A Comparative Analysis of Data Center Network Architectures

Fan Yao, Jingxin Wu, Guru Venkataramani, Suresh Subramaniam
Department of Electrical and Computer Engineering,
The George Washington University, Washington, DC, USA
{*albertyao, jingxinwu, guruv, suresh*}@*gwu.edu*

***Abstract*—Advances in data intensive computing and high performance computing facilitate rapid scaling of data center networks, resulting in a growing body of research exploring new network architectures that enhance scalability, cost effectiveness and performance. Understanding the tradeoffs between these different network architectures could not only help data center operators improve deployments, but also assist system designers to optimize applications running on top of them. In this paper, we present a comparative analysis of several well known data center network architectures using important metrics, and present our results on different network topologies. We show the tradeoffs between these topologies and present implications on practical data center implementations.**

I. Introduction

The computer industry has been actively building large scale data centers that deliver enormous computation power and storage capacity needed by data-intensive applications [1], [2]. Clusters consisting of tens of thousands of nodes have become common in recent years. As network sizes increase, lowering the cost of the overall system infrastructure and achieving higher level of performance have become first order concerns for data center operators.

Data center architectures often have different end goals that require optimization of different characteristics [3], [4]. If the workload is compute-intensive, data centers need to be equipped with powerful nodes. For communication-intensive workloads, data center networks play a critical role in delivering performance while making sure that costs are affordable.

Data center network performance can typically be characterized using well known metrics such as bandwidth, reliability, throughput, power consumption, latency and cost [1], [2], [5]. Some of these metrics are interrelated. For example, cost is dependent on a variety of factors, e.g., power consumption, data center servers, network switches, cables [6], and so on. Data center *network* infrastructure plays an increasingly important role in influencing cost and performance of the overall system, something that has been conventionally underestimated as it does not directly contribute to data center profits. Recently, researchers have come up with different proposals for building cost-effective network architectures [1], [7].

Most prior work have proposed network architectures and topologies with specific goals or have analyzed network architectures based on some single metric [1], [2], [5]. To the best of our knowledge, a holistic comparative analysis of various network architectures is absent in the literature. While cost comparison analysis is useful to analyze different data center architectures [2], we note that quantifying and comparing other dimensions such as scalability, performance and power can yield further insights.

In this work, we conduct a comparative analysis of several representative data center network architectures. We present a list of key metrics to depict performance and cost, and analyze our representative architectures in terms of these metrics. The specific contributions of our work are:

1) We comprehensively compare contemporary popular and representative data center topologies by analyzing significant metrics in data centers including scalability, latency and hop counts, path diversity, cost and power.
2) We evaluate network throughputs of different topologies under typical data center traffic patterns using mininet network simulator [8]. To the best of our knowledge, this is the first work that compares influences of various data center topologies on overall system throughput.
3) We summarize cost and performance characterized by various metrics and give recommendations for practical data center topology implementation based on different network sizes.

II. Data Center Architecture Classification

We adopt the classification from [2] to categorize data center networks into three classes: switch-based, server-based and hybrid architectures. In switch-only architectures, packet forwarding is implemented using switches, whereas server-only architectures use servers for packet forwarding. Servers in server-only architectures have dual functions: (i) to run applications, and (ii) to forward packets between servers. The third class, hybrid architectures, utilize both switches and servers for packet forwarding.

In this work, we study three switch-only data center architectures (multi-tiered network, fat tree, and flattened butterfly); one sever-only architecture (Camcube), and one hybrid architecture (BCube). We briefly review these architectures in this section.

