On the Network Power Effectiveness of Data Center Architectures

Yunfei Shang, Dan Li, *Member, IEEE*, Jing Zhu, and Mingwei Xu, *Member, IEEE*

Abstract—Cloud computing not only requires high-capacity data center networks to accelerate bandwidth-hungry computations, but also causes considerable power expenses to cloud providers. In recent years many advanced data center network architectures have been proposed to increase the network throughput, such as Fat-Tree [1] and BCube [2], but little attention has been paid to the power efficiency of these network architectures. This paper makes the first comprehensive comparison study for typical data center networks with regard to their *Network Power Effectiveness(NPE*), which indicates the end-to-end *bps per watt* in data transmission and reflects the tradeoff between power consumption and network throughput. We take switches, server NICs and server CPU cores into account when evaluating the network power consumption. We measure NPE under both regular routing and power-aware routing, and investigate the impacts of topology size, traffic load, throughput threshold in power-aware routing, network power parameter as well as traffic pattern. The results show that in most cases Flattened Butterfly possesses the highest NPE among the architectures under study, and server-centric architectures usually have higher NPEs than Fat-Tree and VL2 architectures. In addition, the sleep-on-idle technique and power-aware routing can significantly improve the NPEs for all the data center architectures, especially when the traffic load is low. We believe that the results are useful for cloud providers, when they design/upgrade data center networks or employ network power management.

Index Terms—Network power effectiveness, data center network, switch-centric architecture, server-centric architecture, power-aware routing

1 Introduction

Cloud computing realizes the dream of "computing as a utility". It employs the Infrastructure as a Service (IaaS) model, in which customers outsource their computing and software capabilities to third-party infrastructures and pay for the service usage on demand. Compared with the traditional computing model that uses dedicated, in-house infrastructure, cloud computing has many advantages, including economies of scale, dynamic provisioning, and low capital expenditures. It allows customers to establish and rapidly expand a global presence in minutes rather than days or months, with the "pay-as-you-go" charging model. The success of cloud computing is exemplified by a growing number of companies that provide cloud services, such as Amazon's Elastic Compute Cloud (EC2) [3], Google's Google App Engine [4], Microsoft's Azure Service Platform [5], Rackspace's Mosso [6], and GoGrid [7].

However, the giant data centers providing cloud services are both bandwidth hungry and power hungry. For bandwidth, typical applications in cloud data centers, such as MapReduce [8] and GFS [9], generate high volumes of "west-eastern" traffic, which the traditional tree based network architecture cannot well support. For power, the huge amount of data centers' power consumption not only brings considerable economic costs to cloud providers, but also affects the sustainable growth of cloud computing, due to the challenges imposed by power delivery to and heat removal from giant data centers. The power usage of the network part is becoming increasingly remarkable in data centers, since the power management on servers is relatively mature. It has been shown that, with proper power control on servers, the proportion of network power consumption can reach up to 50 percent of the whole data center, if the data center system is lowly utilized [10].

In order to increase the network capacity of data centers, many advanced data center network architectures with high link redundancy have been proposed recently, such as Fat-Tree [1], VL2 [11], Flattened Butterfly (or FBFLY, for short) [12], BCube [2], DCell [13] and FiConn [14]. But the introduction of more and more switches/links in these network architectures even aggravates the power consumption of data center networks. Therefore, it is of high importance to understand the data center network power efficiency, which is translated to the economic investment of cloud providers.

With the industrial support of energy-efficient Ethernet (EEE) switches [15], smart power management can be expected for data center networks, by putting interfaces or even the entire switches into sleep when there is no traffic at all. We can also design power-aware routing (PAR) schemes to prolong the sleeping intervals of switches/ports for maximum power saving. Therefore, we are interested in not only the gross power usage of a data center network when all the

- *Y. Shang is with the Department of Computer Science and Technology, Tsinghua University, and the Naval Academy of Armament, Beijing, China. E-mail: shangyunfei2008@163.com.*
- *D. Li, J. Zhu, and M. Xu are with the Department of Computer Science and Technology, and the Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing, China. E-mail: {tolidan, xmw}@tsinghua.edu.cn, zjinn@yahoo.cn.*

Manuscript received 9 Apr. 2014; revised 10 Dec. 2014; accepted 29 Dec. 2014. Date of publication 15 Jan. 2015; date of current version 9 Oct. 2015. Recommended for acceptance by Y. Yang.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TC.2015.2389808

switches/ports are on, but also the power efficiency when the sleep-on-idle (SoI) technique and power-aware routing are implemented.

In this work, we focus on a specific metric of power efficiency for data center network architectures, i.e., *Network Power Effectiveness(NPE)*, which is calculated as the aggregate network throughput divided by the total network power consumption. NPE indicates the end-to-end *bps per watt* (or equally, *bit per joule*) in data transmission. Here we are interested in the *end-to-end* aggregate network throughput rather than the transportation capacity of individual switches, as the focus here is the performance of the network when it shuffles data for the upper-layer applications. Hence, NPE reflects the tradeoff between power consumption and network throughput in data centers. The higher NPE is, the more power efficient the data center architecture can be. For cloud providers, NPE is an important factor when considering data center network architecture design or upgrade, in addition to the other factors such as network diameter, aggregate throughput, bisection bandwidth and equipment cost.

We for the first time make a comprehensive comparison study on the NPEs of the recently proposed data center network architectures. We choose three switch-centric architectures, namely, Fat-Tree [1], VL2 [11] and Flattened Butterfly [12], and three server-centric architectures, namely, BCube [2], DCell [13] and FiConn [14]. In order to make a fair comparison, we choose topologies with similar network diameters and consider a wide range of server numbers they support. We take switches, server NICs and server CPU cores into consideration when calculating the network power consumption. We then measure the NPEs for both regular routing (RR) and power-aware routing, and investigate the impacts of topology size, traffic load, throughput threshold in power-aware routing, network power parameter as well as traffic pattern.

We obtain the following findings from the study. First, for the network power consumption of data centers, FBFLY, DCell and FiConn consume lower gross network power than the other three architectures when all switches are on. Besides, the power consumption of server CPU cores used for packet processing and forwarding is non-trivial in server-centric architectures. Second, the sleep-on-idle technique and power-aware routing can effectively help improve the NPEs of all the data center architectures under study, especially when the traffic load is low. Third, the topology size and the throughput threshold in power-aware routing have no obvious impact on the NPEs of data center networks. Finally, in most cases FBFLY possesses the highest NPE among the six architectures under study, and server-centric architectures have better NPEs than Fat-Tree and VL2, though a little worse than FBFLY. The parameter setting in the network power model has no obvious impact on this observation.

The remainder of this paper is organized as follows. Section 2 introduces background knowledge and related work. Section 3 discusses the methodology we use in the comparison study. Section 4 presents the comparison results. Finally, Section 5 concludes the paper.

2 Background and Related Work

In this section, we introduce the background and related works on network architecture, power model, as well as power saving techniques in data centers.

2.1 Data Center Network Architecture

In traditional data centers, servers are connected by a tree hierarchy of switches, with low-end ones at the lowest level and increasingly larger and more expensive ones at higher levels. This kind of tree architecture cannot satisfy the high bandwidth requirement of cloud computing, and recently many novel data center network architectures have been proposed to increase network capacity. These architectures can be divided into two categories, namely, switch-centric and server-centric. The former puts the interconnection and routing intelligence on switches, while the latter introduces data center servers into the networking part and applies them to packet forwarding.

Switch-centric architectures. The switch-centric architecture proposals either use a totally new switch infrastructure to replace the tree structure, or make enhancements upon the tree to improve the bisection bandwidth. Fat-Tree [1] and VL2 [11] both use low-end, commodity switches to form a three-layer Clos network. Each server still uses one NIC port to connect an edge-level switch. Both the two architectures can provide an oversubscription ratio of 1:1 for all the servers in the network. Compared with Fat-Tree, VL2 use higher-speed switches, e.g., those with 10 GE ports, in higher levels of the Clos network to reduce wiring complexity. Flattened Butterfly [10], [12] is usually used for high-radix networks. It is a multi-dimensional architecture and the servers in a same dimension are fully connected. FBFLY can provide more abundant paths than a traditional butterfly [10] and enjoy more cost-efficient than Clos networks. Similarly, Dragonfly [16], [17] also uses high-radix routers and takes a group of them as a virtual router. Dragonfly can increase the effective network radix and enjoy a low network cost and high scalability.

