J. Parallel Distrib. Comput. 96 (2016) 45–74

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

The features, hardware, and architectures of data center networks: A survey

Tao Chen, Xiaofeng Gao *, Guihai Chen

Shanghai Key Laboratory of Scalable Computing and Systems, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China

HIGHLIGHTS

- We give a survey on the features and hardware of Data Center Networks.
- We thoroughly analyze the topology designs and architectures of DCNs.
- We provide both qualitative and quantitative analyses on the features of DCNs.

ARTICLE INFO

Article history:
Received 3 August 2015
Received in revised form
15 April 2016
Accepted 10 May 2016
Available online 18 May 2016

Keywords:
Data center network
Architecture
Hardware
Topology

ABSTRACT

The rapid development of cloud computing in recent years has deeply affected our lifestyles. As core infrastructures of cloud computing, data centers have gained widespread attention from both the academia and industry. In a data center, the data center network (DCN) that plays a key role in computing and communication has attracted extensive interest from researchers. In this survey, we discuss the features, hardware, and architectures of DCN's, including their logical topological connections and physical component categorizations. We first give an overview of production data centers. Next, we introduce the hardware of DCN's, including switches, servers, storage devices, racks, and cables used in industries, which are highly essential for designing DCN architectures. And then we thoroughly analyze the topology designs and architectures of DCN's from various aspects, such as connection types, wiring layouts, interconnection facilities, and network characteristics based on the latest literature. Finally, the facility settings and maintenance issues for data centers that are important in the performance and the efficiency of DCN's are also briefly discussed. Specifically and importantly, we provide both qualitative and quantitative analyses on the features of DCN's, including performance comparisons among typical topology designs, connectivity discussion on average degree, bandwidth calculation, and diameter estimation, as well as the capacity enhancement of DCN's with wireless antennae and optical devices. The discussion of our survey can be referred as an overview of the ongoing research in the related area. We also present new observations and research trends for DCN's.

1. Introduction

The term "cloud computing" in the modern sense appeared in a Compaq internal document as early as 1996 [143]. In 2006, Google CEO came up with the concept of "cloud computing" in business [147], which is a model based on the premise that the data services and architecture should be on "cloud" servers. Having the right kind of browser or software in a device (e.g., PC, mobile phone, etc.), you can access to the cloud services freely. Also in 2006, Amazon introduced the Elastic Compute Cloud (Amazon EC2), which is a web service that provides resizable compute capacity in the cloud [7]. In Wikipedia, the term "cloud computing" involves the provision of dynamically scalable and often virtualized resources as a service over the Internet. In science, cloud computing is synonymous to distributed computing over a network, which means the ability to run a program or application simultaneously on many interconnected computers [174]. Generally, cloud computing is a service model where tenants can acquire resources on demand based on service-level agreements (SLAs). Depending on the level of resources, cloud service models can be categorized into Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS), and Anything-as-a-Service (XaaS, X can stand for network, database,

* Corresponding author.
E-mail address: gao-xf@cs.sjtu.edu.cn (X. Gao).

http://dx.doi.org/10.1016/j.jpdc.2016.05.009
0743-7315/

communication, etc.). Data-Center-as-a-Service (DCaaS) is also an important cloud service mode. Data center providers offer places and customized guidances for tenants to construct their data centers with their own equipments. Among these cloud service models, IaaS is the most basic one where providers offer computers (physical or virtual machines) and other resources placed in (part of) a building known as a "data center".

Although the concept of "data center" was proposed in the 1990s, the characteristic features and requirements of a data center actually appeared at the beginning of the very first computer operation [17]. In the early 1960s, the lowest-level (i.e., Tier 1) data center had been deployed, probably a computing center of some laboratory in a university [164]. Nearly 30 years after, in the mid-1990s, the highest-level (i.e., Tier 4) data center was constructed. The name "data center" was used when the Tier 4 data center was developed. Telecommunications Infrastructure Standard for Data Centers (i.e., ANSI/TIA-942-2005) defines data center as: a building or portion of a building whose primary function is to house a computer room and its support areas [161]. Google defines a data center as a building where multiple servers and communication equipment are co-located because of their common environmental requirements and physical security needs, and for ease of maintenance [19]. Based on the two definitions, the computer room is the core physical environment of a data center, which consists of computing equipments for data processing and other areas offering support services to the computer room. Support services mainly comprise the power supply system (including backup power system), cooling system, lighting system, cabling system, fire protection system, and security system. In spite of these highly automated support systems, the staff is also essential to handle routine work and emergencies.

Data centers are the main infrastructures to support many applications, such as cloud service, supercomputing, and social networks. A data center is a huge building consisting of various areas, among which the data center network (DCN) plays a pivotal role in computing and communication. DCN connects the physical components of data centers (e.g., servers, switches) in a specific topology with cables and optical fibers, and the efficiency and performance of a data center greatly depend on the DCN. Since SIGCOMM (the flagship conference of the ACM Special Interest Group on Data Communication) first set a session on data center networking in 2008, the architecture design has become a very active research field to improve the efficiency and performance of DCN's. Many novel architectures have been designed and presented, and many novel devices have been attached to DCN's, such as wireless antennas and optical switches. In just a few years, several surveys on DCN's have been presented. However, all these surveys are not comprehensive, and the limitations include the following four aspects:

1. These surveys hardly include any introduction and discussion on the hardware in DCN's, such as switches, servers, storage devices, racks and cables, which are highly essential for designing DCN architectures. Making these information available provides a benefit for the research communities to understand the DCN's thoroughly.
2. Although several surveys presented some simplified and partial quantitative analyses on the performance of DCN architectures, comprehensive quantitative analyses are scarce, which are more helpful for the researchers to understand DCN's in depth.
3. These surveys only discussed several aspects of DCN's without an overall perspective. A comprehensive survey will benefit the researchers in future.
4. These surveys focused almost exclusively on the wired architectures of DCN's, whereas wireless and optical architectures are hardly proposed.

In this paper, we comprehensively focus on the features, hardware, and architectures of DCN's, including their logical topological connections and physical components categorizations. We first give an overview of production data centers. Next, we introduce the hardware of DCN's, including switches, servers, storage devices, racks and cables used in industries, which are highly essential for designing DCN architectures. And then we thoroughly analyze the topology designs and architectures of DCN's from various aspects, such as connection types, wiring layouts, interconnection facilities, and network characteristics based on the latest literature. Finally, the facility settings and maintenance issues for data centers are also briefly discussed.

Specifically and importantly, we provide both qualitative and quantitative analyses on the features of DCN's, including performance comparisons among typical topology designs, connectivity discussion on average degree, bandwidth calculation, and diameter estimation, as well as the capacity enhancement of DCN's with wireless antennae and optical devices. Our survey can be referred as an overview of the ongoing research in the related area. We also present new observations and research trends.

The rest of this paper is organized as follows. Section 2 provides an overview of production data centers. Section 3 introduces the hardware of DCN's used in industries. Section 4 summarizes the architectures of DCN's, then offers the comparisons and future research directions. Section 5 introduces the considerations of the support systems of DCN's. Section 6 concludes the survey.

2. An overview of production data centers

Nowadays, production data centers (DC's) have become indispensable for large IT companies. An overview can provide a benefit for the research communities to better understand production DC's comprehensively. In this section, we first introduce several representative production DC's, then focus on their main features, including size, infrastructure tiers, and modularity. Finally, we introduce green DC's as the new trend.

2.1. Representative production data centers

Large IT companies constructed several production DC's to support their business. Others are rented out to provide services to medium-sized and small-sized enterprises that cannot afford their own DC's.

Google owns 36 production DC's globally, 19 of which are in America, 12 in Europe, 3 in Asia, 1 in Russia, and 1 in South America, as shown in Fig. 1(a) [148]. These DC's support Google services, such as searching, Gmail, and Google Maps. In 2016–2017, Google DC's will be constructed in Oregon USA, Tokyo Japan and other ten countries and regions.

The prototype of Google's first DC in Fig. 1(b), *BackRub* was once located in the dorm of Larry Page (one of Google's founders) [58]. Although it was simple and crude, BackRub had met basic requirements of Google searching at that time.

Google cost nearly $600 million to build the first DC in 2006, say, Portland Dalles Data Center. It is a pair of 94,000-square-foot DC's that sit on the banks of Columbia River, and is powered by the Dalles Dam [13]. Two four-story cooling towers are used to low the water temperature, and the water vapor is shown in Fig. 1(c) [59]. Google announced another $600 million to build a new DC with 164,000 square feet in Dalles in 2013, and opened it in 2015 [101].

Another Google's typical DC is Georgia Douglas County Data Center, as shown in Fig. 1(d). It provides services for the key business such as searching, Gmail, Maps [59]. Finland Hamina Data Center is reconstructed from a paper mill, which utilizes sea water along pipelines of the paper mill to control the data center temperature, as shown in Fig. 1(e) [59].

(a) Map of Google Data Centers.

(b) BackRub.

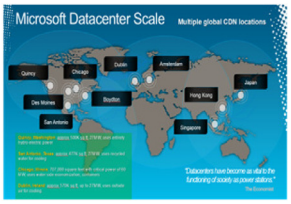
(c) Dalles Data Center.

(d) Douglas Data Center.

(e) Hamina Data Center.

(f) Mayes County Data Center.

Fig. 1. Google data centers.



(a) Map of Microsoft Data Centers.

(b) Quincy Data Center.

(c) San Antonio Data Center.

(d) Chicago Data Center.

(e) Dublin Data Center.

(f) Boydton modular Data Center.

Fig. 2. Microsoft data centers.

Google cost over 2 years and €250 million to build Belgium Saint-Ghislain Data Center, which opened in 2010 [60]. It is the first Google DC worldwide to operate entirely without refrigeration. Instead, it utilizes an advanced evaporative cooling system, which draws grey water (relatively clean wastewater) from a nearby industrial canal. Google planned to invest €300 million to upgrade the facilities for meeting the growing demand of online services in 2013 [89]. Oklahoma Mayers County Data Center has two DC buildings with over $700 million investment, where modular cooling units control the temperature, as shown in Fig. 1(f) [61].

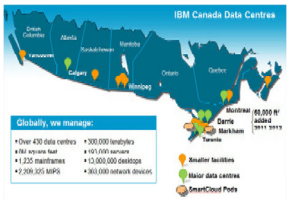
Microsoft also owns production DC's in America, Europe and Asia, as shown in Fig. 2(a) [16]. It built Washington Quincy Data Center with an area of 75 acres in 2007, as shown in Fig. 2(b) [132,163]. Quincy Modular Data Center was online in 2011, which covers 93,023 square feet and utilizes green technologies [128]. In late 2013, Microsoft approved a corporate budget of $11 million to purchase a 200 acre land in Quincy to build a large-scale DC with completion expected in early 2015 [163].

Microsoft established San Antonio Data Center in 2008, which occupies about half a million square feet and costs $550 million, as shown in Fig. 2(c) [130,52]. It costs 8 million gallons of recycled water each month as a part of the cooling system. Illinois Chicago Data Center in Fig. 2(d) [126] was one of the biggest DC's ever in the world, which covers more than 700,000 square feet and costs $500 million. Fifty six 40-foot shipping containers (each contains 1800–2500 servers) are located on the first floor, and the number will grow with additional demands. On the second floor, servers are placed at four traditional raised-floor rooms (each with 12,000 square feet). Cooling water along about seven-mile pipelines keep the DC in a low temperature.

Dublin Data Center in Fig. 2(e) [128] is the biggest oversea DC of Microsoft. It covers 303,000 square feet, and achieves cooling by natural wind for saving energy. It expands with a new 112,000 square feet to place modular DC's (MDC's). Boydton MDC with 316,300 square feet can quickly meet customer demands for cloud services, as shown in Fig. 2(f) [128].

Other large IT companies also own production DC's. IBM, for instance, has always been devoted to building smarter DC's. IBM manages over 430 DC's worldwide, with an overall size of up to 8 million square feet, as shown in Fig. 3(a) [85]. In Canada, IBM built DC's in collaboration with the government and universities, such as Barrie Cloud Data Center (an MDC) in Fig. 3(b) [84], which covers up to 100,000 square feet and significantly improves the power usage effectiveness (PUE) by innovative technologies. In 2014, IBM announced a $1.2 billion commitment to build 15 new DC's in 15 countries in five continents, except in Africa and Antarctica [24].

Amazon also owns DC's globally, which not only support the e-commerce business but also the services for worldwide



(a) IBM Canada Data Centers.

(b) IBM Barrie Cloud Data Center.

(c) Map of Amazon Data Centers.

(d) HP Tulsa Data Center.

(e) Dell Quincy Data Center.

(f) Apple Maiden Data Center.

Fig. 3. Other data centers.

Table 1
The size of data centers.

Size	Covering area (ft^2)	Examples
Huge	More than 100,000	Microsoft Quincy
Large	20,000 to 100,000	Oracle Austria
Medium	5000 to 20,000	Sinopec group
Small	2000 to 5000	SJTU

enterprises, governments, and startup companies by Amazon Web Service (AWS), as shown in Fig. 3(c) [102].

HP's Oklahoma Tulsa Data Center in Fig. 3(d) covers 404,000 square feet with 4 data halls (each 40,000 square feet). It installs a reflective roof system for avoiding sunlight to increase the temperature, and an innovative cooling system to keep DC running stably. It can withstand a Force 5 tornado [36].

Dell's Quincy Cloud Data Center in Fig. 3(e) covers an area of 40,000 square feet and costs $3.6 million in the first phase [90]. Dell also owns other DC's in India and China.

Apple has all DC's powered by 100% renewables [9]. It owns DC's in Maiden (North Carolina), Austin (Texas), Prineville (Oregon), Newark (California), Reno (Nevada), Cork (Ireland) and Munich (Germany). North Carolina Maiden iCloud Data Center in Fig. 3(f) covers 500,000 square feet, which supplies 20 megawatts of power with a 100-acre solar farm [184,10].

2.2. Size of production data centers

To satisfy the growing demand for cloud services, production DC's are growing in size incredibly quickly. Generally, the size of a DC is decided by the covering area and the volumetric ratio (the ratio of covering area and the number of racks). According to the covering area, DC's can be classified into huge, large, medium, and small sizes in Table 1.

According to the covering area, the top 10 largest DC's in the world by 2015 are listed in Table 2, which range from 750,000 to 6.3 million square feet [38]. The other mega DC's are listed in [42], which contain multi-facility campuses or mixed-use buildings where DC space co-exists with large third-party office space (i.e. big-city carrier hotels).

According to the number of racks, we can also divide DC's into huge scale (>10,000 racks), large scale (3000–10,000 racks), and medium and small scale (<3000 racks). The volumetric ratio is a more effective indicator that can reflect the utilization of a DC. The lower the volumetric ratio is, the higher the utilization is. In Table 3, we show the volumetric ratio of several famous DC's.

Table 2
Top ten largest data centers.

Name	Covering area (ft^2)
Range International Information Group	6,300,000
Switch SuperNAP	3,500,000
DuPont Fabros Technology	1,600,000
Utah Data Centre	1,500,000
Microsoft Data Centre	1,200,000
Lakeside Technology Center	1,100,000
Tulip Data Centre	1,000,000
QTS metro Data Center	990,000
NAP of the Americas	750,000
Next Generation Data Europe	750,000

Table 3
The volumetric ratio of several famous data centers.

Name	Covering area (ft^2)	Number of racks	Volumetric ratio
Google Dalles	200,000	9090	2.0
Oracle Austria	82,000	2280	3.3
Cisco Richardson	34,000	1151	2.7

2.3. Infrastructure tiers of data centers

Tier classification is important for DC planners in preparation for the construction budget. Uptime Institute originally defined the four data center infrastructure tiers in the white paper entitled "Tier Classifications Define Site Infrastructure Performance" [164]. Telecommunications Infrastructure Standard for Data Centers (TIA-942-2005) also adopts the definitions from this white paper [161]. In general, the tier level of a DC depends on that of the weakest system. For instance, if the power system is rated at tier 3 while the cooling system is rated at tier 2, the DC is rated at tier 2. We summarize tier requirements and common attributes in Table 4 [164,161].

2.4. Modular data centers

Traditional production DC's are mainly located in fixed buildings, which generally take several years to construct. DC's should be easy to transport and deploy to satisfy the flexible business requirements. Therefore, the concept of Modular data center (MDC) was put forward, which is placed in a shipping-based container. A modular container (usually 20 or 40 ft) is a small DC

(a) Sun modular Data Center.

(b) IBM modular Data Center.

(c) HP Pod modular Data Center.

(d) 21 ViaNet group modular Data Center.

Fig. 4. Examples for modular data centers.

Table 4
Tier requirements and common attributes.

	Tier 1	Tier 2	Tier 3	Tier 4
Number of delivery paths	1	1	1 active and 1 passive	2 active
UPS redundancy	N	$N+1$	$N+1$	$2N$
Continuous cooling	Load density dependent	Load density dependent	Load density dependent	Yes
Concurrently maintainable	No	No	Yes	Yes
Fault tolerance (single event)	No	No	No	Yes
Compartmentalization	No	No	No	Yes
Availability/downtime (hours/year)	99.67%/28.8	99.75%/22.0	99.98%/1.6	99.99%/0.8
Months to plan and construct	3	3–6	15–20	15–30
First deployed	1965	1970	1985	1995

that includes servers, storage devices, networks devices in racks, UPS, cooling system, and so on. It is flexible to run independently or to be connected with other containers to build a larger data center to meet different business requirements. Whether or not to choose MDC depends on the demands of data center planners and customers. The pros and cons of MDC's versus traditional data centers are listed as follows.

Pros. First, MDC can be easily and rapidly deployed to meet the requirements of customers. It is plug-and-play to reduce the period of deployment, and can run immediately after connecting with power, water, and the Internet. Second, it can support around six times more servers (400–2000 servers) than a traditional DC in the same space. Third, it can achieve a low Power Usage Effectiveness (PUE) to cut power cost by using an optimal cooling system. Finally, it can run at full capacity to reduce the CAPEX and OPEX of vendors and tenants.

Cons. First, server compatibility is a drawback of MDC. MDC's are produced by different vendors and equipped with exclusive servers, so they are difficult to modify after a life span of 10 years. Second, its space is compact, which is hard for maintenance. Third, its price is still expensive. Finally, its deployment needs enormous space and large cranes.

In 2006, Sun first presented Blackbox MDC in Fig. 4(a) [129]. A Blackbox contains 8 19-in racks supporting up to 2240 servers and 3PB storage. IBM portable data center in Fig. 4(b) can travel by trunk or other transports [173]. It can support up to 798 1U (1U = 1.75 in = 44.45 mm) servers or 1,596 blade servers placed in 19-in racks, and the total power is about 410 kilowatts (kW). The interior of HP Performance-Optimized Data center (POD) is shown in Fig. 4(c) [137]. A 20-foot POD can support about 1500 computing nodes, 29 kW per rack, with 10 50-unit racks, whereas a 40-foot POD can support 27 kW per rack, with 22 50-unit racks. The first containerized Internet data center of China has been running in the 21ViaNet Group data center campus in Beijing since 2010, as shown in Fig. 4(d) [125]. A 40-foot container can contain over 1000 servers.

Many cloud DC's today cloud be called mega data centers (Mega DC's), which have tens of thousands servers costing tens of megawatts of power with peak workloads. A modular/micro data center (MDC), by contrast, only has thousands of servers (generally 1000–4000) costing thousands of kilowatts with peak workloads [63], which usually placed in agile and cheap shipping containers [72].

Table 5
Several data centers with low PUE value.

Name	PUE
HP EcoPOD Data Center	1.05
Facebook Prineville Data Center	1.07
Yahoo New York Data Center	1.08
Capgemini Merlin modular Data Center	1.08
Google Saint-Ghislain Data Center	1.16
Microsoft Dublin Data Center	1.25

2.5. Green data centers

A green DC is an energy efficiency DC, which employs energy-saving technologies (e.g. modular design, advanced power unit), green management, and renewable resources.

Power Usage Effectiveness (PUE) is a key indicator of energy efficiency, proposed in 2006, which is a ratio of energy used by computing equipment to the total energy used by the data center (including computing equipment, cooling, and other overhead). The ideal value of PUE is 1, which means energy is completely cost by the computing equipment. However, DC's can rarely reach this value due to the cooling system. In Table 5, we list several DC's with low PUE value ($\leqslant$1.08). These DC's achieve low PUE values by different technologies. HP EcoPOD is a MDC, as shown in Fig. 5(a) [76]. Compared with legacy data center designs, HP's self-compensating Adaptive Cooling technology reduced 95% facilities energy without decreasing the peak performance.

Facebook Prineville Data Center in Fig. 5(b) is powered by a large-scale solar array [131]. Yahoo!'s New York Data Center in Fig. 5(c) is angled to take advantage of Buffalos microclimate with 100% outside air to cool the servers. It is called Computing Coop because it looks like something that chickens live in [178]. Capgemini Merlin MDC in Fig. 5(d) applies adiabatic cooling technology (a method of evaporative cooling), which makes outside fresh air pass through a wet filter to reduce the temperature [43]. A modular only needs 10 kW for cooling. Google Belgium Saint-Ghislain Data Center in Fig. 5(e) has an advanced evaporative cooling system, which draws grey water (relatively clean wastewater) from a nearby industrial canal without water chillers [60]. Microsoft Dublin Data Center in Fig. 5(f) utilizes air economizers to increase cooling efficiency, and recycles over 99% of all wastes [128].

(a) HP EcoPOD Data Center.

(b) Facebook Prineville Data Center.

(c) Yahoo New York Data Center.

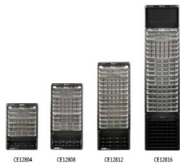
(d) Capgemini Merlin modular Data Center.

(e) Google Belgium Saint-Ghislain Data Center.

(f) Microsoft Dublin Data Center.

Fig. 5. Some data centers with low PUE value.

(a) Cisco Nexus 7000 series data center switches.



(b) Cloud Engine 12 800 series high-performance switches.

(c) RG-N18000 series data center switches.

(d) Arista 7500E series data center switches.

(e) Cisco Nexus 3064 series switches.

(f) Arista 7050QX series data center switches.

Fig. 6. Switches in cloud data centers.

Several DC's have attempted to use renewable energy resources to reduce operating costs and carbon emissions. Facebook has settled a large array of solar panels at Oregon Data Center to supplement electricity usage [131]. Apple already produces 100% renewable energy (i.e., solar, wind, hydro, and geothermal) to power all its DC's [9]. For example, Maiden Data Center fitted a 40 MW co-located solar farm (includes two 20 MW solar photovoltaic facilities) and a 10 MW fuel cell that runs on biogas. Prineville Data Center is powered by locally sourced renewable resources, including wind, hydroelectric, and solar. Microsoft is aggressively considering to purchase long-term renewable power, invest in renewable energy projects, such as wind and methane, and connect DC's directly to innovative energy sources [127]. In summary, renewable energy resources should be utilized by more DC's to achieve the objectives of energy saving and emission reduction.

3. Hardware of data center networks

Hardware is practically physical components, which is highly essential for designing DCN's. Making these information available provides a benefit for the research communities to understand the DCN's thoroughly. The performance requirements of hardware in DCN's have become increasingly higher as the demands of cloud services grows. In this section, we will introduce hardware used in DCN's, including switch, server, storage, rack and cable.

3.1. Switch

Switches are the backbone of many DCN architectures. For instance, fat-tree [5] and VL2 [62] are both three-layer network architectures, including core switch layer, aggregation switch layer, and edge switch (Top of Rack, ToR) layer.

Owing to switch technology evolves rapidly, we present six kinds switches used in cloud DC's for reference only, as shown in Fig. 6. Four kinds of **core** switches are Cisco Nexus 7000 Series data center switches [34], Huawei Cloud Engine 12 800 Series switches [80], Ruijie RG-N18000 Series switches [145], and Arista 7500E Series switches [12], as shown in Fig. 6(a)–(d), respectively. Two kinds of **ToR** switches are Cisco Nexus 3064 Series [33] and Arista 7050QX Series switches [11], as shown in Fig. 6(e); (f), respectively. The main performance parameters, including switching capacity, forwarding performance, and number of line-speed ports, are listed in Table 6.

