Small versus Large: Switch Sizing in Topology Design of Energy-Efficient Data Centers

Indra Widjaja, Anwar Walid
Bell Labs, Alcatel-Lucent
Murray Hill, NJ, USA

Yanbin Luo, Yang Xu, H. Jonathan Chao
Polytechnic Institute of New York University
Brooklyn, NY, USA

***Abstract*—Saving power in datacenter networks has become a pressing issue. While in operation, ElasticTree and CARPO can save power consumed by a fat-tree network by using sleep mode where some components such as ports and switches are turned off when traffic demand in the network is relatively moderate. In this paper, we propose a new approach by exploring the design stage of a datacenter network and focus on how to choose the right switch size that can potentially save the most power during the expected operation of the network. We also consider speed scaling where the power of a switch can be varied by adjusting its processing rate according to its traffic demand. We use analysis and simulation to investigate the power-saving performance of different switch sizes, power-saving modes and traffic demand patterns. Our findings with sleep mode reveal that deploying a large number of small switches is more power-efficient than a small number of large switches when the traffic demand is relatively moderate or when servers exchanging traffic are in close proximity. With speed scaling, the reverse is generally true.**

I. Introduction

High energy consumption has become a serious concern in the design of large-scale enterprise data centers which aim to provide reliable and scalable computing infrastructure for massive Internet services. In addition to high electricity bills and negative environmental implications, increased power consumption may lead to system failures, as data centers increasingly deploy new high-density servers, while their power distribution and cooling systems are approaching their peak capacity. Energy proportional computing [1] has emerged as a new paradigm for the design and operations of datacenter servers where the energy consumption is made to scale with the CPU speed using dynamic voltage and frequency scaling (DVFS) [2] (also known as speed-scaling). More recently, there is new awareness and efforts in tackling energy consumption at the datacenter communication network, which consists of the switches and the links that interconnect the servers [3], [4], [5].

In this paper our goal is to enable energy-proportional datacenter communication networks. In particular, we are interested in making the amount of energy consumed proportional to the traffic intensity (offered load) in the network. Prior work [3] and [4] focused on developing algorithms for dynamically adjusting the set of active links and switches in a particular datacenter topology, namely the fat-tree topology, to satisfy changing datacenter traffic loads. Our contributions center on developing fundamental insights into key structural and scaling properties of the datacenter network that facilitate energy-proportional communications and on how the current and future technologies impact and modify these properties. Such insights are useful in guiding the design and deployment of future datacenter topologies and the analysis of different competing alternatives. Our results are derived based on formulation of a network design optimization problem whose solution provides the optimal size and topology of the network that supports certain number of servers and their traffic loads. Other important elements in the design optimization are the power consumption models of the ports and switches and the level of safety margin (additional capacity beyond normal levels) to handle unpredictable traffic surges.

Recent measurement studies of switch power consumption [6] show that turning on the switch consumes most of the power; going from zero to full rate increases power by less than 8%. Therefore, with today's technology, our best option for saving energy in the network is to manage the non energy-proportional network components intelligently. In particular, a switch or port is opportunistically turned off, referred to here as "sleep mode" operation, during periods of low traffic demands to achieve most of the power-saving benefits. Thus, a network of non energy-proportional components can act as a load-proportional ensemble. There are design choices for building a communication network to support certain number of servers, where each design choice specifies the number of switches, their sizes and their interconnection topology. With the sleep mode operation, we show that a topology with many small switches is more energy-efficient than a topology with few large switches when servers are communicating in close proximity or when the traffic demand is low.

The advances made in making energy consumption of servers load-proportional using speed-scaling will likely help improve the energy dynamic range of the switches and ports, and make inter-server communication energy cost more proportional to the amount of data being transmitted. [5] gives a proposal for dynamically tuning individual plesiochronous links to match the required load while consuming as little power as possible. [5] also highlights opportunities for future energy-proportional switch chips. As Ethernet switches typically come in different sizes, it is natural to ask whether a specific switch size can optimize a given objective function during the operation of a data center. In this paper, we use energy as the objectve function we want to minimize. Our

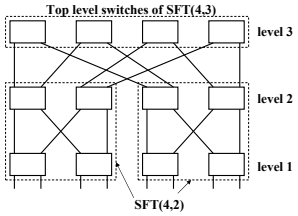

Fig. 1. SFT$(4,3)$ constructed from two SFT$(4,2)$'s.

study reveals that when switches operate in speed-scaling mode, the optimal topology favors few large switches, which is contrary to our findings with the sleep-mode operation.