Instead of completely giving up the traditional tree architecture, the tree enhancement approaches introduce additional shortcut links into the tree architecture, either with 60 Gbps wireless links [18], or with reconfigurable optical circuits [19]. The reason behind is that the richly-connected data center network architectures, such as Fat-Tree and VL2, aggressively provide excessive network devices and links for full bisection bandwidth. However, it is usually unnecessary because the traffic load in mega data centers is far below the peak value in most of the time. Therefore, the tree enhancement architectures have better flexibility and a lower cost to transmit traffic flows.

Since FBFLY, Flyways and c-Through need to employ optical or wireless switching devices, which have different network power models from electronic switching devices, we will leave the comparison study of those architectures to future work.

Server-centric architectures. In server-centric data center architectures, each server uses multiple NIC ports to join the network infrastructure and participate in packet forwarding. In DCell [13], a recursive, level-based structure is designed to connect servers via mini-switches and multiple

server NIC ports. DCell is highlighted by its excellent scalability, i.e., the number of servers supported increases double-exponentially with the number of server NIC ports. FiConn [14] goes one step further to limit the number of server NIC ports as two, since most of current data center servers have two built-in ports. Compared with DCell, FiConn not only eliminates the necessity to add server NIC ports during data center expansion, but also reduces the wiring cost. The downside is that the network capacity in FiConn is lower than DCell.

BCube [2] targets at building a data center container, typically with 1k∼4k servers. BCube is also a recursive structure. Each server uses multiple ports to connect different levels of switches. The link resource in BCube is so rich that 1:1 oversubscription ratio is guaranteed. MDCube [20] designs an architecture to interconnect the BCube-based containers. The inter-container connection and routing in MDCube are closely coupled with the intra-container architecture, so as to provide high bisection width and great fault tolerance.

2.2 Power Model for Data Center Networks

Switches and server NICs are the two main parts for network power consumption in switch-centric data center architectures. server CPU cores also contribute to packet processing and forwarding, and accordingly influence network power consumption in server-centric data center architectures. Therefore, we should consider the server CPU power consumed for packet forwarding when calculating the total network power consumption in server-centric architectures.

Eq. (1) shows the total amount of network power consumption in both a switch-centric architecture and a server-centric architecture. $\mathcal{I}$ and $\mathcal{J}$ denote the sets of switches and server NIC ports respectively in the data center, and $\mathcal{L}$ denotes the set of server CPU cores used for packet processing and forwarding in server-centric data centers. $U_i(i \in \mathcal{I})$ and $V_j(j \in \mathcal{J})$ denote the power consumption of a switch i and that of a server NIC port j respectively. E_l and $Y_l(l \in \mathcal{L})$ denote the power of a server CPU core used for network processing and forwarding at maximum utilization and the utilization ratio of the CPU core, respectively. Here, we simply assume that a CPU core in a server is energy proportional to its utilization [21], [22]. In Eq. (1), we do not consider the power consumption of cables, since they are shown to occupy only a very small portion of the total power in the network [23] although with a non-ignorable deployment expenditure

$$P = \begin{cases} \sum_{i\in\mathcal{I}} U_i + \sum_{j\in\mathcal{J}} V_j, & \text{switch-centric} \\ \sum_{i\in\mathcal{I}} U_i + \sum_{j\in\mathcal{J}} V_j + \sum_{l\in\mathcal{L}} (E_l * Y_l), & \text{server-centric.} \end{cases} \tag{1}$$

The IEEE Energy Efficient Ethernet standard suggests three energy states for Ethernet ports on switches [15], i.e., Active, Normal Idle (N_IDLE), and Low-Power Idle (LP_IDLE). The Active state consumes high power when sending packets. The N_IDLE state does not transmit packets but consumes same or less power than the Active state. The LP_IDLE state consumes almost no power by putting the ports into deep sleep. Rate adaptation, i.e., changing the operating rate of a port according to the traffic load, is not recommended by EEE, since its saving is moderate compared to putting the port into sleep. Therefore, we assume that an Ethernet port can go to sleep when it is idle for a short period of time, i.e., sleep-on-idle, and a sleeping port can be waken up when a packet arrives, i.e., wake-on-arrival (WoA). SoI and WoA only take tens of microseconds to transition, which is tolerable for almost all kinds of applications [15]. These technologies are already implemented in Cisco switches [24]. When all the ports in a switch are asleep, we can also put the entire switch into sleep, so as to save the power consumption on the switching fabric, fans, and other parts, which are relatively fixed. Therefore, we use Eq. (2) to calculate the power consumption for a switch i, where C_i denotes the total number of ports in the switch, M_i denotes the number of sleeping ports, Q_i denotes the power consumption of an active port, and T_i denotes the fixed amount of power consumption in the switch. Here, we assume all active ports on a switch have the same power consumption, as current advanced data center topologies, such as Fat-Tree [1] and BCube [2], usually employ commodity switches equipped with homogeneous network ports to interconnect servers

$$U_i = \begin{cases} 0 & \text{if } M_i = C_i \\ T_i + (C_i - M_i) * Q_i & \text{if } M_i < C_i. \end{cases} \tag{2}$$

2.3 Power Saving Techniques in Data Center Networks

Recently there are many research works studying power-saving mechanisms for data center networks. They can be generally divided into two categories: developing energy-efficient network devices and designing power-aware routing. The consistent theme is to let the power usage of data center networks be proportional to their traffic load. When we compare the power characteristics of six typical data center architectures later, we will apply both the sleeping technique of network devices and power-aware routing into them, and study the impacts of the two techniques on improving the network power effectiveness of these architectures.

Energy efficient network devices. Many novel energy-saving techniques have been proposed in order to improve the energy usage efficiency of individual network devices these years. The network component sleeping and dynamical adaptation technologies are two main methods. Nedevschi et al. [25] argued that putting idle network elements into a sleep mode and dynamically adapting the rate of network ports to their forwarding loads can effectively save the power consumption of network devices. Later, they studied two implementation mechanisms of network component sleeping: wake-on-lan and assistant proxy processing, and proposed an effective proxy framework in [26]. Furthermore, Gupta and Singh [27] proposed a method of detecting idle and under-utilized links to save energy with little sacrifice on network delay and packet loss. Gunaratne et al. [28] investigated the optimal strategies of tuning the transmission rate of links in response to the link utilization and buffer queue lengths of switch ports. Ananthanarayanan and Katz [29] designed a shadow port as the packet

TABLE 1
Number of Servers and Network Diameters in the Five Architectures

Data center architecture	Number of servers	Network diameter
Fat-Tree(n)	$N = n^3/4$	6
VL2(n)	$N = 5n^2$	6
FBFLY(n, k)	$N = n^k$	$k + 2$
BCube(n, k)	$N = n^{k+1}$	$2 * (k + 1)$
DCell(n, k)	$(n + \frac{1}{2})^{2^k} - \frac{1}{2} < N < (n + 1)^{2^k} - 1, k > 0$	$3 * 2^k - 1$
FiConn(n, k)	$N \geq 2^{k+2} * (n/4)^{2^k}, n > 4$	$3 * 2^k - 1$

processing proxy of sleeping ports and used a time window prediction scheme to effectively reduce the power consumption of switches.

