demand from 18s to 45s. Suppose we periodically predict flow future demand in every 5s and the time now is 20s. According to the last 5s traffic pattern, we can analyse the traffic flow have fallen to a low traffic demand level from a heavier demand. So there is a high probability that the flow will maintain a low traffic demand in the period of 20s$\sim$25s. Oppositely, we also can predict the flow will

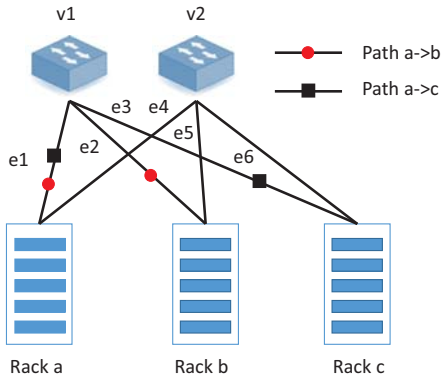


Fig. 3. Simple Topology

have a higher traffic demand during the period of 45s∼50s. According to the observation of the picture, we can find out it is reasonable for the controller to predict the incoming flows traffic demands based on their historical pattern in a short-term period.

III. Problem Formulation

In this section, we give the mathematic formulation of flow consolidation problem.

There are two main parts that constitute the power consumption of DCNs, switches and ports. Because each switch port directly connects to a link in DCNs, for convenience, we treat the power cost by links as that cost by switch ports. We use $P_S(u)$ and $P_L(e)$ denote the power consumption of an idle switch u and the power consumption of a link e, respectively. Suppose the number of powered on switches is N_S and that of powered on links is N_E. Our object is minimizing the total network energy consumption $P_N = P_S(u) \times N_S + P_L(e) \times N_E$.

Suppose there are k flows need to be transmitted in period T, we denote a flow set $F = \{F_1, F_2, ..., F_k\}$, defined by $F_i = (a_i, b_i, d_i)$, where a_i denotes the flow's source rack, b_i denotes the flow's destination rack. d_i denotes the flow traffic demand, respectively. Our formulation uses the notations listed in the TABLE I.

TABLE I. Notations and Definitions

Notations	Definitions
V	the switch set $\{V_1, V_2, ..., V_m\}$
E	the link set $\{E_1, E_2, ..., E_n\}$
E_u	the link set that connected to switch u
Y_u	a binary variable denotes whether switch u is powered on
Y_e	a binary variable denotes whether link e is powered on
$P_S(u)$	the power consumption of switch u, S denote switch
$P_L(e)$	the power consumption of link e, L denote link
P_e	the path set that across link e
$R(p)$	a binary variable denotes whether path p load any traffic
$d(a,b)$	the demand of traffic from rack a to rack b
$P(a,b)$	all available paths from rack a to rack b
$x(p)$	the traffic size upon path p
$c(e)$	the link capacity of link e

We denote the topology of a data center network as a graph $G(V, E)$. Suppose there are m switches and n links in the network, $V = \{V_1, V_2, ..., V_m\}$ denotes all the switches and $E = \{E_1, E_2, ..., E_n\}$ denotes all the links that connect to the switches, respectively. $d_{a,b}$ denotes the traffic demand from rack a to rack b.

In order to make it easier to understand, we give a simple network topology as shown in Figure 3. There are two switches ($v1$,$v2$) and six links ($e1, e2, e3, e4, e5, e6$) in the network. Three racks ($Rack\ a, Rack\ b, Rack\ c$) are interconnected by these equipments. There are two available paths for any two of the racks in the network. Let P_e denote the set of paths from rack to rack across link e. Take the network in Figure 3 for example, there are two paths across the link $e1$ (path a→b and path a→c). Suppose there are k hops in path p, which means path p contains k links. Let $p(e1, e2, ..., ek)$ denote the path, thus the path set across $e1$ in the Figure 3 is $\{p(e1, e2), p(e1, e6)\}$.