In the next section, we provide the preliminaries on generalized fat-trees. We then present power consumption analysis with sleep mode in Sec. III followed by speed-scaling mode in Sec. IV. Experiments using NS-3 are investigated in Sec. V. We also evaluate the impact of energy-saving mechanisms on quality-of-service measures such as packet loss rate and packet delay. Finally, we conclude the paper in Sec. VI.

II. Preliminaries

Although our methodology is applicable to any topology, we focus on fat-tree networks as they are widely deployed in data centers. A fat-tree maintains constant bisection bandwidth as one traverses from the switches at the bottom of the tree to the switch at the root [7]. Prior work on fat-tree datacenter networks only considers a configuration with 3 levels of switches [3][4]. In this section, we describe the construction of a fat-tree network with an arbitrary number of levels. This allows us to pose a deeper power-consumption problem at the *design stage* where one can choose the right topology with the right switch size so that power consumption is reduced during the expected *operation stage*. We also address speed-scaling mode while prior work only deals with sleep mode.

We adopt some notations in [8][9]. A fat-tree, FT(k,n), consists of of n levels of k-port switches, where k is a multiple of 2. We call level-n switches for those at the top and level-1 switches for those that connect to servers at the bottom. A fat-tree can be constructed by connecting each of the (level-n) switches of FT(k,n) with k sub-fat-trees SFT$(k,n-1)$'s that have $n-1$ levels of switches. In general, SFT(k,l), $1<l<n$, can be recursively constructed by connecting each of the top-level (level-l) switches of SFT(k,l) with $k/2$ SFT$(k,l-1)$'s. At level 2, each of the top-level switches of SFT$(k,2)$ is connected to $k/2$ SFT$(k,1)$'s, where each SFT$(k,1)$ is just a single switch with $k/2$ down-links connecting to servers. Fig. 1 shows an example of SFT$(4,3)$.

Note that each of the $k/2$ SFT$(k,1)$'s has $k/2$ up-links. Thus, the total number of up-links from $k/2$ SFT$(k,1)$'s to $k/2$ top-level switches of SFT$(k,2)$ is $(k/2)^2$. Generally, it can be easily seen by induction that the total number of up-links from $k/2$ SFT$(k,l-1)$'s to $(k/2)^{l-1}$ top-level switches of SFT(k,l) is $(k/2)^l$, for $1<l<n$. Since there are k SFT$(k,n-1)$, the total number of up-links from k SFT$(k,n-1)$'s to the top-level switches of FT(k,n) is $2(k/2)^n$. Thus, the number of switches at level n is $2(k/2)^n/k=(k/2)^{n-1}$. For $l\neq n$, the number of switches at each level is $2(k/2)^{n-1}$. Thus, the total number of switches in FT(k,n) is $(2n-1)(k/2)^{n-1}$ and the total number of servers (racks) supported is $N_h=2(k/2)^n$. Note that for a given number of servers N_h, one can choose different pairs of (k,n) to support the servers. Generally, it may not be possible to find the pairs that give the same value $2(k/2)^n$. In this case, some sub-fat-trees SFT$(k,\cdot)$'s of FT(k,n) can be removed if the number of servers is less than those that can be supported by the 'full' fat-tree.