A. Switch-only topologies

a) Multi-tiered Network: Multi-tiered design is a traditional data center architecture that is commonly used in many medium-to-large enterprises. A three-tiered topology (see Figure 1) contains core switches at the root level, aggregation switches at the middle level, and access level switches connected to the hosts. In this work, we assume that all the core level and aggregation level switches use 40 GigE ports. Each access switch uses several GigE ports connecting hosts as well as one 10 GigE uplink to an aggregation switch. A basic parameter for multi-tiered networks is the oversubscription ratio. To the best of our knowledge, there is no standard definition of oversubscription ratio; researchers tend to come up with their own definitions that may be specific to topologies. For example, in a tree-like topology, oversubscription ratio is typically defined as the ratio of bandwidth for downlinks to the bandwidth for uplinks. We adopt a more generalized definition: oversubscription is the ratio of network injection bandwidth to network capacity. Specifically, network injection bandwidth is the largest flow sizes that end hosts could inject to the network, and network capacity [9] is the maximum load on the minimum bisector under uniform random traffic. Note that the oversubscription for multi-tiered topology is highly configurable by varying the number of uplinks and downlinks for each access level and aggregation level switch [10].

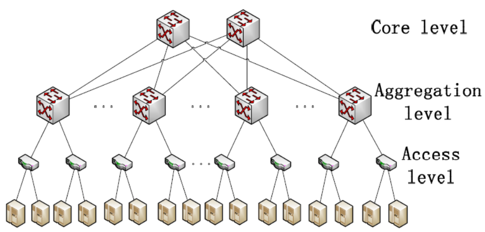

Fig. 1. A multi-tiered network.

b) Fat Tree Network: The design of fat tree network is motivated by the fact that the price differential between high-end switch (switches with higher link bandwidth or higher number of ports) and low-end switches is considerably large. The main idea behind the construction of fat tree topology is to replace the high-end switches in multi-tiered topology by interconnecting several low-end switches. The main difference in fat tree is that all of the aggregation level and core level switches are replaced with interconnections of a set of low end switches. Each subset is called a pod in Figure 2. As the number of uplinks and downlinks for each pod are equal, fat tree has full *bisection bandwidth*. Bisection bandwidth is the maximum bandwidth that can be transferred across the midpoint of the system [9]. Moreover, since all the switches in the fat tree topology are inexpensive low-end access level switches, this network topology is supposed to be highly scalable and economical.

c) Flattened Butterfly Network: Flattened butterfly topology [11] was originally proposed for on-chip interconnection network. The k-ary n-flat flattened butterfly (FBFLY) is a multi-dimensional symmetric topology which takes advantage of high-radix switches to create low-diameter networks. Here, n denotes the dimension of the topology and k is the number of switches in each dimension. Figure 3 shows an 8-ary 2-flat FBFLY. Each square in the figure represents a switch, and each of the 8 switches interconnects with the other 7 switches. In addition, each switch links with 8 host nodes (i.e., servers). A k-ary-n-flat flattened butterfly is constructed from a k-ary-$(n-1)$ flattened butterfly and a k-ary-2 flattened butterfly. For instance, an 8-ary 3-flat FBFLY can be constructed by copying the 8-ary 2-flat 8 times, then interconnecting each switch in one group with the corresponding 7 switches (one in each of the other 7 groups).

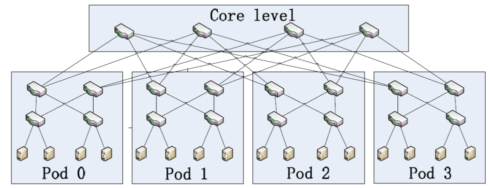

Fig. 2. Fat tree network.

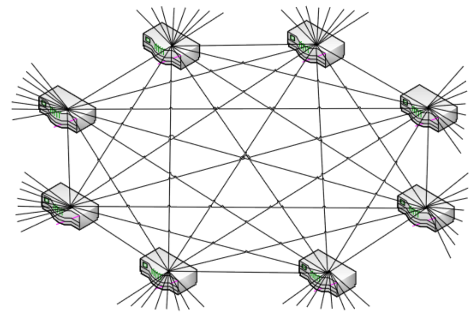

Fig. 3. Flattened butterfly.

B. Server-only topology

d) Camcube: In server-based data center architectures, the data center is created using a set of servers, where each server typically has a multi-core processor, and a high-performance network interface card (NIC) with multiple ports. The servers are not only end hosts, but also perform packet forwarding and routing. The camcube is a type of torus topology in which each server port is connected directly to another port on another server. The topology of camcube is a 3D Torus. The prototype has 27 severs $(3 \times 3 \times 3)$ [12].

