An Approach for Energy Efficient Deadline-Constrained Flow Scheduling and Routing

Keke Fan, Ying Wang, Junhua Ba, Wenjing Li, Qi Li
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China,
Email: fkk_bupt@163.com, {wangy, bug, wjli}@bupt.edu.cn

***Abstract*—With the development and extensive use of the network, its huge energy consumption has attracted more and more attentions. We mainly focus on the network-level energy-saving through the means of flow scheduling and routing, where flow means a certain amount of data that has to be transmitted from a specified source to a specified destination with hard deadline-constraint. We discuss the deadline-constrained flow scheduling and routing problem. Then we propose our EEDFSR method to reduce the active power consumption based on the SDN architecture. The proposed method can be used in both DCN and general topologies. The simulation result has shown that our EEDFSR has ascendancies in reducing the active power consumption of networks which have adequate connections.**

***Index Terms*—SDN, scheduling and routing, energy efficient, deadline constrained.**

I. INTRODUCTION

With the prosperous development of the Internet and the rapid development of network users and various services, the Internet plays a more and more important role in people's lives, and the ensuing is the huge energy consumption problem of the network. Research shows that power consumption of the Internet accounts for 10% of total global electricity consumption [1], of which only the Internet data centers will consume 8% of total global electricity consumption by 2020 [2]. Internet energy consumption is mainly concentrated in network infrastructure, servers and cooling equipment. In the case of data centers, network infrastructure energy consumption accounts for 20% of the energy consumption of the entire data center [3]. With the deployment of energy-efficient servers and the continuous improvement of cooling technologies, the proportion of network infrastructure energy consumption to total Internet energy consumption will continue to increase. However, networks are typically designed for peak traffic, and redundant configurations are used to increase the robustness and reliability of the network, which is contrary to energy savings. Therefore, more and more researchers have paid much attention to this issue and made a lot of efforts.

At present, research results of network energy-saving technologies can be mainly divided into two categories: equipment-level energy-saving technology and network-level energy-saving technology. The former mainly studies the energy consumption factors of network hardware devices from the perspective of hardware design and implementation, and studies how to reduce the energy consumption of a single network device. According to the implementation and complexity, it can be divided into device sleep technology [4-6], dynamic adaptation technology [7-8] and new hardware design technology [9]. The latter mainly studies the routing and scheduling of data flows in the entire network from the perspective of network planning and protocols, in order to study how to effectively reduce the energy consumption of the entire network. And it can be divided into energy-saving topologies, energy-efficient routing, and energy-efficient flow scheduling according to different implementations and granularities. The network-level energy-saving technology adjusts the overall power of the network based on the equipment-level energy-saving technology, thereby realizing the overall power and service load adaptation of the network. In short, device-level energy-saving technology is the foundation of network-level energy-saving technology, and network-level energy-saving technology ultimately achieves energy-saving effects through the operation of network equipment. This paper focuses on network-level energy-saving technologies.

A lot of efforts have been paid in network-level energy saving and they can be divided into three aspects: firstly, topology. The topology can be modified according to the graph features of the topology, and the energy consumption can be reduced by the dormant node and the subset of links. Secondly, network routing. The choice of data center network routing protocols affects network energy consumption and network performance. The trade-off relationship between them should be analysed, and how to design effective energy-efficient routing protocols under network performance constraints is studied. Thirdly, network flow scheduling. This means to study how to combine the scheduling strategy of network flow with energy-efficient routing strategy to achieve more flexible and effective energy consumption in data center network.

In this paper, we mainly focus on the energy-efficient flow scheduling and routing with hard deadlines and propose a solution for this problem. The main contribution of our paper are as follows:

- Our work is based on a power-saving topology where the topology is considered to have been tuned up and no longer considers topology changes. Then, a comprehen-

978-3-903176-15-7 © 2019 IFIP

sive resource allocation method for routing and scheduling is proposed for common flows. Our EEDFSR method combines off-line routing calculation, initial allocation, route adjustment, speed scaling.
- EEDFSR can not only be used in a Fat-tree topology, but also a general network (with enough connection). The result shows that our EEDFSR performs well in saving active energy consumption.

The rest of this paper is as follows. Section II discusses the related work done by seniors. Section III describes the power model for topology links and the flow scheduling and routing problem. Section IV introduces the solution and the simulation results are presented in Section V. At the end of this paper, we conclude our work in Section VI.