III. Power Consumption with Sleep Mode

A. Formulation for General Datacenter Networks

We use the term sleep mode to indicate that a given network component can be turned off when there is no traffic through it. The optimization problem for minimizing power consumed by the network can be formulated as an integer linear program. Formally, let a datacenter network be represented as a graph $G=(N,L)$, where N is the set of switches and L is the set of unidirectional links. Let M be the set of server-to-server unidirectional traffic with each flow $m\in M$ of rate d^m entering an ingress switch s^m and exiting an egress switch t^m. If some of the traffic m is routed through the link from switch i to switch j, we let x_{ij}^m represent the bandwidth consumed on the link (i,j) by the traffic. We assume the capacity of the link $(i,j)\in L$ is C_{ij}. Usually, $C_{ij}=C_{ji},\forall i,j$. Let X_{ij} denote a binary variable that is equal to 1 if link (i,j) is enabled and 0 otherwise. In practice, $X_{ij}=X_{ji}$ as the link/port is bidirectional. Also, let Z_i denote a binary variable that is equal to 1 if switch i is enabled and 0 otherwise, $\forall i\in N$. To quantify the power consumption, we further let P_i^s denote the power of switch $i\in N$ and let $P_{i,j}^l$ denote the power to turn on link (i,j), which may be associated to a given port at switch i. The power of a switch consists includes all components (chassis, switching fabric, line cards, etc.), except ports (links).

For a given traffic demand matrix, the optimization problem that minimizes power consumption is as follows.

$$\text{minimize} \sum_{(i,j)\in L} P_{i,j}^l X_{ij} + \sum_{i\in N} P_i^s Zi \tag{1}$$

subject to

$$\sum_{j\in N:(i,j)\in L} x_{ij}^m - \sum_{j\in N:(j,i)\in L} x_{ji}^m = \begin{cases} d^m, & i=s^m, m\in M \\ -d^m, & i=t^m, m\in M \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$\sum_{m\in M} x_{ij}^m \le C_{ij}, \quad \forall (i,j)\in L. \tag{3}$$

$$X_{ij}=X_{ji}, \quad \forall (i,j)\in L. \tag{4}$$

$$\sum_{m\in M} x_{ij}^m/C_{ij} \le X_{ij}, \quad \forall (i,j)\in L. \tag{5}$$

$$X_{ij}\le Z_i, \quad \forall (i,j)\in L. \tag{6}$$

$$Z_i \le \sum_{m\in M, j\in N:(i,j)\in L} x_{ij}^m, \quad \forall i\in N. \tag{7}$$

Eq. (1) defines the objective function to be minimized. Eq. (2) ensures flow conservation. While Eq. (3) represents the bandwidth constraint on the link, Eq. (4) states the natural bi-directional property of the activity of a link. Eq. (5) ensures that a link is enabled when there is a non-zero flow through the link, while Eq. (6) ensures that a switch is enabled when one of its ports is enabled. Finally, Eq. (7) ensures that a switch is deactivated when there is no flow through it. The above optimization problem has been implemented in AMPL and solved by CPLEX. Unfortunately, the problem is NP-complete for integer flows. We thus explore an alternative method that can efficiently solve a large problem size.

B. Efficient Computation for Fat-Tree Networks

1) Power Consumption Analysis: The formulation in the previous subsection allows for different switches to have different number of ports, different links to have different capacities, and the topology to be arbitrary. On the other hand, a fat-tree topology exhibits high regularity. In particular, links typically have the same capacity, switches have the same size, and the topology is regular. We take advantage of the regularity of a fat-tree network to decide whether to turn on/off switches or ports. The following approach gives the exact same results for power consumption as the approach in the previous subsection, but comes with much less computational burden.

From the description in Sec. II, note that each of the top-level switches of SFT(k, l) (or FT(k, n) at level n) has a link to each of the SFT($k, l-1$)'s. This observation allows us to view the top-level switches of a given SFT(k, l) as *one logical switch*, LS(k, l), consisting of multiple switches. Each logical switch LS(k, l) can continue to maintain the connectivity to all its children LS($k, l-1$)'s as long as at least one switch in LS(k, l) remains on. The number of switches that can be turned off depends on the load of the logical switch.