In addition, **optical** switches have gained great attention in recent years. An optical switch enables signals in optical fibers or integrated optical circuits to be selectively switched from one circuit to another. Several literatures [50,168,28] discuss some issues that utilizing optical switches in DCN's. Owing to expensive optical transceivers and long latency, DCN's have not been deployed with large-scale optical switches. However, in high-performance computing (HPC) systems such as Blue Gene/Q, PERCS/Power 775 and P7-IH, optical modules have been widely used [20]. More detailed analyses and comparisons about optical interconnects for future DCN's can be referred to [92,91].

3.2. Server

Servers are the core physical components of DCN architectures, which process, analyze, store, and transmit massive data and directly determine the performance of DCN's. According to the shape, servers can be categorized into three types, i.e., tower servers, rack servers, and blade servers.

Table 6
The performance parameters of switches.

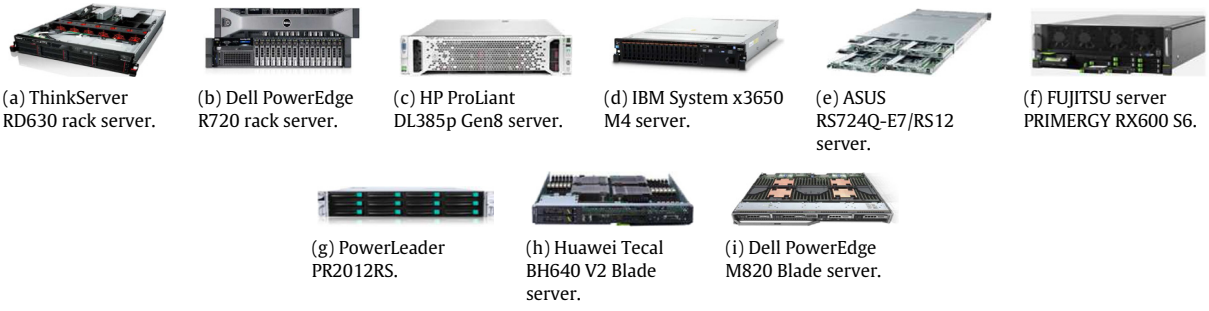
Name	Switching capacity (Tbps)	Forwarding performance	Number of line-speed ports
Cisco Nexus 7000 Series	17.6[a]	1.44–11.5 bpps[b]	32 100 GbE[c], 192 40 GbE or 768 10 GbE
Huawei CloudEngine 12 800 Series	16–64	4.8–19.2 bpps	96 100 GbE, 288 40 GbE or 1152 10 GbE
Ruijie RG-N28000 Series	32–96	11.5–17.3 bpps	96 100 GbE, 288 40 GbE or 1152 10 GbE
Arista 7500E Series	Over 30	up to 14.4 bpps	96 100 GbE, 288 40 GbE or 1152 10 GbE
Cisco Nexus 3064 Series	1.28	950 mpps	48 10 GbE or 4 40 GbE
Arista 7050QX Series	2.56	1.44 bpps	96 10 GbE or 8 40 GbE

[a] Tbps = Terabits per second.
[b] bpps = billion packets per second.
[c] GbE = Gigabit Ethernet.

Table 7
The main performance parameters of servers.

Name	Type[a]	Processor[b]	Mem.[c]	IS[d]	Network interface
RD630	2U	E5-2600	320 GB	24 TB	3 1GE
R720	2U	E5-2600 or 2600v2	768 GB	32 TB	1 10GE & 2 1GE
Proliant Gen8	2U	6200 or 6300	768 GB	Hot plug	4 1GE
System X3650	2U	E5-2600or 2600v2	768 GB	24 TB	2 10GE & 4 1GE
RS724Q-E7	2U	E5-2600 or 2600v2	512 GB	Hot plug	2 1GE
PRIMERGY RX600	4U	E7-2800, 4800 or 8800	2 TB	Hot plug	4 1GE
PR2012PS	2U	E5-2600 or 2600v2	512 GB	48 TB	2 1GE
Tecal BH640	Blade	E5-4600	768 GB	2 T	2 1GE
M820	Blade	E5-4600 or 4600v2	3 TB	4.8 TB	2 10GE & 4 1GE

[a] 2U, 4U are the heights of rack servers.
[b] ProLiant Gen8 server is with AMD Opteron™ processors, and the rest are all with Intel® Xeon® processors.
[c] Mem. = Memory.
[d] IS = Internal storage.



(a) ThinkServer RD630 rack server. (b) Dell PowerEdge R720 rack server. (c) HP ProLiant DL385p Gen8 server. (d) IBM System x3650 M4 server. (e) ASUS RS724Q-E7/RS12 server. (f) FUJITSU server PRIMERGY RX600 S6.

(g) PowerLeader PR2012RS. (h) Huawei Tecal BH640 V2 Blade server. (i) Dell PowerEdge M820 Blade server.

Fig. 7. Servers in cloud data centers.

Tower servers are first used in DC's, of which the shape and performance are larger and several times higher than those of a PC. Several tower servers can satisfy the requirements of small-scale business. However, it is not appropriate for a cloud DC due to the large shape and poor flexibility.

Rack servers are the mainstream servers used in modern DCN's. A rack server is a standard space-saving and maintainable host placed in a rack. A rack can contain several servers, which are arranged like drawers. Compared with tower servers, rack servers have advantages in space occupation and management. However, they have poor heat dissipation and high cabling complexity due to dense placements.

Blade servers are blade-like, low-cost High Availability, High Density servers designed for applications in communication, military, medical, and so on. They support hot plug feature, which significantly reduces the maintenance time of cluster computing. Blade servers have attracted increasing research attention in recent years, and may become the next-generation mainstream servers.

According to Moore's Law, the performance of computers will double each 18 month. As the server hardware evolving so rapidly, we introduce nine kinds of servers in DCN's for reference only, as shown in Fig. 7. Seven kinds of **rack** servers are ThinkServer RD630 [106], Dell PowerEdge R720 rack server [47], HP ProLiant DL385p Gen8 Server [79], IBM System x3650 M4 Server [87], ASUS RS724Q-E7/RS12 server [14], FUJITSU Server PRIMERGY RX600 S6 [53], and PowerLeader PR2012RS Mass Storage Server [139], as shown in Fig. 7(a)–(g), respectively. Two kinds of **blade** servers are Huawei Tecal BH640 V2 Blade Server [82] and PowerEdge M820 Blade Server [45], as shown in Fig. 7(h); (i), respectively. The main performance parameters are listed in Table 7, including type, processor, memory, internal storage and network interface.

Beyond procuring the standard servers from the vendors, some large IT companies, such as Google and Facebook, customize their servers as needed for high performance and low cost. Lu et al. [103] proposed ServerSwitch, a programmable and high performance platform for implementing BCube [66], in which a commodity server and a commodity, programmable switching chip are connected via the PCI-E interface.

3.3. Storage

Network-attached storage (NAS) and Storage area network (SAN) are traditionally two main types of storage systems in data centers [151]. NAS is file-oriented storage network that provides storage access for a heterogeneous group of computer systems. The storage elements are attached directly to a Local Area Network (LAN). SAN is a high-speed storage network that provides enhanced access to consolidated, block-level or file-level data for servers. Fiber Channel (FC), Internet Small Computer

(a) EMC Symmetrix VMAX 40 K. (b) HP 3PAR StoreServ 10 000 storage. (c) Huawei OceanStor N8500. (d) NetApp FAS6200 series. (e) IBM System Storage N7950T.

Fig. 8. Storage systems in data centers.

(a) Emerson Network Power DCF Optimized Racks. (b) Siemon V600 Data Center Server Cabinet. (c) Black Box Freedom Rack Plus with M6 Rails. (d) PowerEdge 4820 Rack Enclosure. (e) HP 11642 1075 Mm Shock Universal Rack.

Fig. 9. Five kinds of racks in cloud data centers.

Table 8
The performance parameters of storage systems.

Name	Storage type	Storage capacity (PB)	Cache capacity
EMC VMAX 40 K	SAN	4	2 TB
HP StoreServ	SAN	3.2	768 GB
Huawei OceanStore	NAS	15	192 GB
NetApp FAS6200	SAN or NAS	4	1 TB
IBM System Storage	SAN or NAS	5	192 GB

System Interface (iSCSI), and Fiber Channel over Ethernet (FCoE) all support SANs.

We introduce five kinds of typical storage systems, which are EMC Symmetrix VMAX 40 K [48], HP 3PAR StoreServ 10 000 Storage [78], Huawei OceanStor N8500 [81], NetApp FAS6200 Series [135], and IBM System Storage N7950T [86], as shown in Fig. 8(a)–(e), respectively. In Table 8, we show the main performance parameters, including storage type, storage capacity and cache capacity.

As the volume of data in data centers rapidly increasing in recent years and unstructured data (e.g., video, photo, and voice) are growing faster than ever before, the centralized management in SAN/NAS may not be adaptable in cloud DC's. A distributed and effective management for mass storage is required to support for better cloud services. The distributed storage systems building by common servers with the help of RDMA over Ethernet [175], such as Windows Azure Storage [26] and Amazon Simple Storage Service (Amazon S3) [6], have been the new trend. The other new trend could be Software Defined Storage (SDS), which includes pools of storage with data service characteristics that may be applied to meet the requirements specified through the service management interface [158].

3.4. Rack

Racks are indispensable key components in DCN's. A rack can support server, switch, and storage devices for easy management and space saving. There are two types of racks, i.e., open racks and cabinets. Open racks are easy to install, configure, and manage, which are categorized into two-post and four-post racks. Four-post racks are superior than two-post racks in cabling. In contrast to open racks, cabinets are more secure and stable. Generally, the height of a rack is between 42 and 48 U (1 U = 1.75 in = 44.45 mm), the width is between 600 and 800 mm, and the depth is between 1100 and 1200 mm. The standard width of devices placed in a rack is 19 in (=482.6 mm). We pick five kinds of 19 in racks used in DCN's, which are Emerson Network Power DCF Optimized Racks [49], Siemon V600 Data Center Server Cabinet [152], Black Box Freedom Rack Plus with M6 Rails [25], Dell PowerEdge 4820 Rack Enclosure [44], HP 11642 1075 mm Shock Universal Rack [77], as shown in Fig. 9(a)–(e), respectively. Black Box Freedom Rack is a four-post open rack, and the rest four are all cabinets. In Table 9, we list the main parameters, including height, width, depth and static load capacity.

3.5. Cable

Cables are essential to DCN architectures, which interconnect the other components (switches, servers and storage devices) and transport electricity or optical signals. Cables are generally categorized as copper and fiber according to the medium. It is crucial to choose proper cables for different applications. The considerations include required useful life of cables, the data center size, the cabling system capacity, and recommendations

Table 9
Comparison of rack parameters in cloud data centers.

Name	Height	Width (mm)	Depth (mm)	Static load capacity (lbs)
Emerson Racks	42 or 48 U	600 or 800	1100 or 1200	3000[a]
Siemon Cabinet	42 or 45 U	600	1000 or 1200	3000
Black Box Rack	45 U	500	450–1000	2500
PowerEdge Rack	48 U	600 or 750	1000 or 1200	2500
HP 11642 Rack	42 or 48 U	600 or 800	1075 or 1200	3000

[a] 3000 lbs ≈ 1360 kg.

Table 10
Cables properties with different Ethernet standards.

Standard	Medium	Distance	Wavelength
10GBASE-CX4	Twinaxial	25 m	N/A
10GBASE-T	CAT5e/6/7 UTP[a]	100 m	N/A
10GBASE-S	MMF[c]	300 m	850 nm
10GBASE-L	SMF[d]	10 km	1310 nm
10GBASE-LX4	MMF or SMF	300 m or 10 km	1310 nm
10GBASE-E	SMF	40 km	1550 nm
10GBASE-ZR	SMF	80 km	1550 nm
40GBASE-KR4	Backplane	1 m	N/A
40GBASE-CR4	STP[b]	7 m	N/A
40GBASE-SR4	MMF	100 m	850 nm
40GBASE-LR4	SMF	10 km	1310 nm
100GBASE-CR10	STP	7 m	N/A
100GBASE-SR10	MMF	100 m	850 nm
100GBASE-LR10	SMF	10 km	1310 nm
100GBASE-ER10	SMF	40 km	1310 nm

[a] UTP, Unshielded Twisted Paired.
[b] STP, Shielded Twisted Paired.
[c] MMF, Multimode Fiber.
[d] SMF, Single-mode Fiber.

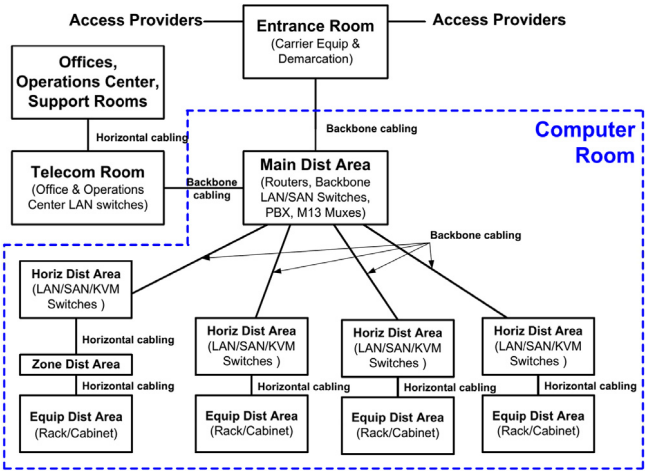


Fig. 10. A typical DCN including cabling system.

or specifications of equipment vendors. A typical DCN including the cabling system is shown in Fig. 10 [161], which depicts the relationship between the elements of a data center and how they are configured to create the total system. Backbone cabling and horizontal cabling are the two main kinds of cabling methods. The Ethernet standards (10GBASE, 40GBASE, and 100GBASE) in Table 10 specify the medium, maximum transmission distance, and wavelength of cables.

4. Architectures of data center networks

Data center networks (DCN's) interconnect the physical components of data centers to support the cloud services. With the dramatic increase in tenants, DCN's must be able to interconnect hundreds of thousands or even millions of servers and provide sufficient bandwidth to ensure the quality of cloud services, and also need to be flexible, reliable, and have high density to ensure that the various applications run steadily and efficiently.

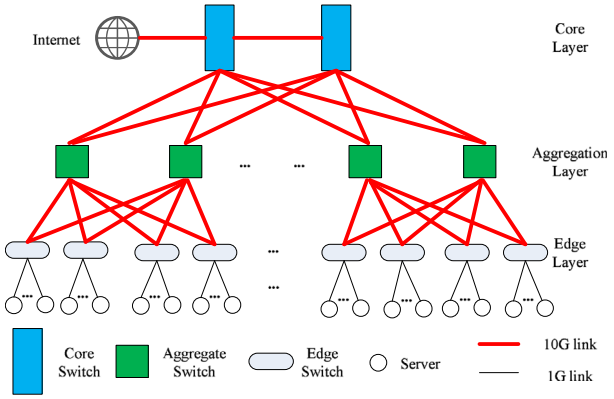


Fig. 11. Traditional data center network architecture.

A traditional DCN has a three-layer, multi-rooted tree-like architecture, as shown in Fig. 11 (adapted from the figure by Cisco [35]). It generally consists of core, aggregation, and edge layer switches in the top-down manner. The uplinks of switches in the core layer connect the data center to the Internet. The switches in the core and aggregation layers interconnect to logically build bipartite graphs with 10G links. The servers are connected directly to the switches in the edge layer with 1G links.

Traditional DCN's cannot meet the increasing demand of cloud services because it is not designed for cloud data centers. It has several inherent disadvantages as follows.

Limited bandwidth. Oversubscription usually occurs when using traditional DCN to reduce operation cost. For example, eight downlinks of a top of rack switch (ToR switch) can be routed to only one uplink, so the bandwidth of a server is really limited. When the workloads reach the peak, the core switches may become bottlenecks, which lead to the performance of the traditional DCN abruptly degrading and make it at the risk of being in a crash.

Poor flexibility. The port number of core switches determines the maximum number of servers supported in the multi-rooted tree-like architectures. If more servers are needed for business when the ports of core switches are all occupied, the present switches must be replaced with new ones with more ports. This kind of incremental deployment, however, is time consuming and costly.

Low utilization. The traditional DCN's are generally divided into multiple domains in layer 2 to ensure security and manageability. This set-up results in massive fragmentation of resources, which are not suitable for large-scale cloud computing. The traditional DCN's also statically assign specific machines and fixed bandwidths for various applications according to their maximum flow rates. Consequently, numerous resources are idle at most of the time and resource utilization is quite low.

Complex cabling. Once the scale of the traditional DCN expand to a large size, the number of cables can be enormous. Cabling becomes a heavy and complex task as servers increasing. The cabling and cooling system will face a great challenge.

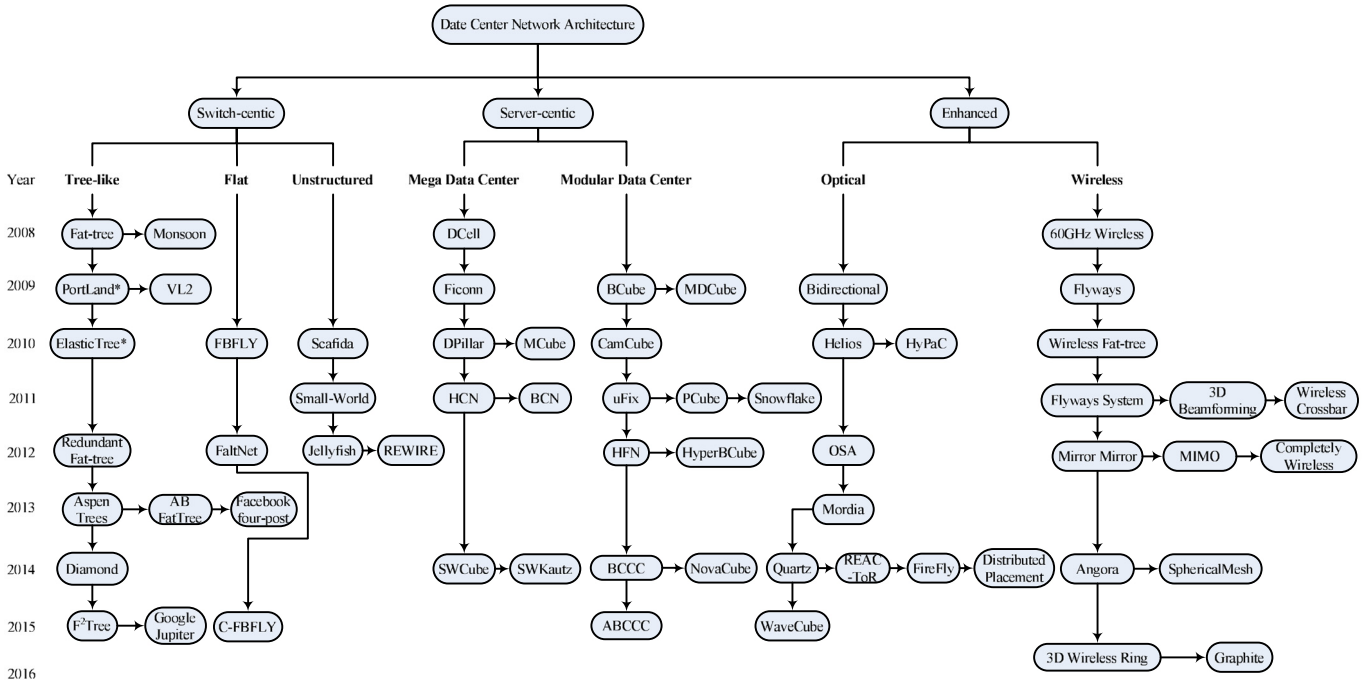


Fig. 12. A chronological tree of data center network architectures. Note: Portland* and ElasticTree* are not topologies, but are two important improvements to Fat-tree.

High cost. The total cost of DCN's includes hardware cost and energy cost. The switches in the core and aggregation layers are usually enterprise-level switches that are very expensive and power hungry, which result in higher Capital Expenditure (CAPEX) and Operating Expense (OPEX).

The modern DCN's should avoid the disadvantages of traditional DCN's and have full bandwidth, good scalability, high utilization, easy cabling, and low cost to provide high-quality cloud services.

This section focuses on the architectures of DCN's proposed from literatures in recent years. A chronological tree comprehensively illuminates the development history of the DCN architectures in timely order, as shown in Fig. 12. It covers almost all the literatures on DCN architectures in major conferences and journals. We categorize those architectures into three categories according to their structural features, say, *switch-centric*, *server-centric*, and *enhanced architectures*. Switch-centric architectures mainly consist of tree-like, flat and unstructured architectures according to their structural features. Server-centric architectures means that servers are responsible for networking and routing, whereas switches without modification are used only for forwarding packets. Switch-centric and server-centric architectures are all adoptable in Mega DC's and MDC's. However, some server-centric architectures are originally designed for MDC's, such as BCube [66] and MDCube [176]. Therefore, server-centric architectures are further divided into two subcategories, say, server-centric architectures for Mega DC's and MDC's. Enhanced architectures include optical and wireless architectures. We would introduce each branch in detail first and then compare their pros and cons in Section 4.4.

4.1. Switch-centric architectures

In switch-centric architectures, the switches are enhanced to accommodate networking and routing requirements, whereas the servers are almost all unmodified. According to the structural properties, switch-centric architectures can be divided into tree-like, flat and unstructured architectures.

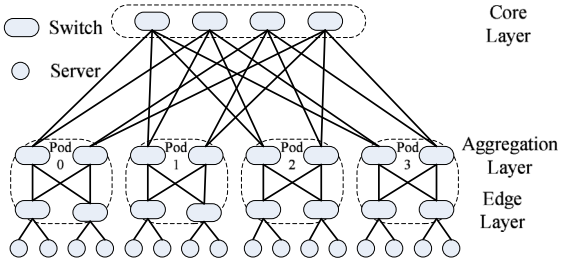


Fig. 13. A fat-tree architecture ($k = 4$).

4.1.1. Tree-like switch-centric architectures

In a tree-like switch-centric architecture, the switches are interconnected to form a multi-rooted tree. Typical architectures include Fat-tree [5], VL2 [62], Diamond [160], Redundant fat-tree [68], Aspen Trees [167], F10 [119], F^{2}Tree [30], FaceBook's "four-post" [51], and Google's Jupiter [154].

Fat-tree [5] is a three-layer multi-rooted tree with core, aggregation and edge layers, as shown in Fig. 13. It is constructed only by commodity switches (all with 1 Gbps ports) to support the aggregate bandwidth for the tens of thousands of servers in DC's. Fat-tree is a folded Clos network [37], and its design may be motivated from [105]. A k-ary fat-tree topology consists of $(\frac{k}{2})^2$ k-port core switches, k pods, and $\frac{k^3}{4}$ servers. Each core switch interconnects a pod with a port (the ith port of a core switch is connected to pod i). In each pod, $\frac{k}{2}$ k-port switches in the aggregation layer interconnect $\frac{k}{2}$ ones in the edge layer to logically form a complete bipartite graph. The $\frac{k}{2}$ ports of the ith switch in the aggregation layer in any pod is connected to the ith $\frac{k}{2}$ core switches. Each switch in the edge layer is connected to $\frac{k}{2}$ servers. With enough core switches, the fat-tree can guarantee a 1:1 over-subscription to support nonblocking communication between servers and significantly improve the performance of DCN.

Fat-tree achieves an even-distribution traffic pattern by two-level prefix lookup routing table, which makes a high bisection bandwidth and spreads traffic as evenly as possible. Those

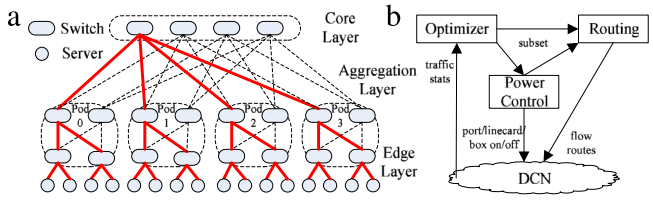


Fig. 14. (a) An example ElasticTree topology. (b) ElasticTree system diagram.

incoming packets that match the first-level prefix but not any second-level suffixes are directly routed to the corresponding output port. Those outgoing packets that match the first-level prefix and a second-level suffix would be matched the suffix, then routed to the outport port. Many production DC's have taken advantages of such design, for example, **Cisco's** Massively Scalable Data Center (MSDC) employs **fat-tree** on the Nexus 7000 series platform to make the multiple paths available between any two servers [32].