Power-aware routing. The basic idea of proposed power-aware routing schemes currently is to aggregate network flows into a subset of network topology, so as to put more idle network devices and components into the sleep mode for energy saving [30]. Vasic et al. proposed a combination method of energy-critical path pre-computation and on-line traffic engineering to make a better tradeoff between network energy conservation and routing calculation complexity [31]. Chabarek et al. [32] used the power consumption model of routers and the optimization theory to investigate energy-saving network and protocol designs. Avallone and Ventre [33] designed a new power-aware online routing scheme, which used the fewest number of network devices and links to provide routing services while meeting multiple performance constraints of network flows. Abts et al. designed low-power, highly-scalable data center topologies, and exploited the link's dynamic range for energy saving [10]. Moreover, ElasticTree [34] targeted at saving network energy consumed by computation-constrained flows in data centers, and dynamically used a limited number of core-level switches in Fat-Tree network to accommodate the network traffic rates. Complementarily, Shang et al. focused on energy saving of bandwidth-constrained flows, and proposed a power-aware flow scheduling algorithm in data centers [35].

2.4 Comparison of Data Center Network Architectures

With regard to the comparison of data center network architectures, the work in [21] may be the closest to our work. Both their work and ours use a fair comparison method to equalize the topology setting in different data center architectures, but they focus on diverse metrics. Specifically, the work in [21] was concerned with a network deployment expenditure, which presented a one-time capital investment by operators when building data center networks. In contrast, our work focuses on comparing the network energy usage efficiency, which poses an impact on the daily operational costs of DCNs. Both network deployment costs and energy efficiency are the key metrics for operators when building and operating DCNs.

Gyarmati and Trinh [36] studied the topology characteristics of different data center networks as well as their energy consumption profiles. They analyzed the relationship between the topology parameters and the power consumption of data centers, and provided some useful topology design suggestions. In [36], the power consumption of data center architectures contains both server power and network device power, and is calculated without using any power-saving technique. On the contrary, we consider switches, server NICs and server CPU cores when evaluating network power consumption. We use a specific metric, i.e., NPE, to compare different data center network architectures, and study the impact of two typical power-saving techniques on improving NPEs in different data center architectures.

In a prior work, we made a comparison study on the power proportionality of different data center network architectures [37], which indicated the power variation behaviors of data center networks with the traffic load changing. The comparison results in [37] showed how much network power could be conserved by sleep-on-idle and power-aware routing techniques in different architectures. In this paper, we propose a new metric, i.e., NPE, which is calculated as the aggregate network throughput divided by the total network power consumption. NPE reflects the tradeoff between power consumption and network throughput, and can be taken as an effective indicator to evaluate the power usage efficiency of data center network architectures.

3 Methodology

Cloud providers concern both the power consumption and network throughput of data center network architectures. As aforementioned, we use NPE as one metric of power efficiency in this paper, which indicates the *bps per watt* (or equally, *bit per joule*) for data transmission. NPE is calculated as the aggregate network throughput of a network architecture divided by the total network power consumption.

We choose Fat-Tree, VL2 and FBFLY as the representative of switch-centric architectures, while BCube, DCell and FiConn as the representative of server-centric ones. In the following sections, we present the topology setting and routing scheme used in the comparison study, respectively.

3.1 Topology Setting

Table 1 shows how the number of servers and network diameter of a data center architecture vary with the topology parameters, where n is the number of ports per switch and k is the number of recursive levels. We can find that for Fat-Tree and VL2, the topology size is only determined by n; while for FBFLY, BCube, DCell and FiConn, the topology size depends on both n and k. In order to make a fair comparison, we try to set the network diameters of the six architectures similar, which indicates similar end-to-end delays for applications running on these data center architectures. Note that the network diameters of both Fat-Tree and VL2 are 6. Hence, we set $k = 2$ in BCube, which also gets a network diameter of 6. In both DCell and FiConn, we set $k = 1$, which results in a network diameter of 5. In FBFLY, we set $k = 3$ to get a network diameter of 5. Therefore, in the following comparison study, we fix the parameter k for FBFLY, BCube,

DCell and FiConn, and only vary the parameter n for all the six networks to investigate the impact of topology size.

We use switches with 1 GE ports in Fat-Tree, BCube, FBFLY, DCell and FiConn. For VL2, we choose switches with 10 GE ports as the intermediate and aggregate switches, and switches with 1 GE ports as the TOR switches. All the servers in the six architectures use 1 GE NIC ports.

3.2 Routing Algorithm

We use two types of routing algorithms in the study, namely, regular routing and power-aware routing.

Regular routing. RR uses the default routing strategies in the six data center architectures. The basic idea is to make sufficient usage of the multi-path characteristic of the "richly-connected" network topologies, which can alleviate the network bottleneck and achieve as high network throughput as possible.

Power-aware routing. In PAR, we make the tradeoff between network throughput and power consumption. We follow the power-aware routing algorithm proposed in [35]. The basic idea is to use the minimum number of switches and links in routing to meet a predefined network throughput requirement. Algorithm 1 shows the computation process of PAR.

Algorithm 1. Power-Aware Routing: PAR(G, F, W)

Input:
G: data center topology;
F: set of traffic flows;
W: throughput threshold.
Output:
R: set of power-aware paths for all flows in F.

```
1  R ← RR(G, F);
2  H_0 ← ComputeThroughput(G, R, F);
3  do
4     G' ← G, R' ← R;
5     G ← RemoveSwitch(G, R, F);
6     R ← RR(G, F);
7     H ← ComputeThroughput(G, R, F);
8     A ← H/H_0;
9  while(A ≥ W);
10 do
11    R ← R'
12    G' ← RemoveLink(G', R', F);
13    R' ← RR(G', F);
14    H ← ComputeThroughput(G', R', F);
15    A ← H/H_0;
16 while (A ≥ W);
17 return R;
```

In Algorithm 1, we predefine *throughput threshold*, W, which means the *percentage* of the lowest network throughput the data center operator can tolerate over the maximum network throughput when all switches are on. We first use RR to calculate the routing (Line 1) and get the basic network aggregate throughput (Line 2). Then, we use RemoveSwitch, which is to turn off or put the entire switch into sleep, to remove eligible switches in turn from the topology (Line 5) until when W is violated (Lines 8-9). In the switch elimination phase, we compute the total throughput of flows traversing each active switch, and preferentially eliminate the active switch carrying the lightest traffic load. Moreover, we usually eliminate multiple switches per round to accelerate the elimination process. Note that we run the fault-tolerant routing in RR to get the network throughput (Line 6-7) when some switches are removed from the topology. After that, we further remove the links from the remaining topology (Line 12), which is to put a port into sleep, until when W is violated (Lines 15-16). In the link elimination phase, we calculate the utilization ratio of each link connected to active switches. The utilization ratio of a link is the total throughput of the flows traversing the link to its capacity. We preferentially eliminate the link with the lowest utilization ratio from the topology. Besides, in the switch and link elimination process, we should not break the connectivity of network topologies.

4 Results

In this section, we conduct simulations to compare the NPEs of the six data center architectures. We are generally interested in the following major issues for each architecture. First, what is the gross network power consumption without SoI or PAR? Second, what effect does SoI have on improving the NPEs of data center architectures? Third, when SoI is enabled, what are the impacts of topology size, traffic load and PAR (with different throughput thresholds) on NPE? Finally, what is the impact of different parameter settings in the network power model on NPE?

4.1 Simulation Setup

We simulate the topology structures and routing algorithms of the six architectures with C++ codes and run them on a desktop with 2.80 GHz Intel Duo E7400 processors and 2 GB of RAM. We use TCP traffic flows in our simulations, as they are dominant in the current data center networks. Also, we assume that the rate of traffic flows is limited by network [38], i.e., the rate of traffic flows depends on the available network bandwidth resources and the other competing flows with it on the bottleneck link. Therefore, we use a well-known max-min fairness model [39] to calculate the throughput of each flow.

4.1.1 Traffic Pattern

We use both one-to-one and all-to-all communication patterns to generate the data center traffic. For a selected set of servers, in one-to-one communication, a server randomly chooses another server in the set and only sends data to it; while in all-to-all communication, each server sends data to all the other servers in the set. The typical application in data centers to generate one-to-one traffic pattern is file backup, and that to generate all-to-all traffic pattern is MapReduce [8]. Other traffic patterns are also possible in data centers (e.g., one-to-many traffic), but generally they are between the two traffic patterns in our study. Full traffic loads indicate all servers in the network participate in communication, and p percent traffic load indicates only p percent servers are chosen randomly to send/receive traffic with a one-to-one or all-to-all traffic pattern.