We can get the traffic size upon the link e is the sum of the traffic size of all the paths in set P_e. Let x(p) denote traffic size upon the path p and $c(e)$ denote the link e's capacity. We guarantee the traffic upon a link can not exceed the link's capacity and we can get the capacity constrains:

$$\sum_{p \in P_e} x(p) \leq c(e) \tag{1}$$

In our work, we consolidate flows onto a subset of network devices and power off the unused ones. We use a binary variable Y_u to denote whether the switch u is powered on and a binary variable Y_e to denote whether the link e is powered on, respectively. So the number of powered on switches $N_S = \sum_{u \in V} Y_u$ and the number of powered on links is $N_E = \sum_{e \in E} Y_e$. So we can get the total power consumption in the data center network as $P_N = \sum_{e \in E} Y_e \times P_L(e) + \sum_{u \in V} Y_u \times P_S(u)$.

For energy saving in DCNs, we deactivated the links without carrying any flow and flows are restricted to only those links that are powered on. In our formulation, we let a binary variable $R(p)$ denotes whether there exist flow across path p. We can figure out whether a link e has loaded any flow by summing up the $R(p)$ in set of P_e (path set that across link e). If all the $R(p)$ in set P_e is 0, means that all the paths across link e do not carry any flow and there will not exist any flow that passes through link e. If the link e does not carry any flow, the link e will be powered off. Besides, flows are restricted to the powered on links. If a link e is powered off, $Y_e = 0$, any path across link e can not carry any traffic, $R(p) = 0$. Thus, we can get the constrains:

$$\forall e \in E, \forall p \in P_e,\ Y_e \leq \sum_{p \in P_e} R(p),\ R(p) \leq Y_e \tag{2}$$

If all the links connected to switch u are powered off, it is useless to power on the switch u because there is no flow passes through this switch. On the other hand, if switch u is powered off, all the links connected to switch u also should

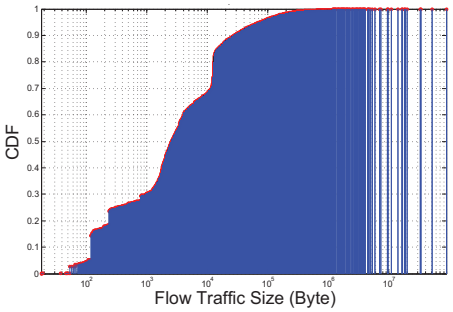


Fig. 4. CDF of Flow Traffic Size

be powered off. Thus, the constrain is:

$$\forall u \in V,\ \forall e \in E_u, Y_e \leq Y_u,\ Y_u \leq \sum_{e \in E_u} Y_e \tag{3}$$

As mentioned before, the objective of our system is to minimize the sum of device power cost for the entire network during each scheduling period T. Flows in DCNs have several routing options because there may exist more than one path that connect two racks. However, due to TCP packet reordering effects, we prevent flow splitting. Therefore, we have the following optimization problem with constraints that prevent flows from getting split.

$Minimize$:

$$\sum_{e \in E} Y_e \times P_L(e) + \sum_{u \in V} Y_u \times P_S(u)$$

subject to

$$R(p) \in \{0,1\},\ \sum_{p \in P_{a,b}} R(p) = 1 \tag{4}$$

$$\forall p \in P_{a,b},\ \forall i,\ x(p) = d_{a,b} \times R(p) \tag{5}$$

$$\text{Constraints}\ \ (1),(2),(3) \tag{6}$$

$P_{a,b}$ denotes the set of paths connecting rack a and rack b. For the constrains (4), the flow should choose one and only one path from the set of paths connecting rack a and rack b for preventing being split. For the constrains (5), the traffic size $x(p)$ between rack a and rack b upon the path p is equal to either its full demand or zero.

To solve the above optimization problem mathematically, we desire the optimal solution that satisfies the above constrains. However, finding the optimal flow assignment for integer flows alone is known to be a NP-complete problem. We use a linear programming tool to determine the consolidation to get a near-optimal solution in the first step. But this algorithm include an integer step that has been proved to be exponential [12]. To reduce the computation complexity, we will present our heuristic consolidation algorithm in the latter sections.

IV. Design of ProCons Framework

In this section, we first give an analysis of two types of flows (elephant flows and ant flows) in the DCNs. Then, propose the principles of the design of the ProCons framework and introduce the method to predict the incoming flows traffic demands. Finally, we design a light weight heuristic algorithm *PCA* to accomplish the traffic consolidation.