C. Hybrid topology

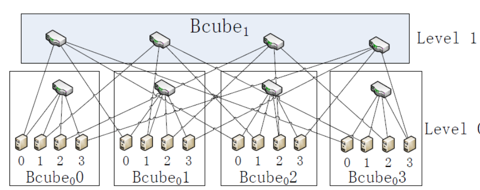

Fig. 4. BCube$_1$.

e) BCube: The BCube architecture [13] uses both switches and servers for routing traffic. BCube$_k$ is recursively defined from BCube$_0$, which contains n servers and an n-port switch, each server connecting to a switch port. There are no direct connections between any two servers. A BCube$_k$ network can be constructed using n BCube$_{k-1}$ topologies and n^k n-port switches. In BCube$_k$, there are $k+1$ levels, and $N = n^{k+1}$ servers and $k+1$ ports for each server, each port connecting a switch at each level. Figure 4 shows an example BCube architecture.

III. Comparison Metrics

In this section, we define and obtain the performance metrics for each of the topologies.

A. Scalability

Scalability is the ability of a system, to handle a growing amount of work in a capable manner or its ability to be enlarged to accommodate that growth [5]. To compare the scalability of different topologies, we need to set oversubscription of the topologies to be the same. As we will discuss later, oversubscription ratio is the major factor that influences network scalability; it is basically a metric to quantify how network bandwidth is shared among all hosts.

In this paper, we set the oversubscription ratio of each topology to 1:1 for comparison purposes. An 1:1 oversubscription indicates that all hosts have the capability to communicate with other hosts with full link bandwidth. The main impediment to scalability is switch port count. Accordingly, we evaluate scalability using two metrics: (a) how does the size of the data center network (i.e., the number of hosts the network can support) change with switch port counts?; and (b) how many switch ports are needed on average per host (i.e., normalized to the size of the data center)? In the following, we analyze the considered topologies and obtain these metrics for them.

(a) For multi-tiered network, assume there are e core switches and f aggregation switches. From the architecture of this network, we can see that each core switch has f ports, and each aggregation switch has e uplink ports.

To achieve 1:1 overall oversubscription, both the aggregation level and access oversubscriptions need to be set as 1:1 [10]. Thus, the number of downlink ports for each aggregation switch is also e; the total number of ports for each aggregation switch is then $2e$. Since we assume that core and aggregation switches are of the same type with the same port count and same link capacity (40GigE), we have $f = 2e$. Then, the number of access switches is $ef = \frac{f^2}{2}$. Also, to get 1:1 oversubscription at access level, we set 4 downlinks (10GigE) to hosts and one uplink (40GigE) to aggregation switch. (The number of links can also be set to other values, such as 2 uplinks and 8 downlinks, without affecting the results; for simplicity, our choice is one uplink and 4 downlinks.) Then the number of hosts can be written as $N = 4\frac{f^2}{2} = 2f^2$.

On the other hand, assuming that one 40GigE port can be viewed as four 10GigE ports, then the number of switch ports per host can be written as $\frac{4\frac{f}{2}f+4f\cdot f+\frac{f}{2}f\cdot 8}{2f^2} = 5$.

(b) For fat tree topology, the network's oversubscription ratio is a fixed 1:1. Assume f is the port count per switch, with link capacity 10GigE. The relationship between number of hosts and switch port count is $N = \frac{1}{4}f^3$. Therefore, the number of switch ports per host is $\frac{\frac{5}{4}f^2\cdot f}{\frac{f^3}{4}} = 5$.

(c) For k-ary n-cube FBFLY which has c endpoints per switch (each switch can connect to c hosts), we have the property that when $k = c$, the oversubscription ratio is $1:1$. Thus the number of hosts is $N = ck^{n-1} = k^n$, and the port count per switch is $f = (k-1)(n-1)+c = kn-n+1$; and $k = \frac{f+n-1}{n}$. Since when $n = 5$, the network can achieve good scalability, we choose this value for our comparison. Thus, the relationship between the number of hosts and switch port count can be written as $N = k^n = \left(\frac{f+n-1}{n}\right)^n = \left(\frac{f+4}{5}\right)^5$. On the other hand, the number of switch ports per host is $\frac{fk^{n-1}}{ck^{n-1}} = \frac{f}{k} = \frac{5f}{f+4}$. Since $f = 5\sqrt[5]{h}-4$, we can get that the number of switch ports per host as $\frac{5f}{f+4} = 5 - \frac{4}{\sqrt[5]{N}}$.