II. RELATED WORK

In this section, some related works are summarized. Before that, we need to introduce the characteristics of the SDN network, which is the basis of many methods. In an SDN network, the data plane is separated from the control plane, where the control plane is centralized and complex network functions are centralized on a centralized controller. The controller has a macro view of the entire SDN network and provides a programmable interface to control the entire network, so the controller has global knowledge of network status information. The controller calculates the rules required for forwarding traffic for the switching device, and sends it to each switch in the network through the control path, and the switch forwards the traffic only according to the rules. In addition, the controller can directly manage network tasks and perform device configuration without the need to add additional software or hardware to the switching components. This feature can be used to implement a rate configuration method to determine the link that needs to be rate configured in a coordinated and centralized manner.

A. Flow Routing

From the perspective of network flow, energy-efficient routing technology adjusts the spatial distribution of network traffic load by scheduling the transmission path of network flows, and then implements energy-saving technologies by using device-level energy-saving technologies.

Elastic-Tree is a Fat-Tree based energy-saving strategy proposed by B Heller et al. [10]. The basic idea is to monitor the traffic in the data center network in real time. Under the constraints of network performance and fault tolerance, some algorithms are used to select the subnet with the least energy consumption to transmit the load traffic. While Energy-aware Routing for General Topologies (ERGT) is more versatile than the Elastic-Tree solution and can be applied to typical topologies [11-12]. The traffic size of each node is calculated, and the node will be deleted in the order of increasing value until the total network throughput rate drops to the predefined performance threshold. Similarly, [M] proposes a heuristic greedy algorithm that closes unnecessary network nodes and links, thereby approximating the energy-optimized network topology.

Some research aims at improving the link utilization to reduce energy consumption. An idea of fairly sharing routing approach is proposed, in which the flows can fairly share the bandwidth of the bottleneck links [10] [13]. Therefore, the bottleneck links can be fully used and the utilization of these links can be improved. But this approach may result in low utilization of non-bottleneck links, wasting energy. To realize higher utilization of all links, a two-phase energy-efficient flow routing algorithm is given by Zhang et al. to find the path of the flow one by one [14].

Differently, a new solution has been discussed, in which flows always exclusively use the bandwidth of links in its routing path [15]. And this solution has been proved to be more efficient than regular fair-sharing routing approach owing to higher link utilization in high-radix data center networks. However, this solution is not suitable for flows which may often compete with others for bandwidth to finish flow transmission within deadline.

B. Flow Scheduling

The energy-efficient flow scheduling technology is further based on the energy-saving routing technology. The transmission path and transmission time of each flow are scheduled to adjust the distribution of network traffic load in time and space, and then the device-level energy-saving technology is used to realize network energy conservation.

A typical scheduling problem is to transmit a set of flows within their deadlines, which is called DCFS [16]. The idea of this algorithm is based on greedy strategy. The interval with the highest current energy density is continuously greedily selected, and then the flow is scheduled by the Earliest Deadline First (EDF) strategy in this interval. Obviously this method only suitable for static traffic scheduling, all flows should be clearly and unchanged in the whole scheduling process. Yao et al. proposed the DEDFS method to improve the application scope of this idea [17]. In their enactment, flows gradually coming with their own deadline, and the previous flows should be calculated whether they have been transmitted already. Then MCF is used at certain interval. This DEDFS need less information of all flows in entire intervals in advance than MCF, but the routing path must be necessary and unchangeable in both DEDFS and MCF.

As a matter of fact, we usually can not know much about burst traffic in network, so solutions consider both routing and scheduling are more reasonable and effective to meet variable requirements. Xu et.al. propose a BEERS method ([18] [19]), the main idea of which is to parse the flow when the flow arrives, then decide a routing path and the start time of each flow. The method is aiming at aggregating traffic on as few links as possible so that they can turn off other idle devices to reduce power consumption, using the power-down strategy [20] [21].

However, the research object may not be the kind of flow mentioned before. A special type of flow which is called Coflow is defined and an algorithm named RAPIER is described [22]. RAPIER adopts the well-known Minimum Remaining Time First (MRTF) policy as its basic scheduling policy.