PortLand [136] is a scalable, efficient, fault tolerant layer 2 routing, forwarding and addressing protocol for DCN's, especially for multi-rooted topologies (e.g., fat-tree). PortLand treats a DCN with more than 100,000 servers as a single plug-and-play fabric, where each server includes several virtual machines (VMs). A logically centralized fabric manager supports Address Resolution Protocol (ARP) resolution, fault tolerance, and multicast. For achieving efficient forwarding, routing, and VM migration, Hierarchical Pseudo MAC (PMAC) addresses are allocated to servers (a unique address for each server).

ElasticTree [75] is an energy-saving system for DCN's. DCN's have the full capacity to deal with peak workloads, whereas at most of the time their workloads are relatively low, and numerous network devices and links are underused or idle. ElasticTree minimizes the network subnet size to appropriately support current traffic patterns by directly turning down the switches and links that are not required now as much as possible. A simple ElasticTree topology is shown in Fig. 14(a). In contrast to a full fat-tree with all active 20 switches and 48 links, the ElasticTree only turns 13 switches and 28 links on (a subtree highlighted in bold solid lines), saving the network power by 38%. However, simply turning on/off switches and links might reduce the performance and reliability of DCN's in a rapidly changing environment. ElasticTree system in Fig. 14(b) is composed of optimizer, routing, and power control modules.

Diamond [160] is an improved fat-tree architecture with only core and edge k-port switches. A Diamond network is divided into two parts by a cutting line, as shown in Fig. 15. In each part, $\frac{k}{2}$ core switches connect to $\frac{k^2}{2}$ edge switches; in each pod, k edge switches connect to servers directly (a fat-tree pod contains $\frac{k}{2}$ aggregation and $\frac{k}{2}$ edge switches). Diamond reduces the average path length by 10% than that of a fat-tree while supporting the same number of servers.

VL2. Greenberg et al. [62] proposed VL2 (Virtual Layer Two Networking), which is also a three-layer folded Clos network. The embryo of VL2 is **Monsoon** [64]. In contrast to Fat-tree, VL2 interconnects D_I-port intermediate switches, D_A-port Aggregate switches, and Top-of-Rack (ToR) switches to support $20 \cdot (\frac{D_I D_A}{4})$ servers, as shown in Fig. 16. The three-layer core network can be regarded as a huge layer-2 switch. The $\frac{D_A}{2}$ intermediate switches interconnect the D_I aggregate switches to logically form a complete bipartite graph. Each ToR switch is connected to 2 aggregate switches and 20 servers. VL2 also has a 1:1 over-subscription guarantee. The cabling complexity of VL2 is lower than that of fat-tree due to its high-capacity links, while the routing cost of high-level switches are more higher than that of fat-tree. VL2 uses a

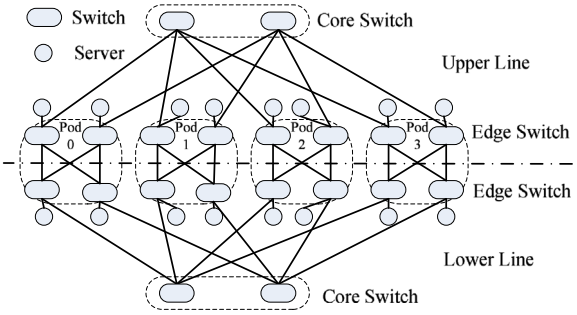


Fig. 15. A Diamond topology ($k = 4$).

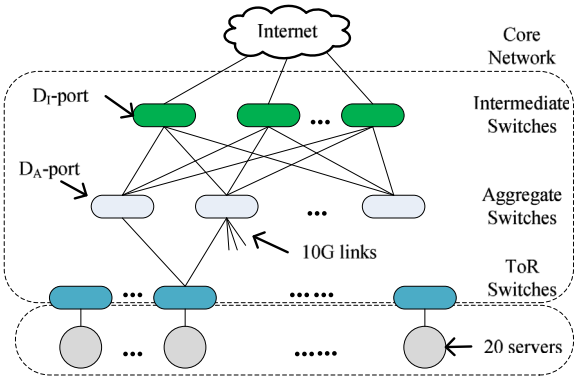


Fig. 16. An example VL2 network architecture.

layer-3 routing fabric to implement a virtual layer-2 network. It employs Valiant Load Balancing (VLB) to relieve the load unbalance in DCN's, which treats switches and servers as a whole resource pool and dynamically assigns IP addresses and workloads to servers.

GRIN, a simple and cheap improvement to existing DCN's (e.g., VL2, multihomed topology), connects servers in the same rack with free ports to form "neighbours" to maximize the bandwidth for each server [3]. **Subways** is another cheap solution to connect servers to the neighboring ToR switches. It achieves decreased congestion, improved load balancing, and better fault tolerance [124].

Redundant fat-tree [68] reduces the cost of nonblocking multirate multicast DCN's. A k-redundant fat-tree means every server has other $k - 1$ redundant servers, which must be connected with different ToR switches. By theoretical analysis, the sufficient condition on the number of core switches for nonblocking multicast communication can be significantly reduced when the fat-tree DCN is k-redundant.

Aspen trees [167] are a set of modified multi-rooted trees to balance the fault tolerance, scalability and cost in DCN's. The main goal is greatly decreasing the re-convergence time when link failures occur in a multi-rooted fat-tree. Aspen trees achieve fault tolerance by adding redundant links between the adjacent levels of the fat-tree. An n-level, k-port aspen tree is a set of n-level modified fat-trees consisting of k-port switches and servers with Fault Tolerant Vector (FTV). A Level i (denoted as L_i) pod means the maximal set of L_i switches connected to the same set of L_{i-1} pods, and an L_1 pod only has an L_1 switch. For instance, FTV $= \langle 1, 0, 0\rangle$ means that an aspen tree has 1-fault tolerance at L_4, i.e., there are 2 links between pairwise L_4 and L_3 switches. A 4-level, 4-port aspen tree with FTV $= \langle 1, 0, 0\rangle$ is shown in Fig. 17. When any of the two dashed links fails, a robust routing mechanism ensures that the switch s can reroute packets to other links within a short time. Compared with a fat-tree in the same scale, aspen tree only supports half of the servers because of the redundant links. If the same number of servers need to be supported, more switches should be involved.

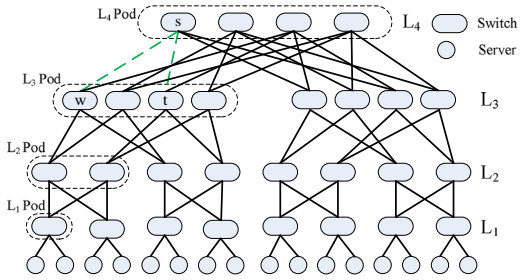


Fig. 17. A 4-level, 4-port aspen tree with FTV = ⟨1, 0, 0⟩.

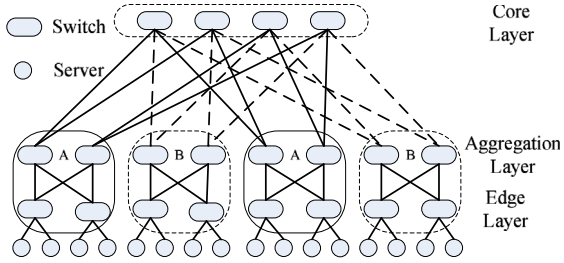


Fig. 18. An example AB FatTree topology ($k = 4$).

F10 [119] is a fault-tolerant engineered system for addressing the failures in DCN's, which consists of AB FatTree, failover and load balancing protocols and a failure detector. The core of F10, AB FatTree has many favorable properties similar as fat-tree, while has much better recovery properties than fat-tree by introducing a limited amount of asymmetry. An example AB FatTree in Fig. 18 has two types of subtree pods (called type A and B) that are wired to their parents (core switches) in two different ways (solid lines for type A and dash lines for type B). The asymmetry existing between core and aggregation layers provides a benefit for failure recovery.

Based on AB-FatTree, a series of failover protocols are designed to cascade and complement each other. F10 can almost instantaneously achieve local rerouting and load balancing, even though multiple failures occur. The simulation results show that following network link and switch failures, F10 has less than $\frac{1}{7}$th the congestion packet loss of PortLand for UDP traffic. A MapReduce trace-driven evaluation shows that F10 yields a median application-level 30% speedup than PortLand due to lower packet loss.

F^{2}Tree [30] is a fault-tolerant solution, which can significantly reduce the failure recovery time in multi-rooted tree-like DCN's. F^{2}Tree only rewires a small amount of links to improve path redundancy, and changes a few switch configurations to reroute locally. F^{2}Tree connects switches in the same pod of the aggregation or core layer to from a ring, which could increase immediate backup links for a certain link to ensure the packet forwarding when a failure occurs. The experimental results show that F^{2}Tree can significantly reduce the failure recovery time by 78% compared to Fat-tree.

FaceBook's "four-post" architecture [51] consists of "Fat Cat" aggregation switches (FC), cluster switches (C), rack switches (R) and servers in racks, as shown in Fig. 19. Compared to Fat-tree, a 160G protection ring (10G × 16) is connected to four FC switches and an 80G protection ring (10G × 8) is connected to four C switches in each cluster. The four-post architecture greatly eliminates service outages caused by network failures through its additional connections. The FC switch tier reduces traffic through the expensive links between the clusters. However, the very large and costly modular C and FC switches restrict scalability, which become potential bottlenecks. Facebook has considered many alternative architectures (e.g., 3D Torus or Fat-tree) to solve the specific networking challenges.

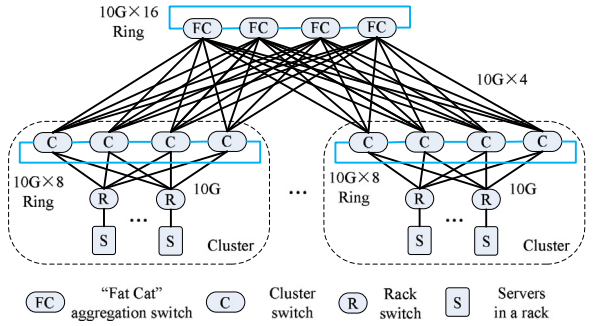


Fig. 19. Facebook's "four-post" DCN architecture.

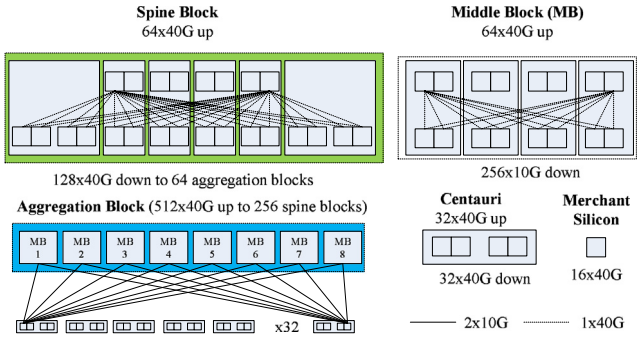


Fig. 20. The building blocks in the Jupiter topology.

Google's Jupiter. Singh et al. [154] introduced Google's five generations of DCN's based on Clos topology in the last decade. The newest generation Jupiter, a 40G datacenter-scale fabric equipped dense 40G capable merchant silicon. The building blocks in the Jupiter topology are shown in Fig. 20. A Centauri switch is employed as a ToR switch (including 4 switch chips). Four Centauri composed a Middle Block (MB) for use in the aggregation block. The logical topology of an MB is a two-stage blocking network. Each ToR chip connects to 8 MBs with 2 × 10G links to form an aggregation block. Six Centauris are used to build a spine block. There are 256 spine blocks and 64 aggregation blocks in Jupiter.

There are several regular but **Non-Tree** switch-centric DCN's. **MatrixDCN**, a matrix-like approximate non-blocking network, includes row switches, column switches and access switches [159]. The row and column switches connect the access switches in each row and column to form a matrix-like structure. For example, a 2 × 3 MatrixDCN has 2 row switches, 3 column switches and 6 access switches. A MatrixDCN is easy to expend or shrink the scale with similar switch/server ratio of Fat-tree. **Hypernetworks**, a novel method of constructing large switch-centric DCN's using fixed port number switches, first constructs large direct hypergraphs based on hypergraph theory and transversal block design theory, and then converts direct hypergraphs into indirect hypergraphs [141]. Compared to Fat-tree, hypernetworks could significantly support more servers using the same number of switches.

Discussion. Tree-like switch-centric architectures have balanced traffic loads, robust fault-tolerance, and multi-routing capabilities. However, they still have several disadvantages. First, the three or more layers switches increase the cabling complexity and constrain the network scalability. Second, the security and fault tolerance of commodity switches are poor compared to high-level switches. Third, the centralized manager could severely affect DCN performance when it was down due to the bursty traffic.

4.1.2. Flat switch-centric architecture

Flat switch-centric architectures flatten the three or more switch layers down to two or only a single switch layer, which

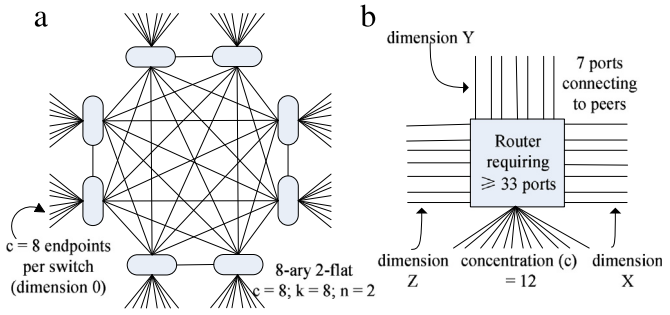


Fig. 21. (a) The logical illustration of an 8-ary 2-flat FBFLY. (b) A 33-port router to implement an 8-ary 4-flat with a concentration of 12 (oversubscription of 3:2).

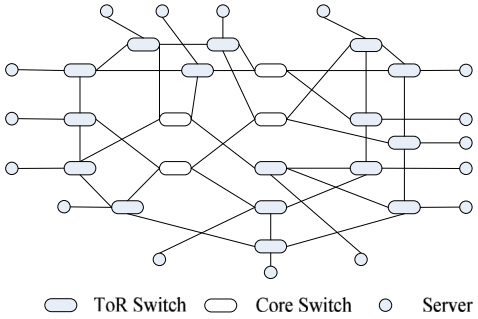


Fig. 22. A Jellyfish with 16 servers and 20 4-port switches.

simplify the management and maintenance of DCN's. Typical architectures include FBFLY [1], FlatNet [115], and C-FBFLY [39].

FBFLY. Abts et al. [1] used the k-ary n-flat FBFLY to build energy proportional DCN's, which was inspired from flattened butterfly (networks based on high-radix switches) [98,99,4]. Energy proportional means the power consumption is more proportional to the traffic amount in DCN's. For example, the links with a maximum bandwidth of 40 Gbps can be detuned to 20, 10, 5, or 2.5 Gbps in different traffic scenarios. FBFLY is a multidimensional directed network, similar to a torus (k-ary n-cube). Each high-radix switch ($\geqslant$64 ports) interconnects servers and other switches to form a generalized multidimensional hypercube. A k-ary n-flat FBFLY is derived from a k-ary n-fly conventional butterfly. The number of supporting servers is $N = k^n$ in both networks. The number of switches is nk^{n-1} with port number $2k$ in the conventional butterfly, and is $\frac{N}{k} = k^{n-1}$ with port number $n(k-1)+1$ in the FBFLY. The dimension of FBFLY is $n-1$. Fig. 21(a) shows an 8-ary 2-flat FBFLY with 15-port switches. Although it is similar to a generalized hypercube, FBFLY is more scalable and can save energy by modestly increasing the level of oversubscription. A 33-port switch with a concentration (number of servers, denoted as c) of 12 (changing from 8 to 12) in an 8-ary 2-flat FBFLY is depicted in Fig. 21(b). The size of the FBFLY can scale to $ck^{n-1} = 12 \times 8^{(4-1)} = 6144$ from the original size of $8^4 = 4096$. The level of oversubscription is moderately raised form 1:1 (8:7) to 3:2 (12:7).

FlatNet [115] is a scalable 2-layer architecture. The first layer contains an n-port switch and n servers, and the second layer consists of n^2 1-layer FlatNet with $2n^2$ switches and n^3 servers. Given an equal sized servers supported, the cost of a 2-layer FlatNet on number of links and switches are roughly 2/3 and 2/5 that of a 3-layer Fat-tree, while still offering comparable performance. FlatNet is also fault-tolerant and load-balanced due to its 2-layer structure and the effective routing protocols.

Colored-FBFLY (C-FBFLY) [39] is an improved optical architecture based FBFLY, which reduced the cabling complexity by an order of magnitude without increasing the control plane complexity. It transforms a full mesh with k long inter-rack cables in each dimension of the k-ary n-flat FBFLY into a "pseudo"-mesh with just k shorter cables. Specifically, C-FBFLY first replaces the grey transceivers in the switches in each dimension of FBFLY with dense wavelength division multiplexing (DWDM, or colored) transceivers, then connects all colored transceivers to an optical arrayed wave-guide grating router (AWGR) through a layer of multiplexers and demultiplexers on each end, which results in an optical star network with the AWGR in the center.

Discussion. A flat architecture with two layers or less switches is a feasible way to reduce network delay. Compared with the folded Clos network, FBFLY has the same number of servers and bisection bandwidth with approximately half the switches, less cables and 64% power consumption. However, the expensive cost of high-radix switches, the problem of single-point failure, and increased control plane complexity are the drawbacks. C-FBFLY improved the cabling complexity by adding optical devices.

4.1.3. Unstructured switch-centric architectures

Unstructured switch-centric architectures are irregular or asymmetric architectures that are feasible for DCN's. Solutions for addressing, routing, and load balancing were presented in arbitrary networks in recent years [97,162,133]. Typical architectures include Scafida [69], Small-World [150], Jellyfish [156], and REWIRE [41].

Scafida, a scale-free network inspired architecture for DCN's, reduces the average path lengths compared to other topologies with the same server numbers [69]. **Small-World** is an unorthodox topology for DCN's, which replaces several links with random links in a ring, 2D torus, or 3D hexagon torus, while limiting the degree of each node to 6 [150]. The two architectures both employ random links. However, they require the correlation among links, which is unknown when the networks expand.

Jellyfish [156] is an architecture based on random graphs, which can incrementally expand the size of DCN's. In contrast to Fat-tree, ToR switches with the same or different port counts in Jellyfish logically form a random graph to achieve a flexible network size. A simple approach to produce random graphs is described as follows. First, non-neighbor pairs of ToR switches with idle ports are randomly selected and connected with a link. Then the same operation is repeated until no new links could be added. For example, a switch with more than two free ports denoted as (p_1, p_2) (including the scenario that a new switch is added to the network) is inserted into an existing link (x, y), then new links (p_1, x), (p_2, y) are added. A Jellyfish with 20 4-port switches and 16 servers is shown in Fig. 22.

Compared to Fat-tree, Jellyfish can support 27% more servers at full bandwidth with the same switching equipment when the number of servers is lower than 900, and the advantage increases with the port count of switches. The average path length in Jellyfish is shorter than Fat-tree, and the diameter is at least the same with Fat-tree.

REWIRE optimally rewires the links existing in a given tree-like DCN with heterogeneous switches by Simulate Anneal Arithmetic [41], which proved that an unstructured topology also could effectively support DCN's. REWIRE maximizes the bisection bandwidth and minimizes the end-to-end latency based on satisfying user-defined constraints and properly modeling the cost of DCN's. The evaluation results show that REWIRE significantly outperforms previous proposals. However, when new switches are adding, the scalability of REWIRE is still under investigation.

Discussion. Unstructured architectures with random links have been proven to provide the low latency and high bandwidth for DCN's. However, the random links could greatly increase the complexity of cabling and routing when physically constructing a large-scale DCN.

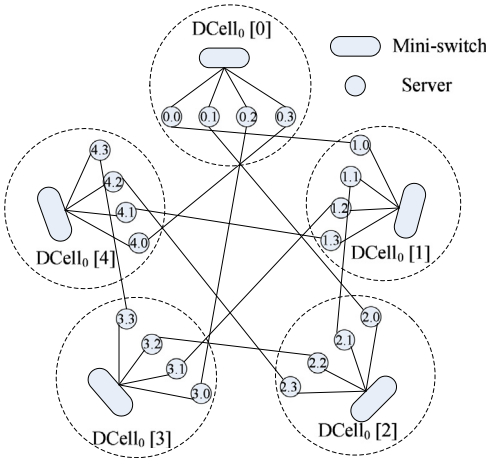


Fig. 23. A $DCell_1$ network architecture with $n = 4$.

4.2. Server-centric architectures

In server-centric architectures, servers are responsible for networking and routing, whereas commodity switches without modification are used only for forwarding packets. Servers are generally more programmable than switches to achieve more effective routing schemes, such like ServerSwitch [103]. Server-centric architectures are usually multi-level recursively defined structures, where a high-level structure consists of several low-level structures connected in a well-defined manner. In this subsection, we survey server-centric architectures designed for mega DC's and MDC's.

4.2.1. Server-centric architectures for mega DC's

Typical server-centric architectures for mega DC's include DCell [67], FiConn [107], DPillar [104], MCube [171], HCN and BCN [65], and SWCube and SWKautz [111], which are described as follows.

DCell [67] is a recursively defined architecture constructed by mini-switches and servers with multiple Network Interface Cards (NICs), which effectively deals with a sharp increase in servers of DCN's. A high-level DCell is logically a complete graph constructed by low-level DCell's. In a k-level $DCell_k$, g_k is the number of $DCell_{k-1}$, and t_k is the total number of servers. The recursive formulae of g_k and t_k are: $g_k = t_{k-1} + 1$, $t_k = g_k \times t_{k-1}$ ($k \geqslant 1$). Initially, $DCell_0$ contains an n-port switch and n servers ($g_0 = 1$ and $t_0 = n$). A $DCell_1$ consists of 5 $DCell_0$ with $n = 4$, as shown in Fig. 23. DCell is easy to scale. For example, a $DCell_3$ can support up to 3,263,442 servers ($k = 3$ and $n = 6$). However, the recursively defined manner leads to high cabling complexity as k increases.

DCell employs a near-optimal, distributed routing protocol, DCell Fault-tolerant Routing protocol (DFR), including DCellRouting and DCellBroadcast. Specifically, DFR handles link, server, and rack failures by three techniques of local-reroute, local link-state, and jump-up, respectively.

FiConn [107] is a recursively defined and low-cost interconnection architecture constructed by mini-switches and dual-port servers (active/backup ports). A server uses the active port to connect with a mini-switch, and the backup port for expansion. Similar to DCell, a high-level FiConn is a complete graph logically built by low-level FiConn's. Initially, a 0-level $FiConn_0$ contains an n-port switch and n servers. The active port (called $level_0$ port) directly connects to the switch with a $level_0$ link. Half of the backup ports of a $FiConn_0$ are reserved to build $level_1$ links with other $FiConn_0$'s, while another half are reserved to build higher-level links. Generally, a $FiConn_k$ consists of $(\frac{c}{2} + 1)$ $FiConn_{k-1}$'s, where c is the number of backup ports in a $FiConn_{k-1}$, and the number of servers is

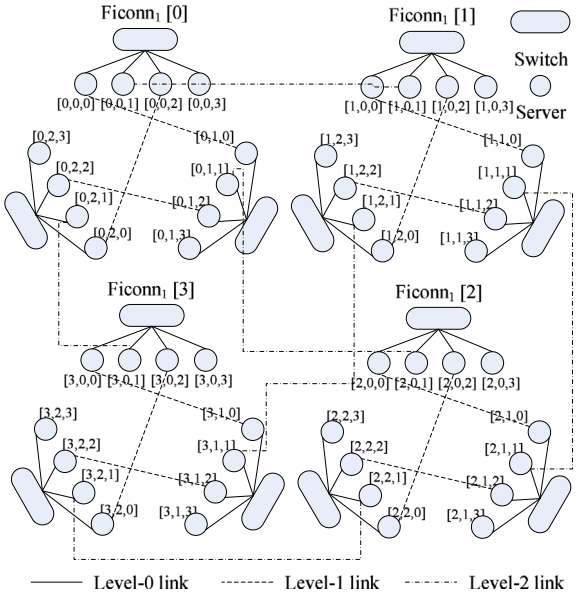


Fig. 24. A $FiConn_2$ network architecture with $n = 4$.