To make the traffic analysis closer to real data centers, we use the results in [11], which gives the average number of concurrent flows on a server during a day. We find that each server will send or receive one flow in 1/3 of the one-day

TABLE 2
Number of Switch Ports and Server NIC Ports in Each Architecture with N Servers

Data center architecture	Number of 1G switch ports	Number of 10G switch ports	Number of NIC ports
Fat-Tree	$5N$	0	N
VL2	N	$2N/5$	N
FBFLY($k=3$)	$3N$-$2N^{\frac{2}{3}}$	0	N
BCube($k=2$)	$3N$	0	$3N$
DCell($k=1$)	N	0	$2N$
FiConn($k=1$)	N	0	$3N/2$

time, and thus compare the NPEs of the six architectures during this period in Section 4.8. In the simulations, we take the maximum number of network flows as 100 percent traffic load and change the percentage to simulate different traffic loads. The traffic matrix is generated randomly between servers.

4.1.2 Topology Size

In the following simulations, we try to use data center architectures with the similar size, except when studying their power characteristics under different topology sizes. For one-to-one traffic, we use the topologies containing ~8,000 servers. Specifically, we use Fat-Tree with $n = 32$ (8192 servers), VL2 with $n = 40$ (8,000 servers), FBFLY($n = 20, k = 3$) (8,000 servers), BCube($n = 20, k = 2$) (8,000 servers), DCell ($n = 90, k = 1$) (8,190 servers) and FiConn ($n = 126, k = 1$) (8,064 servers), where n and k denote the number of ports per switch and the number of recursive levels respectively. But for all-to-all traffic, it takes unaffordable long time to calculate the network throughput if we use a large network size. Hence we set ~100 servers in the network. Specifically, we use Fat-Tree with $n = 8$ (128 servers), VL2 with $n = 4$ (80 servers), FBFLY($n = 5, k = 3$) (125 servers), BCube ($n = 5, k = 2$) (125 servers), DCell($n = 10, k = 1$) (110 servers) and FiConn ($n = 14, k = 1$) (112 servers). Note that the topology settings may not be well matched with realistic switches, but we use them to make a fair comparison study among these topologies.

4.1.3 Power Data

For switch power, we take the Cisco Nexus Series switches [40] as an example, i.e., a 1 GE port and a 10 GE port consuming 2W and 20W, respectively. Besides, we take a Nexus 2224TP switch to analyze the power composition of a typical switch [40], and assume the power consumed by all ports of a switch occupies half of the total power of the switch in our evaluations. Here, we take the power composition percentage as an example to evaluate the NPEs of data center architectures. For the power consumption of server NIC ports and CPU cores, we choose Intel server adapter [41] as the example, i.e., a 1GE NIC port consumes 2W, and an Intel Atom series processor consumes 5W at maximum utilization [42] [43]. We will alter the parameters of the network power model and study the variance in Section 4.7.

4.2 Gross Network Power

We first calculate the gross network power consumption of each data center architecture when there is no power management technique at all, i.e., without using SoI or PAR. Also, we assume the server CPU cores used for packet forwarding are at full utilization in server-centric data center architectures, i.e., a CPU core will consume maximum power of 5W. Hence, the gross network power only depends on the topological characteristic, regardless of routing schemes or traffic loads.

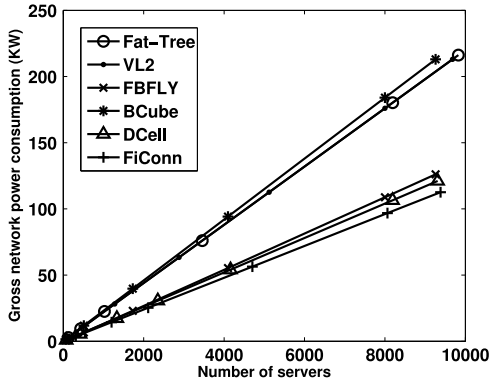


Fig. 1. Gross network power consumption of the data center architectures without using SoI or PAR.

We note an interesting feature of all the data center architectures under study, i.e., the numbers of switch ports and server NIC ports are linear with the number of servers in most of the architectures under study. We show their relationship in Table 2. Given N servers in the network, the numbers of switch ports for DCell and FiConn are both N, since each server only connects to one switch port. BCube uses more switch ports because each server uses more than one ports to connect to switches. Fat-Tree uses the most switch ports since switches are hierarchically interconnected, and VL2 uses many 10 G switch ports. The number of switch ports in FBFLY is not linear with the number of servers, because besides connecting the servers, the switches are also used to connect to every peer switch in different dimensions of FBFLY. As for the number of server NIC ports, it depends on the number of interfaces each server uses to connect to the network. Hence, server-centric architectures have much more server NIC ports than switch-centric ones.

From the numbers in Table 2, we can easily calculate the gross network power for each data center architecture, by summing up the powers of switches (including the fixed part and switch ports), server NICs and server CPU cores. The result is shown in Fig. 1. It suggests that for every data center architecture under study, the gross network power increases linearly with the number of servers. Under the same topology size, Fat-Tree, VL2 and BCube consume higher gross network power, and this is the cost paid to provide an oversubscription ratio of 1:1. FBFLY, DCell and FiConn have lower network power, with the penalty of low network capacity.

Then we turn to the composition of gross network power, to which switches, server NICs and server CPU cores contribute. We set the number of servers in each architecture as ~8,000 and check the percentage of network power consumption each part accounts for. Fig. 2 shows the result. It demonstrates that the power consumption of server CPU cores used for packet processing and forwarding cannot be neglected in server-centric architectures. The server CPU

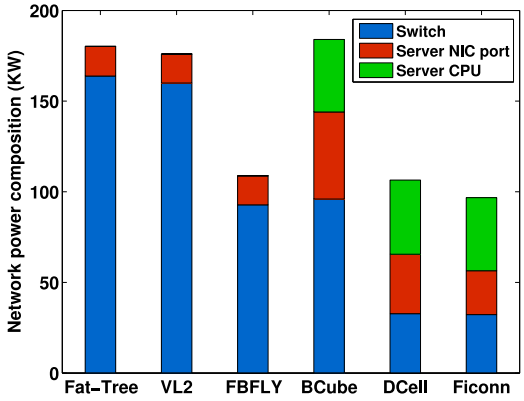
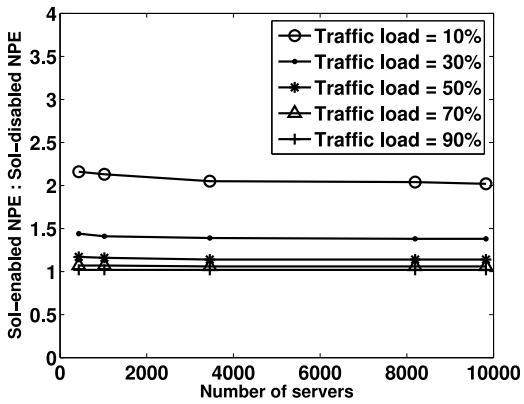


Fig. 2. Network power composition of data center architectures.

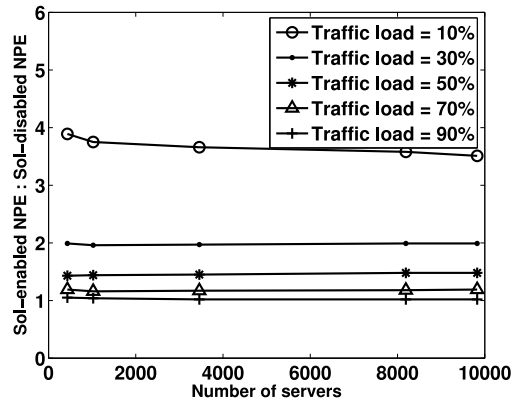
cores account for 22 percent of the total network power in BCube and about 40 percent in DCell and Ficonn. Besides, we observe that switches consume dominating network power in switch-centric architectures. The percentage even arrives at 90 percent in Fat-Tree and VL2 and 85 percent in FBFLY.

