Interconnection Structures, Management and Routing Challenges in Cloud-Service Data Center Networks: A Survey

https://doi.org/10.3991/ijim.v12i1.7573

Ahmad Nahar Quttoum
The Hashemite University, Jordan
Quttoum@hu.edu.jo

Abstract—Today's data center networks employ expensive networking equipments in associated structures that were not designed to meet the increasing requirements of the current large-scale data center services. Limitations that vary between reliability, resource utilization, and high costs are challenging. The era of cloud computing represents a promise to enable large-scale data centers. Computing platforms of such cloud service data centers consist of large number of commodity low-price servers that, with a theme of virtualization on top, can meet the performance of the expensive high-level servers at only a fraction of the price. Recently, the research in data center networks started to evolve rapidly. This opened the path for addressing many of its design and management challenges, these like scalability, reliability, bandwidth capacities, virtual machines' migration, and cost. Bandwidth resource fragmentation limits the network agility, and leads to low utilization rates, not only for the bandwidth resources, but also for the servers that run the applications. With Traffic Engineering methods, managers of such networks can adapt for rapid changes in the network traffic among their servers, this can help to provide better resource utilization and lower costs. The market is going through exciting changes, and the need to run demanding-scale services drives the work toward cloud networks. These networks that are enabled by the notation of autonomic management, and the availability of commodity low-price network equipments. This work provides the readers with a survey that presents the management challenges, design and operational constraints of the cloud-service data center networks..

Keywords—Cloud-based Data Center Networks; Structures of Data Center Networks; Network Management; Routing in Data Center Networks.

1 Introduction

Over the last few decades, we lived and still living a huge Internet era and a big rise in the Web-based technologies that drive the theme of data centers to be more strategic than ever. Data Center Networks (DCNs) are mainly proposed to provide appropriate network structures with associated protocols that can interconnect differ-

ent servers holding varying applications, all together to act as a one single network [1].

In many organizations, the heartbeat of their business lies in data centers, where different parties (i.e. employees, partners, and customers) physically rely on the same data and network resources of a single DCN to interact, collaborate, and create services. As a consequence, such a theme receives a great attention from the Information technology (IT) specialists to enhance business processes, accelerate change, and improve productivity. Managers of DCNs face several challenges in satisfying such objectives, while demands on their networks are growing rapidly, and the needs became emerging to meet the economic and technical growth we are living nowadays. Mainly, an efficient DCN should provide (1) balanced network capacities [2]; (2) low-cost equipments; (3) high degree of scalability; and (4) reliability, where DCNs must be reliable with a substantial level of tolerance against network failures. Therefore, for DCNs managers, the challenge now is: (1) to efficiently utilize the network resources and maximize the number of provided services using the same amount of resources, while maintaining a reliable network state that is robust enough to link or server faults; (2) to provide scalable cost-effective interconnection structures that can accommodate large servers' populations, along with efficient bidirectional bandwidth capacities between the network components.

Today, most institutions still build redundant sites as backups, and usually, data on such secondary sites are manually replicated and managed. Although such backup sites represent an insurance policy in the case of failure, they also represent a non performing asset at most of the time [3]. This is considered a waste of time and power. However, by introducing the concept of *virtualization* [4], resources of a

DCN and its backup sites can be turned to ongoing available resources that can function in distributed scenarios. Regardless of the location, with such virtualization scheme, DCNs can provide lower costs, with higher performance and better reliability for its data and applications. In this direction, research in *cloud service* DCNs is tackling the issue of improving the services provided by DCNs. Existing interconnection structures, routing protocols and Traffic Engineering techniques, are all in the way to support virtualization management schemes.

DCNs interconnection structures play a great role in overcoming the aforementioned challenges, and provide for better virtualized cloud services while reducing the costs and networks' failure probability. These structures which define how the network components (i.e. servers, switches, transits, links) to be interconnected, and the characteristics of each component. Traditional interconnection structures usually come in the form of hierarchical trees that interconnects a set of connecting devices through a set of links [5]. The specifications and characteristics of such elements may vary, and hence, both performance and cost of the whole DCN may also get affected.