A. Elephant Flows and Ant Flows

The flows in DCNs mainly are small-data flows (ant flows) that randomly burst and short-lived. These flows mainly occupy 80% amount of traffic flows but may carry less than half traffic bytes [13] [14] [15]. Besides the ant flows, large-data flows (elephant flows) only is a small set of traffic flows in DCNs and usually are long-lived and consume lots of bandwidth. The number of the elephant flows is small but can have a big and long-lived affection on the performance of the network. For some applications, the completion time usually is decided by the time when last request flow accomplishes. Thus, the long-lived elephant flows can cause great impact on the total working time of network equipments.

The Figure 4 shows the flow data size distribution in the real data center network provided by [10]. We analyse the total flows trace in the data center network within one hour and give the CDF of the flow data size. As shown in the figure, 85% of flows's data size is under 50KB and these ant flows usually can be accomplished in a short time.

According to the analysis, the ProCons will distinguish the incoming flows by their size and only schedule the elephant flows (the size over 50KB). There are three reasons as follows: First, ant flows usually have a higher delay sensibility and it will cause a certain delay for them if they are scheduled by the consolidation scheme. Second, because of their short living period, the ant flows have less effect on the working time of the network equipments. Third, it is unpracticable to predict the incoming traffic demands of ant flows because they are randomly burst. Thus, The ant flows will use their default routing such as ECMP and will be scheduled with no delay. The ECMP [16] strategy schedule the flows by the following way: for each flow, first compute all the shortest paths for it and then randomly assign one path for its routing. The flows can quickly find a path using the ECMP strategy and take scarce latency cost in routing. The method that only schedule the elephant flows can greatly reduce the computing complexity of the ProCons scheduler and provide a guaranteed service quality for the ant flows.

B. ProCons Framework and Prediction Method

With the appearance of the Software Defined Network (SDN) technology, operators of the DCNs can originally design the protocols and employ them to the network easily. ProCons applies the SDN technology to consolidate the flows in DCNs and uses a central controller to monitor the whole network. The central controller connects all the servers and switches and responses to route the path to each incoming flow. ProCons takes three steps to optimize the power consumption in DCNs, Flow Filtration, Demand Prediction and Flow Consolidation.

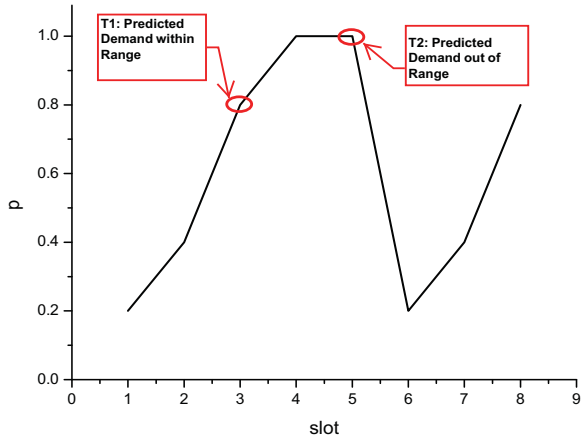


Fig. 5. The Variation of The Probability Metric p

In the Flow filtration step, the central controller of the network divide the incoming flows into two categories, the elephant flows and ant flows, according to their traffic size in last transmission period. The flow whose traffic size is over 50KB traffic demand will be grouped as elephant flow and placed in a queue that will be scheduled by the central controller. The flows with smaller traffic size that under 50KB will be routed by the default ECMP method immediately.