4.3 Impact of SoI

Hereafter we take *traffic and routing* into account to calculate the NPEs of the data center architectures. We first study the impact of SoI on the NPE. To differentiate, we define two kinds of NPEs. One is *SoI-disabled NPE*, which does not put idle switches/ports into sleep; while the other is *SoI-enabled NPE*, which enables sleep-on-idle. We make comparison between them by measuring the ratio of SoI-enabled NPE over SoI-disabled NPE.

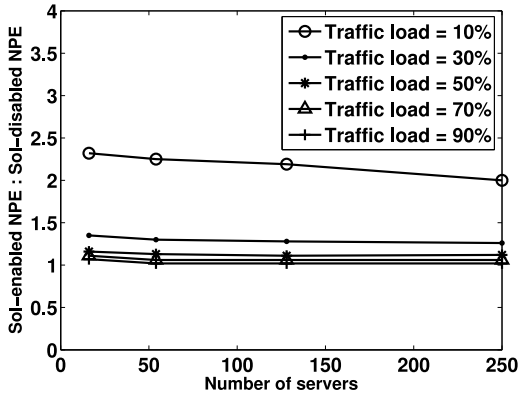


(a) RR

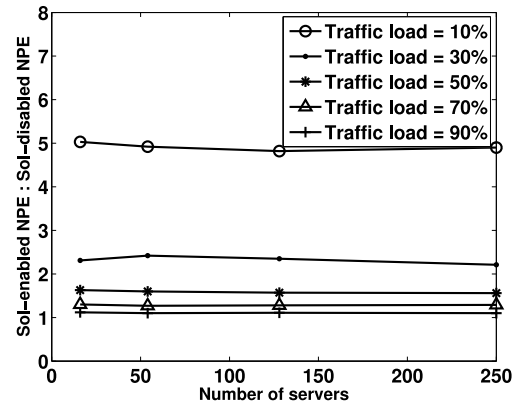


(b) PAR

Fig. 3. Ratio of SoI-enabled NPE to SoI-disabled NPE in Fat-Tree against topology sizes for one-to-one traffic.



(a) RR



(b) PAR

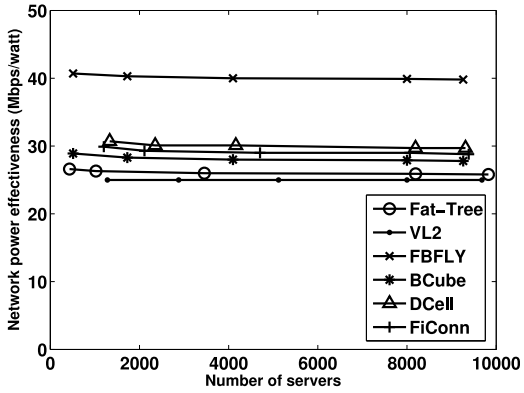
Fig. 4. Ratio of SoI-enabled NPE to SoI-disabled NPE in Fat-Tree against topology sizes for all-to-all traffic.

Figs. 3 and 4 show the results in the Fat-tree architecture for one-to-one and all-to-all traffic, respectively. We vary the traffic load from 10, 30, 50, 70, to 90 percent, and consider both RR and PAR. We change the number of servers from 432 to 9,826 for one-to-one traffic, and from 16 to 250 for all-to-all traffic (running all-to-all traffic in larger topology will cause unaffordable computation time). In PAR, we set the throughput threshold as 95 percent. From the figures, we find that the impact of SoI on the NPE is remarkable for whatever topology size, especially when the traffic load is low and PAR is used. It follows our intuition, because low traffic load and PAR will result in more idle switches in the network, which wastes power if SoI is disabled. Take Fig. 4b as an example. When using PAR for all-to-all communication in Fat-Tree and the traffic load is 10 percent, enabling SoI will bring about five times higher NPE than disabling SoI.

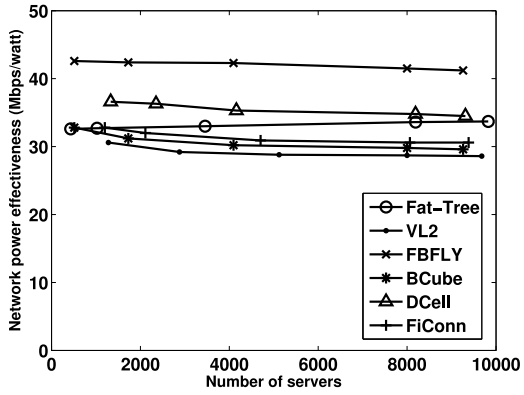
We have similar results for the other five data center architectures under study, but do not show the figures here. For all the following sections, we only consider *SoI-enabled NPE*.

4.4 Impact of Topology Size

In this section, we investigate the impact of topology size on the NPEs of the six data center architectures, by setting the

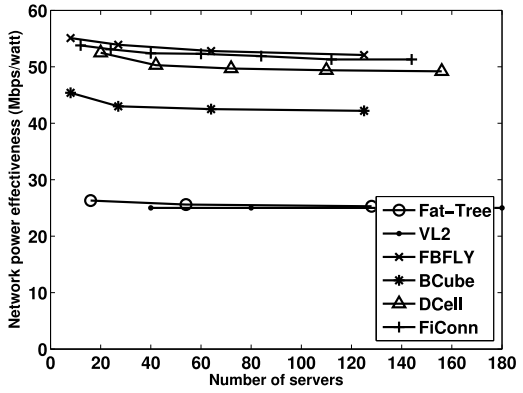


(a) RR

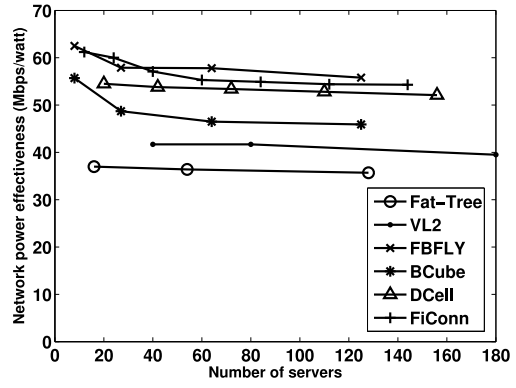


(b) PAR

Fig. 5. NPEs of the data center architectures against topology sizes for one-to-one traffic with the traffic load of 50 percent.



(a) RR



(b) PAR

Fig. 6. NPEs of the data center architectures against topology sizes for all-to-all traffic with the traffic load of 50 percent.

traffic load as 50 percent and the throughput threshold in PAR as 95 percent.

We first consider the one-to-one traffic by varying the number of servers from 400 to 10,000, the simulation results of which are shown in Fig. 5. Generally, we find that the NPE decreases slowly with the growth of topology size, but the overall impact of topology size on NPE is not obvious. This is because when fixing the traffic load percentage, both the aggregate throughput of network flows and the total network power consumption grow with the increase of topology size, but their relative ratio changes little. Furthermore, PAR always achieves higher NPE than RR. Taking the VL2 architecture containing 1,280 servers as an example, its NPE is 25 Mbps/W in RR but 31 Mbps/W in PAR.

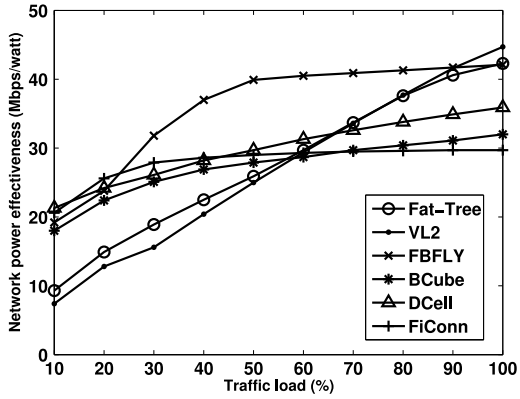
Then we turn to all-to-all traffic. In this case we vary the topology size within 200 servers. Fig. 6 shows the results under RR and PAR. The general trend of each curve in the two sub-figures indicates that the topology size has no obvious impact on the NPE of all-to-all traffic, either. For instance, in DCell, when the number of servers increases from 20 to 156, its NPE decreases by 8 and 6 percent under RR and PAR, respectively; in Fat-Tree and VL2, their NPEs almost do not change with the variance of topology size. Moreover, taking both Figs. 5 and 6 into account, FBFLY shows obviously higher NPEs than the other five architectures, especially for the one-to-one traffic.