Routing is another crucial player in exploring the capacities of the DCNs structures [6]. Hence, several DCN routing protocols have been proposed in the literature. In general, such protocols differ from that of the Internet, where in DCNs, the routing protocols are specially designed to accommodate the DCNs topologies. In general, most basic routing schemes seek routes between any pair of network nodes with certain conditions (e.g. shortest routes, or other traffic metrics), in DCNs, routing is a bit

more sophisticated where it requires further constraints to be taken into consideration like energy and throughput. As this can be considered as a Traffic Engineering (TE) problem, the focus in DCNs is on the internal routing schemes (intra-routing), since most if the communication patterns of a DCN are internal ones [7]. Therefore, same in interconnection structures, the design of TE solutions in DCNs

should take into account the principles of reliability, load-balancing, and energy efficiency. In this study, we survey the state of the art of the research propositions that targeted the theme of DCNs through the last few years.

1.1 Paper Organization

The reminder of this survey is organized as follows: Section 2 discusses the cloud service DCNs, while Section 3 presents and compares the different interconnection DCNs structures. In turn, Section 4 provides an overview of the routing protocols and the TE techniques in DCNs. Finally, Directions for Open Issues and Future Research are presented in Section 5, and then Section 6 concludes the paper.

2 Cloud Service Data Center Networks

Within the IT community, *cloud computing* has emerged as a stunning theme that provides new management schemes to accommodate the growing challenge and the *dynamic* change in service demands, in efficient and cost-effective ways [8]. Relying on the traditional networks is not satisfying any more, where nowadays the trends are all toward dynamic *scalable* networks that can efficiently satisfy the changing demands and the varying workloads. The interest in cloud computing is increasing,

however, there still a kind of confusion in many areas as to what does cloud computing really mean? How it differs from the traditional enterprise networks? What are the benefits of adopting such a theme?, and what about the risks or side-effects if it is applied for the next generation management technologies?

2.1 What is Cloud Computing?

The theme of cloud computing can be basically defined as a computing style that employs high-level IT resources in *scalable-virtualized* scheme, in order to provide a wide platform for computing services [9]. Cloud networks deliver *services* rather than computing products, where the employed resources, software, and information are provided to the end-stations (i.e. users) as *utilities*. Hence, the users do not need to know how the network is implemented, how it is managed, or even what technologies are used. What concerns them is only that they have access to a reliable computing systems that can meet their applications requirements in a cost-effective way. Such architectures facilitate a dynamic on-demand access to a shared pool of reliable and highly available network resources, with easier management, and a pay-per-use pricing scheme!

2.2 Cloud vs. Traditional Computing Architectures

Fairly recently, the traditional distributed computing architectures were dominant in supporting most enterprise IT services. In such architectures, network resources are *physically* partitioned into several portions assigned to exclusively serve certain applications. This approach might serve well and provide for good performance, however, it requires significant investments and accurate forecasting techniques to efficiently utilize the provisioned network resources in a way to reduce the corresponding resource costs. On the other hand, the emergence of the *virtualization technologies* provides new management methodologies, these methodologies that allow high-levels of *autonomic management* for *virtually* partitioned resources [10]. With such virtualization scheme, as depicted in Figure 1, DCNs' managers started to have the ability to utilize their resources better and maximize the provided services'

agility. Accordingly, by virtually partition the resources of their physical server machines into several Virtual Machines (VMs), DCNs' managers could consolidate varying applications all at once in the same physical server machine [11], [12]. Indeed, in a cloud service DCN that contains thousands of physical server machines, with a modest amount of *virtualization* applied on top, such network can provide capacities for *millions* of end nodes running varying applications. Distributing the different applications over the VMs is done in a way that enables better utilization of the whole resources in the hosting physical machine. Resources like CPU, memory, and disk space. Hence, applications that use more CPU are consolidated with others that use less CPU but more memory or disk spaces, and so on [13]. The success of such virtualization technologies opened the path for developing reliable virtualized data center architectures [14] called *cloud data center networks*. Through such networks, the physical resources are leveraged to provide agile, and a scalable on-demand access to a pool of different services and IT applications. Not like traditional networks, in the cloud, a high-level of autonomic resource management is applied through a suite of virtualization software to handle dynamic demands fluctuations and changes in the network state.

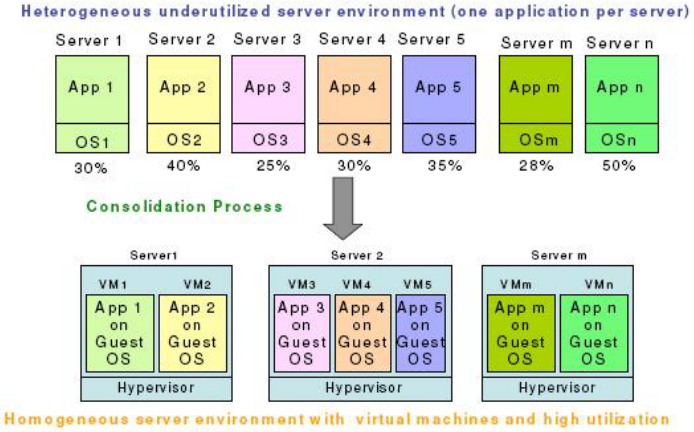

Fig. 1. Server's Resource Virtualization [11]