In the second step, ProCons will probabilistic predict the incoming traffic demand according to the historical traffic metric. In this step, the prediction mainly depends a critical mutative metric p. Let D_p denote the flow previous slot traffic demand of the flow and D'_p denote the previous slot traffic demand of the D_p. We assume the incoming flow traffic demand $D = D_p * (1-p) + D'_p * p$. Inspired by the congestion control method of TCP, we define the variation function of the metric p as flows:

$$p = \begin{cases} 1, & D_p \leq 2D'_p \cap D_p \geq D'_p/2 \cap p' > 0.5 \\ p' * 2, & D_p \leq 2D'_p \cap D_p \geq D'_p/2 \cap p' \leq 0.5 \\ p_0, & D_p > 2D'_p \cup D_p < D'_p/2 \end{cases} \quad (7)$$

p' is the probabilistic metric in the previous slot. We set p_0 as the initiate value of p. Figure 5 shows the function line of the variation of the probabilistic metric p. In the slot T1, the previous predicted traffic demand D_p is in the range of $(1/2D'_p, 2D'_p)$, it indicates that the demand we have predicted in the previous slot is appropriate and there is a tendency that the incoming flow traffic size may be as similar as the traffic in previous slot. Thus, the previous slot traffic size will have a greater influence on the incoming flow traffic and we modulate the probability $p = p' * 2$. In the slot T2, the previous predicted traffic demand D_p is out of the range of $(1/2D'_p, 2D'_p)$. It indicates that the flow traffic has a intense variation. Therefore, the influence of the previous traffic demand will be reduced and p just fall down to its initiate value p_0.

C. Working example

In order to get clear idea of how ProCons works, here we give an example of the traffic consolidation principle in ProCons as follows. Figure 6 shows a partial Fattree [17] network topology which contains 15 switches and 12 servers. In this topology, there are four servers have communication missions and marked with "A", "B", "C", "D", respectively. Suppose the capacity of the each link in this network is 1 and three flows need to be transmitted in the network, flow "A→C" with predicted 0.8 demand, flow "A→D" with 0.1 predicted demand and flow "B→D" with 0.6 predicted demand. There exist three available pathes between any two inter-pod servers. Assume the mentioned three flows arrive at the same time and the default routing of each flow as shown in Figure 6 (a). We dye the switches that need to be used with blue color and the switches that are not occupied by the flows with yellow color. In the figure, the total number of the working switches is 12 and the amount of the links carrying load is 16. Therefore, we can get the default routing method will cost $P = P_S(u) \times 12 + P_L(e) \times 16$ power in the transmission period, $P_S(u)$ and $P_L(e)$ denote the power consumption of an idle switch u and the power consumption of a link e as mentioned before.

Unlike the default method, the ProCons will consolidate the flow traffic in a more energy-efficiency way according to the predicted demand. Due to the capacity limitation, congestion will be caused when the aggregated flow traffic size over the capacity of the link. So the ProCons will consolidate the flows with appropriate traffic sizes onto one link. As shown in the Figure 6 (b), the controller of the Procons will schedule the three flows orderly. Because there is no traffic in the network at the beginning, the controller will randomly choose a path for flow A→C. For the next flow A→D, the controller will be prone to choose a path with links carrying traffic load and guarantees the aggregated traffic size under the capacity of each link on the path. Due to the sum of the flow size of flow A→C and A→D is 0.8+0.1=0.9<1, the controller of the ProCons will consolidate these two flows onto the same core switch and these flows will share serval links on their path. Due to the capacity limitation, the controller will not choose the core switch that occupied by the flow A→C for flow B→D. Thus, the controller need to open a new core switch and place the flow onto the path as marked by the blue triangle. The controller will shut down the switches that not used (as marked with gray color) after assignment. We can account the total energy consumption under the ProCons strategy is $P = P_S(u) \times 9 + P_L(e) \times 13$ and energy consumption can be greatly reduced under the ProCons strategy compared to the default method.

D. Probabilistic Consolidation Algorithm

We have proposed the ProCons framework and the method to probabilistically predict the incoming flow traffic demand in last section. We now present a light weight heuristic algorithm **PCA**(**P**robabilistic **C**onsolidation **A**lgorithm) that performs an energy-aware consolidation according to the predicted traffic with low computational overhead. The principle of PCA is greedily assigning as many traffic flows as possible to a single