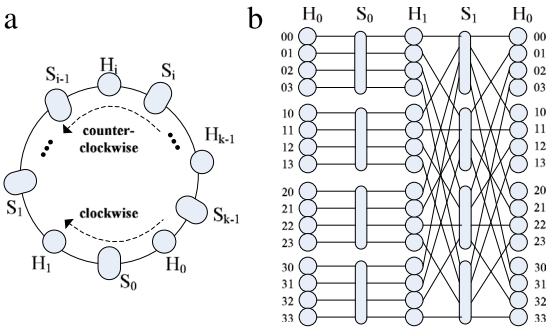


Fig. 25. DPillar. (a) The vertical view. (b) The 2D view.

$N_k = N_{k-1}(\frac{N_{k-1}}{2^k} + 1)$, $k \geqslant 1$. A 2-level $FiConn_2$ ($n = 4$) is shown in Fig. 24. If $n = 48$, it can support up to 361,200 servers.

FiConn employs Traffic-Aware Routing (TAR), which is a greedy approach to set up the traffic-aware path hop-by-hop on each intermediate server. In TAR, each server balances the traffic volume between its two outgoing links. The source server always selects the outgoing link with higher available bandwidth to forward the traffic.

DPillar [104] is a server-centric architecture constructed only by dual-port servers and low-cost layer-2 switches, which is scalable to support any number of servers. Similar to multistage interconnection networks, a DPillar(n, k) consists of k n-port switch columns and k server columns. The $2k$ columns of servers and switches logically form the cylindrical surface of a pillar in the 3D view, and the k server columns and k switch columns are alternately placed along a cycle in a vertical view, as shown in Fig. 25(a), which are denoted as $H_0 \sim H_{k-1}$ and $S_0 \sim S_{k-1}$, respectively.

Each server column involves $(\frac{n}{2})^k$ servers, and each switch column contains $(\frac{n}{2})^{k-1}$ switches. Thus, a DPillar(n, k) contains $k(\frac{n}{2})^k$ servers and $k(\frac{n}{2})^{k-1}$ switches. For instance, a DPillar(8, 2) network in 2D view is shown in Fig. 25(b), where 4 servers in column H_0 (labeled 00, 01, 02, 03) and 4 servers in column H_1 with the same labels form a group. If the 0th digit is removed, the rest labels are all 0, and the 8 servers are all connected to the switch labeled 0 in column S_0. A simple and efficient routing is sufficient for the symmetric DPillar architecture. The packet forwarding process in DPillar has a helix phase and a ring phase. In the helix phase, the packet is forwarded from the source to

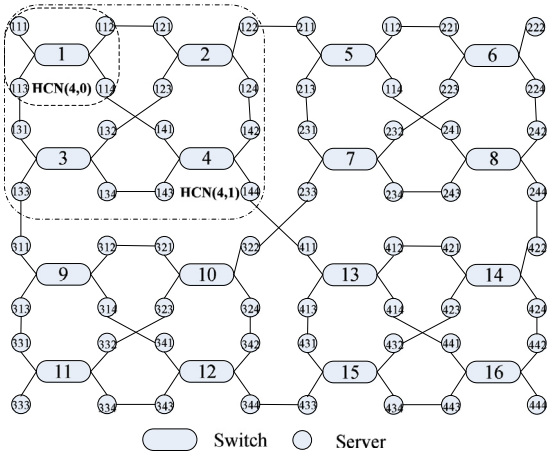


Fig. 26. An example $HCN(n, h)$ with $n = 4$ and $h = 2$.

an intermediate server whose label is the same as that of the destination. In the ring phase, the packet is forwarded from this intermediate server to the destination.

MCube, a low-cost, high-performance and fault-tolerant architecture, which is logically a modified cuboid with several unit cubes [171]. In an MCube, the vertices are replaced by mini-switches and a dual-port server is inserted into each edge. A 3D $n \times n \times n$ MCube contains $(n + 1)^3$ mini-switches and $3n(n + 1)^2$ servers with n servers in each dimension. For example, a $3 \times 3 \times 3$ MCube involves 64 mini-switches and 144 dual-port servers.

HCN and **BCN**. Guo et al. [65] proposed two symmetric and scalable network architectures, Hierarchical Irregular Compound Network (HCN) and Bidimensional Compound Network (BCN). Similar to FiConn, both HCN and BCN use dual-port servers and n-port commodity switches to construct DCN's.

HCN is a recursively defined architecture. Generally, an i-level HCN(n, i) $(i \geqslant 1)$ consists of n HCN$(n, i - 1)$'s, and reserves n servers for the interconnection at $(i+1)$-level. Therefore, HCN(n, i) can support up to n^{i+1} servers. For example, an HCN(4, 2) in Fig. 26 consisting of 4 HCN(4, 1)'s.

BCN is a bidimensional network architecture. In the first dimension, it is a multi-level HCN, and in the second dimension it is a 1-level regular compound graph. In BCN, servers are divided into two groups, master servers and slave servers, which are both directly connected with switches. The backup ports of master servers are used in the first dimension, whereas the second ports of slave servers are used in the second dimension. Let BCN$(\alpha, \beta, h, \gamma)$ denote a bidimensional BCN, where α is the number of master servers, β is the number of slave servers, h is the level of the BCN in the first dimension, and γ is the level of the BCN selected as the unit cluster in the second dimension. For instance, BCN(4, 4, 1, 0) is shown in Fig. 27, where master server, slave servers, and switches are denoted as air circles, solid circles, and rounded rectangles, respectively.

BCN achieves fault-tolerant routing by local reroute and remote reroute. Local reroute is used in BCN$(\alpha, \beta, h, \gamma)$ $(h < \gamma)$, where the source server immediately identifies all available candidate servers of the destination server, and picks up one such server as the relay. If the relay fails, it will select another server as a new relay to forward packets. Remote reroute is used in BCN$(\alpha, \beta, h, \gamma)$ $(h \geqslant \gamma)$. When a link failure occurs, if at least one slaver server and associated links are available, the packets are sent to another slave server connected with the same switch, then to the destination server.

SWCube and **SWKautz**, two low diameter, scalable, and fault-tolerant architectures built with commodity switches and dual-port servers [111]. SWCube logically is a modified generalized

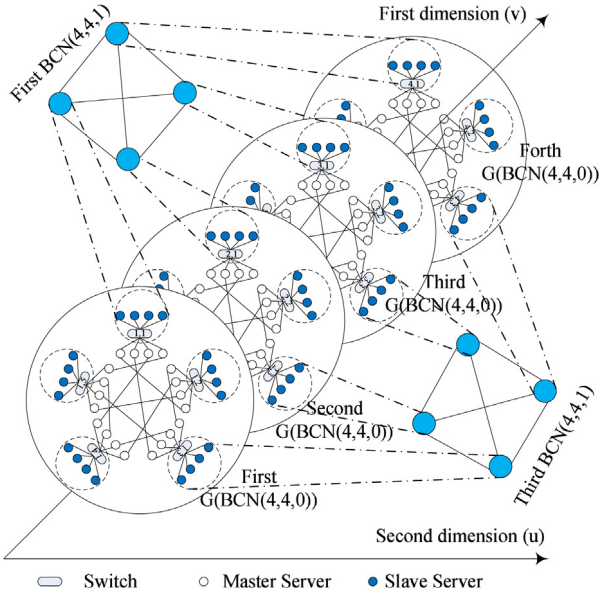


Fig. 27. An example of $BCN(4, 4, 1, 0)$.

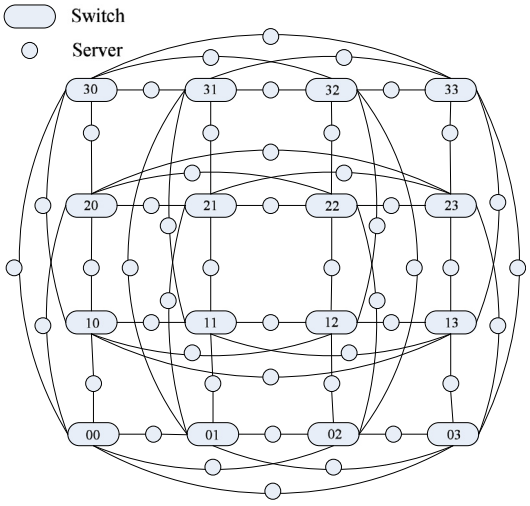


Fig. 28. A 2D SWCube with 4 switches in each dimension.

hypercube, where the vertices are replaced by switches, and a dual-port server is inserted into each edge. Let r_i denote the number of switches in the ith dimension of SWCube. A k-dimension SWCube contains $\frac{1}{2}(\prod_{i=1}^{k} r_i)(\sum_{i=1}^{k}(r_i-1))$ servers and $\prod_{i=1}^{k} r_i$ switches. The diameter of a SWCube(r, k) is $k + 1$. An example 2D SWCube is shown in Fig. 28.

Similar to SWCube, SWKautz displaces the vertices of Kautz directed graph [96] with n-port switches, and inserts a server into each directed link between switches. KA(r, k) denotes a k-dimensional Kautz directed graph with $r + 1$ symbols. SWKautz $(\frac{n}{2}, k)$ denotes a k-dimensional modified Kautz directed graph with $\frac{n}{2}+1$ symbols, which involves $(\frac{n}{2}+1)(\frac{n}{2})^k$ servers and $(\frac{n}{2}+1)(\frac{n}{2})^{k-1}$ switches. The diameter of SWKautz$(\frac{n}{2}, k)$ is also $k + 1$. A KA(2, 3) and an SWKautz(2, 3) are shown in Fig. 29.

Discussion. Server-centric architectures for mega DC's can significantly handle a shape increase in servers due to their recursive features. However, the cabling complexity would be extremely high as the level increases.

4.2.2. Server-centric architectures for modular DC's

As MDC's are increasing in popularity, the basic component of building DC's gradually changes from a rack to a shipping

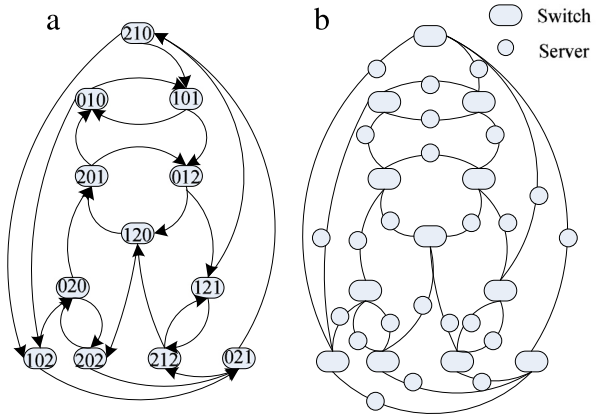


Fig. 29. (a) KA(2, 3). (b) SWKautz(2, 3).

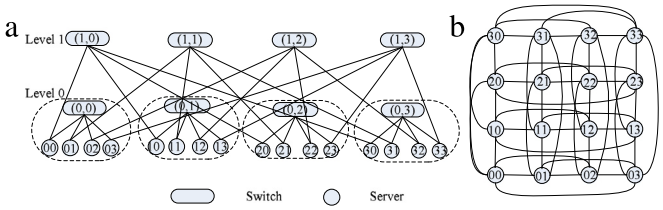


Fig. 30. (a) A BCube_1 with $n = 4$. (b) A generalized hypercube equivalent to BCube_1 with $n = 4$.

container. Typical architectures for MDC's include BCube [66], MDCube [176], PCube [83], uFix [108], Snowflake [123], HFN [46], Hyper-BCube [116], BCCC [109], ABCCC [113], CamCube [2] and NovaCube [169,170].

BCube [66] is a low latency, full bandwidth architecture specifically for MDC's, which supports different communication patterns (e.g., one-to-one, one-to-several, one-to-all and all-to-all). The cabling complexity is relatively low as just thousands of servers. BCube is a recursively defined architecture built by commodity mini-switches and multi-port servers. Initially, in a 0-level BCube_0, n servers are directly connected to an n-port mini-switch. Generally, BCube_k is constructed by n BCube_{k-1}'s and n^k n-port switches ($n \leqslant 8$), which contains n^{k+1} servers and $(k+1)n^k$ switches. An example BCube_1 with $n = 4$ is shown in Fig. 30(a), in which there are several fault-tolerant equal-cost links between any two servers. BCube is actually a generalized hypercube [21], as shown in Fig. 30(b).

In the load-balanced, fault tolerant BCube Source Routing, the source determines the routing path of a packet flow by sending probe packets over multiple parallel paths, and the destination returns a probe response. The source selects a best path with maximum available bandwidth and minimum end-to-end delay to forward the flow. It periodically makes a path selection to adjust the path for network failures.

MDCube [176], Modularized Data Center Cube network, interconnects BCube-based containers with optical fibers to construct mega DC's. MDCube achieves the inter-container architecture by the high-speed up-link interfaces of the commodity switches in BCube-based containers, greatly reducing the cabling complexity. It is logically a generalized hypercube, in which a BCube-based container is a vertex. The number of containers in an m–d MDCube is the product of the containers in each dimension ($m = 1, 2, \ldots$). For instance, if BCube_1 uses 4-port switches, then a 1-d MDCube supports 5 containers (20 servers), and 2-d MDCube supports 9 containers (36 servers), as shown in Fig. 31(a) and (b). Generally, if a BCube_1 uses 48-port switches, then a 1-d MDCube can support up to 97 containers (222,488 servers), and a 2-d MDCube can contain up to 2401 containers (5,531,904 servers).

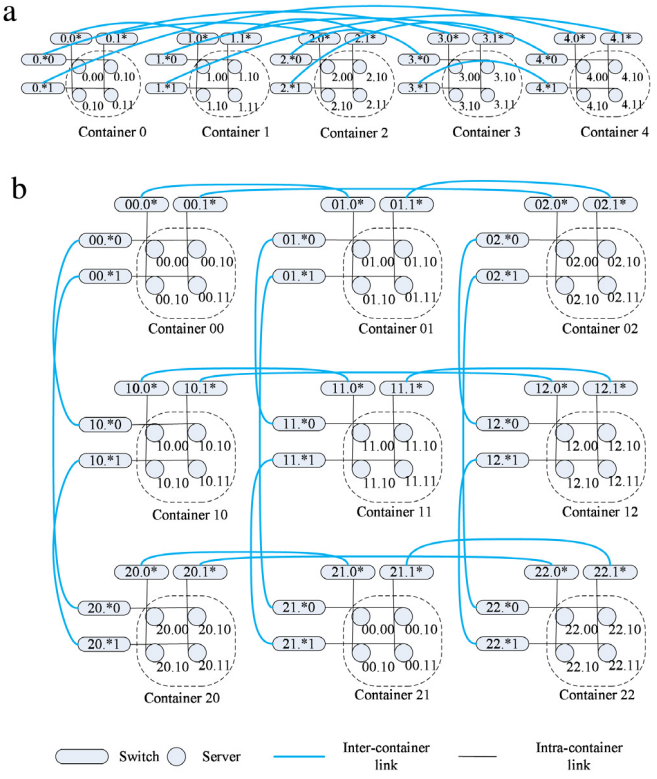


Fig. 31. (a) 1-d MDCube. (b) 2-d MDCube.

MDCubeRouting is used to check the tuples of the container ID to reach the destination container, where the order is determined by a permutation. Then BCubeRouting handles the routing between servers and switches within a container. However, MDCubeRouting is not load balanced or fault-tolerant. Thus, a Detour Routing for load balance initiates the routing by a random, container-level jump to a neighbor container, then uses MDCubeRouting to adjust the first random jump at the final step.

PCube [83] is an elastic power efficiency DCN. Similar to ElasticTree, PCube dynamically shuts down several switches in the hypercube-like DCN's (e.g., BCube and MDCube) for saving energy to satisfy different traffic demands.

uFix [108] is a network architecture to construct mega DC's by interconnecting heterogeneous containers (such as Fat-tree, FiConn, and BCube), where each server in containers must reserve several available NIC ports to directly interconnect servers of other containers to reduce rewiring cost. Compared to other well-defined hierarchical topologies, uFix adopts a natural and smooth network-extension mode. A 1-level uFix is shown in Fig. 32. Similar to MDCube, the intra-container routing is controlled by Fat-tree, FiConn, or BCube, while the inter-container routing is determined by the uFix proxy table.

Snowflake [123], a recursively defined scalable architecture, is inspired by the Koch snowflake, which expands in the Koch snowflake way [100]. For instance, a Snow_0 with $n = 3$ (n is the port number of switches) is shown in Fig. 33(a), where three virtual links denotes as dotted lines. A Cell is with only two virtual links (similar to the extend mode in Koch snowflake), as shown in Fig. 33(b). A Snow_1 is shown in Fig. 33(c), where three virtual links in Snow_0 are replaced by three Cells. Consequently, there are 6 new real links between switches in Cells and servers in Snow_1. Generally, k-level Snow_k ($n \geqslant 2$) is constructed by replacing virtual and real links in Snow_{k-1} with Cells. The number of servers in Snow_k is $n(n+1)^k$, where $n \in [3, 8]$. The ratio of severs to switches in BCube_k is $n : (k+1)$, whereas it is always $n : 1$ in Snow_k.

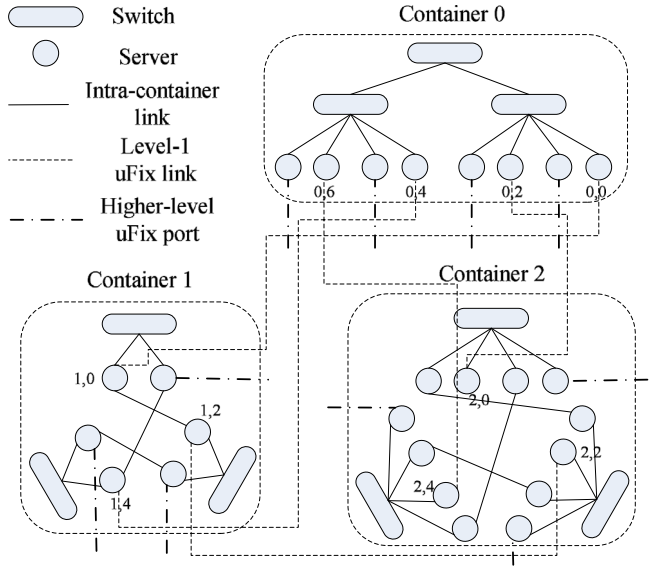


Fig. 32. A 1-level uFix network architecture.

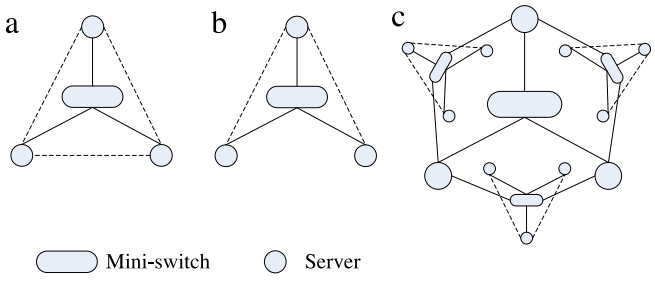


Fig. 33. (a) A $Snow_0$. (b) A Cell. (c) A $Snow_1$.

HFN is a Hyper-Fat-tree architecture designed for MapReduce applications [46]. Similar to BCube, HFN is a recursively defined architecture, where several low-level HFNs construct a high-level HFN. Compared to BCube, the lowest-level HFN is a Fat-tree-like redundant architecture. A 0-level $HFN_0(n, m)$ contains n master servers, n switches and $n \times m$ worker servers from the top down. The n masters servers and n switches logically construct a bipartite graph, while a switch connects with m worker servers. An HFN_i consists of n HFN_{i-1}'s and n^i switches ($i \geqslant 1$). The switches directly connect with master servers in HFN_0's.

Hyper-BCube is a cost-effective and scalable architecture for DCN's, which combines the advantages of DCell and BCube architectures while avoiding their limitations [116]. A 1-level Hyper-BCube is the same as a $DCell_0$. The k-level Hyper-BCube is composed by n^2 $(k-1)$-level Hyper-BCube, including kn^{2k-2} switches and n^{2k-1} servers. Hyper-BCube achieves a tradeoff between the excessive scalability of DCell and high cost of BCube. Given an equal sized servers supported, the cost of Hyper-BCube on number of links and switches is roughly 1/2 that of BCube, while still offering comparable performance.

BCCC, a BCube Connected Crossbars [109], is a recursively defined structure built upon BCube and Cube Connected Cycles (CCC) [140]. BCCC consists of element switches, crossbar switches, and dual-port servers. A BCCC(n, k) can be seen as a BCube(n, k) with each server replaced by $(k+1)$ servers connected to a crossbar switch. An element switch has n ports, while a crossbar switch has $(k+1)$ ports and is used to connect different elements. An element means n server connecting to an element switch (the same as a BCube(4, 0)). Each server in an element connects to the element switch by the first port, and the second port is used to connect with a crossbar switch for expansion. BCCC(4, 1) is shown in Fig. 34. BCCC has good scalability. When it expands, we only need to add new components without modifying the existing system. Advanced BCube Connected Crossbars (**ABCCC**) [113] is a more general BCCC, in which an element is a BCube(2, 1).

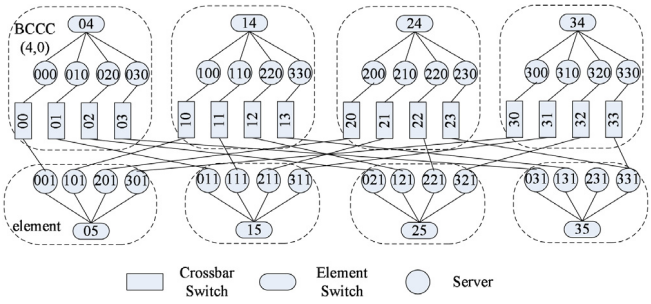


Fig. 34. BCCC(4, 1) is composed of 4 BCCC(4, 0)s along with 4 elements BCube(4, 0).

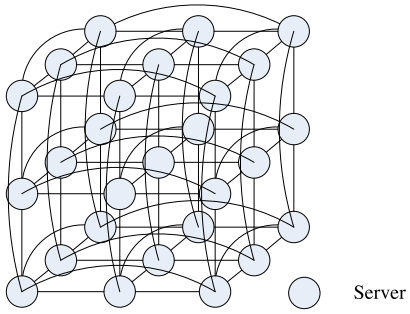


Fig. 35. A 3D Torus with 27 servers.

CamCube was designed to build an easier platform for distributed applications in MDC's [2]. CamCube is a k-ary 3-cube (also known as 3D torus, which is employed by a number of supercomputers on the TOP500 list), where each server directly connects with 6 neighbor servers, as shown in Fig. 35. In each dimension, k servers logically form a ring. CamCube can support up to k^3 servers. Each server is assigned an address (denoted as (x, y, z) coordinate), which indicates its relative offset from an arbitrary origin server in 3D torus, and is fixed during the lifetime of the server. **NovaCube**, an architecture based on regular Torus topology, which adds many jump-over links between servers [169,170]. A probabilistic oblivious routing algorithm (PORA) is carefully designed to enable NovaCube to achieve low average path length, low latency and high throughput.

Discussion. Server-centric architectures for MDC's employ servers as both computation units and packet-forwarding devices. These architectures could flexibly connect the servers and switches to different scales to meet the requirements of different scenarios. However, the cabling complexity is high due to recursively defined architectures.