4.5 Impact of Traffic Load

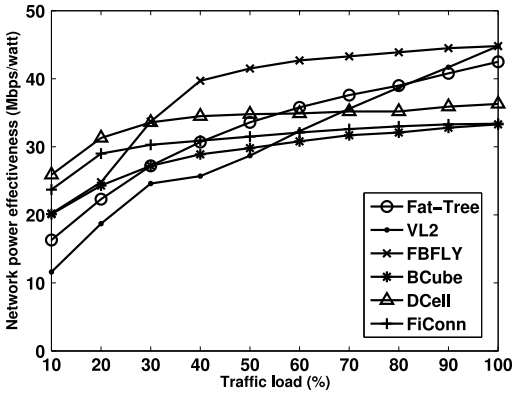
Next we study the impact of traffic load on NPE. Again, we set the throughput threshold in PAR as 95 percent.

Fig. 7 shows the results for one-to-one traffic. We observe that the traffic load significantly impacts the NPE for every architecture. For example, under RR in Fat-Tree or VL2, the NPE with 90 percent traffic load is four times more than that with 10 percent traffic load. Besides, we observe that in most cases, FBFLY with RR or PAR can achieve the highest NPE among the architectures under study in one-to-one traffic. Because FBFLY can provide abundant transmission paths for traffic flows with a low cost. As for the other five architectures, when the traffic load in one-to-one traffic is lower than 60 percent, the three server-centric architectures have obviously higher NPEs than Fat-Tree and VL2 architectures. But when the traffic load is higher than 60 percent, two switch-centric networks perform a little better, due to the high network capacity Fat-Tree and VL2 can provide. Moreover, PAR always achieves higher NPEs compared with RR, especially under low traffic load. For example, In the DCell architecture, the NPE is 21 Mbps/W in RR and 26 Mbps/W in PAR under the traffic load of 10 percent.

The simulation results for all-to-all traffic are shown in Fig. 8. Again we find that the NPE is much higher when the traffic load is high. In both RR and PAR, FBFLY usually has a higher NPE than the other architectures, especially under high traffic load. Besides, the NPEs of Fat-Tree and VL2 are obviously lower than those of server-centric ones, no matter what the traffic load is. For example, when using PAR, the

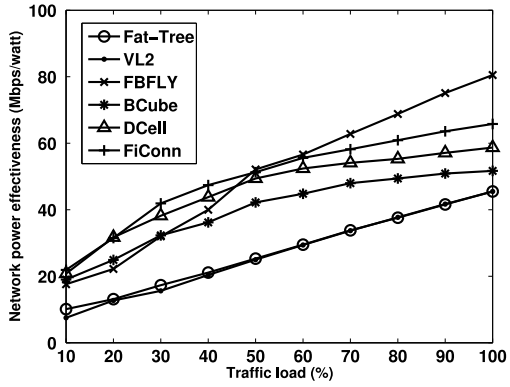


(a) RR

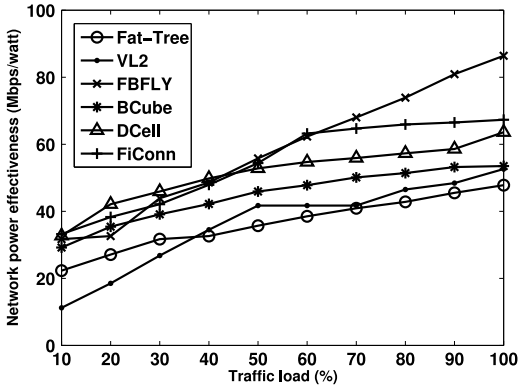


(b) PAR

Fig. 7. NPEs of the data center architectures against traffic loads under the one-to-one traffic pattern.



(a) RR



(b) PAR

Fig. 8. NPEs of the data center architectures against traffic loads under the all-to-all traffic pattern.

NPE in three server-centric architectures is three times more than that in VL2 under the 10 percent traffic load. Similar with the results in Fig. 7, PAR can significantly improve the NPE from RR.

One important reason for the significant impact of traffic load on NPE is that, data center switches have a fixed part of power consumption, which is not proportional to the traffic load. Therefore, when the number of flows is more, the data center network is more power efficient by amortizing the fixed power consumption on switches.

4.6 Impact of Throughput Threshold in PAR

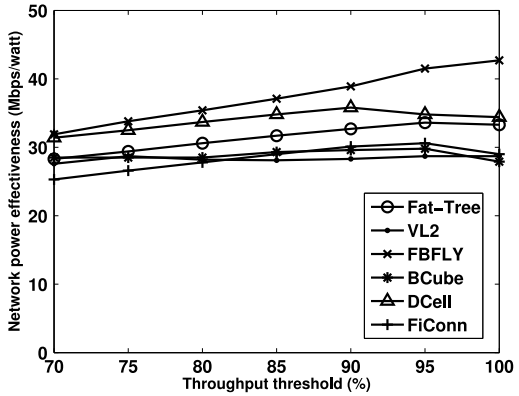
In PAR, we adjust the network throughput threshold to reflect the tradeoff between network performance and power consumption. We evaluate the impact of throughput threshold on the NPEs of data center architectures in this section. The traffic load is set as 50 percent for both one-to-one and all-to-all traffic.

Fig. 9 shows the results. It suggests that the throughput threshold has no significant impact on the NPEs of most architectures for both the traffic patterns. In most cases, the NPE of 95 percent threshold is a little higher than that of 100 percent threshold. But when we set lower thresholds, the NPE has little changes. The implication is that, in practice we can set a high throughput threshold in PAR, which achieves high network throughput without sacrificing the NPE.

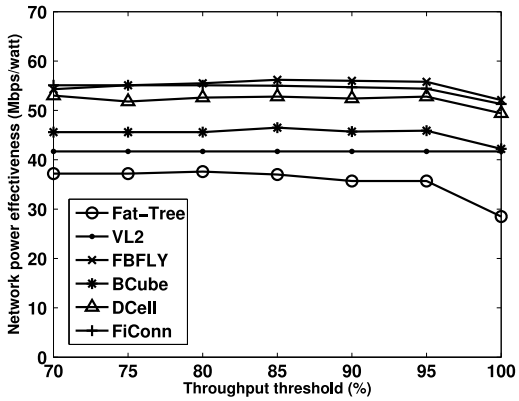
4.7 Impact of Network Power Parameter

Now we adjust parameter settings in the network power model, in particular, the power consumption ratio between a switch port and a server NIC port, and then recalculate the NPEs. It makes sense because cloud providers may purchase switches and servers from different vendors, and their power consumption also changes over time with hardware innovations. We define *APC ratio* as the ratio of the amortized power consumption (including the fixed part) of a switch port over that of a server NIC port. In all our evaluations above, the APC ratio is 2:1, since we assume 2W for a switch port and 2W for a server NIC port, but the other fixed part in a switch occupies 50 percent of the total power. In this group of simulations, we set the power consumption of a server NIC port as 2W, but vary the APC ratio from 1:1, 2:1, 3:1, 4:1 to 5:1. The traffic load is 50 percent and the throughput threshold in PAR is 95 percent.