Our work also focuses on both routing and flow scheduling based on the speed-scaling policy. Note that the "flow" here, means that a certain amount of data that has to be transmitted from a specified source to a specified destination with hard deadline-constraint. Simulation result shows that our EEDFSR has a superior performance and it can show greater advantages in complex topology such as data center network.

III. MODEL ESTABLISHMENT

In this section, we introduce the energy consumption model of network topology and the general model for the deadline-constrained flow energy saving problem.

A. *Energy Consumption Model*

A network G can be seen as a set of nodes and links, in other words, G={N,L}, where N represents all switches and hosts, L expresses all links. It is known that the energy consumption of a switch is not proportional to the traffic load but to the dynamic part which together with the fixed part constitutes the power consumption of a switch [18]. When a switch is working, the fixed part always consumes constant power, while the dynamic part usually refers to the ports [23]. And we use classical queueing model for links, that is, a link is modelled as a forwarding unit with buffers at its two ends.

We mainly discuss the scheduling and routing problem in network, which often expressed as the choice of paths and speed of flows. Our method aims at distributing traffic load across links as well as reducing transmission rate as much as possible in order to improve the utilization and reduce the active energy consumption. We mainly consider the power consumption of network components such as ports and links which are the main power consumers that can be manipulated for energy conservation. The power consumption of the ports at the ends of a link is also abstracted into the power consumption of the link for the ease of exposition.

For the energy consumption model, we have used the power function which is derived from the widely adopted speed scaling model [19]. For each link e ∈L, a power consumption $f_e(x_e)$ is given with respect to the transmission rate x_e of link e. Formally, for every link a function is given which is expressed as below:

$$f(x) = \begin{cases} 0, & x_e = 0 \\ \sigma + \mu x_e^{\alpha}, & 0 < x_e \leq C \end{cases} \tag{1}$$

where σ, μ, α are constants associated with link type. σ is the idle power for maintaining link state, C means the maximum transmission rate of a link and $\alpha > 1$ because the function is super-additive.

B. *Deadline-Constrained Flow Energy Saving Problem*

We model the application requirements in the network as a set of data flows that are transmitted from the source to the destination and have strict time requirements. Different from the assumption from Ref. [13] in which each flow can only follow one single path, we do not have this restriction because in our scheduling process, flows may change routing path in order to avoid bandwidth contention. This action makes our solution more efficient when the network is not so congested. Let a set $F = \{f_1, f_2, ...f_n\}$ represent all flows that have to be transmitted on the network during the whole time interval T . Each flow $f_i \in F$ has its input parameter set as below:

r_i and d_i: the release time and the deadline of f_i;

p_i and q_i: the source and the destination of f_i;

w_i: the amount of data that needs to be transmitted.

So the whole time interval T can be represented as $T = [min_{0<i\leq n} r_i, max_{0<i\leq n} d_i]$, and for each flow f_i, it can only be transmitted during $[r_i$,$d_i]$ with bandwidth b_i.

$$\forall i \in [1, n], b_i \geq w_i/(d_i - r_i) \tag{2}$$

What we need to do is to find one or more suitable path(s) and decide the transmission rate and transmission time for each flow, also taking into account energy saving goals. As has been mentioned before, we will not change the topology during our process, which means the idle power consumption of the active link is a constant. So we mainly analysis the dynamic energy consumption for transmission, which can be expressed by Equation (3).

$$\phi_f = \int_{min_{0<i\leq n} r_i}^{max_{0<i\leq n} d_i} \sum_{e\in L} \mu(x_e(t))^{\alpha} dt \tag{3}$$

Here $x_e(t)$ is the transmission rate of link e at time t. Our goal is to transmit all flows and try to minimize ϕ_f. The routing and scheduling procedure will be introduced and evaluated in the following sections.

IV. ENERGY EFFICIENT DEADLINE-CONSTRAINED FLOW SCHEDULING AND ROUTING METHOD

Our method (EEDFSR) consists of four main steps, each of which completes the transfer of a part of flows except the first one. After the cycles of this four steps, all flows can be handled and the routing and scheduling process will be terminated. We give the flow chart of EEDFSR (shown in Fig. 1), and introduce the detailed process of EEDFSR in this section.