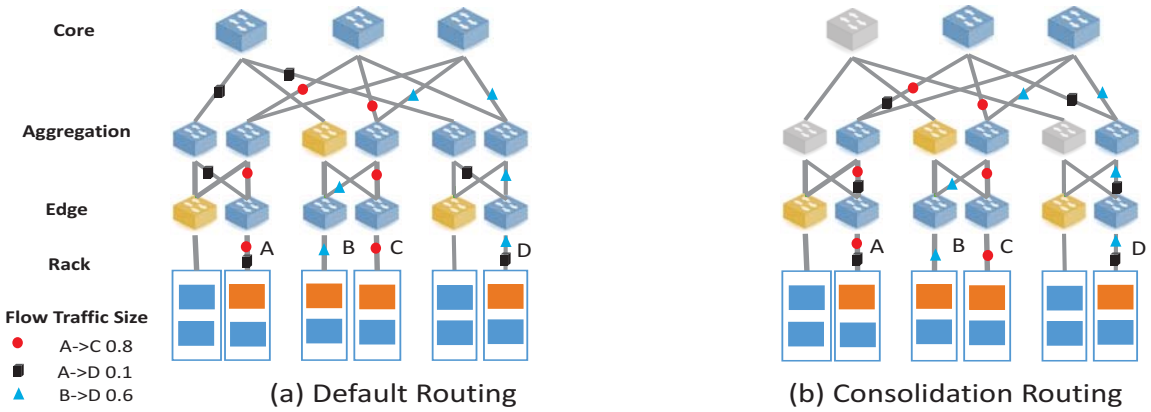
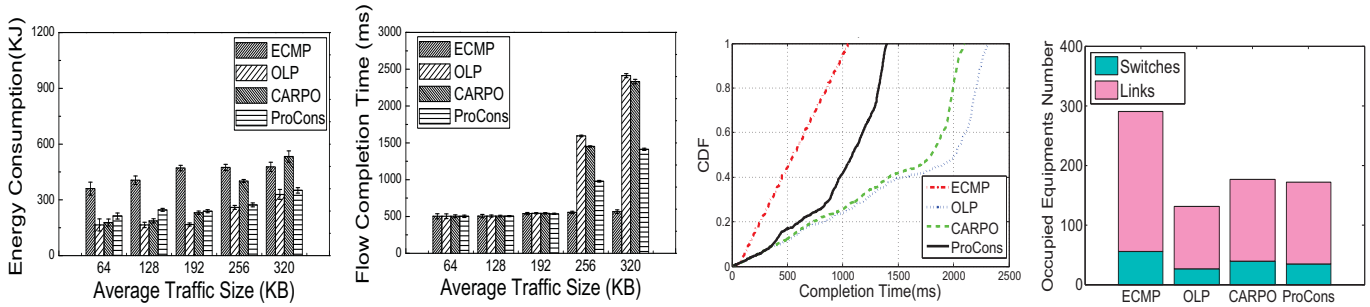
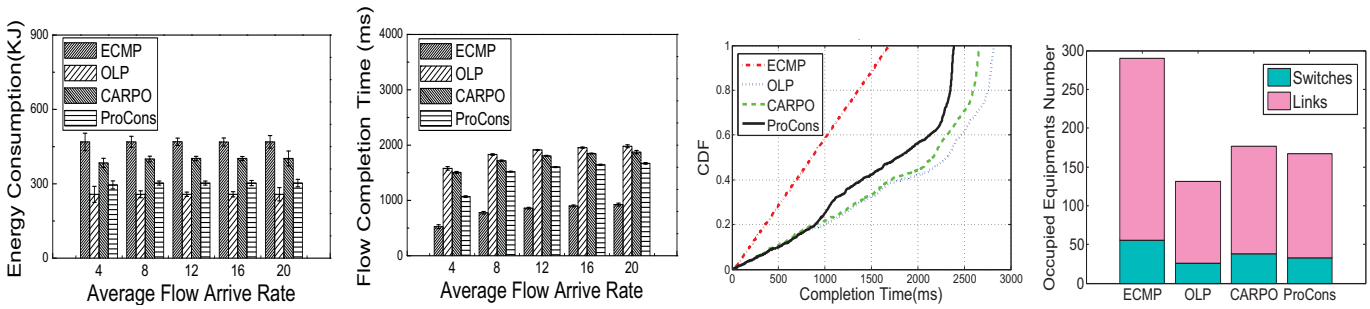


Fig. 6. Energy Saving Routing by Consolidating Flows



(a) Energy Consumption (b) Flow Completion Time (c) CDF of Flow Completion Time (d) Occupied Equipments Number