(d) For BCube$_k$ topology, we know that the total number of switches is $(k+1)n^k$, where n is the port count of a switch. The total number of switch ports is then $(k+1)n^k \cdot n$. The relationship between number of hosts and switch port count is $N = n^{k+1}$. From this, the number of switch ports per host is $\frac{((k+1)n^k\cdot n)}{n^{k+1}} = k+1$. For BCube$_2$, $N = n^3$, and the number of switch ports per server is 3. For BCube$_3$, $N = n^4$, and the number of switch ports per server is 4.

(e) Since scaling a 3D-torus Camcube would result in undesirable network performance due to long routing paths as opposed to other topologies [2], [14], we exclude this topology for scalability studies.

The results of the scalability comparison among different topologies are shown in Figure 5 and Figure 6 for the two metrics. As we can see from Figure 5, FBFLY outperforms the other two switch-based topologies by supporting more hosts for a given switch port count. For example, when the port count per switch is set to 64, the number of hosts for FBFLY is nearly $8\times$ the number of hosts that can be supported by the fat tree, and $60\times$ the number of hosts in multi-tiered. The hybrid architecture, BCube$_2$, also has better scalability compared to fat tree and multi-tiered topologies. Figure 6 shows the efficiency of port utilization for each topology. We can conclude that BCube$_2$ requires the least number of switch ports per host (or server) in the data center.

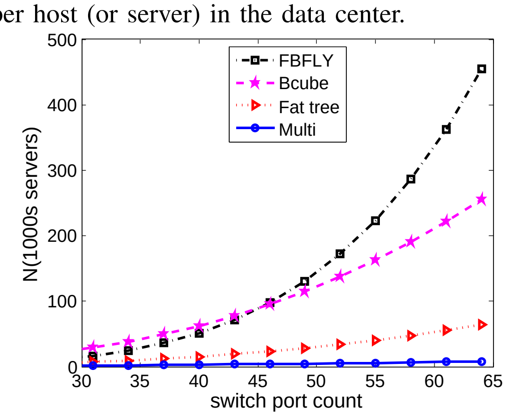

Fig. 5. Number of hosts vs. switch port count.

We also analyze the impact of oversubscription ratio on the scalability of a network. Using multi-tiered data center network as an example, we set the oversubscription to 4:1 and 9:1. For oversubscription 4:1 (aggregation level 2:1 and access level 2:1), the number of hosts as a function of ports per core switch is $N = \frac{16}{3}f^2$. For oversubscription 9:1 (aggregation level 3:1 and access level 3:1), the number of hosts as a function of ports per core switch is $N = 8f^2$. The scalability comparison of different oversubscription values for multi-tiered network is shown in Figure 7. We notice that at higher oversubscription ratio such as 9:1, multi-tiered network can scale to larger number of hosts making it ideal for large scale data centers.

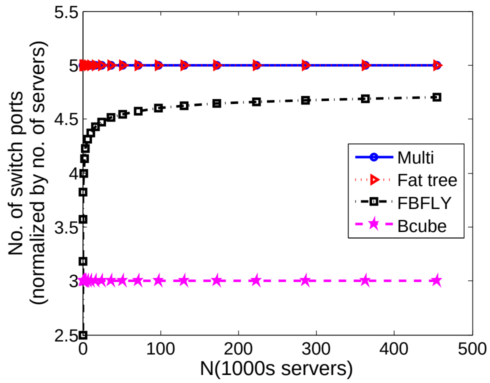

Fig. 6. Number of switch ports per host vs. number of hosts.