Let the capacity of each k-port switch be denoted by $C_s(k) = k\,C_p$, where C_p is the port/link capacity. Since the number of switches per logical switch at level l is $(k/2)^{l-1}$, the maximum capacity of a logical switch is $C_{LS}(k,l) = (k/2)^{l-1}C_s(k)$, when all switches and ports are turned on. Let the aggregate traffic rate from a given LS(k, l) to its parent LS($k, l+1$) be denoted by $R^u(k,l)$ and the aggregate traffic rate from the parent LS($k, l+1$) to the LS(k, l) be denoted by $R^d(k,l)$. Then, $R(k,l) = \max\{R^u(k,l), R^d(k,l)\}$ is the aggregate traffic that needs to be supported by the *up-link* ports of LS(k, l) or the corresponding SFT(k, l). The number of up-link ports that are to be turned on at LS(k, l) for minimum power consumption is

$$N_p^u(R,k,l) = \max\{\lceil R(k,l)/C_p \rceil, 1\}, \forall\, l < n,$$

with at least one port turned on to maintain connectivity to all servers. The rest of the up-link ports of LS(k, l) can be turned off and the corresponding *down-link* ports at the parent LS($k, l+1$) are also turned off. As the parent LS($k, l+1$) has multiple children, the parent will turn off its corresponding down-link ports for each child. Let $N_p^d(R,k,l+1)$ denote the sum of all down-link ports that are to be turned on at a parent LS($k, l+1$) to incur minimum power consumption. At each level l, the number of switches at logical switch LS(k, l) that are to be turned on is

$$N_s(R,k,l) = \begin{cases} \max\{\lceil N_p^d(R,k,l)/k \rceil, 1\}, & l = n, \\ \max\{\lceil N_p^d(R,k,l)/(k/2) \rceil, 1\}, & 1 < l < n, \\ 1, & l = 1, \end{cases}$$

where we assume that at least one switch per logical switch is turned on to maintain connectivity.

The calculation for power consumption with sleep mode is illustrated in Algorithm 1. The computational complexity is $O((k/2)^n)$, which is extremely low as it is linear in the number of servers. In the algorithm, we assume that the switches and their down-link ports at level-1 are always turned on to maintain connectivity to all servers.

Algorithm 1 Power consumption with sleep mode

```
1: INPUT: k, n and traffic demand matrix.
2: OUTPUT: Power consumption.
3: Construct a tree with each node representing a logical switch.
4: Route all the traffic demand.
5: for each level l from 1 to n
6:   for each LS(k, l) from 0 to 2(k/2)^{n-l} - 1
7:     Retrieve saved result from children for N_p^d(R, k, l), if l > 1.
8:     Compute N_s(R, k, l), if l > 1.
9:     Turn off unused down-links and switches, if l > 1.
10:    Compute N_p^u(R, k, l).
11:    Turn off unused up-links.
12:    Update the corresponding N_p^d(R, k, l+1) at LS(k, l)'s parent.
13:      and save the result for steps 7-9, if l < n.
14:  end for
15: end for
16: Compute power based on the number of enabled switches and links.
```

In the computation of $N_p^u(R,k,l)$, we assume that the traffic is packed to minimize the number of up-link ports used. To have efficient packing where each up-link port from 0 to j_l^{max} ($j_l^{max} < (k/2)^l$) will have nearly equal load, we adopt an approach where each switch applies Valiant Load Balancing (VLB) among the active ports. If each of SFT(k, l)'s, $l = 1, \cdots, n-1$, packs its up-link traffic to its left-most ports and the interconnections to SFT($k, l+1$) form a perfect-shuffle, then the traffic will be packed to the left most switches of SFT($k, l+1$); among those that carry traffic, the traffic will also be packed to the left most down-link ports.

C. Power-Consumption Evaluation

We evaluate fat-tree networks for different configurations using Algorithm 1. For computing power consumption, we categorize a switch into two components: chassis and ports. The chassis includes the switching fabric and other modules such as line cards, processing modules, etc. It has been shown that the power consumption of a multi-stage switching fabric scales according to $O(k \log k)$, while a crossbar-type switching fabric scales according to $O(k^2)$ [10], where k is the number of ports. For other modules such as line cards, it is reasonable to assume that the power consumption scales according to $O(k)$. Suppose there is a choice of several switch sizes where the smallest one is equipped with k_{min} ports. Then, the chassis

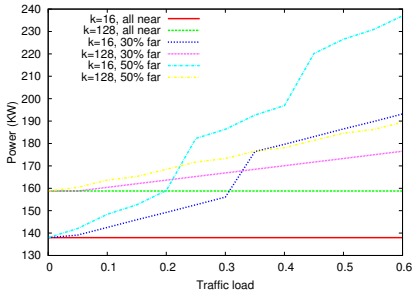

Fig. 2. Power consumption with sleep mode under near-far traffic.