A. *EEDFSR*

When the procedure is triggered, the first step is the off-line step. This step has nothing to do with the flows, just find a feasible path between any two nodes according to the network topology. Then with the arrival of the flows, the second step named preliminary assignment should be started. This step is to find a path that satisfies the bandwidth demand of the flow. If the second step succeeds, the flow can be routing in this path and scheduling with the assigned bandwidth. However, if the second step fails, then the third step named routing change step should be started. The third step is to determine whether the new flow can be provided with a bandwidth-conforming transmission path by modifying the path of a flow currently being transmitted. If feasible, then a flow is selected to modify its transmission path, and a new flow can be assigned to the path. While if the third step fails, then EEDFSR should jump to the fourth step named speeding up step. The fourth step is to make full use of the remaining bandwidth of each link on the path to speed up the transmission of all the flows on this path.

The detailed process of these four main steps will be introduced later.

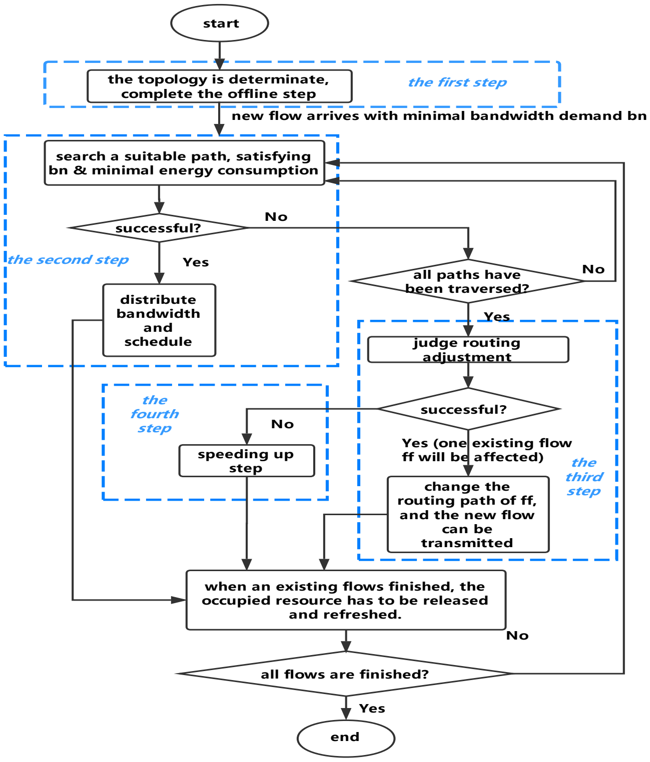


Fig. 1: The flow chart of EEDFSR

The first step is carried out before transmitting. We call it off-line step, in which EEDFSR iterates through all feasible paths between any two nodes in the entire topology and sort them by the path length (or hop counts) in ascending order. Then it traverses each link on each path to find the available bandwidth, and on each link the minimum value of the available bandwidth is measured as the available bandwidth of the path; the path information is again sorted and saved in descending order of available bandwidth. Relatively, the following steps belong to on-line step in which path information needs to be updated and modified in real time.

Then when the routing and scheduling work starts, EEDFSR proceeds to the step of preliminary assignment. When a new flow f_i arrives, EEDFSR seeks for a path from p_i to q_i that satisfies its minimum bandwidth requirement b_i=w_i/(d_i-r_i) and brings the least increase in energy consumption. The power consumption function of links is $f(x_e) = \sigma + \mu x_e^{\alpha}$, so we can assume that the original energy consumption is $P = \sigma + \mu v_0^{\alpha}$ (as Equation 1 explains, σ is the idle power, μ and α is the related parameter, v_0 is the current transmission speed on the link), after the new flow with its speed of v added into the link, the new energy consumption change to $P^{'} = \sigma + \mu(v_0 + v)^{\alpha}$. The path we seek is the one can minimize the $\Delta P = \mu[(v_0 + v)^{\alpha} - v_0^{\alpha}]$. If this step is successfully done, the flow can get its required bandwidth and be scheduled on the chosen path. If this step fails, the algorithm will jump to the third step, which is called routing change step.