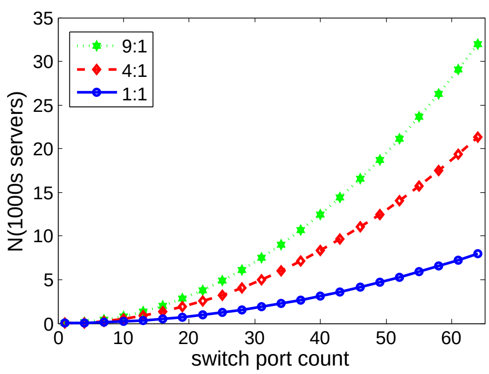

Fig. 7. Scalability of multi-tiered network for different oversubscription ratios.

B. Path Diversity

Path diversity is an important metric for two reasons: a multiplicity of paths can improve load balance and enhance throughput by distributing the traffic load, and the network is more immune to link and switch failures. In this paper, we define path diversity as the number of different shortest paths between a pair of hosts [9]. We consider both the maximum (over all host pairs) number of shortest paths between a pair of hosts, and the average number (over all host pairs) of shortest paths. These paths are not necessarily disjoint. As a second measure, we consider the number of *disjoint* paths (not necessarily shortest), both maximum and average over all host pairs. In order to maintain fairness of the comparison in terms of path diversity, we need to set oversubscription of the networks the same. Here we set it to be 1:1 for all architectures to equalize the performance.

For multi-tiered topology, we know that if the number of core switches is e, the number of hosts is $N = 2f^2 = 8e^2$ with 1:1 oversubscription ratio. By analyzing the topology, we deduce some parameters: $p = 12e^2$ is the number of pairs of hosts under the same access switch, $q = 16e^2(e-1)$ is the number of pairs of hosts under the same aggregation switch but different access switches, and $r = \frac{8e^2(8e^2-1)}{2}$ is the total number of host pairs. Then, the average number of different shortest paths is $\frac{p\cdot 1+q\cdot 1+(r-p-q)\cdot e}{r}$, and the maximum path diversity is e.

For k-ary fat tree topology, we know that the number of hosts is $N = \frac{1}{4}k^3$. We also deduce parameters p, q, r for this topology. Let $p = \frac{k^3}{8}(\frac{k}{2}-1)$, $q = \frac{k^4}{16}(\frac{k}{2}-1)$, and $r = \frac{k^3}{8}(\frac{k^3}{4}-1)$. Then the average number of shortest paths is $\frac{p\cdot 1+q\cdot \frac{k}{2}+(r-p-q)\cdot \frac{k^2}{4}}{r}$, and the maximum path diversity is $\frac{k^2}{4}$.

For FLBLY, we have chosen $n = 5$. Hence, the maximum number of shortest paths is $(n-1)! = 24$. The average number is calculated using the number of shortest paths between pairs of hosts that are in the same or adjacent dimension.

For BCube, we have chosen $k = 2$, since a BCube$_2$ network is already capable of connecting desired number of hosts for our comparison. The maximum number of different shortest paths is $k + 1 = 3$. The average number of shortest paths is computed by exhaustively calculating the number of shortest paths for each host pair in the BCube. A comparative analysis of different topologies is shown in Figures 8 and 9.

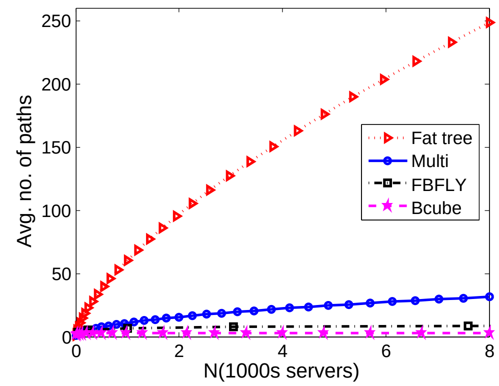

Fig. 8. Average number of shortest paths between a pair of hosts.

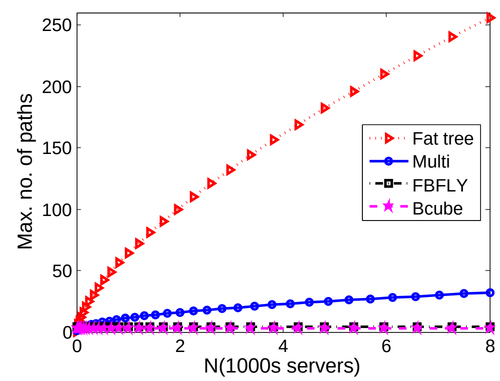

Fig. 9. Maximum number of shortest paths between a pair of hosts.