power of a k-port switch, when the switch is turned on, can be expressed as

$$P^{s(k)} = \eta_1(k/k_{min}) + \eta_2(k\log(k))/(k_{min}\log(k_{min}), \quad (8)$$

and the chassis power is zero if the switch is turned off. The first term represents the scaling due to other modules and the second term represents the scaling due to a switching fabric. Note that we take a more conservative power model for the switching fabric.

In the following, we assume that the parameter values in Eq. (8) are $\eta_1 = 50$W and $\eta_2 = 50$W. In addition, each port has a capacity $C_p = 10$ Gbps and consumes 2W when it is turned on (active). A port consumes zero power if it is turned off. We observe that the general qualitative results do not change when we experimented with other combinations of the above parameter values.

For the traffic patterns, we decided to the model described in previous work ([3][4]) to maintain consistent qualitative results. Specifically, we consider the model where each server i, for $i = 0, \cdots, N_h - 1$, sends traffic to a fixed server j at a rate given by

$$\lambda_{i,j} = \begin{cases} d, & \text{if } j = (i+z) \bmod N_h \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

We set $z = 1$ for *near* traffic and $z = N_h/2$ for *far* traffic in Eq. (9). A mixture of the two consists of far traffic of rate αd and near traffic of rate $(1-\alpha)d$, where α is the percentage of far traffic over the total traffic.

Fig. 2 compares the powers consumed by two different fat-tree configurations that support the same number servers $N_h = 8192$: FT$(16, 4)$ and FT$(128, 2)$. While FT$(16, 4)$ needs 3584 small switches (16-port), FT$(128, 2)$ can be deployed with 192 large switches (128-port). As can be seen from the figure, if $\alpha = 0$ (all near traffic), the power consumed with small switches is always less than that consumed with large switches. This is because only a low percentage of small switches need to be turned on and these small switches consume less power than the large ones. If $\alpha = 0.3$ (30% far traffic), small switches still consume less power than large switches when the traffic load (d/C_p) is below 0.35 approximately. Beyond that, many more small switches need to be turned on and the power consumption with small switches becomes larger than that with large switches.

IV. Power Consumption with Speed-Scaling Mode

A. Formulation for General Datacenter Networks

With *speed scaling* or rate adaptation [11], the frequency at which a network component (e.g., a switch) operates is increased or decreased to match the offered load. The power consumption of a switch i is modeled as an affine function to reflect power proportionality [11] and is given by

$$P_i^s(x) = a_i x + b_i, \quad (10)$$

where a_i and b_i are some constants and x is the traffic load at the switch. Since a port generally cannot be subjected to speed scaling, we assume that a port can only be turned on or off. The optimization problem with speed scaling can be formulated as follows.

$$\text{minimize} \sum_{(i,j)\in L} P_{i,j}^l X_{ij} + \sum_{i\in N,(j,i)\in L, m\in M} P_i^s(x_{j,i}^m + d^m) \quad (11)$$

subject to

$$\sum_{j\in N:(i,j)\in L} x_{ij}^m - \sum_{j\in N:(j,i)\in L} x_{ji}^m = \begin{cases} d^m, & i = s^m, m \in M \\ -d^m, & i = t^m, m \in M \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

$$\sum_{m\in M} x_{ij}^m \le C_{ij}, \quad \forall (i,j) \in L. \quad (13)$$

$$X_{ij} = X_{ji}, \quad \forall (i,j) \in L. \quad (14)$$

$$\sum_{m\in M} x_{ij}^m / C_{ij} \le X_{ij}, \quad \forall (i,j) \in L. \quad (15)$$

The objective function in Eq. (11) consists of port power using sleep mode and switch power using speed-scaling mode. The interpretations of Eq. (12)-Eq. (15) are the same as before.