Why existing proposals for traditional DCNs do not work for cloud service DCNs? As mentioned above, there are many differences between both traditional and cloud DCNs. Differences are mainly related to management and architecture. Hence, it is expected that existing proposals for better traditional DCNs' designs and management may not work for the cloud DCNs, why? In fact there are many reasons:

- First, big portion of the costs in the traditional DCNs goes to the operational IT staff expenses, where due to partial automation [15], a ratio of 1 : 100 can describe the number of IT staff members to the number of server machines. This indicates more expenses to be paid to the IT staff. Moreover, this can lead to higher rates of human-based errors that cause large impact on the performance. In the case of cloud DCNs, the scenario is different, where due to mandatory automation requirements [16], the ratio of IT staff members to the network servers goes down to 1 : 1000. Consequently, the way the operational expenses are distributed in the two types of networks is different, and so, proposals for cost reduction in the traditional networks may not work well for the cloud ones.
- Second, cloud DCNs are usually built to support large size networks that interconnect hundreds of thousands of servers. Such environments necessitate the use of economic commodity servers, not like that in the traditional DCNs that contain less number of servers which can only be covered, comparatively, with non-commodity ones.
- Third, optimizing a traditional DCNs is usually represented by using less physical spaces, and less number of machines. Hence, optimization structures may follow the scale-up forms (i.e. north to south). This is usually expensive and requires replacing the commodity machines by other high-level expensive ones. The scenario in cloud DCNs is completely different, where the space is not a priority, and the scale takes the form of scale-out (i.e. east to west) using commodity machines that allow for scalable low-price interconnection structures. Such structures that distribute the workload over larger number of cheap network machines [17].

Traditional enterprise networks are moving toward the cloud, and so, proposed solutions for better cloud networks' structures are expected to enhance the traditional networks as well.

2.3 Benefits of Adopting Cloud Computing

Benefits of cloud services are many, and the adoption of such technology is driven by many factors. In DCNs, in addition to the *massively scalable* networks provided by the cloud service providers, such networks allow for delivering different applications with reliable, economic driven, and changing traffic patterns. Moreover, intelligent cloud networks can enable their providers to offer new applications and services that open the way for new markets. In general, the following are of the main benefits that a cloud DCN can provide to the end-users.

Economic Drivers: Starting by economics, the shrinking IT budgets along with the increasing demand for dynamic IT services, represents one of the leading drivers for adopting such cloud technologies. Building and maintaining a facility with thou-

sands or more servers is a quit expensive issue, isn't it? It is, certainly! Knowing that the biggest portion of DCN costs comes from the installed servers. According to [18], and as shown in Table 1, servers costs is dominant with around 45% of a total cost of a DCN, compared to only 15% to the other network equipments (i.e. switches, links, transits, ..., etc). So, why would anyone panic with building such networks that absorb high costs and time-consuming efforts for development and configuration, when instead, we can run our applications *now* and services on *someone else's machines* or network? Do we need to pay for a full DCN when we can pay only for the exact amount of resources we use?

Table 1. How Cost is Distributed in a DCN [18]

Average Cost	Component	Sub-Components
45%	Servers	CPU, Memory, Storage
25%	Infrastructure	Power Outlets, Cooling
15%	Power Utilities	Electrical Costs
15%	Networking Equipments	Switching Machines, Links, Transits

Scalability: From the market-perspective, the ability to support a rapid growth of dynamic service demands, without compromising the network's efficiency and the network cost is a critical issue. Converting both infrastructure and operational costs into a *scalable expenses* that reflect the actual use of the resources is a promising option for many operators, especially those interested in *getting more while spending less* in their infrastructures [19]. Moreover, the *logically* infinite on-demand capacities of the cloud DCNs represents another attractive feature that provides fast support for any immediate-demanding applications to be deployed easily, without complex management and time consuming operations.

Resource Utilization: It also provides for efficient resource utilization based on the real-time *dynamic* demands of the provisioned applications. In cloud DCNs, operators have the ability to meet their changing demands that varies between low to peak load states. This delivers better use of the available resources, reduce blocking rates, and cost of the provided services [18].