A second measure we choose to quantify path diversity is the number of disjoint paths between a pair of hosts. We exclude the link connecting hosts and access switches when computing number of disjoint paths. For multi-tiered topology, the maximum and average number of disjoint paths are both 1. For fat tree topology, the maximum number of disjoint paths is $\frac{k}{2}$, and the average is $\frac{p\cdot 1+q\cdot \frac{k}{2}+(r-p-q)\cdot \frac{k}{2}}{r}$. For recursive or multiple-level topologies like flattened butterfly and BCube, we can determine the number of disjoint shortest paths for any pair of nodes by utilizing the Hamming distance between the source node and destination node, when the nodes are labeled by its base and dimension (or level) in the topology. Note that the minimum number of disjoint shortest paths for all these topologies is 1. The results are shown in Figure 10.

For both cases, the average number of shortest paths for fat tree is higher with increasing number of hosts, making it ideal for fault tolerance purposes.

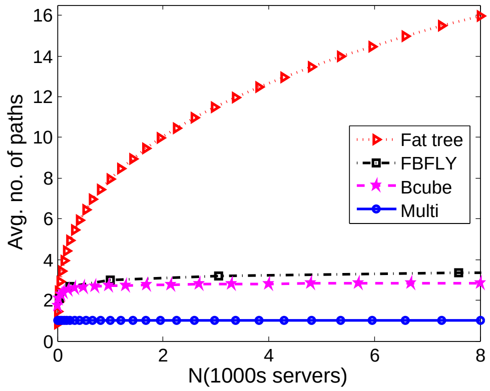

Fig. 10. Average number of disjoint shortest paths between a pair of hosts.

For the Camcube, we can easily see that the number of different shortest paths is $n!$ where $n = \log_2 N$ is the dimension of the Camcube. However, we omit plotting results for Camcube due to the same reason mentioned in III-A.

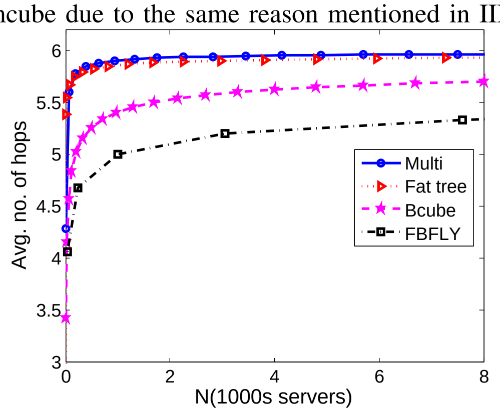

Fig. 11. Average hop count.

C. Hop Count

The average hop count is the average number of hops on the shortest path between a pair of hosts (averaged over all host pairs). This metric is useful in deducing packet latencies. *Similar to the comparison with respect to path diversity, we choose an oversubscription ratio of 1:1 to normalize the performance, and the parameters are the same as those for path diversity comparison.* For multi-tiered and fat tree topologies, we use the same parameters p, q and r that we defined earlier for the path diversity comparison. The average hop count for multi-tiered topology is $\frac{p\cdot 2+q\cdot 4+(r-p-q)\cdot 6}{r}$, and the maximum hop count is 6. The average hop count for fat tree topology is $\frac{p\cdot 2+q\cdot 4+(r-p-q)\cdot 6}{r}$. The maximum hop count is 6. For FBFLY and BCube, the hop counts are calculated using Hamming distance. The comparison results are shown in Figure 11. We notice that for smaller number of hosts, FBFLY and BCube have lower hop count than others making them more suitable for smaller scale data centers.

D. Throughput

In a network, throughput (or accepted traffic) is the rate (bits/sec) at which traffic is delivered to the destination nodes [9]. In our experiment, we show normalized throughput over maximum achievable injection bandwidth. We performed simulations using Mininet [8], a software network emulator, and networkx [15] that statically analyzes network constructs and features. We ran our simulations on a Xeon x5472 3.0GHz quad-core CPU machine with 8G DRAM.