algorithm 1 Routing Change Step

```
Input: new flow $flow$, $flow$‘s path set $paths$, current flow set $flows$
Output: the path $path$ and the speed $speed$ of $flow$
1: for each $path' \in paths$ do
2:     for each $flow' \in flows$ do
3:         the min bandwidth of $flow$ is $bandwidth$
4:         the speed of $flow'$ is $speed'$
5:         available bandwidth of $path'$ is $bandwidth‘$
6:         $paths'$=FindPath($flow'$,$speed'$)-p
7:         if $(paths' \neq \emptyset)\&\&(speed' + bandwidth' \geq bandwidth)$ then
8:             $path$=$path'$
9:             $speed$=$bandwidth$
10:            for each $path'' \in paths'$ do
11:                $path'$= $path''$ when $path''$ minimizes the increased energy consumption in $paths'$
12:            end for
13:            $return\ \{path,\ speed\}$
14:        end if
15:    end for
16: end for
17: execute $The\ Fourth\ Step$
```

When the procedure comes to the third step, it can be considered that there is no path that satisfies the bandwidth demand.

This step consists of two parts (shown in Algorithm 1), one is path evaluation and the other is routing adjustment. Firstly, all paths between the source and destination should be traversed to evaluate whether the bandwidth contention can be eliminated by modifying the path of the flow currently being transmitted (line 3- line 7). If the judgement condition is satisfied (line 7), then change the routing path of the chosen flow. After the contention is eliminated, the flow can be transmitted on this path (line 8- line 13). If failed, then jump to the fourth step.

If both the preliminary assignment and the routing change step fail to route and schedule, the speeding up step is the final action, in which we choose a path with least contention and speed up all the flows on this path to make all flows can be transmitted within their own deadline. The core of this part is to calculate the speed of flows, but before introducing it, I think I have to explain a flaw that seems to exist. Speeding up needs bandwidth, so we must make sure the available bandwidth of bottleneck links can be used to transmit flows under the assumption that the former steps can not find a suitable path. On the one hand, network capacity must be enough even if it is not "rich connected". On the other hand, if the available bandwidth is not enough to solve transmission, then this problem is not the level of our consideration, it may involve network congestion, etc..

The set of speed is calculated by the next steps (shown in Algorithm 2).

On the moment t_{now}, we assume that i flows need to be transmitted, so the basic speed should be $v_0 = \sum w_{remain}/(maxdeadline - t_{now})$. We sort the deadlines of flows in ascending order, then we have n ($n \le i$) demarcation point, and the current moment t_{now} can be noted as $deadline_0$. Next, calculate the speed v_i in n time intervals formed by t_{now} with the n deadlines, which is expressed as $v_i = \sum_{f_i \in F_i} w_{remain}/(deadline_i - t_{now})$(line 4). Finally, separately compare v_1, v_2, ..., v_{n-1} with v_0. If $v_i(1 \le i \le n-1) \le v_0$, the transmission rate in interval $[deadline_{i-1}, deadline_i]$ will be set as v_0. Otherwise, the speed will be set as v_i (line 5). As a result, the transmission interval and the corresponding rate are determined.

However, the fourth step is a little controversial. In actual situations, the transmission rate on the link may not be able to make any desired changes, more likely to be a split-gear design. Then the corresponding coping strategy is that the highest priority flow starts transmission according to the lowest gear that is not less than the calculated rate. After the transmission, the current highest priority flow is selected to transmit according to the selection mechanism. But in simulations of methods based on speed-scaling policy, this problem is often ignored in order to have a more intuitive description of the design of speed.

algorithm 2 Calculate Speed For Flows

```
Input: a flow $flow$, $flow$'s path $path$
Output: $flow$'s speed $speed$
1: $speed = \sum w_{remain}/(maxdeadline - t_{now})$
2: for each $interval_i \in flow's[start, end]$ do
3:     for each $link \in path$ do
4:         $speed[i] = \sum_{f_i \in F_i} w_{remain}/(deadline_i - t_{now})$
5:         transmit flow in $interval_i$ in speed of max{speed[i],$speed$}
6:         update $flow_i.w_{remain}$ and record the interval with the speed on it.
7:     end for
8: end for
```

B. Example

After the detailed introduction of EEDFSR, a brief example is given in this part. Table I shows that three flows are going to be transmitted on a small network topology with their information. And the network topology as well as the transmission routing of flow is shown in Fig.2.