Figs. 10 and 11 show the results for one-to-one and all-to-all traffic, respectively. An important observation is that FBFLY always enjoys the highest NPE among the six architectures under different APC ratios, which indicates that FBFLY is not only cost-effective but also a better tradeoff between network throughput and power consumption. We further compare the NPEs of the other five architectures for both traffic patterns respectively. In Fig. 10, when the APC ratio is larger than or equal to 2:1, server-centric architectures have obviously higher NPEs than Fat-Tree and VL2. But when the APC ratio is 1:1 in

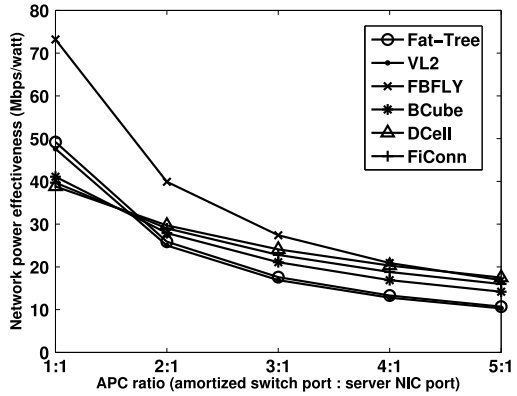


(a) One-to-one traffic pattern

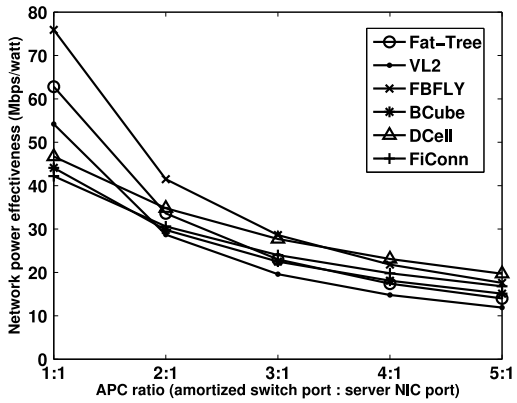


(b) All-to-all traffic pattern

Fig. 9. NPEs of the data center architectures against throughput thresholds in PAR with the traffic load of 50 percent.



(a) RR



(b) PAR

Fig. 10. NPEs of the data center architectures against APC ratios for one-to-one traffic with the traffic load of 50 percent.

one-to-one traffic, the NPEs of the two switch-centric architectures are a little higher. It is because that Fat-Tree and VL2 can achieve high network capacity with much more switch ports, and the power costs they pay are mitigated when the switch port consumes relatively less power. But for all-to-all traffic in Fig. 11, the NPEs of server-centric architectures are always higher than those of Fat-Tree and VL2. In most cases, DCell and FiConn possesses higher NPEs than BCube, Fat-Tree and VL2 architectures.

4.8 Random Traffic Pattern

In the simulations above, we compare the NPEs of the six architectures under two typical traffic patterns, i.e., one-to-one and all-to-all. In this section, we study the case of random traffic pattern. Specifically, we take the maximum number of flows traced by [11] as 100 percent traffic load and vary the flow percentage to simulate different network loads. The traffic matrix is generated randomly between servers.

We show the evaluation results with RR and PAR in Figs. 12a and 12b, respectively. Similar with the comparison results above, FBFLY usually enjoys a higher NPE under the random traffic pattern than the other five architectures, no matter using RR or PAR. The NPEs of DCell and FiConn are obviously higher than those of Fat-Tree and VL2, though a little worse than FBFLY. Again, we observe that PAR can always significantly improve the NPE from RR in each architecture.

4.9 Discussions

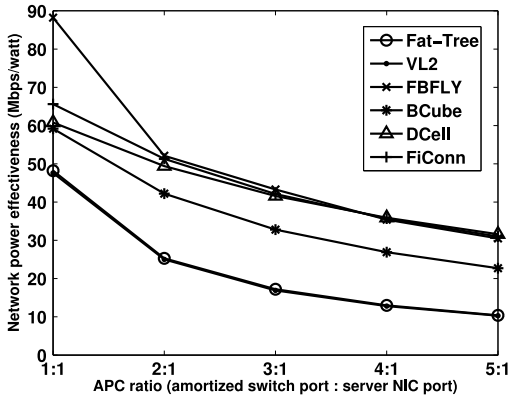
Based on the comparison studies above, we get the following findings about the NPEs of the typical data center architectures under study.

First, for network power in data centers, FBFLY, DCell and FiConn consume lower gross network power than the other three architectures when all switches are on. In server-centric architectures, the power consumption of server CPU cores used for packet processing and forwarding is non-trivial and should be taken into account in network power calculation.

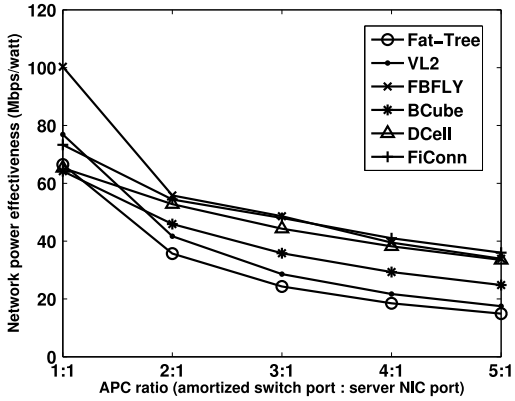
Second, both SoI and PAR can effectively improve the NPEs of all the data center architectures under study, especially when the traffic load is low.

Third, the topology size and throughput threshold in PAR have no significant impact on the NPEs of the data center architectures. It indicates that we can set a high throughput threshold in PAR, which achieves high network throughput without sacrificing the NPE.

Finally, in most cases FBFLY possesses the highest NPE among the six architectures under either RR or PAR. Besides, three server-centric architectures have better NPEs than Fat-Tree and VL2, though a little worse than FBFLY. The parameter setting in the network power model has no obvious impact on the result, except when a server NIC port consumes more power than an amortized switch port.

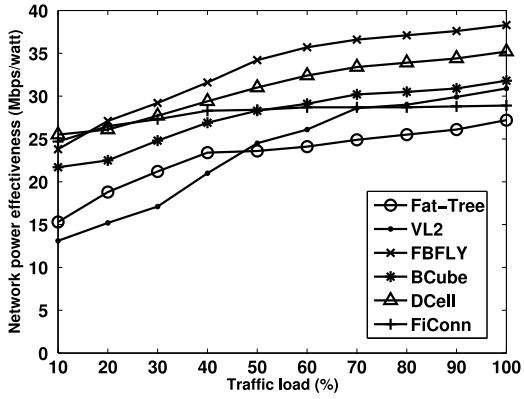


(a) RR

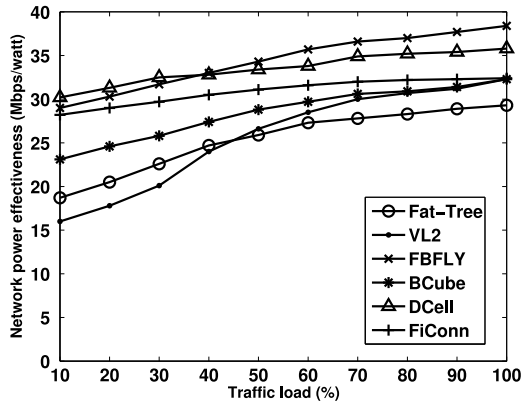


(b) PAR

Fig. 11. NPEs of the data center architectures against APC ratios for all-to-all traffic with the traffic load of 50 percent.



(a) RR



(b) PAR

Fig. 12. NPEs of the data center architectures against number of traffic loads under the random traffic pattern.

5 Conclusion

NPE is an important metric with little attention paid to in data center network design. We made a comprehensive comparison study on the NPEs of six typical data center architectures in this paper. By setting fair simulation environments, we first studied the gross network power consumption; then we checked the impacts of SoI, topology size, traffic load, PAR (with different throughput thresholds), network power parameter and traffic pattern on the NPE, for each data center architecture. The results suggest that SoI and PAR have significant impact on improving the NPE, especially when the traffic load is low. FBFLY usually enjoys the highest NPE among the architectures under study, and server-centric architectures have better NPEs than Fat-Tree and VL2 architectures. We hope that the results can be helpful for data center operators when they design/upgrade their data center architectures or employ power management techniques.

Acknowledgments

The work was supported by the National Key Basic Research Program of China (973 program) under Grant 2014CB347800, the National Natural Science Foundation of China under Grant No. 61170291, No. 61432002, No. 61133006, the National High-tech R&D Program of China (863 program) under Grant 2013AA013303, and Tsinghua University Initiative Scientific Research Program. Dan Li is the corresponding author of this paper.