Ease of Maintenance: Another attractive feature of these cloud DCNs is the ease of maintenance. As an advantage of virtualization, cloud network architectures are built form less *physical* machines compared to the ordinary computing environments. Intuitively, a network with less hardware devices requires less efforts for maintenance and management. This not only reduce the time of maintenance, but also the number of IT technicians needed to handle the integrity of the network. It is worth to mention that such a cloud scheme allows for a *win-win scenario*, in which, the cloud providers can set up their networks to run the required applications in a cheaper, easier to manage, time-saving, and reliable manner. This lies in the interest of the end-users. Same way, these cloud providers are making money from such type of business!

2.4 Risks for Adopting the Cloud DCNs

Although cloud service DCNs are proposed to benefit the traditional enterprise users, there still few points of stress that may negatively impact the performance and the compliance of the service level agreements in such type of networks. For cloud service providers, to provide efficient services to the DCNs' end-users, features like performance, availability, and security are all of top importance.

Virtualized networks are proposed to enhance performance and availability, however, a non-efficient virtualization scheme of the servers' physical resources may increase that servers latency, and decrease the its reliability [20]. This can badly affect the applications' performance, and the systems' availability. Hence, the way in which the servers' resources are *virtually partitioned* among the provided applications is crucial, and therefore, such a process must be done with a high level of concern.

Security or the applications integrity is also another point of challenge in cloud service DCNs. The idea in cloud DCNs is to integrate varying applications, and their related computing environments in single servers. However, and not to mention the increase in heat and power consumption, such integration or consolidation within the same physical machine magnifies the problem of *single point of failure*. This deepens the security threats for such points, affecting its reliability, availability, and performance [9].

However, let us be more optimistic, relying on the fact that, by days, the offered security protections provided by the service providers are only getting better. This gives us the hope that opportunities of better management technologies are on the way, and next generation DCNs are promising to provide more robust deployments.

3 Interconnection Structures for Data Center Networks

The growth of proficiency in building clusters of commodity PCs has enabled the theme of integrating the provisioning process of both, computation storage and computation power in a cost-efficient scheme. In large institutions like universities, clusters can consist of thousands of nodes. Building a communication structure for such high-scale clusters can follow one of two options. First, to use high-level hardware components with specialized protocols like that in Myrinet [21]. Such an option can deliver high-scale clusters that can interconnect thousands of nodes, while providing high bandwidth capacities between the connected entities. However, these high-level (non-commodity) hardware components impose *expensive* funding, and usually require *special configurations* to be compatible with the TCP/IP applications [22]. The second option is to use cheaper hardware components (i.e. commodity Ethernet switches) to handle the interconnections among the cluster nodes. This allows deploying familiar management infrastructures, without any modifications in the network applications and operating systems.

One major problem in building *high-scale* clusters is the poor aggregate bandwidth capacities in the network [22], where such bandwidth capacities do not scale well with the cluster size, and unfortunately, achieving better capacities comes with a non-linear increase in cost that depends on the cluster size. More precisely, bandwidth

capacities in large size clusters may become *oversubscribed* by certain percentages due to the network hierarchy and the different physical specifications of the network components. Even when employing high-level hardware components, statistics show that resulting topologies could only support 50% of the network-edge aggregate bandwidth capacities.

Accordingly, the option of building such communication structures using commodity hardware structures can be considered the *dominant*. Accordingly, network architects are working to design efficient interconnection structures that deliver high performance networks, along with low-cost infrastructures compared to that of today's high-end solutions. In this section, we survey the state of the art in regard to the *interconnection structures* in DCNs.

Our focus will mainly target those structures that employ commodity designs, for which we will discuss how it works, show their topologies, and present their switching techniques. Moreover, we will also provide a comparison in terms of the supported bandwidth capacities, and its associated cost metrics.

3.1 Background

Traditional interconnection structures usually come in the form of a hierarchal tree that consists of routing and switching elements. The specifications and characteristics of such elements may vary, and hence, both performance and cost of the whole DCN may also get affected. As an example, one structure may employ the commodity GigE switches, while it is 10 GigE switches in other structures [24].

Efficient DCN structures need to satisfy the following design goals:

- *Scalability*; DCN structures need to be designed in a way that easily allows for network expansion and dynamic changes [25]. This involves the ability to smoothly accommodate future upgrades of servers and any other networking equipments.
- *Reliability*; any proposed DCNs structures must be reliable enough, with high degrees of tolerance against network failures.
- *Cost-efficiency*; DCNs in general need to provide for cost reduction in terms of both, network assets and power requirements.
- *Resource capacities*; to avoid blocking and bottleneck states, DCNs should have the ability to provide for high aggregate capacities.