At present, we have results for switch-only topologies, as the simulator requires considerable reconfiguration effort to make it work for other topologies. We compare our topology results with a star topology where every pair of hosts are connected through a single non-blocking switch using a dedicated pair of links. In our experiments, each topology has 16 hosts with 10 Mbps link capacity (except for the aggregation level links with capacity 40 Mbps in multi-tiered topology).

We study the topologies using several types of workloads: hotspot, random and stride [16]. Each host in Mininet runs a *shell* program and communication among programs is modeled for the above-mentioned workload patterns. The average throughput for the workloads in each architecture is measured by averaging over three independent runs. The results for the three types of workloads (random, stride and hotspot) are shown in Figure 12. Note that although fat tree and multi-tiered networks have the same physical bisection bandwidth, the achievable throughput for fat tree is much better. This is due to the larger number of disjoint paths between nodes in fat tree, that could potentially balance the load and reduce traffic congestion.

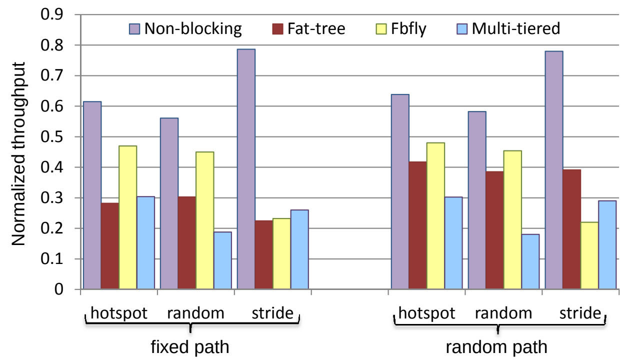

Fig. 12. Throughput comparison for different topologies. Fixed path: statically choose one path from all the available paths; Random path: generate a random number as an index to the available paths.

E. Cost

We model the cost of a network topology as the capital cost for network infrastructure. For switch-based topologies, network components include NICs, switches, and cables. For server-based data centers, the cost involves switches (if any), NICs and CPU cores (that are responsible for packet forwarding). Table 1 shows the costs of a data center with approximately 4K hosts using different topologies. Assume that P_x is the price per 40GigE port, P_y is the price for a 10GigE port, C is the price for 10GigE NIC port, and N is the number of servers.

For multi-tiered network, with a core level switches, b aggregation level switches and c access switches, the switch cost is: $(a+b)bP_x + c(4P_y + P_x)$. The NIC cards cost would be $4.5k \cdot C$ in our setting.

Fat tree has a constant number of ports per server, namely 5. Therefore, the switch cost is $5NP_y$, and the cost for NIC cards is NC. In this setting, N is equal to 4.3K.

In FBFLY, we set $k = c$ in order to achieve full bisection bandwidth. The number of ports per switch is $P_t = (k-1)(n-1) + c$. The cost for switches is $P_t K^{n-1} P_y$, and the cost for NIC cards is NC.

For BCube with base N and K levels, the number of switches is $N^K(K+1)$. The number of ports per switches are also N. We use BCube_2, so $K = 2$. Therefore, each server has 3 NIC ports, and the cost for NIC cards is $3N^{K+1}$.

Camcube is a 3D torus, so we assume a 16 base 3-cube torus. Each server has 6 NIC ports.

For the cost numbers, we refer to Popa et al. [2] for the price of network components and also consider the data revealed by Mudigonda et al. [6]. We assume that switch cost is $400 per 10GigE switch port and $2500 per 40GigE switch port, $150 per Ethernet NIC port, and $200 per CPU core. For BCube, we assume two extra cores per server, for Camcube, we assume 4 packet forwarding cores.

TABLE I
COST COMPARISON OF DATA CENTER TOPOLOGIES.