4.3. Enhanced architectures

Optical devices and wireless antennas could provide the capacity enhancement for DCN's.

4.3.1. Optical architectures

In 2009, the US Department of Energy estimated that 75% of IT energy and facility total energy can be saved if all-optical networks are deployed in DC's [165]. A recent CIR report indicated that optical technology has become increasingly interesting and promising in DCN's [31]. First, an optical network is an on-demand connection-oriented network whose flexibility is higher than that of a traditional Ethernet network. Second, optical

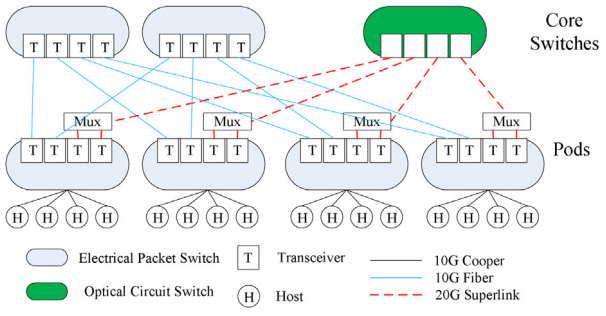


Fig. 36. Helios network architecture.

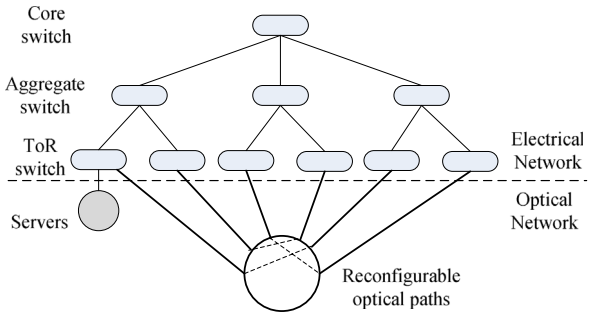


Fig. 37. HyPaC network architecture.

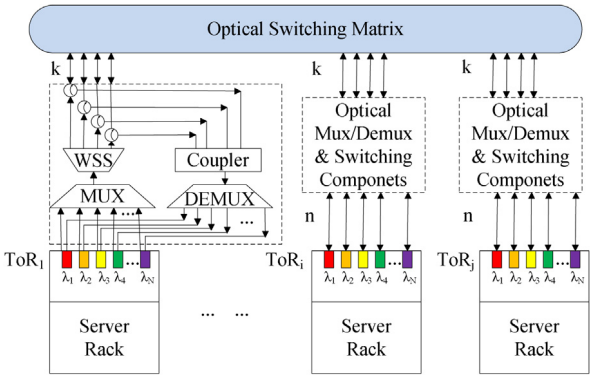


Fig. 38. OSA network architecture.

circuits can supply higher bandwidths over longer distances, and thus cost less power than copper cables. Third, optical switches with high-radix ports bring less heat than electrical ones and reduce cooling cost. Compared to 10GBase-T cabling (using copper interconnect), using optical interconnect will save roughly 150 million dollars in electrical power expenses in over 10 years [20]. Thus, optical switch-centric architectures would provide many benefits for DCN's. Typical optical architectures include Helios [50], HyPaC [168], REACToR [122], OSA [28], Mordia [138], Quartz [118], FireFly [71], Distributed Placement [177], and WaveCube [29].

Helios [50] is a 2-level multi-rooted tree architecture for MDC's to achieve low cost, low energy, and low complexity. Similar to Fat-tree, Helios is a tree-like topology constructed by core and pod switches, but it employs a hybrid packet and circuit switched architecture. In Helios, the core switches consist of traditional electrical packet switches and MEMS-based (Micro-Electro-Mechanical Systems) optical circuit switches, which are mutually complementary. The electrical packet switches are suitable for carrying out busty flows between servers in different pods, whereas the optical circuit switches could support low-fluctuation inter-pod flows. An example of Helios is shown in Fig. 36. Copper links are used to connect servers to pods, while fiber links and superlinks (using Wavelength Division Multiplexing, WDM) interconnect the pod and core switches. A core centralized topology manager first predicts the true inter-pod traffic demands according to monitored real-time communication patterns, and then computes a new topology and reconfigures circuits and uplinks to maximize throughput.

HyPaC [168] is a hybrid electrical/optical architecture, as shown in Fig. 37. HyPaC conserves the traditional three-layer tree-like topology as an electrical network, and connects all ToR switches to optical circuit switches to form an auxiliary optical network.

c-Through, a prototype system based on HyPaC, contains a control plane and a data plane. The control plane first estimates rack-to-rack traffic demands, then dynamically reconfigures circuits to accommodate the new demands. The data plane isolates the electrical and optical networks, and dynamically demultiplexes traffic from servers or ToR switches onto the circuit or packet path. In c-Through, when a circuit between two racks is available, the optical path has a higher priority than the electrical path.

REACToR is a hybrid packet/circuit ToR switch [122]. Using high-speed optical transceivers, the current packet-based 10 Gbps DCN could be upgraded to 100 Gbps. When rapid and bursty traffic changes occur at the server side, REACToR could react within hundreds of microseconds, which is a few orders of magnitude faster than previous hybrid solutions.

OSA [28,157] is a flexible optical switching architecture (OSA) for container-based DCN's, which is composed of MEMS-based optical circuit switches and ToR electrical switches. An OSA architecture is shown in Fig. 38, where electrical signals are sent from the rack servers and converted to optical signals through the optical transceivers in ToR switches. Wavelength Selective Switch (WSS) maps the optical signals to different ports according to different wavelengths. Optical Switching Matrix executes transmissions of the optical signals between different ports. OSA employs the optical circulator to support two-way communication in one circuit, which increases the utilization of high-priced optical ports. Under different traffic demands, OSA dynamically shifts the link capacities by reconfiguring optical devices at runtime.

Helios and HyPaC only provide one-hop high-capacity optical circuits, while OSA adopts a new multi-hop circuit switching through multiple cheaper optical circuits to support overall network connectivity for small flows and bursty communications. OSA also relieves the ToR switch hotspots, which performs better than previous hybrid architectures.

Mordia, an optical circuit switching prototype, where the switching time is at microsecond time scales [138]. A control plane based on a circuit scheduling method named Traffic Matrix Scheduling is used to achieve a microsecond end-to-end reconfiguration time.

Quartz, a low latency component for DCN's, is logically a complete mesh of optical switches, and physically a ring network [118], which can replace different parts of a hierarchical or random network. Quartzes could be substitutes for core switches to achieve lower switching delays, or replacements for aggregation and edge switches to reduce congestion-related delays due to cross-layer traffic.

Xiao et al. [177] presented a **distributed placement** with optical switches and racks in a given DCN. A network node is a component set including content and core switches, or aggregation switches and racks with ToR switches. Content and ToR switches are electrical, whereas core and aggregation switches are optical. Different component sets are connected by optical links and fibers to form a folded Clos network. Content switches classify traffic into external and internal traffic. The distributed placement of optical switches can reduce the power and cooling cost, the cabling complexity, and the external traffic overhead. However, it leads to additional transmission delay and internal traffic overhead. A

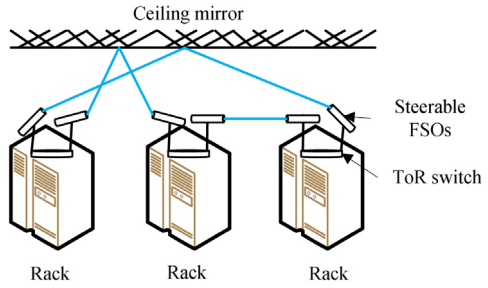


Fig. 39. High-level view of FireFly network.

heuristic algorithm for node distribution are proposed to minimize the total cost.

WaveCube, a scalable, fault-tolerant, and high-performing optical architecture for DCN's [29], which removes MEMS, a potential bottleneck, from traditional optical designs for scalability purpose. WaveCube is fault-tolerant since the single point failure is eliminated and every pairwise ToRs have multiple node-disjoint paths. WaveCube uses multipathing and dynamic bandwidth assignment to achieve high performance.

FireFly [71], a **wireless** architecture with free-space **optics** (FSO), can offer tens of Gbps data rate over long distances only using low transmission power without zero interference. The high-level view of FireFly network is shown in Fig. 39. FireFly is an inter-rack network scheme with only ToR switches, where all wireless links are reconfigurable. The servers in different racks communicate with each other by steerable FSO reflected with ceiling switchable mirrors or Galvo mirrors. FireFly can provide significant benefits such as low equipment cost and low cabling complexity. A prototype has been built to demonstrate the feasibility of FireFly.

Discussion. Optical switch-centric architectures are proved to be feasible in DCN's. However, these architectures still has several limitations. First, an approximate 10 ms latency of reconfiguration would affect latency-sensitive applications, such as online search. Second, the equipment such as electrical switches with optical transceivers are really expensive (a few hundred dollars per port). Third, the scale of an optical architecture is still under study. More detailed analyses and comparisons on the optical interconnects for DCN's could be referred to [92,91].

4.3.2. Wireless architectures

Wireless architectures employ wireless antennas in 60 GHz frequency band, in which the theoretical data transfer rate is up to 7 Gbps. Typical architectures include 60 GHz wireless technologies [142], Flyways [93,70], Wireless Fat-tree [166], 3D beamforming [181,182], Wireless crossbar [94,95], Completely wireless data center [149], Angora [183], Graphite [180], Spherical mesh [112], and 3D wireless ring [40].

Ramachandran et al. [142] first explored the possibility of using 60 **GHz wireless technologies** in DCN's, say, substituting wireless links operating in the 60 GHz frequency band for wired cables to reduce cabling complexity [88]. The authors categorized the patterns of wireless communication in DC's into Line-of-Sight (LOS) between racks, indirect Line-of-Sight with reflectors, and multi-hop Non-Line-of-Sight (NLOS), as shown in Fig. 40. Servers between two racks can choose LOS paths, indirect LOS paths with ceiling-mounted reflectors, or multi-hop compounded paths to communicate. Servers in the same rack can communicate along indirect LOS paths with rack-mounted reflectors.

Flyways. Kandula et al. [93] handled hotspots in oversubscribed DCN's by adding *flyways* (60 GHz wireless links) in an on-demand way, which resulted in a hybrid wired/wireless architecture. The simulation results showed that if flyways are placed appropriately,

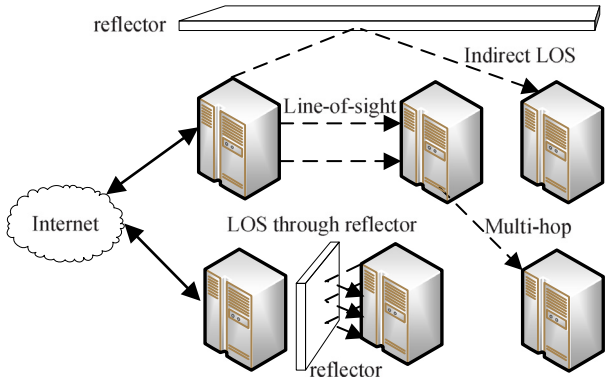
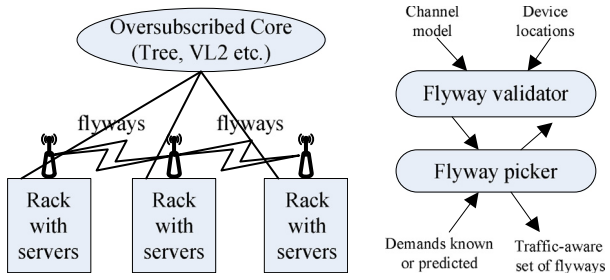


Fig. 40. A wireless data center with 60 GHz links.



(a) A flyways augmented network. (b) Flyway controller.

Fig. 41. A flyways augmented network and its controller.

the network performance will be improved by over 50%. Subsequently, Halperin et al. [70] further designed a *flyways system* based on 60 GHz wireless technologies to mitigate hotspots in oversubscribed DCN's such as a tree-like topology with an 1:2 oversubscription ratio. A flyway augmented network is shown in Fig. 41(a), whose backbone is the wired oversubscribed DCN. A 60 GHz wireless device with a steerable directional horn antenna is placed on top of a ToR switch, which makes several flyways work concurrently. For providing extra link capacity to alleviate hotspots, a centralized controller monitors traffic patterns in DCN's, then manages the beams of the 60 GHz wireless devices to build flyways between ToR switches, as shown in Fig. 41(b). Flyways could improve the performance of network-limited applications with predictable traffic workloads by 45%.

Wireless Fat-tree. Vardhan et al. [166] explored 60 GHz wireless technologies to build a **wireless Fat-tree** DCN. The authors designed a transceiver based on beamforming and beam steering technologies, which provides point to point LOS links between servers. Compared to the rack arrangement of two parallel rows, a hexagonal rack arrangement is more suitable for wireless DCN by theoretical analyses. Two node placement algorithms are designed to emulate wireless three-layer tree and Fat-tree architectures only with LOS links.

3D beamforming was presented to improve the transmission range and concurrent number of 60 GHz wireless links in DCN's [181,182], which sets up indirect LOS path by utilizing ceiling-mounted reflectors to interconnects the 60 GHz wireless devices placed on the ToRs that cannot directly communicate. A sender (TX) with a horn antenna transmits its signal toward some points on the ceiling-mounted reflector, which then bounces off the signal to the receiver (RX), as shown in Fig. 42. By this way, the sender and receiver could bypass obstacles to talk "directly" without multi-hop relays. 3D beamforming technology could extend link range while raising the number of parallel links.

Wireless crossbar. Katayama et al. [94,95] explored to build a robust **wireless crossbar** switch architecture with steered-beam

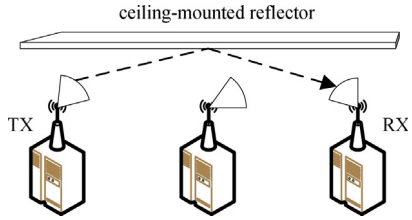


Fig. 42. 3D beamforming.

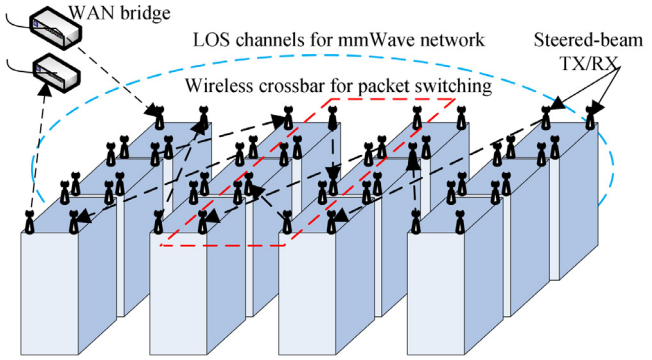


Fig. 43. The wireless crossbar packet switches is configured via LOS channels on top of the server racks.

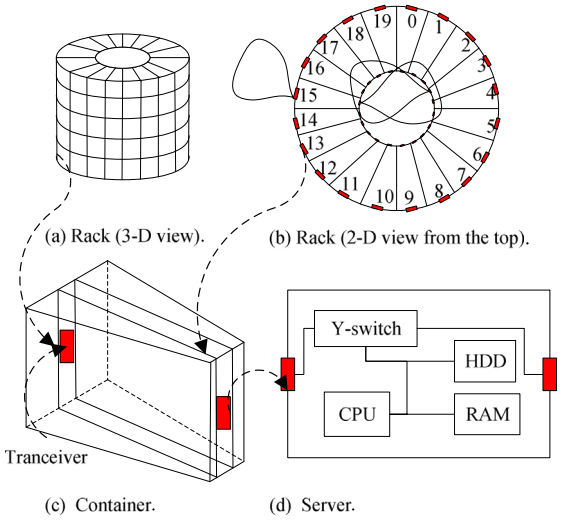


Fig. 44. Rack and server design of Wireless Data Center.

mmWave links. Compared to wired cabling, a hybrid approach was proposed, where cables are only used to interconnect within a rack or between racks in the same row. The wireless nodes with steered-beam transmitter and receiver (TX/RX) on top of two adjacent racks compose a wireless crossbar, as shown in Fig. 43. The authors use multiple unblocked wireless LOS channels to increase the bandwidth and decrease the interference of multiple rows, by which the racks can communicate directly with low latency. The wireless crossbar greatly reduce the cabling complexity, and server installation and reconfiguration costs.

Completely wireless data center. Shin et al. [149] proposed a fully wireless architecture based on 60 GHz wireless technologies for DCN's, which removes all wired links but power supply cables. A new cylindrical rack is designed as the basic unit of the fully wireless architecture, of which different views is shown in Fig. 44(a); (b). A rack consists of numerous servers with two transceivers in prism-shaped containers, as shown in Fig. 44(c); (d). A server could communicate with other servers in the same rack or in different racks owing to the cylinder design. The communication topology between servers is logically considered as a mesh of Cayley graphs, which supports intra-rack and inter-rank intensive interconnections, as shown in Fig. 45. Therefore, the completely wireless data center is also known as Cayley data center. The evaluation results showed that Cayley DCN could achieve higher aggregate bandwidth, lower latency, and lower cost than that of Fat-tree and conventional DCN's.

It is relatively easy to construct a Cayley DCN since there is no need to use enormous cables. The daily maintenance routines are placing and replacing the physical components. However, the Carley DCN still has some limitations. First, there is interference from different transmitting directions of transceivers due to features of 60 GHz wireless signals. Second, MAC layer contention derived from sharing the wireless channels could decrease the overall performance. Third, the multi-hop feature of the Carley DCN leads to a poor scalability.

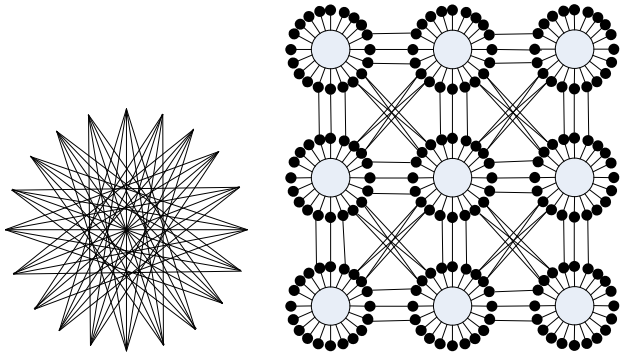
Angora [183], a low-latency and robust wireless architecture with 3D beamforming radios based on Kautz graphs, reduces the paths between any two racks. An example path from rack 012 to rack 213 in the Angora overlay is shown in Fig. 46(a). The bold solid arrows are 3D beamforming links. The corresponding path in Kautz graph ($d = 3, k = 3$) is shown in Fig. 46(b). Angora avoids link coordination by pre-tuned antenna direction under different traffic patterns. The properties of Kautz graphs make all racks within a small constant hops.

Spherical Mesh [112], a wireless architecture for DCN's, could greatly mitigate link blockage by putting antennas on top of racks at different heights, and decrease the network diameter by splitting the whole mesh into several equivalent units. A Spherical Mesh is shown in Fig. 47(a). Solid points with different shapes represent antennae at the different heights. In 3D view, the antennas are all located in a spherical surface, as shown in Fig. 47(b).

3D Wireless Ring. Cui et al. [40] proposed **Diamond**, a hybrid wired/wireless architecture for DCN's, which equipped radios on all servers. In Diamond, all links between servers are wireless, and links between a server and its ToR switch or between two ToR switches are wired. The low-cost scalable 3D Ring Reflection Spaces (RRSs) nest the streamlined wired herringbone to provide abundant concurrent wireless links by multi-reflection of radio signals over metal. A real 60 GHz-based testbed is built to prove the feasibility of Diamond. The top and side views of the 3D wireless ring in Diamond ($N = 3$ rings and $H = 3$ layers) are shown in Fig. 48(a); (b). The rings (i.e., several concentric regular polygons) are constructed by racks (vertices) and flat metal reflectors (edges) standing vertically to the ground. The layers are formed by the servers inside different racks at the same height. If a source server S attempts to communicate with a destination server D in different layers at different racks, it could achieve that by reflecting the radio signals over the metal reflections.

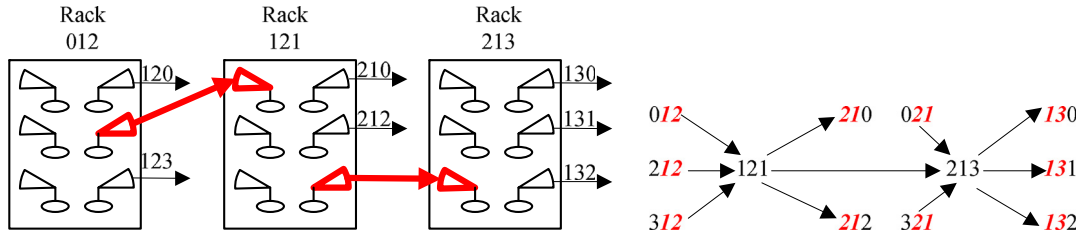
Han et al. [74] proposed the RUSH framework, which minimizes the network congestion in hybrid DCN's by jointly routing flows and scheduling wireless (directional) antennas. Though the scheduling problem is NP-hard, the RUSH algorithms offer guaranteed performance bounds. A survey was presented about wireless technologies for DCN's [172]. To our best knowledge, it is the first comprehensive survey on 60 GHz wireless technologies in DCN's.

Graphite [180], a flexible wireless architecture, properly solves the problem of link blockage by placing horn antennas in different layers, as shown in Fig. 49(a). By the propagation distance of 60 GHz wireless technologies, a server can communicate with as many other servers as possible in Graphite. The two-layer deployment of antennas in 2-D view from the top is similar to the molecular structure of graphite, as shown in Fig. 49(b), where the discs and



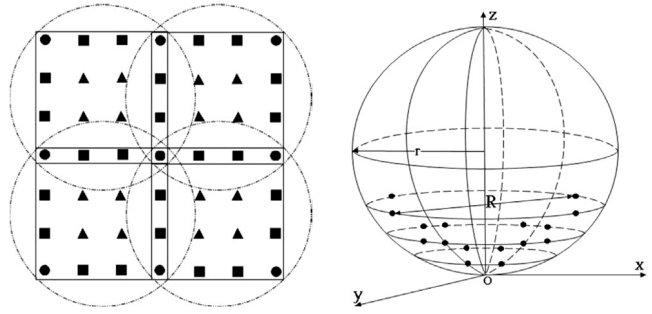
(a) Intra-rack. (b) Inter-rank.

Fig. 45. Cayley data center topology.



(a) The Angora overlay: an example path. (b) Two nodes in a Kautz graph ($d = 3$, $k = 3$).

Fig. 46. High-level view of Angora network.



(a) Spherical Mesh. (b) The 3D view of Spherical Mesh.

Fig. 47. A Spherical Mesh. (a) 2D view (b) 3D view.

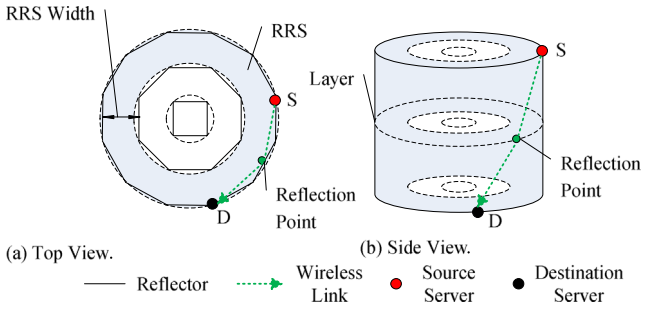


Fig. 48. Top and side views of the 3D wireless ring in Diamond ($N = 3$ rings and $H = 3$ layers).

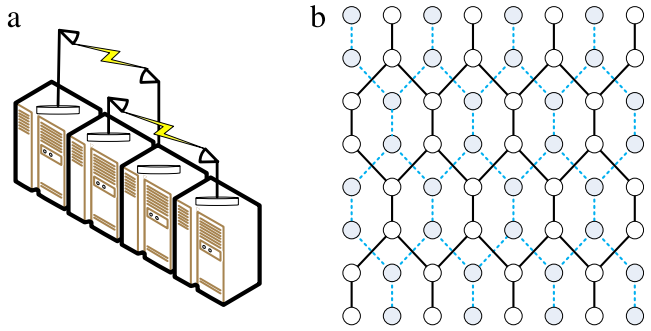


Fig. 49. (a) Place antennas in different layers. (b) The deployment of antennas in 2-D view from the top.

circles denote the antennas in different layers. Through theoretical analyses and simulations, Graphite was proved to be a feasible wireless architecture for DCN's.