The routing and scheduling process is as below:

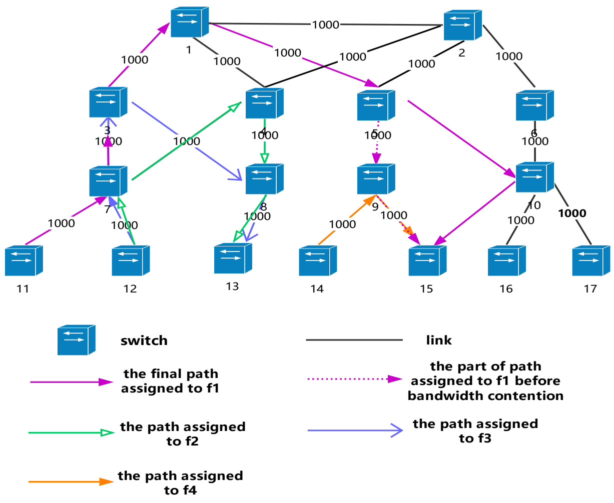


Fig. 2: Example of a small application of EEDFSR

TABLE I: FLOW INFLORMATION

Flow ID	Input information					Output information	
	r/s	d/s	p	q	w/M	path	speed/(M/s)
1	0	4	11	15	800	$11 \to 7 \to 3 \to 1 \to 5 \to 9 \to 15$; $11 \to 7 \to 3 \to 1 \to 5 \to 10 \to 15$	200
2	0	3	12	13	900	$12 \to 7 \to 4 \to 8 \to 13$	300;600
3	1	2	12	13	800	$12 \to 7 \to 3 \to 8 \to 13$	800
4	3	4	14	15	900	$14 \to 9 \to 15$	900

1) Before starting transmitting, the EEDFSR searches for all available paths between any two points for the later path planning.
2) At the moment 0, f_1 and f_2 arrives, the EEDFSR finds a path($11 \to 7 \to 3 \to 1 \to 5 \to 9 \to 15$) for f_1 and distributes 200 M/s as its transmission bandwidth; after computing and comparing,the EEDFSR finds the other path($12 \to 7 \to 4 \to 8 \to 13$) for f_2, and distributes 300 M/s for it.
3) At the moment 1, f_3 arrives, but the available bandwidth of link 12-7 and 8-13 are less than the minimal demand of f_3. The routing change step is also failed, so we take the speeding up step on the two contention links. The remaining data of f_2 is 600 M, so the speed on link 12-7 and 8-13 in time interval [1, 2]should be max{(600+800)/2, 800/1}=800 M/s, and because of the earlier deadline, f_3 will be scheduled earlier. And in time interval [2, 3], the speed on the two links is 600/1=600 M/s. So the speed of f_2 in [2, 3] is 600 M/s. The path assigned to f_3 is $12 \to 7 \to 3 \to 8 \to 13$.
4) At the moment 3, f_4 arrives, but the available bandwidth of link 9-14 and 9-5 are both less than the minimal demand of f_4. But the contention can be solved by switching the partial path of f_1 from 5 to 15. So in the interval [3, 4], the routing path of f_1 is changed to another choice as $11 \to 7 \to 3 \to 1 \to 5 \to 10 \to 15$.

And the speed of f_1 is still 200 M/s and the speed of f_4 is the minimal demand of it (900 M/s).

V. SIMULATION & EVALUATION

In this section, the simulation environment is described and the performance of the proposed method is also analysed.

A. *Simulation Environment*

We build a simulator with the EEDFSR implemented in Java and take the energy model as 36W for fixed power of active switches, the active power consumption functions is selected as x^2 (refer to Ref. [16]), in other words, $\alpha = 2$ in (1).

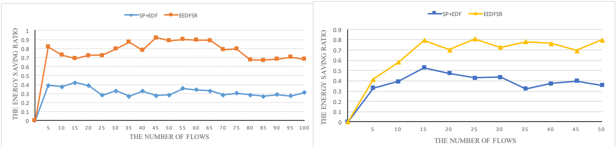
Before our simulation, two input elements are explored: the network topology and the input flow information. For the former, we separately adopt a general topology with enough connectivity (shown as Fig. 2) and a Fat-Tree topology (pod= 4) with the capacity of each link is 1000 M/s and evaluate the applicability of this two methods. For the later, the flows are generated randomly by a host to another one. We consider the overall time interval is [1, 50] and as we assume no prior knowledge on the flows, we select release times and deadlines of flows randomly in [1, 50]. The number of flows ranges from 5 to 100 in fat-tree topology and 5 to 50 in the general topology. The amount of data from each flow is given by a random rational number.