Fig. 7. Performance under Different Traffic Sizes



(a) Energy Consumption (b) Flow Completion Time (c) CDF of Flow Completion Time (d) Occupied Equipments Number

Fig. 8. Performance under Different Flow Arrival Rates

path with capacity limitation. Algorithm 1 presents the pseudo code of PCA. The input of our algorithm include the flow list F, the total link capacity c in period T and the path set PL for each available path. Note that all the flow should be inseparable and each flow can take one and only one path. The order of the path in the PL is from left to right based on the network topology. In Algorithm 1, Line 1 gets the predicted traffic demand of each flow according to the prediction method mentioned in Section IV-B. Lines 2-12 response to assign all the flows to their path. Line 5 finds out whether the total traffic size after adding flow f_i to path j is over the capacity of any link across that path. If there is no capacity violation, Line 6 will choose path j for f_i and the Line 7 will remove the f_i from the flow list F. Finally the remain capacity of each link across the path j will minus the demand of the flow f_j as shown in Line 8 and return the route path PF for each flow in the end. The complexity of PCA depends on the number of flows and switches. Suppose the number of flows is M and the number of switches is N, the worst case of the complexity of the algorithm is $O(MN^2)$.

V. Evaluation

In this section, we compare the performance of ProCons with other traffic consolidation strategies. First we present the

Algorithm 1 PCA: **P**robabilistic **C**onsolidation **A**lgorithm

Input:
Flow set $F = \{f_i\}$ with N flows, each flow with traffic demand d; Link list $L = \{l_i\}$, each link with capacity c in period T; Path set PL with all available path.

Output: The final route path PF of each flow

$\forall f_i \in F, d[i] = ProPredict(F)$
for $F \neq NULL$ **do**
  for $j = 1$ to m **do**
    for $\forall f_i$ can pass through path j **do**
      if $c[k] - d[i] \geq 0, \forall l[k] \in PL[j]$ **then**
        $PF[j] = PF[j] \cup \{f_i\}$
        $F = F - \{f_i\}$
        $c[k] = UPDATE(PATH[j], d[i])$
      end if
    end for
  end for
end for
return PF

setup and methodologies of our evaluations and then give the analysis of the simulation results.

A. Evaluation Setup and Methodologies

In this subsection, we first introduce the setup of our evaluation. According the measurement from [18], we set the power consumption of a switch is 151W and the power consumption of a port is 11W. Our simulations are performed by a flow-level simulator implemented with C++ and use the setting of TCP as that utilized in [19]. Our simulator apply a three layer Fattree [17] with eight pods as the basic network topology.

For the comparative analysis, we simulate the performance of our scheme ProCons and three other scheduling algorithms (*CARPO, OLP, ECMP*) proposed by other researches [7] [2] [16].

- CARPO is a flow consolidation scheme that consolidates flows according to the correlations among them. They observed flows traffic have different variance and use this characteristic to consolidate the flows that do not peak at exactly the same time.
- OLP uses a greedy scheme that just place the flows to the leftmost path with sufficient capacity without any traffic demand consideration. Paths are chosen in a deterministic left-to-right order.
- ECMP is the default random routing strategy applied in DCNs.

As we analyse that the elephant flows can take more than half of the total traffic in the data center, we filter the large flows whose traffic sizes are larger than 50KB and take these flows as the input of our simulator. We present the default values of parameters have been involved in the simulation in Table II. During the simulation, when one factor is changed, other factors are set to the default values.

TABLE II. Evaluation Parameters Setup

	Default	Range
Number of large flows per second	12	$4 \sim 20$
Average traffic size of a large flow (KB)	192	$64 \sim 320$

B. Simulation Results Analysis

1) Performance of different traffic size: In this subsection, we evaluate the both energy and the network performance of each flow scheduling scheme under different traffic sizes. In Figure 7, we apply four indexes to evaluate the performance of the flow scheduling schemes. The Figure 7 (a) shows the energy consumption of the whole network under different schemes. We can see the ECMP generally cost most energy because it schedules the flows without any energy-aware consideration. The OLP consumes minimal energy among these schemes because it just arbitrarily consolidates the flows onto their leftmost path. However, this method may cause serious congestion and we can see the flows scheduled by OLP have a high transmission latency in the Figure 7 (b).