Discussion. According to the literatures published in recent years, it is feasible and effective to use 60 GHz wireless technologies in DCN's. By taking advantage of multidimensional space, we could greatly increase the overall performance of DCN's and even build a completely wireless data center.

4.4. Architecture comparison and discussion

In this subsection, we will compare DCN architectures qualitatively and quantitatively according to three classifications, say, compare switch-centric and server-centric architectures, compare optical architectures, and then compare wireless architectures. We firstly provide some tags for each architecture,

Table 11
Qualitative comparison of switch-centric and server-centric architectures.

Architecture	Designers	Year	Scalability	Bisection bandwidth	Cabling complexity	Cost	Fault tolerance	Energy efficiency	Traffic control	Prototype
Fat-tree	UC San Diego	2008	Medium	High	Medium	Low	Medium	High	Centralized	⋆
VL2	Microsoft	2009	High	High	Medium	Medium	Medium	High	Centralized	⋆
FBFLY	Google	2010	Medium	High	High	High	Low	High	Centralized	
Aspen Trees	Microsoft, Google UC San Diego	2013	Medium	High	High	Medium	High	Medium	Centralized	
AB FatTree	Washington U.	2013	Medium	High	High	Low	High	High	Centralized	⋆
Diamond	BJTU	2014	Medium	High	Medium	Low	Medium	Medium	Centralized	
DCell	Microsoft, Tsinghua UC Los Angeles	2008	Quite high	Medium	Quite high	High	Medium	Low	Distributed	⋆
Ficonn	Microsoft, UC Los Angeles	2009	Quite high	Medium	High	High	Medium	Low	Distributed	
Dpillar	UMASS, NWPU	2010	Quite high	Medium	High	Medium	Medium	Medium	Centralized	
MCube	NEU	2010	Low	High	High	Medium	High	Low	Centralized	
HCN	HUST, NUDT	2011	High	Medium	High	High	Medium	Medium	Distributed	
BCN	HUST, NUDT	2011	High	Medium	Quite high	High	Medium	Medium	Distributed	
SWCube	Temple U.	2014	Medium	High	High	Medium	Medium	Medium	Centralized	
SWKautz	Temple U.	2014	Medium	High	High	Medium	Medium	Medium	Centralized	
BCube	Microsoft, Tsinghua, PKU, HUST, UCLA	2009	Low	High	High	Medium	Medium	Medium	Centralized	⋆
MDCube	Microsoft	2009	High	High	High	Quite high	Medium	Medium	Centralized	⋆
CamCube	Microsoft	2010	Low	High	High	High	High	Medium	Centralized	⋆
uFix	Tsinghua, Uni-goettingen	2011	High	High	High	Quite high	Medium	Medium	Centralized	⋆
Snowflake	USTC, AHNU	2011	Quite high	Medium	Quite high	High	Low	Low	Distributed	
HFN	NUDT, McGill U., SJTU	2012	Medium	High	High	Medium	High	Low	Centralized	

depicting their designers, "birth" years, and other information. Then we select several critical metrics to reflect their features and performance. For qualitative comparisons, we focus on the most typical features including the network scalability, bisection bandwidth, and traffic control, etc., and some special indices for optical and wireless architectures. While for quantitative comparisons, we rigorously define notations and parameters, and then provide precise mathematical formulas to compute the fundamental metrics for network evaluations. We also illustrate some representative architectures in detail for clarification. Finally, we discuss the virtualization, traffic management, power consumption, and many other issues and promising research directions related to the design and analysis of DCN architectures.

Note that, usually DCN architectures are personalized for specific application scenarios, different client requirements, or special types of hardware. Moreover, data center operators are generally silent to share the actual requirements of their applications, making it difficult to evaluate the practicality of any particular architecture [144]. Hence, it is hard to make a completely objective comparison for DCN architectures. Correspondingly, instead of giving a perhaps controversial assessment, we try our best to provide leveled marks for qualitative comparison and parameterized formulas for quantitative comparisons in general sense. Our comparisons does not mean that some architectures are always having relatively worse properties, especially when facing their original design scenarios.

4.4.1. Switch-centric and server-centric architectures

First, a **qualitative** comparison of major switch-centric and server-centric architectures was conducted in Table 11. In this table, we choose 8 metrics to evaluate the performance of each architecture, including *scalability*, *bisection bandwidth*, *cabling complexity*, *cost*, *fault tolerance*, *energy efficiency*, *traffic control*, and *prototype*. The *scalability* of DCN architectures is the capability of being expanded to accommodate an increasing amount of workloads, which is constrained by the port number of switches and the recursively defined ways of connection. If a network is segmented into two equal parts, *bisection bandwidth* is the bandwidth available between the two parts, which is a performance metric in the worst-case scenario. *Cabling complexity* is a measure of the number of long inter-switch cables required to construct a DCN. Long cables require planning and overhead cable trays, and is more difficult to install and maintain than short intra-rack cables or cables that cross between adjacent racks. The *cost* of DCN's is composed of the cost of power and all physical components (including switches, servers, storage, racks and cables), which is an important part of CAPEX and OPEX of DC's. *Fault tolerance* is the capability that enables a DCN to continue operating properly when failures occur among some of its components. *Energy efficiency*, also known as efficient energy use, is the goal to reduce the amount of energy required to provide cloud services. *Traffic control* is the process of managing, controlling or reducing the network traffic

to achieve low latency and low packet loss rate, which is generally divided into two modes, centralized control and distributed control. Whether the *prototype* of an architecture has been implemented shows the feasibility of the architecture used in a real DCN. Note that, in the qualitative comparison, the leveled marks consist of low, medium, high, and quite high. If an architecture has a prototype, the corresponding cell is filled with a ⋆.

We illustrate some representative architectures in detail for clarification. For example, **Fat-tree**, a switch-centric architecture, was proposed in 2008 by UC San Diego, of which the *scalability* is determined by the port-count of commodity switches. Fat-tree could provide 1:1 oversubscription ratio, full *bisection bandwidth*, and multiple equal-cost end-to-end paths between any two servers. **ElasticTree** has proved the feasibility of about 50% energy savings on Fat-tree using *energy efficiency* technology [75]. By increasing redundancy and asymmetry between different levels in the symmetric structure of Fat-tree, **Aspen Trees** and **AB Fat-tree (F10)** could achieve higher *fault tolerance* than that of Fat-tree. The *cost* of a Fat-tree is relatively lower than that of **VL2** due to low-end commodity switches, while the *cabling complexity* is higher than that of VL2 due to only 1 Gbps ports. Fat-tree adopts a **centralized** controller, and has been evaluated in a **testbed**. **Cisco** has used the Fat-tree architecture in their production data centers [32], which further demonstrated the feasibility of Fat-tree in real DCN's.

BCube, a server-centric architecture, was presented in 2009 by Microsoft. As designed for Modular Data Centers, the *scalability* of BCube is lower than that of **Fat-tree** and **DCell**, which is determined by the recursively defined ways of connection and the port number of mini-switches. BCube employs servers as *intermediate nodes* for routing and forwarding packets, which may limit the transition of servers to power-off mode for *energy efficiency*. Therefore, when the CPU utilization is high, the energy efficiency techniques are relatively difficult to be used in BCube. As multiple equal-cost paths between any pair servers in BCube, the *bisection bandwidth* is higher than that of **DCell**. The *cabling complexity* is higher than that of **Fat-tree** due to the features of the recursively defined architecture. BCube also employs a number of low-end servers with multiple NICs and commodity switches, so the *cost* is relatively medium. A *prototype* of BCube also has been implemented in a testbed.

Second, a **quantitative** comparison of the major switch-centric and server-centric architectures is also conducted, and the results are given in Table 13. In this table, we choose 4 critical metrics to evaluate the performance of each architecture [67], and the meanings of the 4 metrics are listed as follows.

- **Server Degree**: Server degree is the number of NICs connected with other switches by links. Small server degree means fewer links, lower cabling complexity, and lower deployment overhead.
- **Network Diameter**: Diameter is the longest of shortest distances between any pair of nodes in the same network, which means any node is reachable by all other nodes within "diameter" hops. A small diameter generally lead to efficient routing and fewer hops between any two servers in DCN's.
- **Bisection Bandwidth (BiW)**: Bisection Bandwidth (BiW) is the bandwidth available between two equal-sized partitions of a network in the worst-case scenario. A large BiW value mean better fault tolerance and better network capacity.
- **Bottleneck Degree (BoD)**: BoD is the maximum number of flows over a single link in an all-to-all communication scenario. A small BoD value means that the network traffic is dispersed over all the links to achieve better load balancing.

We adopt a unified symbol system to denote the metrics for ease of comparison. The symbols and their descriptions are given in Table 12, including N_e, N_w, n, h, k, m, f and so on. We use N_e and N_w to denote total supported servers and total used switches, respectively. N_e reflects the *scalability*, and the N_w/N_e means the *cost per server on switches*. We assume the port-counts of switches are the same as n. The tree architecture is logically a complete k-ary tree with height h. The level numbers of all recursively defined architectures are all k. We take m to denote the dimension number of multi-dimensional architectures, and employ f to show the duplicate connection count of Aspen Trees. **Note that**, in Table 13, the N_e in each row denotes the value of cell on column $\boldsymbol{N_e}$ in the same row. For example, in the "Fat-tree" row, the BiW is $\frac{N_e}{2}$, i.e., $\frac{n^3/4}{2}$.

Table 12
Symbol descriptions for switch-centric and server-centric architectures.

Symbol	Description
N_e, N_w	Total servers and total switches
n	Port number of switches, $n \geqslant 2$
h	Height of Tree and Aspen Trees, $h \geqslant 0$, level of a BCN in the first dimension
k	Level, ary or column number, $k \geqslant 0$
m	Dimension number, $m \geqslant 0$
f	Duplicate connection count of Aspen Trees
α, β	Number of master (slave) servers in the level-0 BCN
γ	Level of the unit BCN in the second dimension
r^i	Number of switches in the i dimension
s	Max value of each symbol in SWKautz
d_0	The highest diameter of a container
a	Number of master servers in the level-0 HFN, number of switches in the level-0 HFN
b	Number of worker servers in the level-0 HFN
L_i	The level i in tree-like architectures
c_i	The number of links from an L_i switch to L_{i-1} switches per pod
g_i	The number of L_{i-1} pods to which each L_i switch connects
p_i	The number of pods at L_i
e_i	The number of switches per L_i pod

We also illustrate some representative architectures in detail for clarification. For instance, a 3-layer **Fat-tree** can accommodate $\frac{n^3}{4}$ servers and $\frac{5n^2}{4}$ switches. As a server only connects to a ToR switch, the *server degree* is 1. The *diameter* is 6 due to the 3-layer tree-like structure. Fat-tree could achieve 1:1 *oversubscription* to overcome the *bottleneck problem* in the tree architecture (the *BiW* is only 1). As the symmetry, if a Fat-tree is divided into two equal groups, the *BiW* is $\frac{N_e}{2}$. In an all-to-all scenario, if we assume any of servers in fat-tree communicate with the other $N_e - 1$ servers simultaneously, the *BoD* is $N_e - 1$. The general terms of different levels in switch-centric tree-like architectures are proposed in Table 14 [117]. The symbol descriptions are also shown in Table 12. The value of c indicates the density of links between two adjacent levels, which affects the fault tolerance of the architectures. For example, **Fat-tree** ($c_2 = c_3 = \cdots = c_n = 1$) has a poor *fault tolerance* compared to **Aspen Trees** ($c_2 \cdot c_3 \ldots \cdot c_n > 1$) and **VL2** ($c_2 = c_3 = \cdots = c_{n-1} = 1$ and $c_n > 1$), whereas Fat-tree supports *the most number of servers (scalability)* when switch port-count n and level k are the same.

The number of servers of **DCell** is proportional to $O(n^{2^k})$, whereas that of **BCube** is proportional to $O(n^{k+1})$, which means the *scalability* of **DCell** is higher. The N_w/N_e of BCube is $\frac{k+1}{n}$, while that of DCell is $\frac{1}{n}$, which means the *cost per server on switches* in BCube is higher and BCube needs more switches and cables than DCell does. The *server degree* of DCell is $k+1$ and the same as that of BCube, which means the number of NICs is $k+1$. The *diameter* of DCell $<2^{k+1} - 1$ due to the way of expansion. The *diameter* of BCube is $k+1$ owing to the hierarchical structure, which is lower than that of DCell. As the unbalanced traffic in DCell, the 0-level links carry higher traffic than the links in other levels and the *BoD* is proportional to $O(N_e \log N_e)$. As $k+1$ parallel equal-cost paths between any two servers in a BCube$_k$, it is relatively easy to spread out the traffic equally along all the links, and the *BoD* is $\frac{N_e-1}{k+1}$, which is lower than that of DCell.

Table 13
Quantitative comparison of wired architectures with electrical switches.

Architecture	N_e	N_w	Degree	Diameter	BiW	BoD
Tree	n^h	$\frac{n^h-1}{n-1}$	1	$2\log_n N_e$	1	$(\frac{n-1}{n^2}){N_e}^2$
Fat-tree	$\frac{n^3}{4}$	$\frac{5n^2}{4}$	1	6	$\frac{N_e}{2}$	N_e-1
VL2	$5n^2$	$\frac{n^2+6n}{4}$	1	6	$2N_e-20$	N_e-1
FBFLY	k^m	k^{m-1}	1	$m+1$	$\frac{N_e}{4}$	N_e-1
Aspen Trees	$\frac{n^h}{2^{h-1}f}$	$\frac{(h-\frac{1}{2})n^{h-1}}{2^{h-2}f}$	1	$2n$	$\frac{N_e}{2}$	N_e-1
Diamond	$\frac{n^3}{4}$	$\frac{5n^2}{4}$	1	6	$\frac{N_e}{2}$	N_e-1
DCell	$\in((n+\frac{1}{2})^{2^k}-\frac{1}{2},(n+1)^{2^k}-1)$	$\frac{t_k}{n}$	$k+1$	$<2^{k+1}-1$	$\frac{N_e}{4\log_n N_e}$	$<N_e\log_n N_e$
Ficonn	$\geqslant((\frac{n}{4})^2 2^{k+2})$	$\frac{N_k}{n}$	2	$<2^{k+1}-1$	$>\frac{N_e}{4*2^k}$	$2^k N_e$
Dpillar	$k(\frac{n}{2})^k$	$k(\frac{n}{2})^{k-1}$	2	$k+\lfloor\frac{k}{2}\rfloor$	$\frac{N_e}{k}$	$3k\frac{N_e-1}{2}$
MCube	$3k(k+1)^2$	$(k+1)^3$	2	$3k$	$\frac{N_e}{3k}$	$\frac{N_e-1}{2}$
HCN	n^{k+1}	n^k	2	$2^{k+1}-1$	$\frac{n^2}{4}$	N_e-1
BCN	$\alpha^h(\alpha+\beta)$ or $\alpha^{h-\gamma}(\alpha^\gamma(\alpha+\beta)(\alpha^\gamma\beta+1))$	$\frac{N_e}{n}$	2 or 1	$2^{h+1}+2^{\gamma+1}-1$	$\frac{\alpha^2}{4}$ or $\frac{\alpha^2-1}{4}$ or $\frac{\alpha^h\beta(\alpha^\gamma\beta+2)}{4}$	$<\alpha^h\beta\log_n(\alpha^h\beta)$
SWCube	$\frac{n}{2}\prod_{i=1}^{k} r_i$	$\prod_{i=1}^{m} r_i$	2	$m+1$	$\frac{N_e}{2}$	$\frac{N_e-1}{2}$
SWKautz	$(\frac{n}{2}+1)(\frac{n}{2})^k$	$(\frac{n}{2}+1)(\frac{n}{2})^{k-1}$	2	$m+1$	$\frac{N_e}{2}$	$\frac{N_e-1}{2}$
BCube	n^{k+1}	$(k+1)n^k$	$k+1$	$k+1$	$\frac{N_e}{2}$	$\frac{N_e-1}{k+1}$
MDCube	$(\frac{(k+1)n^k}{m}+1)^m n^{k+1}$	$(\frac{(k+1)n^k}{m}+1)^m(k+1)n^k$	2	$4k+3+(m-1)(2k+3)$	$\frac{N_e}{2}$	$\frac{tN_e}{k_i}$
BCCC	$(k+1)n^{k+1}$	$(n+k+1)n^k$	2	$2k+2$	$\frac{N_e}{2(k+1)}$	$\frac{N_e-1}{2(k+1)}$
CamCube	k^3	0	6	$\frac{3}{2}\sqrt[3]{N_e}$	$8\sqrt[3]{N_e}$	$N_e\sqrt[3]{N_e}/8$
Snowflake	$n(n+1)^k$	$(n+1)^k$	3 or 1	$2k+2$	6	$\frac{n}{2}(N_e-n)$
HFN	$(b+1)a^{k+1}$	$(k+a)a^k$	1	$k+1+\lfloor\frac{a}{2}\rfloor$	$\frac{a^{k+1}}{2}$ or 2	N_e-1 or $\frac{b^2(a^{k+1}-1)}{2}$

Table 14
Structure of general tree-like switch-centric architectures.

L_i	L_1	L_2	L_i	$\cdots$	L_{k-1}	L_k
c_i	1	c_2	c_i	$\cdots$	c_{k-1}	c_k
g_i	$\frac{n}{2}$	$\frac{n}{2c_2}$	$\frac{n}{2c_i}$	$\cdots$	$\frac{n}{2c_{k-1}}$	$\frac{n}{c_k}$
p_i	$\frac{n^{k-1}}{2^{k-2}\cdot\prod_{j=2}^{k} c_j}$	$\frac{n^{k-2}}{2^{k-3}\cdot\prod_{j=3}^{k} c_j}$	$\frac{n^{k-i}}{2^{k-i-1}\cdot\prod_{j=i+1}^{k} c_j}$	$\cdots$	$\frac{n}{c_k}$	1
e_i	1	$\frac{n}{2c_2}$	$\frac{n^{i-1}}{2^{i-1}\cdot\prod_{j=2}^{i} c_j}$	$\cdots$	$\frac{n^{k-2}}{2^{k-2}\cdot\prod_{j=2}^{k-1} c_j}$	$\frac{n^{k-1}}{2^{k-1}\cdot\prod_{j=2}^{k} c_j}$

Discussion. The main difference between switch-centric and server-centric architectures is the *intermediate nodes* employed for routing and forwarding packets in DCN's. Switch-centric architectures use *layer-3 and layer-2 switches*, while server-centric architectures use *modified servers with multiple NICs*. The results of **qualitative** and **quantitative** comparisons showed that **switch-centric** architectures can provide several nice features, such as *high oversubscription ratios*, *high bisection bandwidth*, *high fault tolerance*, *high load balancing* and so on, while the relatively low *scalability* due to the port number of switches; **server-centric** architectures can offer *high scalability*, *low diameter*, *high bisection bandwidth*, and *high load balancing*, but the relatively high *cabling complexity* due to the recursively defined structures. Any architecture has its advantages and disadvantages. *No architecture can be "perfect" with all desired features. A DCN designer must first better understand the traffic patterns for applications, and then make proper tradeoffs to arrive at a solution for their situations.*

4.4.2. Optical architectures

Many optical interconnects are proposed recently to offer a promising, feasible and high-bandwidth solution for future DCN's. We also make a qualitative comparison of optical architectures in Table 15. In this table, we choose 5 metrics to summary each architecture, including *technology* (*hybrid*, *all-optical*, *WDM*), *connectivity* (*circuit*, *packet*), *capacity*, *scalability*, and *prototype*. We refer to the summary proposed in [91], and add several architectures proposed in 2012–2015. *Technologies* include *hybrid* or *all-optical* interconnections, and *Wavelength Division Multiplexing* (*WDM*). WDM transceivers multiplex the data with separate wavelength to traverse it simultaneously in the fiber for providing higher bandwidth. *Connectivity* means the ways of connection in switches, including *circuit*-based switching and *packet*-based switching. The former has long reconfiguration time (in the orders of few *ms*), which is suitable for DCN's with long-term enormous data transfers, and the later achieves very low latency between any two servers, which is fit to latency-sensitive DCN's. *Scalability* is the capability of being expanded to accommodate a large number of nodes (e.g., ToR switches), which is constrained by the number of optical ports and wavelength channels. *Capacity* means the maximum supported data rate defined by the capacity limitation technology, including Semiconductor Optical Amplifier (SOA), Tunable Wavelength Converters (TWC) and optical MEMS transceivers (Transc.). *Prototype* shows the feasibility of the architecture used in a real DCN. Note that, on the columns, if

Table 15
Summary of optical architectures.

Architecture	Designers	Year	Technology			Connectivity		Capacity	Scalability	Prototype
			Hybrid	All-optical	WDM	Circuit	Packet			
OSMOSIS	IBM	2004		⋆	⋆		⋆	SOA	Medium	
Data vortex	Columbia U.	2006		⋆		⋆		Transc.	Low	⋆
Bidirectional	Columbia U.	2009		⋆	⋆		⋆	SOA	High	⋆
c-Through	Rice U., CMU, Intel	2010	⋆			⋆		Transc.	Low	⋆
Helios	UC San Diego	2010	⋆		⋆	⋆		Transc.	Low	⋆
DOS	UC Davis	2010		⋆			⋆	TWC	Low	⋆
Proteus	UIUC, NEC	2010		⋆	⋆	⋆		Transc.	Medium	⋆
Petabit	Poly-NY	2010		⋆	⋆		⋆	TWC	High	
Space-WL	SSSUPisa	2010		⋆	⋆	⋆		SOA	High	
Polatis	Polatic Inc.	2010		⋆		⋆		Transc.	Low	⋆
OPST	Intune Inc.	2010		⋆	⋆		⋆	TWC	Low	⋆
OSA	Northwestern U.	2012		⋆	⋆	⋆		Transc.	Medium	⋆
Mordia	UC San Diego	2013	⋆		⋆	⋆		Transc.	Low	⋆
Quartz	Waterloo U., UC Berkeley	2014	⋆		⋆	⋆	⋆	Transc.	Medium	⋆
REACToR	UC San Diego	2014	⋆			⋆	⋆	Transc.	Medium	⋆
WaveCube	HKUST	2015		⋆	⋆	⋆		Transc.	High	⋆

Table 16
Summary of wireless architectures.

Architecture	Designers	Year	Technology		Prototype
			Hybrid	All-wireless	
Flyways	Microsoft, UW	2009/2011	⋆		⋆
Wireless fat-tree	UT Dallas	2010		⋆	
3D beamforming	UCSB, XJTU	2011/2012	⋆		⋆
Wireless crossbar	IBM	2011/2012	⋆		⋆
Carlay	Cornell U., Microsoft	2012		⋆	⋆
Angora	UCSB, UCSD, Google, Dartmouth Col.	2014	⋆		⋆
Spherical Mesh	SJTU	2014	⋆		⋆
Graphite	SJTU	2016	⋆		⋆
Diamond (3D Wireless Ring)	Tsinghua, SBU, USTC, BUPT	2016	⋆		⋆

an architecture has the feature, the corresponding cell is filled with a ⋆.

We also illustrate some representative architectures in detail for clarification. For example, **Helios**, a *hybrid* architecture, was proposed in 2010 by UC San Diego. **OSA**, an *all-optical* architecture, was presented in 2012 by Northwestern University. Helios can immediately provide an incremental upgrade for existing DCNs, while OSA offers a *high bandwidth* with low latency and power consumption for future DCNs but needs to plan carefully. The ways of optical connection are both *circuit*-based switching. The two architectures both employ *WDM* and *optical MEMS transceivers*, which can provide any data rate (40 Gbps, 100 Gbps or higher). The *scalability* of Helios is lower than OSA due to the limited optical ports. The two both have *prototypes*.