Due to the following two reasons: 1) the topology won't be changed during our whole routing and scheduling period; 2) the fixed power consumption occupy a relatively large proportion of energy consumption. So we only focus on the active power consumption, not all.

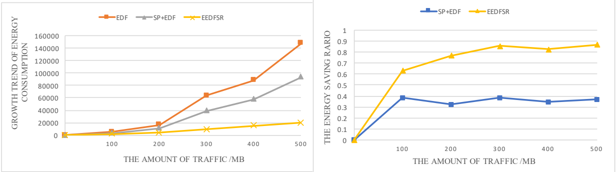
B. *Metrics and Comparison Method*

It must be explained that the energy consumption is mainly computed by the power consumption function in (1) with corresponding transmission time, but to be honest, we can not find the correspondence between this value and the actual power consumption which usually be measured in W, KW or others. So in Fig. 3 (a) (b) (d), we use the energy saving ratio (the energy saved by this method compared to the baseline method as a percentage of the baseline method) to compare the performance of the two methods. And in the vertical coordinates in Fig. 3 (c) are only the computed values without unit of measurement to describe the growth trend of energy consumption.

Since our method is based on topology adjustment, we can increase the energy efficiency by distributing the flows to the entire network topology and adapting the link rate according to the link load. It is not comparable to the methods in the referenced papers. So we choose the EDF method as the baseline method, and compare the energy saving ratio of SP+ EDF with our EEDFSR method in a Fat-tree topology (pod= 4) and a general topology (shown in Fig. 2). But we must note that because of the randomness of path choosing in baseline method, the energy consumption of it is critically depend on the choices of transmitting paths. This may make the same set of input data produce very different results. We control the path length of the choice and use five results to get the average value as the final data to reduce errors.



(a) based on the Fat tree topology (b) based on the general topology



(c) the energy consumption growth trend (d) the impact of traffic size

Fig. 3: Experimental Comparison Chart

C. *Simulation Results and Analysis*

As expected, our EEDFSR outperforms SP+ EDF to a large extent not only in the Fat-tree topology, but also in the general topology with enough connection. (a) and (b) show that the energy saving ratio of EEDFSR can even reach more than 90% in Fat-tree topology and 80% in the general topology, as the energy of SP+ EDF is correspondingly about 40% and 50%. This shows that our approach is generic and not limited to data center networks, but it is clearly more advantageous in data center networks.

As a matter of fact, the amount of traffic on the network is also a big factor. We set the number of flows to 100, and then gradually increase the transmission data of flows to explore the growth trend of energy consumption. We show the result in (c) and (d). It's intuitively that the growth trend is much lower in EEDFSR than others, and the energy saving ratio also shows its superiority than the comparison method.

VI. CONCLUSION

In this paper, we focus on the network-level energy saving problem. We design the EEDFSR method to achieve energy saving, which combines routing with scheduling and is based on the speed-scaling policy. We compare the EEDFSR with the SP+EDF method in a small Fat-tree topology and a general topology with enough connectivity. As expected, it performs much better and has a much higher utilization of links in both two topologies.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China (61501044).

REFERENCES

[1] Kong G, Wang H, Huang F, et al. An Energy-Aware Ant Colony Optimization Routing Algorithm in the Private Network[C]// IEEE, International Conference on High PERFORMANCE Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing. IEEE, 2014:1681-1686.

[2] Gao P X, Curtis A R, Wong B, et al. It's not easy being green[J]. Acm Sigcomm Computer Communication Review, 2012, 42(4):211-222.

[3] Abts D, Marty M R, Wells P M, et al. Energy proportional datacenter networks[C]// International Symposium on Computer Architecture. ACM, 2010:338-347.

[4] Gupta M, Grover S, Singh S. A feasibility study for power management in LAN switches[C]// Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on. IEEE, 2004: 361-371.

[5] Ananthanarayanan G, Katz R H. Greening the Switch[C]//HotPower. 2008.