B. Efficient Computation for Fat-Tree Networks

Efficient computation of the network power consumption with speed scaling is also possible for fat-tree networks. Here, switches employing speed scaling are never turned off. Suppose the power of a switch i receiving traffic at rate λ_i is given by $P_i^s(\lambda_i) = a_i\lambda_i + b_i$. If switches are within a logical switch LS(k, l), then $a_i = a$ and $b_i = b, \forall i \in LS(k, l)$, as these switches are of the same type. The power consumed by a logical switch LS(k, l) that is carrying traffic at rate λ_i at each switch i within the logical switch is given by

$$P^{LS} = \sum_{i\in LS(k,l)} (a_i\lambda_i + b_i) = \sum_{i\in LS(k,l)} (a\lambda_i + b) = a\lambda + bN_{LS},$$

where the aggregate traffic rate is $\lambda = \sum_{i\in LS(k,l)} \lambda_i$ and the number of switches in a logical switch is N_{LS}. The above implies that the power consumed by a logical switch is invariant to how the aggregate traffic rate is distributed among the switches within a given logical switch as long as no switch is overloaded. As a result, the computation for power consumption at a logical-switch level is simplified as we only need to consider the aggregate traffic at a logical switch (but not at the individual switch).

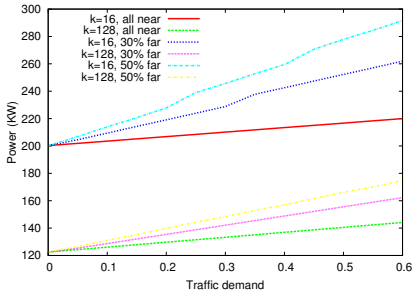

Fig. 3. Power consumption with speed-scaling mode under near-far traffic.

In addition to considering the traffic between LS(k,l) and LS$(k,l+1)$ (i.e., $R^u(k,l)$ and $R^d(k,l)$) as described in Sec. III-B1, we also need to consider the traffic from each source LS$(k,l-1)$ to each destination LS$(k,l-1)$ that passes through a given LS(k,l) in order to account *all traffic entering* LS(k,l). Let the aggregate rate of all traffic entering LS(k,l) be denoted by $\check{R}(k,l)$. We summarize the procedure for implementing speed-scaling mode in Algorithm 2.

Algorithm 2 Power consumption with speed-scaling mode

1: INPUT: k, n and traffic demand matrix.
2: OUTPUT: Power consumption.
3: Construct a tree with each node representing a logical switch.
4: Route all the traffic demand.
5: for each level l from 1 to n
6: for each LS(k,l) from 0 to $2(k/2)^{n-l}-1$
7: Retrieve saved result from children for $N_p^d(R,k,l)$, if $l>1$.
8: Turn off unused down-links, if $l>1$.
9: Compute P^{LS} (power at LS(k,l)).
10: Compute $N_p^u(R,k,l)$.
11: Turn off unused up-links.
12: Update the corresponding $N_p^d(R,k,l+1)$ at LS(k,l)'s parent.
13: and save the result for steps 7-8, if $l<n$.
14: end for
15: end for
16: Compute total power from the switches and the number of enabled links.

C. Power-Consumption Evaluation

We assume that a logical switch LS(k,l) uses speed-scaling and its power consumption [11] is given by

$$P^{LS} = P^{idle} + (P^{max} - P^{idle})\check{R}(k,l)/C_{LS}(k,l),$$

where $C_{LS}(k,l)$ is the capacity of a logical switch. P^{idle} is the power consumption when the logical switch is idle and cannot be eliminated through speed-scaling. P^{max} is the maximum power consumption when the incoming traffic rate reaches the switch capacity. Since a logical switch LS(k,l) consists of $(k/2)^{l-1}$ k-port switches, $P^{max} = (k/2)^{l-1}P^{s(k)}$, where $P^{s(k)}$ is the power consumption of a k-port switch given in Eq. (8).

Fig. 3 shows the power consumption using the same near-far-traffic model as in Sec. III. It is assumed that $P^{idle} = 0.5P^{max}$. The qualitative results remain unchanged over a wide range of P^{idle} values. Because small switches require more ports and there is no opportunity to turn off switches with speed-scaling, a large number of small switches are likely to consume more power than a small number of large switches irrespective of the traffic patterns.