Discussion. Compared to an electrical architecture, an optical architecture provides better energy efficiency but higher latency and poorer scalability. A hybrid architecture attempts to combine the advantages of both electrical and optical architectures, where the electrical part provides transmission of latency-sensitive data and the optical part is in charge of transmission of long-term bulky data. Hybrid architectures can immediately provide an incremental upgrade for existing DCN's, while all-optical architectures offer a high bandwidth with low latency and power consumption for future DCN's but need to plan and design prudently.

4.4.3. Wireless architectures

Long inter-switch cables require planning and overhead cable trays, and is more difficult to install and maintain than short intra-rack cables or cables that cross between adjacent racks. Wireless architectures provide a promising, feasible, and on-demand solution to reduce the long inter-switch cables. A summary of wireless architectures is presented in Table 16. In this table, we choose 2 metrics to summary each architecture, including *technology* (*hybrid*, *all-wireless*), and *prototype*. *Technologies* include *hybrid* or *all-optical* interconnections. *Prototype* shows the feasibility of the architecture used in a real DCN. Note that, on the columns, if an architecture has the feature, the corresponding cell is filled with a ⋆.

We also illustrate some representative architectures in detail for clarification. For example, **Flyways**, a *hybrid* wired/wireless architecture, was proposed by Microsoft and UW in 2009/2011. **3D beamforming**, also a *hybrid* architecture, was presented in 2011/2012 by UCSB and XJTU. **Carlay**, an *all-wireless* architecture, was proposed in 2012 by Cornell University and Microsoft. The three architectures all have been evaluated in *testbeds*.

The **quantitative** metrics of wireless architectures focus on *the total number of wireless links* and *average node degree* other than these metrics of wired switch or server-centric architectures. We also provide a quantitative comparison of Flyways, 3D Beamforming, and Graphite [180]. The symbol descriptions are given in Table 17. In this table, $G(V, E)$ denotes a wireless network. $V = \{ToR_i\}$ is the set of all the ToRs (nodes), and each ToR is connected with a 60 GHz antenna. We focus on 2 metrics, the total number of wireless edges $|E|$ $(=|\{(ToR_i, ToR_j)\}|)$ and average node degree Δ $(=\sum_{v\in V} deg(v)/|V|)$. The larger the parameters $|E|$ and Δ are, the more a node can communicate with the other nodes averagely, which leads to higher connectivity of the wireless DCN's. The results of comparison in Table 18 showed that Graphite has better performance due to the design of multi-layer antennas.

Discussion. With 60 GHz wireless antennas and reflectors, wireless architectures provide a feasible and on-demand solution for DCN's. Hybrid architectures can immediately provide an incremental upgrade for existing DCN's, while all-wireless architectures offers a high bandwidth with low latency and cabling complexity for future DCN's but need to plan and design carefully. As the lack of large-scale testbeds, all-wireless architectures have a long way to go to be used in real DCN's.

Table 17
Symbol descriptions of three wireless architectures.

Symbol	Description
$G(V, E)$	Wireless network
$V = \{ToR_i\}$	Set of ToR switches
$\|E\| = \|\{(ToR_i, ToR_j)\}\|$	Number of links between a pair of ToRs
Δ	Average node degree, $k \geqslant 0$
x, y	Number of racks in a horizontal/vertical line
R	Propagation radius
l	Distance between two adjacent racks horizontally
w	Distance between two adjacent racks vertically
h	Height from the top of rack to the ceiling
h_i	Height from the ith network level, $h_1 = 0$

4.4.4. Discussion and future research directions

In this subsection, we discuss several important issues of DCN architectures and promising research directions in future.

Virtualization is an important issue of DCN architecture. Bari et al. [18] offered a survey of DCN virtualization. Based on DCN architectures, network virtualization optimally schedules the physical resources (switches, servers, and links), which aims to provide low cost, better management flexibility, scalability, resource utilization [27], energy efficiency, security, and performance isolation to meet SLAs between tenants and service providers.

Traffic management is also an essential issue of DCN architectures. Bilal et al. [23] and Zhang and Ansari [179] both provided surveys on traffic management methods in DCN's, mainly including routing algorithms, transmission protocols, flow detection/consolidation, and congestion control strategies [114,55,54]. All these methods aim to achieve the features of low latency, decongestion, and load balance. The research on Software-Defined Network (SDN) and Openflow is already in full swing, which is helpful for better controlling DCN's automatically [15].

Power consumption is another important issue of DCN architectures. Hammadi and Mhamdi [73] and Zhang and Ansari [179] both carried out surveys on issues of reducing power consumption to achieve **green/harmony** data centers, including dynamic voltage/frequency scaling, rate adaptation, dynamic power management, smart power delivery, cooling system, and renewable energy supply. Renewable energy sources (e.g., solar and wind) have inherent features such as intermittently dependent on the local environment and the weather. The literatures [121,120,22,57,110,153,56] explored the possibility and feasibility of utilizing renewable energy in data centers.

Topology Design is an essential issue for the purpose of exploring a general method of designing and analyzing topologies. Singla et al. [155] proposed the first systematic method of designing heterogeneous networks. The non-trivial upper bound on network throughput under uniform traffic patterns for any topology with identical switches is presented by extensive simulations. When the homogeneous/hierarchical topology design may be reaching its limits, using random graphs as building blocks is logically a good choice for incremental deployment of heterogeneous networks, which surprisingly achieves close-to-upper-bound throughput. The VL2 deployed in Microsoft's data centers can increase throughput by 43% with the same equipment by rewiring uniform randomly. Schlinker et al. [146] presented Condor, a rapid and efficient approach of designing DCN's to achieve tradeoffs among many criteria such as energy cost, bandwidth, latency, reliability and expandability. A declarative, constraint-based Topology Description Language (TDL) is employed to enable concise modifiable descriptions of DCN's (e.g., fat-tree, BCube, DCell) that Condor transforms into constraint-satisfaction problems to support rapid synthesis. Condor also supports efficient incremental expansions at live networks. At the very beginning of designing architectures for DCN's, structured topologies may be better choices for data center planners, which have more stable performance and are relatively easier to cabling and maintain than random topologies. When existing structured topologies need to upgrade, the random topologies may be employed partially for incremental deployment.

Optical DCN architectures can be adopted in wired or wireless data centers [71]. They are not used in large-scale deployment scenarios so far because of expensive devices and long latency. However, it may be a good choice for MDC's.

Wireless DCN architectures are worth paying attention to due to low cabling complexity is always a goal of designing DCN's. In recent years, a number of wireless DCN designs have been presented, even fully wireless data centers. However, these designs are still in laboratories. Wireless technologies may be suitable for MDC's according inherent features.

Specialized data centers are operated for particular application scenarios of the specialized businesses at different companies and organizations. Facebook's data centers mainly support the business of social networks [144], in which VMs are not typically employed, each physical server has precisely one role (e.g., Web, DB, Cache, Hadoop, Multifeed), and racks contain only servers of the same role.

5. The facility considerations for data centers

The support parts of DCN's should be taken into consideration carefully. **Cisco** considered that the reliability and sustainability of DCN's intimately depend on fundamental physical facilities, such as the power, cooling, physical housing, cabling, physical security, and fire protection systems [8]. The term Network-Critical Physical Infrastructure (NCPI) denotes the set of facilities as follows.

Power. The power facility consists of the electrical service entrance of the building, main distribution, generators, uninterruptible power supply (UPS) systems and batteries, surge protection, transformers, distribution panels, and circuit breakers.

Cooling. The cooling system includes computer room air conditioners (CRACs) and rack- or row-level cooling devices. The associated subsystems of CRACs are chillers, cooling towers, condensers, ductwork, pump packages, and piping.

Table 18
Quantitative comparison of three wireless architectures.

Architecture	$\|E\|$	Δ
Flyways	$y(x-1)+(y-1)[(2u+1)-u(u+1)], u = \frac{1}{l}\left\lfloor\sqrt{R^2-w^2}\right\rfloor$	$2(1-\frac{1}{x})+2(1-\frac{1}{y})[2u+1-\frac{u(u+1)}{x}]$
3D Beamforming	$\sum_{i=0}^{2}\frac{1+sgn(i)}{2}(y-i)[(2v_i+1)x-v_i(v_i+1)]-\frac{1}{2}xy$, $v_0=\left\lfloor\frac{1}{l}\sqrt{R^2-4h^2}\right\rfloor, v_1=u=\left\lfloor\frac{1}{l}\sqrt{R^2-w^2}\right\rfloor, v_2=\left\lfloor\frac{1}{l}\sqrt{R^2-4h^2-4w^2}\right\rfloor$	$\sum_{i=0}^{2}[1+sgn(i)](1-\frac{i}{y})[(2v_i+1)-\frac{v_i(v_i+1)}{x}]-1$
Graphite	$\frac{1}{4}(5x-8)+\frac{1}{2}(y-1)[(2u+1)x-u(u+1)]+\frac{1}{2}\sum_{i=0}^{2}\frac{1+sgn(i)}{2}(y-i)[(2w_i+1)x-w_i(w_i+1)]$, $u=\left\lfloor\frac{1}{l}\sqrt{R^2-w^2}\right\rfloor, w_0=\left\lfloor\frac{1}{l}\sqrt{R^2-h_2^2}\right\rfloor, w_1=\left\lfloor\frac{1}{l}\sqrt{R^2-h_2^2-w^2}\right\rfloor, w_2=\left\lfloor\frac{1}{l}\sqrt{R^2-h_2^2-4w^2}\right\rfloor$	$(\frac{5}{2}-\frac{4}{x})+(1-\frac{1}{y})[(2u+1)-\frac{u(u+1)}{x}]+\sum_{i=0}^{2}\frac{1+sgn(i)}{2}(1-\frac{i}{y})[(2w_i+1)-\frac{w_i(w_i+1)}{x}]$

Cabling. The cabling considerations include cabling topology, cabling media, and cabling pathways, which aim to optimize the performance and flexibility of DCN's.

Racks and physical structure. Racks hold IT equipment such as servers, switches, and storage. Physical structures such as dropped ceiling, raised floors, and pathways satisfy cabling considerations.

Management. Management employs visual methods to monitor physical facilities to achieve reliability, which includes building management systems, network management systems, element managers, and other monitoring hardware and software.

Grounding. Grounding is an important system that protects for the staff and equipment in a data center from lightning strikes and electrostatic discharge.

Physical security and fire protection. Physical security devices are placed at the room and racks to ensure the security. Fire detection/suppression systems are significant for the data center investment.

Intel also proposed their facility designs for high-density data centers in [134], which is a good choice for efficiently leveraging energy and space to increase the capacity of the computer room. Intel concentrates their considerations on 5 areas as follows.

Air management includes air segregation and automated control sensors. The former means separating supply and return air paths to maintain a constant pressure difference, and providing enough conditioned air to satisfy the actual demand of servers. The later are located in data centers to monitor the power, temperature, humidity, and static pressure.

Thermal management provides extra cooling to protect key devices against residual heat before temperatures raise up to thresholds when the power fails.

Architectural considerations include sealed walls, ceiling, floors, removal of raised metal floors (RMF), increased floor loading specifications, and overhead structured cabling.

Electrical considerations reduce electrical losses by branch circuits, which make electrical systems efficient and reliable.

6. Conclusion

In this paper, we provide a comprehensive survey on the features, architectures, and hardware of DCN's. We first give an overview of production data centers. Next, we introduce the hardware of DCN's, including switches, servers, storage devices, racks and cables used in industries, which are highly essential for designing DCN architectures. And then we thoroughly analyze the architectures of DCN's from various aspects, such as connection types, wiring layouts, interconnection facilities, and network characteristics based on the latest literature. Moreover, we precisely analyze the network features qualitatively and quantitatively, and we also discuss some important issues and new research trends of DCN's. Finally, the facility settings and maintenance issues for data centers are also discussed.

Acknowledgments

This work has been supported in part by the China 973 Projects (2014CB340303, 2012CB316201), China NSF Projects (61472252, 61133006), the Opening Project of Key Lab of Information Network Security of Ministry of Public Security (The Third Research Institute of Ministry of Public Security) Grant number C15602, the Opening Project of Baidu (Grant number 181515P005267). We would like to thank Wei Wei, Xuanzhong Wei, Bo Fu, and Senhong Huang for their contributions on the early versions of this paper, and thank the anonymous reviewers for their valuable comments to improve the quality of our paper.

References

[1] D. Abts, M.R. Marty, P.M. Wells, P. Klausler, H. Liu, Energy proportional datacenter networks, ACM SIGARCH Comput. Archit. News 38 (3) (2010) 338–347.
[2] H. Abu-Libdeh, P. Costa, A. Rowstron, G. O'Shea, A. Donnelly, Symbiotic routing in future data centers, ACM SIGCOMM Comput. Commun. Rev. 40 (4) (2010) 51–62.
[3] A. Agache, R. Deaconescu, C. Raiciu, Increasing datacenter network utilisation with GRIN, in: USENIX NSDI, 2015, pp. 29–42.
[4] J.H. Ahn, N. Binkert, A. Davis, M. McLaren, R.S. Schreiber, HyperX: topology, routing, and packaging of efficient large-scale networks, in: IEEE/ACM SC, 2009, pp. 1–11.
[5] M. Al-Fares, A. Loukissas, A. Vahdat, A scalable, commodity data center network architecture, ACM SIGCOMM Comput. Commun. Rev. 38 (4) (2008) 63–74.
[6] Amazon, Amazon S3, 2016. http://aws.amazon.com/s3/.
[7] Amazon, Announcing Amazon Elastic Compute Cloud, Amazon EC2—beta, 2006. https://aws.amazon.com/cn/about-aws/whats-new/2006/08/24/announcing-amazon-elastic-compute-cloud-amazon-ec2---beta/.
[8] American Power Conversion and Panduit, Facility Considerations for the Data Center, 2005.
[9] Apple, Apple and the environment, 2014. http://www.apple.com/environment/climate-change/.
[10] Apple, Apple Hits Milestone: All Data Centers Powered by 100% Renewables, 2013. http://www.sustainablebusiness.com/index.cfm/go/news.display/id/24713.
[11] Arista, Arista 7050QX series data center switch, 2016. http://www.arista.com/assets/data/pdf/7050QX-QuickLook.pdf.
[12] Arista, Arista 7500 series data center switch, 2014. http://www.aristanetworks.com/media/system/pdf/Datasheets/7500_Datasheet.pdf.
[13] Associated Press, Google proposes new data center in The Dalles, 2013. http://www.columbian.com/news/2013/jun/06/google-proposes-new-data-center-in-the-dalles/.
[14] ASUS, RS724Q-E7/RS12, 2014. https://www.asus.com/Commercial_Servers_Workstations/RS724QE7RS12/.
[15] S. Azodolmolky, Software Defined Networking with OpenFlow, Packt Publishing Ltd., 2013.
[16] R. Bakken, How Microsoft Manages Its Cloud Infrastructure at a Huge Scale, 2011. http://channel9.msdn.com/Events/TechEd/NorthAmerica/2011/COS211.
[17] R. Balodis, I. Opmane, History of data centre development, in: Reflections on the History of Computing, Springer, 2012, pp. 180–203.
[18] M.F. Bari, R. Boutaba, R. Esteves, L.Z. Granville, M. Podlesny, M.G. Rabbani, Q. Zhang, M.F. Zhani, Data center network virtualization: A survey, IEEE Commun. Surv. Tutor. 15 (2) (2013) 909–928.
[19] L.A. Barroso, U. Hölzle, The datacenter as a computer: An introduction to the design of warehouse-scale machines, Synth. Lect. Comput. Archit. 4 (1) (2009) 1–108.
[20] A. Benner, Optical interconnect opportunities in supercomputers and high end computing in: Optical Fiber Communication Conference, 2012, pp. 1–60.
[21] L.N. Bhuyan, D.P. Agrawal, Generalized hypercube and hyperbus structures for a computer network, IEEE Trans. Comput. 100 (4) (1984) 323–333.
[22] R. Bianchini, Leveraging renewable energy in data centers: present and future, in: ACM HPDC, 2012, pp. 135–136.
[23] K. Bilal, S.U.R. Malik, O. Khalid, A. Hameed, E. Alvarez, V. Wijaysekara, R. Irfan, S. Shrestha, D. Dwivedy, M. Ali, et al., A taxonomy and survey on green data center networks, Future Gener. Comput. Syst. 36 (2014) 189–208.
[24] J. Booton, IBM Invests $1.2B Dollars to Build 15 New Data Centers and 'Cloud Hubs', 2014. http://www.foxbusiness.com/technology/2014/01/17/ibm-invests-12b-in-15-new-data-centers-local-cloud-hubs/.
[25] B. Box, Freedom Rack Plus with M6 Rails, 2014. http://www.blackbox.com/Store/Detail.aspx/Freedom-Rack-Plus-with-M6-Rails-45U-19/RM088A.
[26] B. Calder, J. Wang, A. Ogus, N. Nilakantan, A. Skjolsvold, S. McKelvie, Y. Xu, S. Srivastav, J. Wu, H. Simitci, et al., Windows Azure Storage: a highly available cloud storage service with strong consistency, in: Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles, ACM, 2011, pp. 143–157.
[27] B. Cao, X. Gao, G. Chen, Y. Jin, NICE: Network-aware VM consolidation scheme for energy conservation in data centers, in: IEEE ICPADS, 2014, pp. 166–173.
[28] K. Chen, A. Singla, A. Singh, K. Ramachandran, L. Xu, Y. Zhang, X. Wen, Y. Chen, OSA: An optical switching architecture for data center networks with unprecedented flexibility in: USENIX NSDI, 2012, pp. 1–14.
[29] K. Chen, X. Wen, X. Ma, Y. Chen, Y. Xia, C. Hu, Q. Dong, WaveCube: A scalable, fault-tolerant, high-performance optical data center architecture, in: IEEE INFOCOM, 2015, pp. 1–9.
[30] G. Chen, Y. Zhao, D. Pei, D. Li, Rewiring 2 links is enough: Accelerating failure recovery in production data center networks in: IEEE ICDCS, 2015, pp. 569–578.
[31] CIR, 40G Ethernet-Closer Than Ever to an All-Optical Network, 2010.
[32] Cisco, Cisco's Massively Scalable Data Center: Network Fabric for Warehouse Scale Computer, 2010. http://www.cisco.com/c/dam/en/us/td/docs/solutions/Enterprise/Data_Center/MSDC/1-0/MSDC_AAG_1.pdf.
[33] Cisco, Cisco Nexus 3064 Series Switches, 2016. http://www.cisco.com/c/en/us/products/collateral/switches/nexus-3000-series-switches/data_sheet_c78-651097.html.
[34] Cisco, Cisco Nexus 7000 Series Switches, 2013. http://www.cisco.com/c/en/us/products/collateral/switches/nexus-7000-10-slot-switch/Data_Sheet_C78-437762.html.