References

[1] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2008, pp. 63–74.
[2] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "BCube: A high performance, server-centric network architecture for modular data centers," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2009, pp. 63–74.
[3] Amazon Elastic Compute Cloud [Online]. Available: http://aws.amazon.com/ec2/, 2008.
[4] Google App Engine [Online]. Available: https://developers.google.com/appengine/, 2011.
[5] Windows Azure [Online]. Available: www.microsoft.com/windowsazure/, 2010.
[6] Rackspace Cloud [Online]. Available: www.rackspace.com/cloud/, 2006.
[7] GoGrid [Online]. Available: www.gogrid.com/, 2008.
[8] J. Dean and S. Ghemawat,"Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.
[9] S. Ghemawat, H. Gobioff, and S. Leung, "The Google file system," in *Proc. 19th ACM Symp. Oper. Syst. Principles*, 2003, pp. 29–43.
[10] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu, "Energy proportional datacenter networks," in *Proc. 37th Annu. Int. Symp. Comput. Archit.*, 2010, pp. 338–347.
[11] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: A scalable and flexible data center network," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2009, pp. 51–62.
[12] J. Kim, W. J. Dally, and D. Abts, "Flattened butterfly: A cost-efficient topology for high-radix networks," in *Proc. 34th Annu. Int. Symp. Comput. Archit.*, 2007, pp. 126–137.
[13] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "DCell: A scalable and fault-tolerant network structure for data centers," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2008, pp. 75–86.

[14] D. Li, C. Guo, H. Wu, K. Tan, Y. Zhang, and S. Lu, "FiConn: Using backup port for server interconnection in data centers," in *Proc. IEEE Conf. Comput. Commun.*, 2009, pp. 2276–2285.
[15] IEEE P802.3az Energy Efficient Ethernet Task Force [Online]. Available: www.ieee802.org/3/az/index.html, 2010.
[16] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable dragonfly topology," in *Proc. 35th Annu. Int. Symp. Comput. Archit.*, 2008, pp. 77–88.
[17] J. Kim, W. Dally, S. Scott, and D. Abts, "Cost-efficient dragonfly topology for large-scale systems," in *Proc. Opt. Fiber Commun.*, 2009, pp. 1–3.
[18] S. Kandula, J. Padhye, and P. Bahl, "Flyways to de-congest data center networks," in *Proc. 8th ACM Workshops Hot Topics Netw.*, 2009, pp. 1–6.
[19] G. Wang, D. G. Andersen, M. Kaminsky, K. Papagiannaki, T. E. Ng, M. Kozuch, and M. Ryan, "c-Through: Part-time optics in data centers," in *Proc. ACM SIGCOMM Conf.*, 2010, pp. 327–338.
[20] H. Wu, G. Lu, D. Li, C. Guo, and Y. Zhang, "Mdcube: A high performance network structure for modular data center interconnection," in *Proc. 5th Int. Conf. Emerging Netw. Experiments Technol.*, 2009, pp. 25–36.
[21] L. Popa, S. Ratnasamy, G. Iannaccone, A. Krishnamurthy, and I. Stoica, "A cost comparison of datacenter network architectures," in *Proc. ACM 6th Int. Conf. Emerging Netw. Experiments Technol.*, 2010, p. 16.
[22] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007.
[23] J. Baliga, K. Hinton, and R. S. Tucker, "Energy consumption of the internet," in *Proc. Joint Int. Conf. Opt. Internet*, 2007, pp. 1–3.
[24] IEEE 802.3az Energy Efficient Ethernet: Build Greener Networks [Online]. Available: www.cisco.com/en/US/prod/collateral/switches/ps5718/ps4324/white_paper_ c11-676336.pdf, 2011.
[25] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," in *Proc. 5th USENIX Conf. Netw. Syst. Design Implementation*, 2008, pp. 323–336.
[26] S. Nedevschi, J. Ch, S. Ratnasamy, and N. Taft, "Skilled in the art of being idle: Reducing energy waste in networked systems," in *Proc. 6th USENIX Conf. Netw. Syst. Design Implementation*, 2009, pp. 381–394.
[27] M. Gupta and S. Singh, "Using low-power modes for energy conservation in ethernet LANs," in *Proc. IEEE Conf. Comput. Commun.*, 2007, pp. 2451–2455.
[28] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen, "Reducing the energy consumption of ethernet with adaptive link rate (ALR)," *IEEE Trans. Comput.*, vol. 57, no. 4, pp. 448–461, Apr. 2008.
[29] G. Ananthanarayanan and R. H. Katz, "Greening the switch," in *Proc. Conf. Power Aware Comput. Syst.*, 2008, p. 7.
[30] A. Bianzino, C. Chaudet, D. Rossi, and J.-L. Rougier, "A survey of green networking research," *IEEE Commun. Survey Tuts.*, vol. 14, no. 1, pp. 3–20, First Quarter 2012.
[31] N. Vasić, P. Bhurat, D. Novaković, M. Canini, S. Shekhar, and D. Kostić, "Identifying and using energy-critical paths," in *Proc. ACM 7th Conf. Emerging Netw. Experiments Technol.*, 2011, p. 18.
[32] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *Proc. IEEE Conf. Comput. Commun.*, 2008, pp. 457–465.
[33] S. Avallone and G. Ventre, "Energy efficient online routing of flows with additive constraints," *Comput. Netw.*, vol. 56, no. 10, pp. 2368–2382, 2012.
[34] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "ElasticTree: Saving energy in data center networks," in *Proc. 7th USENIX Conf. Netw. Syst. Design Implementation*, Apr. 2010, p. 17.
[35] Y. Shang, D. Li, and M. Xu, "Energy-aware routing in data center network," in *Proc. ACM SIGCOMM Workshop Green Netw.*, 2010, pp. 1–8.
[36] L. Gyarmati and T. A. Trinh, "How can architecture help to reduce energy consumption in data center networking?" in *Proc. 1st Int. Conf. Energy-Efficient Comput. Netw.*, 2010, pp. 183–186.
[37] Y. Shang, D. Li, and M. Xu, "A comparison study of energy proportionality of data center network architectures," in *Proc. 32nd Int. Conf. Distrib. Comput. Syst. Workshop Data Center Perform.*, 2012, pp. 1–7.
[38] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic flow scheduling for data center networks," in *Proc. 7th USENIX Conf. Netw. Syst. Design Implementation*, 2010, p. 19.
[39] D. Nace, N.-L. Doan, E. Gourdin, and B. Liau, "Computing optimal max-min fair resource allocation for elastic flows," *IEEE/ACM Trans. Netw.*, vol. 14, no. 6, pp. 1272–1281, Dec. 2006.
[40] Cisco Nexus data sheet [Online]. Available: http://www.cisco.com/en/US/products/ps9441/Products_Sub_Category_Home.html, 2011.
[41] Intel server adapter data sheet [Online]. Available: http://www.intel.com/content/dam/doc/product-brief/gigabit-et-et2-ef-multi-port-server-adapters-brief.pdf, 2009.
[42] Intel Atom series processor power [Online]. Available: http://en.wikipedia.org/wiki/List_of_CPU_power_dissipation_figures#Intel_Atom, 2011.
[43] Intel Atom series processor [Online]. Available: http://www.intel.co.uk/content/www/uk/en/processors/atom/atom-processor .html, 2012.

Yunfei Shang is currently working toward the PhD degree in the Computer Science Department of Tsinghua University, and is an engineer in Naval Academy of Armament, Beijing, China. His main research interest includes data center networks and green networking.

Dan Li received the PhD degree in computer science from Tsinghua University in 2007. He is an associate professor in Computer Science Department of Tsinghua University, Bejing, China. His research interest includes Future Internet architecture and data center networking. He is a member of the IEEE.

Jing Zhu is currently working toward the PhD degree in the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include Internet architecture and protocol designing, routing strategy designing, and data center network.

Mingwei Xu received the BSc and the PhD degrees in 1994 and 1998, respectively from Tsinghua University. He is a professor in the Department of Computer Science at Tsinghua University. His research interest includes future Internet architecture, Internet routing and virtual networks. He is a member of the IEEE.