V. Simulation Evaluation

A. Simulation Setting

In this section, we turn to simulation to evaluate the impact of power saving on performance. The simulation experiments were conducted on a datacenter testbed built on NS-3. Typically datacenter networks incorporate some level of capacity safety margin to prepare for traffic surges [3]. In such cases, the network may allocate more capacity than essential for normal workload. To implement safety margin, we monitor the utilization of each outgoing port of a switch. If the safety margin is set to θ, then a new port on the same side of the switch will be enabled (opened) when the utilization exceeds $1-\theta$. The corresponding port in another switch will also be enabled to establish the new link.

In the simulation, each server injects a certain number of UDP flows into the fat-tree network with packets of size 1.5KB. To emulate flow arrivals and terminations, we give each flow two states: ON and OFF. The duration of each flow (i.e., the ON state) is an exponentially distributed random variable, which is determined when the flow is generated. The idle time (i.e., the OFF state) of each flow is also an exponentially distributed random variable, decided when the previous ON state finishes. In our simulation, the average ratio of ON period to OFF period is 3.

B. Packet Delay and Loss Rate

Since packet-mode simulation is very time-consuming, we evaluate packet delay and loss rates only on small fat-tree networks with 512 servers using sleep mode. With sleep mode, the fat-tree initially starts as a *thin-tree* when the load is low as the bisection bandwidth decreases at higher levels. It becomes fatter as the load increases. We want to study if the power-saving feature (i.e., sleep mode) adopted by the fat-tree networks will have any adverse impact on performance. Fig. 4 shows the packet delay and packet-loss rates in a 4-level, 512-server fat-tree network FT$(8,4)$. The traffic pattern from each server consists of 50% near traffic and 50% far traffic, and the safety margin is $\theta = 0.1$. We can see that the packet-delay curve shows a sawtooth pattern, with an increasing trend as the load increases. The sawtooth pattern of the delay curve is the consequence of flow consolidation. Packet delay in a fat-tree network is dominated by the most congested link(s). When the traffic load is very low, the system only needs a minimal spanning tree and no congestion occurs. Also, the packet delay is very low. As the traffic load increases, multiple links of the minimal spanning tree starts to be more congested as can be observed by the increasing trend of the packet delay. When the load on the most congested link exceeds a pre-determined threshold (i.e., $1-\theta = 0.9$ in this case), a new path will be activated to relieve the current congestion. That is why there are recurrent drops in the packet delay, forming a sawtooth pattern, as the traffic load increases.

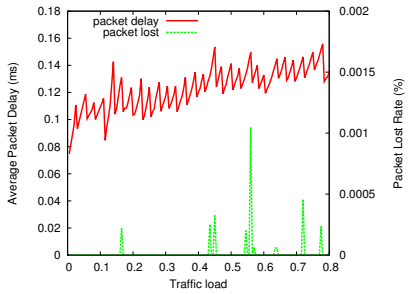

Fig. 4. Packet delay and loss rate of fat-tree network FT$(8, 4)$ (safety margin $\theta = 0.1$, traffic pattern: 50% far).

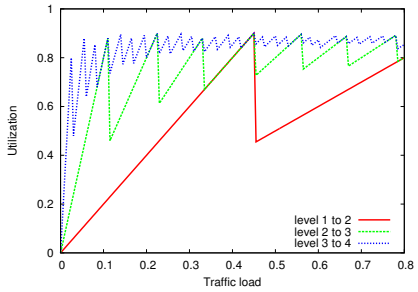

Fig. 5. Average utilizations of links between different levels of fat-tree network FT$(8, 4)$ (safety margin: $\theta = 0.1$, traffic pattern: 50% far).