[35] Cisco, Data Center: Load Balancing Data Center Services SRND, 2004.
[36] H. Clancy, Some like it cool: HP/EDS fellow shares data center best practices, 2009. http://www.zdnet.com/blog/green/some-like-it-cool-hpeds-fellow-shares-data-center-best-practices/5393.
[37] C. Clos, A study of non-blocking switching networks, Bell Syst. Tech. J. 32 (2) (1953) 406–424.
[38] Computer Business Review, Top 10 biggest data centres from around the world, 2015. http://www.cbronline.com/news/data-centre/infrastructure/top-10-biggest-data-centres-from-around-the-world-4545356.
[39] M. Csernai, F. Ciucu, R.-P. Braun, A. Gulyás, Towards 48-fold cabling complexity reduction in large flattened butterfly networks, in: IEEE INFOCOM, 2015, pp. 109–117.
[40] Y. Cui, S. Xiao, X. Wang, Z. Yang, C. Zhu, X. Li, L. Yang, N. Ge, Diamond: Nesting the data center network with wireless rings in 3D space, in: USENIX NSDI, 2016, pp. 657–669.
[41] A.R. Curtis, T. Carpenter, M. Elsheikh, A. López-Ortiz, S. Keshav, REWIRE: an optimization-based framework for unstructured data center network design, in: IEEE INFOCOM, 2012, pp. 1116–1124.
[42] Data Center Knowledge, Largest Data Centers: Worthy Contenders, 2013. http://www.datacenterknowledge.com/special-report-the-worlds-largest-data-centers/largest-data-centers-worthy-contenders/.
[43] Data Center Knowledge, Merlin: Capgemini's Modular Data Center, 2013. http://www.datacenterknowledge.com/merlin-capgeminis-modular-data-center/.
[44] Dell, PowerEdge 4820 Rack Enclosure, 2013. http://www.dell.com/us/business/p/poweredge-4820/pd.
[45] Dell, PowerEdge M820 Blade Server, 2013. http://www.dell.com/us/business/p/poweredge-m820/pd.
[46] Z. Ding, D. Guo, X. Liu, X. Luo, G. Chen, A MapReduce-supported network structure for data centers, Concurr. Comput.: Pract. Exper. 24 (12) (2012) 1271–1295.
[47] Dell, PowerEdge R720 rack server, 2013. http://www.dell.com/us/business/p/poweredge-r720/pd.
[48] EMC, EMC SYMMETRIX VMAX 40K, 2013. http://www.emc-storage.co.uk/emc-symmetrix-vmax-40k-emc-vmax-40k.
[49] Emerson Network Power, DCF Optimized Racks, 2013. http://www.emersonnetworkpower.com/en-US/Products/RacksAndIntegratedCabinets/IndoorRacksAccessories/Racks/Pages/DCFOptimizedRacks.aspx.
[50] N. Farrington, G. Porter, S. Radhakrishnan, H.H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, A. Vahdat, Helios: a hybrid electrical/optical switch architecture for modular data centers, ACM SIGCOMM Comput. Commun. Rev. 40 (4) (2010) 339–350.
[51] N. Farrington, A. Andreyev, Facebook's data center network architecture in: IEEE Optical Interconnects Conference, 2013, pp. 49–50.
[52] I. Fired, Microsoft opens San Antonio data center, 2008. http://www.cnet.com/news/microsoft-opens-san-antonio-data-center/.
[53] FUJITSU, FUJITSU Server PRIMERGY RX600 S6, 2014. http://www.fujitsu.com/fts/products/computing/servers/primergy/rack/rx600/.
[54] X. Gao, W. Li, Y. Chen, W. Liang, G. Chen, L. Kong, Traffic load balancing schemes for devolved controllers in mega data centers, IEEE Trans. Parallel Distrib. Syst. (2016) in press.
[55] X. Gao, W. Xu, F. Wu, G. Chen, Sheriff: A regional pre-alert management scheme in data center networks, in: IEEE ICPP, 2015, pp. 669–678.
[56] Y. Gao, Z. Zeng, X. Liu, P. Kumar, The answer is blowing in the wind: Analysis of powering Internet data centers with wind energy, in: IEEE INFOCOM, 2013, pp. 520–524.
[57] I. Goiri, R. Beauchea, K. Le, T.D. Nguyen, M.E. Haque, J. Guitart, J. Torres, R. Bianchini, GreenSlot: scheduling energy consumption in green datacenters, in: IEEE/ACM SC, 2011, pp. 20–30.
[58] Google, Backrub, 1997. http://pimm.wordpress.com.
[59] Google, Google Data Centers, 2012. http://www.google.com/about/datacenters/gallery/#/places.
[60] Google, Google is proud to call Belgium home to one of our data centers, 2013. http://www.google.com/about/datacenters/inside/locations/st-ghislain/.
[61] Google, Google is proud to call Oklahoma home to one of our data centers, 2013. http://www.google.com/about/datacenters/inside/locations/mayes-county/.
[62] A. Greenberg, J.R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D.A. Maltz, P. Patel, S. Sengupta, VL2: a scalable and flexible data center network, ACM SIGCOMM Comput. Commun. Rev. 39 (4) (2009) 51–62.
[63] A. Greenberg, J. Hamilton, D.A. Maltz, P. Patel, The cost of a cloud: research problems in data center networks, ACM SIGCOMM Comput. Commun. Rev. 39 (1) (2008) 68–73.
[64] A. Greenberg, P. Lahiri, D.A. Maltz, P. Patel, S. Sengupta, Towards a next generation data center architecture: scalability and commoditization in: ACM PRESTO, 2008, pp. 57–62.
[65] D. Guo, T. Chen, D. Li, Y. Liu, X. Liu, G. Chen, BCN: expansible network structures for data centers using hierarchical compound graphs, in: IEEE INFOCOM, 2011, pp. 61–65.
[66] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, S. Lu, BCube: a high performance, server-centric network architecture for modular data centers, ACM SIGCOMM Comput. Commun. Rev. 39 (4) (2009) 63–74.
[67] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, S. Lu, DCell: a scalable and fault-tolerant network structure for data centers, ACM SIGCOMM Comput. Commun. Rev. 38 (4) (2008) 75–86.
[68] Z. Guo, Y. Yang, On nonblocking multirate multicast fat-tree data center networks with server redundancy in: IEEE IPDPS, 2012, pp. 1034–1044.
[69] L. Gyarmati, T.A. Trinh, Scafida: A scale-free network inspired data center architecture, ACM SIGCOMM Comput. Commun. Rev. 40 (5) (2010) 4–12.
[70] D. Halperin, S. Kandula, J. Padhye, P. Bahl, D. Wetherall, Augmenting data center networks with multi-gigabit wireless links, ACM SIGCOMM Comput. Commun. Rev. 41 (4) (2011) 38–49.
[71] N. Hamedazimi, Z. Qazi, H. Gupta, V. Sekar, S.R. Das, J.P. Longtin, H. Shah, A. Tanwer, FireFly: a reconfigurable wireless data center fabric using free-space optics, in: ACM SIGCOMM, 2014, pp. 319–330.
[72] J. Hamilton, Architecture for modular data centers ArXiv Preprint cs/0612110.
[73] A. Hammadi, L. Mhamdi, A survey on architectures and energy efficiency in data center networks, Comput. Commun. 40 (2014) 1–21.
[74] K. Han, Z. Hu, J. Luo, L. Xiang, RUSH: RoUting and scheduling for hybrid data center networks, in: IEEE INFOCOM, 2015.
[75] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, N. McKeown, ElasticTree: saving energy in data center networks in: USENIX NSDI, 2010, pp. 1–16.
[76] HP, HP EcoPOD data center, 2014. http://www8.hp.com/us/en/hp-information/environment/hp-ecopod-data-center.html#.U1vLtLKBTVl.
[77] HP, HP 11642 1075mm Shock Universal Rack, 2013. http://m.hp.com/us/en/products/rack-cooling/product-detail.do?oid=5379434.
[78] HP, HP 3PAR StoreServ 10000 Storage, 2013. http://www8.hp.com/cn/zh/products/disk-storage/product-detail.html?oid=5157544#!tab=features.
[79] HP, HP ProLiant DL385p Gen8 Server, 2013. http://www8.hp.com/us/en/products/proliant-servers/product-detail.html?oid=5249584#!tab=features.
[80] Huawei, CloudEngine 12800 Series High-Performance Core Switches, 2013. http://www.huaweienterpriseusa.com/products/network/switches/data-center-switches/cloudengine-12800-series-high-performance-core.
[81] Huawei, OceanStor N8500, 2013. http://support.huawei.com/en/products/itapp/storage/fc-switch/hw-u_149107.htm.
[82] Huawei, Tecal BH640 V2 Blade Server, 2013. http://enterprise.huawei.com/en/products/itapp/server/e-series-blade-server/hw-149397.htm.
[83] L. Huang, Q. Jia, X. Wang, S. Yang, B. Li, PCube: Improving power efficiency in data center networks, in: IEEE CLOUD, 2011, pp. 65–72.
[84] IBM, IBM Canada Leadership Data Centre unveiled in Barrie, Ontario, 2012. http://dcnonl.com/article/id52112/-ibm-canada-leadership-data-centre-unveiled-in-barrie-ontario.
[85] IBM, IBM in Canada, 2013. http://www-935.ibm.com/services/ca/en/it-services/shared-services-ibm-in-canada.html.
[86] IBM, IBM System Storage N7950T, 2013. http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=PM&subtype=SP&htmlfid=TSD02538USEN.
[87] IBM, IBM System x3650 M4, 2014. http://www.redbooks.ibm.com/abstracts/tips0850.html.
[88] IEEE Computer Society, Wireless LAN medium access control (MAC) and physical layer (PHY) specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band, 2012.
[89] P. Jones, GOOGLE TO EXPAND BELGIUM DATA CENTER, 2013. http://www.datacenterdynamics.com/focus/archive/2013/04/google-expand-belgium-data-center.
[90] P. Jones, R. Miller, Dell Opens Quincy Data Center, 2012. http://www.datacenterdynamics.com/focus/archive/2012/02/dell-opens-quincy-data-center, http://www.datacenterknowledge.com/archives/2012/02/15/dell-opens-quincy-data-center/.
[91] C. Kachris, K. Bergman, I. Tomkos, Optical Interconnects for Future Data Center Networks, Optical Networks.
[92] C. Kachris, I. Tomkos, A survey on optical interconnects for data centers, IEEE Commun. Surv. Tutor. 14 (4) (2012) 1021–1036.
[93] S. Kandula, J. Padhye, P. Bahl, Flyways to de-congest data center networks, in: ACM SIGCOMM HotNets, 2009, pp. 1–6.
[94] Y. Katayama, K. Takano, Y. Kohda, N. Ohba, D. Nakano, Wireless data center networking with steered-beam mmwave links, in: IEEE WCNC, 2011, pp. 2179–2184.
[95] Y. Katayama, T. Yamane, Y. Kohda, K. Takano, D. Nakano, N. Ohba, MIMO link design strategy for wireless data center applications, in: IEEE WCNC, 2012, pp. 3302–3306.
[96] W.H. Kautz, Design of optimal interconnection networks for multiprocessors, Archit. Des. Digit. Comput. (1969) 249–272. NATO Advanced Summer Institute.
[97] C. Kim, M. Caesar, J. Rexford, Floodless in seattle: a scalable ethernet architecture for large enterprises, ACM SIGCOMM Comput. Commun. Rev. 38 (4) (2008) 3–14.
[98] J. Kim, W.J. Dally, D. Abts, Flattened butterfly: a cost-efficient topology for high-radix networks, ACM SIGARCH Comput. Archit. News 35 (2) (2007) 126–137.
[99] J. Kim, W.J. Dally, S. Scott, D. Abts, Technology-driven, highly-scalable dragonfly topology, ACM SIGARCH Comput. Archit. News 36 (3) (2008) 77–88.
[100] H.v. Koch, Koch snowflake, 2015. http://en.wikipedia.org/wiki/Koch_snowflake.
[101] KPTV-KPDX Broadcasting Corporation, Google opens new data center in The Dalles, 2015. http://www.kptv.com/story/28776478/google-opens-new-data-center-in-the-dalles.
[102] R. LeMay, Amazon confirms Sydney CDN node, 2012. http://delimiter.com.au/2012/06/20/amazon-confirms-sydney-cdn-node/.
[103] G. Lu, C. Guo, Y. Li, Z. Zhou, T. Yuan, H. Wu, Y. Xiong, R. Gao, Y. Zhang, ServerSwitch: a programmable and high performance platform for data center networks in: USENIX NSDI, 2011, pp. 1–14.
[104] Y. Liao, D. Yin, L. Gao, Dpillar: Scalable dual-port server interconnection for data center networks, in: IEEE ICCCN, 2010, pp. 1–6.
[105] C.E. Leiserson, Fat-trees: universal networks for hardware-efficient supercomputing, IEEE Trans. Comput. 100 (10) (1985) 892–901.

[106] Lenovo, ThinkServer RD630 Rack Server, 2013. http://shop.lenovo.com/us/en/servers/thinkserver/racks/rd630/.
[107] D. Li, C. Guo, H. Wu, K. Tan, Y. Zhang, S. Lu, FiConn: Using backup port for server interconnection in data centers, in: IEEE INFOCOM, 2009, pp. 2276–2285.
[108] D. Li, M. Xu, H. Zhao, X. Fu, Building mega data center from heterogeneous containers, in: IEEE ICNP, 2011, pp. 256–265.
[109] Z. Li, Z. Guo, Y. Yang, BCCC: An expandable network for data centers, in: ACM/IEEE ANCS, 2014, pp. 77–88.
[110] C. Li, A. Qouneh, T. Li, iSwitch: coordinating and optimizing renewable energy powered server clusters, in: IEEE ISCA, 2012, pp. 512–523.
[111] D. Li, J. Wu, On the design and analysis of data center network architectures for interconnecting dual-port servers, in: IEEE INFOCOM, 2014, pp. 1851–1859.
[112] Y. Li, F. Wu, X. Gao, G. Chen, SphericalMesh: A novel and flexible network topology for 60 GHz-based wireless data centers, in: IEEE/CIC ICCC, 2014, pp. 796–800.
[113] Z. Li, Y. Yang, ABCCC: An Advanced cube based network for data centers, in: IEEE ICDCS, 2015, pp. 547–556.
[114] W. Liang, X. Gao, F. Wu, G. Chen, W. Wei, Balancing traffic load for devolved controllers in data center networks, in: IEEE GLOBECOM, 2014, pp. 2258–2263.
[115] D. Lin, Y. Liu, M. Hamdi, J. Muppala, FlatNet: Towards a flatter data center network, in: IEEE GLOBECOM, 2012, pp. 2499–2504.
[116] D. Lin, Y. Liu, M. Hamdi, J. Muppala, Hyper-Bcube: A scalable data center network, in: IEEE ICC, 2012, pp. 2918–2923.
[117] Y. Liu, X. Gao, G. Chen, Design and optimization for distributed indexing scheme in switch-centric cloud storage system, in: IEEE ISCC, 2015, pp. 1–6.
[118] Y.J. Liu, P.X. Gao, B. Wong, S. Keshav, Quartz: a new design element for low-latency DCNs, in: ACM SIGCOMM, 2014, pp. 283–294.
[119] V. Liu, D. Halperin, A. Krishnamurthy, T. Anderson, F10: a fault-tolerant engineered network in: USENIX NSDI, 2013, pp. 399–412.
[120] Z. Liu, M. Lin, A. Wierman, S.H. Low, L.L. Andrew, Geographical load balancing with renewables, ACM SIGMETRICS Perform. Eval. Rev. 39 (3) (2011) 62–66.
[121] Z. Liu, M. Lin, A. Wierman, S.H. Low, L.L. Andrew, Greening geographical load balancing, in: ACM SIGMETRICS, 2011, pp. 233–244.
[122] H. Liu, F. Lu, A. Forencich, R. Kapoor, M. Tewari, G.M. Voelker, G. Papen, A.C. Snoeren, G. Porter, Circuit switching under the radar with reactor, in: USENIX NSDI, 2014, pp. 1–15.
[123] X. Liu, S. Yang, L. Guo, S. Wang, H. Song, Snowflake: A new-type network sturcture of data center, Chinese J. Comput. 34 (1) (2011) 76–85.
[124] V. Liu, D. Zhuo, S. Peter, A. Krishnamurthy, T. Anderson, Subways: A case for redundant, inexpensive data center edge links, in: ACM CoNEXT, 2015.
[125] L. Luo, CHINA'S FIRST CONTAINERIZED IDC, 2012. http://www.datacenterdynamics.com/focus/archive/2012/04/chinas-first-containerized-idc.
[126] Mircosoft, Microsoft Chicago Data Center, 2009. http://www.datacenterknowledge.com/inside-microsofts-chicago-data-center-microsoft-chicago-the-road-ahead/, http://www.datacenterknowledge.com/inside-microsofts-chicago-data-center/.
[127] Microsoft, Becoming carbon neutral: How Microsoft is striving to become leaner, greener, and more accountable, 2012.
[128] Microsoft, Microsoft Data Centers, 2014. http://www.globalfoundationservices.com.
[129] R. Miller, Sun MD Powers China's Earthquake Readiness, 2008. http://www.datacenterknowledge.com/archives/2008/11/25/sun-md-powers-chinas-earthquake-readiness/.
[130] R. Miller, Microsoft Migrates Azure, Citing Tax Laws, 2009. http://www.datacenterknowledge.com/archives/2009/08/05/microsoft-migrates-azure-citing-tax-laws/.
[131] R. Miller, Facebook Installs Solar Panels at New Data Center, 2011. http://www.datacenterknowledge.com/archives/2011/04/16/facebook-installs-solar-panels-at-new-data-center/.
[132] R. Miller, Microsoft's Energy Practices in Quincy Under Fire, 2012. http://www.datacenterknowledge.com/archives/2012/09/25/microsoft-quincy-energy-practices-under-fire/.
[133] J. Mudigonda, P. Yalagandula, M. Al-Fares, J.C. Mogul, SPAIN: COTS data-center ethernet for multipathing over arbitrary topologies, in: USENIX NSDI, 2010, pp. 265–280.
[134] J. Musilli, B. Ellison, Facilities Design for High-density Data Centers, 2012.
[135] NetApp, NetApp FAS6200 Series, 2013. http://netapp.conres.com/netapp-systems/FAS6200-Series.php.
[136] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, A. Vahdat, PortLand: a scalable fault-tolerant layer 2 data center network fabric, ACM SIGCOMM Comput. Commun. Rev. 39 (4) (2009) 39–50.
[137] F. Pilato, Is that a data center on your trailer? 2010. http://www.mobilemag.com/2010/02/03/is-that-a-data-center-on-your-trailer/.
[138] G. Porter, R. Strong, N. Farrington, A. Forencich, P. Chen-Sun, T. Rosing, Y. Fainman, G. Papen, A. Vahdat, Integrating microsecond circuit switching into the data center, in: ACM SIGCOMM, 2013, pp. 447–458.
[139] PowerLeader, PR2012RS, 2013. http://www.powerleader.com.cn/en/product/productshow.aspx?id=100000024824375&nodecode=105016003002.
[140] F.P. Preparata, J. Vuillemin, The cube-connected cycles: a versatile network for parallel computation, Commun. ACM 24 (5) (1981) 300–309.
[141] G. Qu, Z. Fang, J. Zhang, S.-Q. Zheng, Switch-centric data center network structures based on hypergraphs and combinatorial block designs, IEEE Trans. Parallel Distrib. Syst. 26 (4) (2015) 1154–1164.
[142] K. Ramachandran, R. Kokku, R. Mahindra, S. Rangarajan, 60GHz data-center networking: wireless ⇒ worryless?, Tech. Rep., NEC Laboratories America, 2008.
[143] A. Regalado, Who coined 'cloud computing'? Technol. Rev. (MIT) 9 (1) (2011) 1–10.
[144] A. Roy, H. Zeng, J. Bagga, G. Porter, A.C. Snoeren, Inside the social network's (Datacenter) network, in: ACM SIGCOMM, 2015, pp. 123–137.
[145] Ruijie, Ruijie RG-N18000 Series Data Center Switch, 2013. http://www.ruijie.com.cn/product/Switches/data-center-switch/RG-N18000.
[146] B. Schlinker, R.N. Mysore, S. Smith, J.C. Mogul, A. Vahdat, M. Yu, E. Katz-Bassett, M. Rubin, Condor: Better topologies through declarative design, in: ACM SIGCOMM, 2015, pp. 449–463.
[147] E. Schmidt, A Conversation With Google CEO Eric Schmidt, 2006. http://www.google.com/press/podium/ses2006.html.
[148] E. Schonfeld, Where are all the google data centers? 2008. http://techcrunch.com/2008/04/11/where-are-all-the-google-data-centers/.
[149] J.-Y. Shin, E.G. Sirer, H. Weatherspoon, D. Kirovski, On the feasibility of completely wireless datacenters, in: ACM/IEEE ANCS, 2012, pp. 3–14.
[150] J.-Y. Shin, B. Wong, E.G. Sirer, Small-world datacenters, in: ACM SOCC, 2011.
[151] Siemon, Data Center Storage Evolution, 2014. http://www.siemon.com/us/white_papers/14-07-29-data-center-storage-evolution.asp.
[152] Siemon, Siemon Launches the V600 600 mm Data Center Server Cabinet, 2011. http://www.siemon.com/us/company/press_releases/11-05-31-versapod-600mm.asp.
[153] R. Singh, D. Irwin, P. Shenoy, K. Ramakrishnan, Yank: Enabling green data centers to pull the plug, in: USENIX NSDI, 2013, pp. 143–155.
[154] A. Singh, J. Ong, A. Agarwal, G. Anderson, A. Armistead, R. Bannon, S. Boving, G. Desai, B. Felderman, P. Germano, et al. Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network in: ACM SIGCOMM, 2015, pp. 183–197.
[155] A. Singla, P. Godfrey, A. Kolla, High throughput data center topology design, in: USENIX NSDI, 2014, pp. 29–41.
[156] A. Singla, C.-Y. Hong, L. Popa, P.B. Godfrey, Jellyfish: Networking data centers randomly, in: USENIX NSDI, 2012, pp. 1–14.
[157] A. Singla, A. Singh, K. Ramachandran, L. Xu, Y. Zhang, Proteus: a topology malleable data center network, in: ACM SIGCOMM HotNets, 2010, pp. 1–6.
[158] Storage Networking Industry Association, Software Defined Storage, 2016. http://www.snia.org/sds.
[159] Y. Sun, M. Chen, Q. Liu, J. Cheng, A high performance network architecture for large-scale cloud media data centers, in: IEEE GLOBECOM, 2013, pp. 1760–1766.
[160] Y. Sun, J. Chen, Q. Liu, W. Fang, Diamond: An improved fat-tree architecture for large-scale data centers, J. Commun. 9 (1) (2014) 91–98.
[161] Telecommunications Industry Association, Telecommunications infrastructure standard for data centers, ANSI/TIA-942, Tech. Rep., Telecommunications Industry Association Standards and Technology Department, 2005.
[162] J. Touch, R. Perlman, Transparent interconnection of lots of links (TRILL): Problem and applicability statement, Tech. Rep., RFC 5556, Internet Engineering Task Force, 2009.
[163] I.J. Tu, Port of Quincy selling land to Microsoft for massive data center, 2013. http://blogs.seattletimes.com/microsoftpri0/2013/12/20/port-of-quincy-selling-land-to-microsoft-for-massive-datacenter/.
[164] W. Turner, J. Seader, K. Brill, Industry standard tier classifications define site infrastructure performance, white paper Tech. Rep., The Uptime Institute, 2005.
[165] US Department of Energy, Vision and roadmap: Routing telecom and data centers toward efficient energy use, in: Vision and Roadmap Workshop on Routing Telecom and Data Centers, 2009, pp. 1–27.
[166] H. Vardhan, N. Thomas, S.-R. Ryu, B. Banerjee, R. Prakash, Wireless data center with millimeter wave network, in: IEEE GLOBECOM, 2010, pp. 1–6.
[167] M. Walraed-Sullivan, A. Vahdat, K. Marzullo, Aspen trees: balancing data center fault tolerance, scalability and cost in: ACM CoNEXT, 2013, pp. 85–96.
[168] G. Wang, D.G. Andersen, M. Kaminsky, K. Papagiannaki, T. Ng, M. Kozuch, M. Ryan, c-Through: Part-time optics in data centers, ACM SIGCOMM Comput. Commun. Rev. 40 (4) (2010) 327–338.
[169] T. Wang, Z. Su, Y. Xia, B. Qin, M. Hamdi, NovaCube: A low latency Torus-based network architecture for data centers, in: IEEE GLOBECOM, 2014, pp. 2252–2257.
[170] T. Wang, Z. Su, Y. Xia, B. Qin, M. Hamdi, Towards cost-effective and low latency data center network architecture, Comput. Commun. 82 (2016) 1–12.
[171] C. Wang, C. Wang, Y. Yuan, Y. Wei, Mcube: A high performance and fault-tolerant network architecture for data centers, in: IEEE ICCDA, Vol. 5, 2010, pp. 423–427.
[172] W. Wei, X. Wei, G. Chen, Wireless technology for data center networks, ZTE Technol. J. 18 (4) (2012) 1–6.
[173] Wikipedia, IBM Portable Modular Data Center, 2011. http://en.wikipedia.org/wiki/File:IBMPortableModularDataCenter2.jpg.
[174] Wikipedia, Cloud computing, 2013. http://en.wikipedia.org/wiki/Cloud_computing.
[175] Wikipedia, RDMA over Converged Ethernet, 2016. https://en.wikipedia.org/wiki/RDMA_over_Converged_Ethernet.
[176] H. Wu, G. Lu, D. Li, C. Guo, Y. Zhang, MDCube: a high performance network structure for modular data center interconnection in: ACM CoNEXT, 2009, pp. 25–36.

[177] J. Xiao, B. Wu, X. Jiang, A. Pattavina, H. Wen, L. Zhang, Scalable data center network architecture with distributed placement of optical switches and racks, J. Opt. Commun. Netw. 6 (3) (2014) 270–281.
[178] Yahoo! YAHOO! NEW YORK DATA CENTER AND THE ENVIRONMENT, 2011. http://www.ydatacentersblog.com/blog/new-york/environment/.
[179] Y. Zhang, N. Ansari, On architecture design, congestion notification, TCP incast and power consumption in data centers, IEEE Commun. Surv. Tutor. 15 (1) (2013) 39–64.
[180] C. Zhang, F. Wu, X. Gao, G. Chen, Free talk in the air: A hierarchical topology for 60 ghz wireless data center networks, IEEE Trans. Netw. (2016) submitted for publication.
[181] W. Zhang, X. Zhou, L. Yang, Z. Zhang, B.Y. Zhao, H. Zheng, 3D beamforming for wireless data centers, in: ACM SIGCOMM HotNets, 2011, pp. 1–6.
[182] X. Zhou, Z. Zhang, Y. Zhu, Y. Li, S. Kumar, A. Vahdat, B.Y. Zhao, H. Zheng, Mirror mirror on the ceiling: flexible wireless links for data centers, ACM SIGCOMM Comput. Commun. Rev. 42 (4) (2012) 443–454.
[183] Y. Zhu, X. Zhou, Z. Zhang, L. Zhou, A. Vahdat, B.Y. Zhao, H. Zheng, Cutting the cord: a robust wireless facilities network for data centers, in: ACM MobiCom, 2014, pp. 581–592.
[184] C. Zibreg, iCloud solar farm is nearly complete, 2013. http://www.idownloadblog.com/2012/09/14/icloud-solar-farm-nearly-complete/.

Tao Chen is a Ph.D. candidate in computer science and technology at Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. He received the B.S. degree in software engineering from Harbin Engineering University, China, in 2008; the M.S. degree in computer science and technology from Hohai University, China, in 2011. His research interests include data center networking and distributed computing, especially in the area of wireless data center networks, and virtual machine migration.

Xiaofeng Gao received the B.S. degree in information and computational science from Nankai University, China, in 2004; the M.S. degree in operations research and control theory from Tsinghua University, China, in 2006; and the Ph.D. degree in computer science from The University of Texas at Dallas, USA, in 2010. She is currently an Associate Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. Her research interests include data engineering, wireless communications, and combinatorial optimizations. She has published more than 90 peer-reviewed papers and 6 book chapters in the related area, and she has served as the PCs and peer reviewers for a number of international conferences and journals.

Guihai Chen earned his B.S. degree from Nanjing University in 1984, M.E. degree from Southeast University in 1987, and Ph.D. degree from the University of Hong Kong in 1997. He is a distinguished professor of Shanghai Jiaotong University, China. He had been invited as a visiting professor by many universities including Kyushu Institute of Technology, Japan in 1998, University of Queensland, Australia in 2000, and Wayne State University, USA during September 2001 to August 2003. He has a wide range of research interests with focus on sensor networks, peer-to-peer computing, high-performance computer architecture and combinatorics. He has published more than 200 peer-reviewed papers, and more than 120 of them are in well-archived international journals such as IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, Journal of Parallel and Distributed Computing, Wireless Networks, The Computer Journal, International Journal of Foundations of Computer Science, and Performance Evaluation, and also in well-known conference proceedings such as HPCA, MOBIHOC, INFOCOM, ICNP, ICPP, IPDPS and ICDCS.