Switch Sizing for Energy-Efficient Datacenter Networks

Indra Widjaja, Anwar Walid
Bell Labs, Alcatel-Lucent

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

ABSTRACT

Saving power in datacenter networks has become a pressing issue. ElasticTree and CARPO fat-tree networks have recently been proposed to reduce power consumption by using sleep mode during the operation stage of the network. In this paper, we address the design stage where the right switch size is evaluated to maximize power saving during the expected operation of the network. Our findings 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. We also discuss the impact of sleep mode on performance such as packet delay and loss.

1. 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) (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 [2], [3].

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 [2] and [3] 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 topology designs and 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 communication 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 [4] 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-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 moderate.

2. FAT-TREE NETWORKS

We focus on fat-tree networks as they are widely deployed in data centers. Our methodology, however, is also applicable to other networks. 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 [5]. Prior work on energy-efficient fat-tree datacenter networks only considers a configuration with 3 levels of switches [2][3]. In this section, we describe a fat-tree network with arbitrary number of levels. This is needed so that we can 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 adopt some notations in [6]. A fat-tree, FT(k, n), consists of of n levels of k-port switches, where k is a multiple of 2. The level-n switches are called top-level switches and the level-1 switches are called bottom-level switches that connect to the servers. A fat-tree FT(k, n) can be constructed with k sub-fat-trees SFT$(k, n-1)$'s by connecting each of the top-level (level-n) switches of FT(k, n) with k sub-fat-trees SFT$(k, n-1)$'s. In general, SFT(k, l), $1 < l < n$, is also 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.

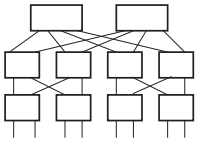

Figure 1: Example of FT$(4, 3)$.

Fig. 1 shows an example of FT$(4, 3)$. It can be shown that 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.

3. POWER CONSUMPTION

We assume that the power consumption of a datacenter network depends on the switch power and link power. We use the term sleep mode to indicate that a given network component can be turned off when there is no traffic through it.

3.1 Model

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) is C_{ij}. Usually, $C_{ij} = C_{ji}, \forall i, j$. Let $X_{ij}, \forall (i, j) \in L$, 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 for switch $i \in N$. We assume that the power for a switch includes all components (chassis, switching fabric, line cards, etc.), except ports (links). We let $P_{i,j}^l$ denote the power to turn on link (i, j), which may also be associated to a given port at switch i.

For a given traffic demand matrix, the optimization problem that minimizes power consumption can be formulated 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

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

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

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

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

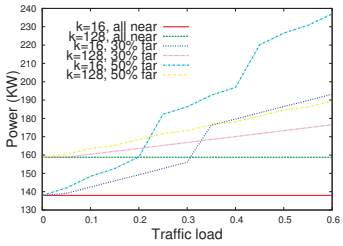

Figure 2: Power consumption with sleep mode under near-far traffic.

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

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

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 formulation has been implemented in AMPL and solved by CPLEX.

3.2 Evaluation

As described above, power is consumed by two groups of components: chassis and ports. The power consumption of a multi-stage switching fabric included in a chassis scales according to $O(k \log k)$, while a crossbar-type switching fabric scales according to $O(k^2)$ [7], 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 size k_{min} ports. Then, the chassis power of a k-port switch, when the switch is turned on, can be expressed as

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

and $P^{s(k)} = 0$ if the switch is turned off. 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. It consumes 2W when it is turned on (active) and zero power when it is turned off. We note that the general qualitative results did not change when we experimented with other combinations of the above parameter values.

For the traffic pattern, we decided to adopt the one described in previous work ([2][3]) to maintain consistent qualitative results. We consider the case where each server i, for $i = 0, \cdots, N_h - 1$, sends traffic to a fixed server j at a rate given by

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

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.

4. SIMULATION EVALUATION

It is obvious that sleep mode will impact the end-to-end QoS such as packet delay and loss. Since accurate analytical models are complex, we discuss the resulting impact through simulation.

4.1 Simulation Setting

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 [2]. 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 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 packet-mode 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 assign each flow two states: ON and OFF. When ON, the duration of a flow 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.

4.2 Packet Delay and Loss Rate

In order to reduce simulation time, we evaluate packet delay and loss rates on fat-tree networks with 512 servers using sleep mode. Our main objective is to study the impact of power saving on performance. Fig. 3 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. 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 shows the average link utilization values at different levels of the fat-tree network. These curves also confirm 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

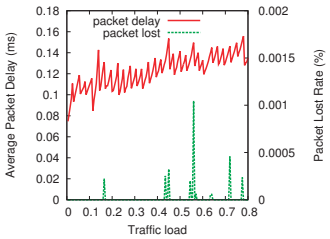

Figure 3: Packet delay and loss rate of fat-tree network.

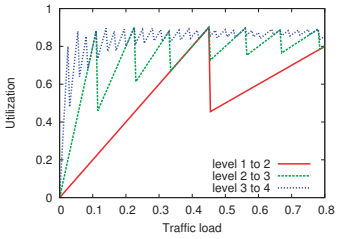

Figure 4: Average utilizations of links at different levels.

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. 3 and the level 3∼4 utilization curve of Fig. 4 always occur at the same value of the traffic load.

5. CONCLUSION

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 results reveal new tradeoffs between small and large switches in terms of power consumption. Our approach can also be extended to the case where the switches operate in speed-scaling mode.

6. REFERENCES

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

[2] 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.

[3] 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.

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

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

[6] 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.

[7] 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.