	Cost(k$)			
	Switch	NIC	CPU core	Total
Multi-tiered	13360	690	0	14050
Fat tree	8806	690	0	9496
FBFLY	5939	690	0	6629
Camcube	0	3686	3276	6962
BCube	614	1843	1638	4095

F. Power Consumption

The topology parameters for power consumption comparison are the same as for the cost comparison. Here, we quote the power consumption of network components from [2]: 12W for 10GigE switch port and 40GigE switch port, 10W for NIC, and 10W per CPU core. Table 2 shows the power consumption of data centers of approximately 4K hosts using different topologies.

TABLE II
POWER CONSUMPTION COMPARISON OF DATA CENTER TOPOLOGIES.

	Power(kW)			
	Switch	NIC	CPU core	Total
Multi-tiered	110	46	0	156
Fat tree	264	44	0	308
FBFLY	178	45	0	223
Camcube	0	245	162	407
BCube	18	122	81	221

G. Trade-offs

In this section, we present an overall comparison of the metrics discussed in the previous sections based on different network sizes. To be more specific, we considered three network scales: small, S ($N \leq 100$), medium, M ($100 < N \leq 1000$), and large, L ($1000 < N \leq 5000$). Table III shows the overall comparison. The letters in the table show the best-performing architecture for a particular metric. Some useful insights can be gained from these results. For example, the multi-tiered architecture is well-suited for small and medium data centers, if power is the major concern. Camcube is cost-effective for small to medium-sized data centers but does not scale well. Both fat tree and flattened butterfly could be used to build large scale high performance data centers. While fat tree has better fault tolerance and latency, flattened butterfly is less expensive and more scalable.

TABLE III
A SUMMARY TABLE OF COMPARISON AMONG TOPOLOGIES.

	Scalability	Latency	Path diversity	Cost	Power
Multi-tiered					S M
Fat tree		L	S M L		
FBFLY	S L	S M		L	L
Camcube				M	
BCube	M			S	

IV. CONCLUSION

In this work, we presented a comparative analysis of several popular data center network topologies such as Fat tree, Multi-tiered networks, Flattened Butterfly, Camcube and BCube. The metrics that we chose for comparison were scalability, path diversity, hop count, throughput and cost. We find that different topologies scale differently for various metrics, and we conclude that data center designers have to consider such characteristics to maximize their performance while minimizing cost and power. As future work, we will study energy optimization strategies and application-specific constraints to better understand data center needs and design.

REFERENCES

[1] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu, “Energy proportional datacenter networks,” in *ISCA*, 2010.

[2] L. Popa, S. Ratnasamy, G. Iannaccone, A. Krishnamurthy, and I. Stoica, “A cost comparison of datacenter network architectures,” in *Co-NEXT*, 2010.

[3] 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 *SIGCOMM*, 2009.

[4] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, “Portland: a scalable fault-tolerant layer 2 data center network fabric,” in *SIGCOMM*, 2009.

[5] M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” in *SIGCOMM*, 2008.

[6] J. Mudigonda, P. Yalagandula, and J. C. Mogul, “Taming the flying cable monster: A topology design and optimization framework for data-center networks,” in *USENIX ATC*, 2011.

[7] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, “Elastictree: Saving energy in data center networks,” in *NSDI*, 2010.

[8] “Mininet,” http://mininet.github.com/.

[9] W. J. Dally and B. P. Towles, *Principles and practices of interconnection networks*. Elsevier, 2004.

[10] “Cisco data center infrastructure 2.5 design guide,” http://www.cisco.com/.

[11] J. Kim, W. J. Dally, and D. Abts, “Flattened butterfly: a cost-efficient topology for high-radix networks,” in *ISCA*, 2007.

[12] H. Abu-Libdeh, P. Costa, A. Rowstron, G. O’Shea, and A. Donnelly, “Symbiotic routing in future data centers,” in *SIGCOMM*, 2010.

[13] 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 *SIGCOMM*, 2009.

[14] P. Costa, T. Zahn, A. Rowstron, G. O’Shea, and S. Schubert, “Why should we integrate services, servers, and networking in a data center?” in *WREN*, 2009.

[15] “Networkx,” http://networkx.github.com/.

[16] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, “Hedera: Dynamic flow scheduling for data center networks,” in *NSDI*, 2010.