As shown in the Figure 7 (a) and Figure 7 (b), compared to ECMP and OLP, the CARPO can achieve a better balance between the network performance and energy consumption. However, the performance can not be improved in a great degree because consolidating flows according to their correlation is a coarse-grained consolidation. Flows do not peak at the same time that may have similar big size or small size and it is inappropriate to consolidate these flows to a single link. Unlike the CAPRO, the ProCons predicts the incoming flow traffic based on the historical traffic matric and can consolidate the flows with a more fine-grained way. As we can see in the Figure 7 (a) and Figure 7 (b), the ProCons can achieve a high energy efficiency flow scheduling with a low latency.

The flow details of the case with average traffic size 192KB are presented in Figure 7 (c) and Figure 7 (d). Figure 7 (c) shows the CDF of the flow completion time distribution in details. As we can see, most of flows scheduled by ProCons are accomplished under 1500ms and the it will cost above 2000ms for the OLP and CARPO scheduling. The ECMP have the lowest latency because of the large amount of equipments occupation. In Figure 7 (d), we treat a equipment with traffic load as an active one during one slot and calculate the total number of occupied equipments in the transmission period T. We can see equipments taken by ProCons nearly 40% less than the ECMP.

2) Performance of different flow arrival rates: We vary the network loads in Figure 8 by scaling the average flow arrival rate. In Figure 8 (a), we can see the energy consumption do not have a significant change for ECMP but have a little increase for three other algorithms. It is because ECMP takes more equipments and is more robust to the heavy traffic. In Figure 8 (b), when the flow arrival increases, flow completion time increases for all the algorithms. When the arrival rate becomes larger in range of (4∼12), the completion time increases quickly for ProCons. However, in the range of (12∼20), the flow completion time of ProCons remains at similar range.

The increasing transmission latency of ProCons is mainly due to there are more flows to schedule in one time and the occurrence of congestion. With the flow arrival rate growing to a certain extend, the impact of the congestion becomes stable and the flow completion time will have a certain up boundary for each algorithm. As we can see in Figure 8 (b), the performances of ProCons under different flow arrival rate are better than the CARPO and OLP.

More specifically, the Figure 8 (c) and Figure 8 (d) shows the flow details of case with flow arrival rate is 20. We can see all the flow completion time become larger compared to the Figure 7 (c) because of the congestion caused by increasing incoming flows. However, with the appropriate traffic consolidation, completion time of ProCons is still lower than the CARPO and OLP. In Figure 8 (d), we can the ProCons still occupies the minimal number of equipments compared to CARPO and ECMP. Unlike OLP, ProCons would not cause serious network congestion and can address an efficient balance between energy saving and network performance even under heavy traffic load.

VI. Related Work

We introduce some related work on traffic engineering in DCNs as well as energy-aware data center networking in this section.

There are mainly two categories of energy-saving technologies in DCNs. First is designing new topologies to use fewer network equipments and guarantee the similar network performance. Flatted butterfly [4] and Pcube [20] are classic energy-aware topologies designed for data center networks. [21] [22] proposed a novel architecture design of data center networks by deploying wireless card on the top of the racks. Second kind of technologies is optimization routing methods for scheduling flows in DCNs. Nedevschi et al. first proposed the the method that energy consumption of network equipments could be improved by applying the sleep-on-idle and rate-adaptation technique by [23]. The most representative work in this type is ElasticTree [2], which studied the method to choose a appropriate subset of links to satisfy traffic loads in DCNs. Similar to the ElasticTree, a flow consolidation method according to their variation correlation named CARPO had been proposed by [7]. [24] proposed a dynamic workload management in distributed data centers. Similar to the our methodology, [25] had done a great work on bandwidth allocation based on the variability of DCNs' traffic. Their algorithm had an efficient performance in guaranteeing the bandwidth for VMs as well as provided fast convergence to efficiency and fairness, and smooth response to bursty traffic. The difference between their work and ours is that their work focused on providing an efficient bandwidth allocation, while our work tries to reduce the energy consumption by traffic consolidation. To the best of our knowledge, this paper is the first one to design a fine-grain prediction method of flow traffic demand according to their historical traffic and dynamically consolidates the flows while guranteeing good quality of network performance.