[6] Gupta M, Singh S. Greening of the Internet[C]//Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications. ACM, 2003: 19-26.

[7] Fisher W, Suchara M, Rexford J. Greening backbone networks: reducing energy consumption by shutting off cables in bundled links[C]// Proceedings of the first ACM SIGCOMM workshop on Green networking. ACM, 2010: 29-34.

[8] Mumey B, Tang J, Hashimoto S. Enabling green networking with a power down approach[C]//Communications (ICC), 2012 IEEE International Conference on. IEEE, 2012: 2867-2871.

[9] Aleksic S. Analysis of power consumption in future high-capacity network nodes[J]. Journal of Optical Communications and Networking, 2009, 1(3): 245-258.

[10] Heller, B., Seetharaman, S., Mahadevan, P., Yiakoumis, Y., Sharma, P., Banerjee, S., & McKeown, N. (2010, April). Elastictree: Saving energy in data center networks. In Nsdi(Vol. 10, pp. 249-264).

[11] Shang, Y., Li, D., & Xu, M. (2010). Energy-aware routing in data center network. (pp.1-8). ACM.

[12] Xu, M., Shang, Y., Li, D., & Wang, X. (2013). Greening data center networks with throughput-guaranteed power-aware routing. Computer Networks, 57(15), 2880-2899.

[13] Li, D., Yu, Y., He, W., Zheng, K., & He, B. (2015). Willow: Saving data center network energy for network-limited flows. IEEE Transactions on Parallel and Distributed Systems, 26(9), 2610-2620.

[14] Zhang, Q., Zhang, X., Peng, J., Zhao, Y., Liu, K., & Li, S. (2016, November). Energy Optimization by Flow Routing Algorithm in Data Center Network Satisfying Deadline Requirement. In Asia-Pacific Services Computing Conference(pp. 117-129). Springer, Cham.

[15] Li, D., Shang, Y., He, W., & Chen, C. (2015). EXR: greening data center network with software defined exclusive routing. IEEE Transactions on Computers, 64(9), 2534-2544.

[16] Wang, L., Zhang, F., Zheng, K., Vasilakos, A. V., Ren, S., & Liu, Z. (2014, June). Energy-efficient flow scheduling and routing with hard deadlines in data center networks. In Distributed Computing Systems (ICDCS), 2014 IEEE 34th International Conference on (pp. 248-257). IEEE.

[17] Yao, Z., Wang, Y., Ba, J., Zong, J., Feng, S., & Wu, Z. (2017, November). Deadline-aware and energy-efficient dynamic flow scheduling in data center network. In 2017 13th International Conference on Network and Service Management (CNSM)(pp. 1-4). IEEE.

[18] Xu, G., Dai, B., Huang, B., & Yang, J. (2015, August). Bandwidth-aware energy efficient routing with sdn in data center networks. In High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conferen on Embedded Software and Systems (ICESS), 2015 IEEE 17th International Conference on (pp. 766-771). IEEE.

[19] Xu, G., Dai, B., Huang, B., Yang, J., & Wen, S. (2017). Bandwidth-aware energy efficient flow scheduling with SDN in data center networks. Future Generation computer systems, 68, 163-174.

[20] Zhao, Y., Chen, K., Bai, W., Yu, M., Tian, C., Geng, Y., ... & Wang, S. (2015, April). Rapier: Integrating routing and scheduling for coflow-aware data center networks. In Computer Communications (INFOCOM), 2015 IEEE Conference on (pp. 424-432). IEEE.

[21] Kaup, F., Melnikowitsch, S., & Hausheer, D. (2014, November). Measuring and modeling the power consumption of openflow switches. In Network and Service Management (CNSM), 2014 10th International Conference on (pp. 181-186). IEEE.

[22] Nedevschi, S., Popa, L., Iannaccone, G., Ratnasamy, S., & Wetherall, D. (2008, April). Reducing Network Energy Consumption via Sleeping and Rate-Adaptation. In NsDI (Vol. 8, pp. 323-336).

[23] Wong, S. W., Valcarenghi, L., Yen, S. H., Campelo, D. R., Yamashita, S., & Kazovsky, L. (2009, November). Sleep mode for energy saving PONs: Advantages and drawbacks. In GLOBECOM Workshops, 2009 IEEE (pp. 1-6). IEEE.