The above phenomenon can be easily understood when we look at Fig. 4 and Fig. 5 together. Fig. 5 shows the average link utilization values at different levels of the fat-tree network. These curves also show a sawtooth pattern. When the traffic load increases, the link utilization values at different levels increase linearly but with different slopes. The differences in slopes are due to the fact that the number of opened links between levels 3∼4 is only $2/k$-th (in this case, $k = 8$) of that between levels 2∼3, while the total traffic loads on both levels are identical. So, when the traffic load increases, the load on links between levels 3∼4 increases much faster than those on links between levels 2∼3 and level 1∼2. When the links between levels 3∼4 become congested, new links between levels 3∼4 will be opened to relieve the congestion. This is the reason that we see the sawtooth pattern occurring first on links between levels 3∼4. It is also intuitively clear that the peaks and valleys that appear in the delay curve of Fig. 4 and the level 3∼4 utilization curve of Fig. 5 always occur at the same value of the traffic load. This is because high link utilization directly causes high packet delay. Since we use a relatively large safety margin value of 0.1 in this scenario, packet loss is generally rare, as shown in Fig. 4.

Impact of Safety Magin. We have also experimented with lower θ values set to 0.05 and 0.1 and observe a significant increase in packet loss rate while the increase in average packet delay is not as substantial.

C. The Impact of Safety Margin on Power Consumption

In Fig. 6, we show the power consumption of the fat-tree network FT$(8, 4)$ when different safety margins are used. We observe that the impact of the safety margin on power consumption is generally minor.

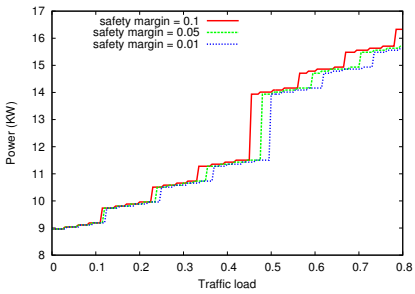

Fig. 6. Power consumptions of fat-tree network FT$(8, 4)$ under different safety margins.

Based on the simulation results above, it is wise to select a moderate safety margin (e.g., $\theta = 0.1$) in the fat-tree network to achieve a good balance between packet delay, packet-loss rate and power consumption.

Impact of Switch Size on Performance. From our simulation study, we observe that large switches with fewer hops tend to incur lower packet delay and loss rate.

VI. Conclusion

In this paper, we presented a methodology at a design stage of deploying a datacenter network. Given the number of servers that need to be supported in a data center and the network topology, our approach determines the right switch size that minimizes the energy consumption of the network during the expected operation of the data center. Our analytical and simulation results reveal that a fat-tree network using sleep mode with a large number of small switches would generally consume less power than the counterpart with a small number of large switches when the traffic demand is low or localized. This observation is generally not true if the network uses the speed-scaling mode for power saving.

References

[1] L. Barroso and U. Holzle, "The case for energy-proportional computing," *IEEE Computer*, vol. 40, pp. 33–37, Dec 2007.

[2] A. Gandh, M. Harchol-Balter, R. Das, and C. Lefurgy, "Optimal power allocation in server farms," in *SIGMETRICS*, pp. 157–168, 2007.

[3] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banarjee, and N. McKeown, "Elastictree: Saving energy in data center networks," in *NSDI*, 2010.

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

[5] D. Abts, M. Marty, P. Wells, P. Klausler, and H. Liu, "Energy proportional datacenter networks," in *ISCA*, pp. 338–347, 2010.

[6] P. Mahadevan, P. Sharma, S. Banerjee, , and P. Ranganathan, "A power benchmarking framework for network devices," in *In Proceedings of IFIP Networking*, 2009.

[7] C. E. Leiserson, "Fat-trees: Universal networks for hardware-efficient computing," *IEEE Trans. on Computers*, vol. 34, pp. 892–901, Oct 1985.

[8] X.-Y. Lin, Y.-C. Chung, and T.-Y. Huang, "A multiple LID routing scheme for Fat-Tree-Based ifiniband networks," in *IPDPS*, 2004.

[9] X. Yuan, wickus Nienaber, Z. Duan, and R. Melhem, "Oblivious routing in fat-tree based system area networks with uncertain traffic demands," *IEEE Trans. on Networking*, vol. 17, pp. 1439–1452, Oct 2009.

[10] V. Eramo1, A. Germoni, A. Cianfrani, E. Miucci, and M. Listanti, "Comparison in power consumption of mvmc and benes optical packet switches," in *IEEE NOC*, pp. 125–128, 2011.

[11] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *NSDI*, pp. 323–336, 2008.