VII. Conclusion

In this paper, we propose an online flow consolidation framework for the network-wide energy saving in DCNs. Different from existing work, our framework takes into account the variance in traffic demand. In details, we observed a light correlation between future traffic demand and the historical traffic size from the analysis of real traffic traces. Based on this feature, we develop a heuristic prediction algorithm that predicts future traffic demand based on historical traffic matrix, and then consolidate flows based on the predicted traffic demand and the capacities of links. We evaluated our solution with real life traffic traces [10] by using a flow-level simulator. The results have demonstrated the effectiveness of our proposed scheme in achieving energy savings and improving data transmission performance.

Reference

[1] "Growth in data center electricity use 2005 to 2010," in *Analytics Press*, 2011.

[2] B. Heller *et al.*, "Elastictree: Saving energy in data center networks," in *NSDI*, 2010.

[3] K. Zheng *et al.*, "Joint power optimization of data center network and servers with correlation analysis," in *IEEE INFOCOM*, 2014.

[4] D. Abts *et al.*, "Energy proportional datacenter networks," in *ISCA*, 2010.

[5] P. Padala *et al.*, "Adaptive control of virtualized resoures in utility computing environments," in *EuroSys*, 2007.

[6] T. Benson *et al.*, "Understanding data center traffic characteristics," in *SIGCOMM*, 2009.

[7] X. Wang *et al.*, "Carpo: Correlation-aware power optimization in data center networks," in *IEEE INFOCOM*, 2012.

[8] M. Wang *et al.*, "Consolidating virtual machines with dynamic bandwidth demand in data centers," in *IEEE INFOCOM*, 2011.

[9] A. Verma *et al.*, "Server workload analysis for power minimization using consolidation," in *USENIX*, 2009.

[10] T. Benson *et al.*, "Network traffic characteristics of data centers in the wild," in *IMC*, 2010.

[11] Karamshuk *et al.*, "On factors affecting the usage and adoption of a nation-wide tv streaming service," in *IEEE INFOCOM*, 2015.

[12] Y. Berstein *et al.*, "The graver complexity of integer programming," in *Annals of Combinatorics*, 2009.

[13] S. Kandula *et al.*, "The nature of data center traffic: measurements and analysis," in *IMC*, 2009.

[14] M. Alizadeh *et al.*, "Data center tcp (dctcp)," in *SIGCOMM*, 2010.

[15] D. Halperin *et al.*, "Augmenting data center networks with multi-gigabit wireless links," in *SIGCOMM*, 2011.

[16] C. E. Hopps *et al.*, "Analysis of an equal-cost multi-path algorithm," in *IETF*, 2000.

[17] M. Al-Fares *et al.*, "A scalable, commodity data center network architecture," in *SIGCOMM*, 2008.

[18] P. Mahadevan *et al.*, "A power benchmarking framework for network devices," in *IFIP Networking*, 2009.

[19] M. Al-Fares *et al.*, "Hedera: Dynamic flow scheduling for data center networks," in *NSDI*, 2010.

[20] L. Huang *et al.*, "Pcube: Improving power efficiency in data center networks," in *IEEE CLOUD*, 2011.

[21] Y. Cui *et al.*, "Wireless data center networking," *Wireless Communications*, vol. 18, pp. 46–53, 2011.

[22] C. Yong *et al.*, "Data centers as software defined networks: Traffic redundancy elimination with wireless cards at routers," *Selected Areas in Communications*, vol. 31, pp. 2658–2672, 2013.

[23] S. Nedevschi *et al.*, "Reducing network energy consumption via sleeping and rate-adaptation," in *NSDI*, 2008.

[24] Z. Guo *et al.*, "Jet: Electricity cost-aware dynamic workload management in geographically distributed datacenters," *Computer Communications*, vol. 50, p. 162C174, 2014.

[25] J. Guo *et al.*, "On efficient bandwidth allocation for traffic variability in datacenters," in *IEEE INFOCOM*, 2014.