In the tree-based structures, a hierarchy of network switches is used to interconnect the hosted servers. The current DCN practice is to use the switch-based tree structure to interconnect the increasing number of servers. At the lowest level of the tree, servers are placed in racks and connected to an edge level rack switch (usually called Top of Rack switch). At the higher levels, ToR switches are interconnected using higher layer switches with capacities to aggregate the traffic of hundreds of ToRs. In this context, it is worth to mention that in such scenarios, those root switches may represent bandwidth bottlenecks that may be central points of failure.

3.2 The Fat-Tree Structure

Driven by the price difference between the commodity and non-commodity network switches, network architects started to have the tendency build their large scale DCNs using many commodity switches rather than fewer non-commodity expensive ones. With such price incentive, and to deliver high rates of bandwidth, in 1950*s* Charles Clos chose to build a multi-stage telephone switching network from interconnected commodity switches [26]. This interconnection scheme is known today as *Clos network*.

As an instance of the Clos network, the fat-tree topology [22], [27] was proposed in the form of a multi-rooted tree that employs commodity Ethernet switches to interconnect the DCNs' servers. In fat-tree, *redundant* aggregation points are used to reduce the problem of bandwidth bottlenecks and central points of failure.

Topology: In the fat-tree, the topology is organized as depicted in Figure 2, where there is a tree-based hierarchy that consists of a set of layered network switches that are used to interconnect a group of network *servers*. Each set of servers is placed in a rack, and each rack has a *edge switch* that interconnects all of the underlying servers together, and to the rest of the network.

Edge and *aggregate* switches are grouped in pods, where a for fat-tree structure that has *k* pods, there are *k* switches in each pod (*k*/2 edge and *k*/2 aggregate switches). Edge switches come with *k* ports, *k*/2 of these ports are used to create direct connections with *k*/2 servers, and the remaining *k*/2 ports are used to get connected with *k*/2 other aggregate switches. In the most higher level of the network structure, $(k/2)^2$ *core* switches are employed to interconnect the whole underlying aggregate switches, each core switch comes with *k* ports that interconnects the underlying *k* pods.

Based on these specifications, a designer can define the number of physical hosts (i.e. servers) a DCN can support based on the switch degree. True, where a fat-tree DCN that is built from *k* port switches can be used to physically interconnect $k^3/4$ servers [22].

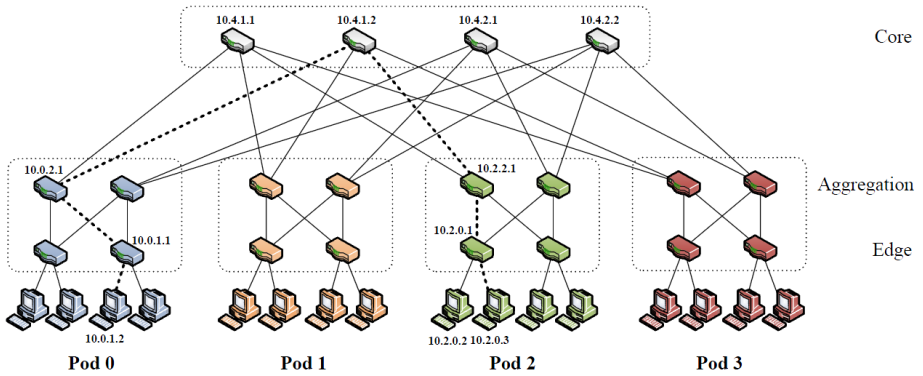

Fig. 2. A Fat-Tree interconnection Structure, with *n* = 4 [22]

Features and Summary: Switches used in the fat-tree topology are all identical, which means that all network levels of a fat-tree DCN come with the same specifications of the switching components[26]. This allows cheap commodity switches, and so *low-cost DCNs*. Hence, it is worth to note that the phrase *hierarchy* in fat-tree refers to the structural level only, not the types of those used equipments. In terms of bandwidth capacities, fat-tree is designed with the intention to support *full bisectional bandwidth* between the network servers by the use of multi-rooted trees [28]. This is assumed to deliver non-blocking communication sessions between the interconnected servers.

Multi-rooted trees are built with multi core switches. Those that interconnect the huge number of aggregate/edge switches that gathers the tree branches (i.e. servers). This means additional costs. What is more, in regard of scaling capacities, fat-tree structures come with limited number of the ports available physically at their switches.

3.3 The DCell Structure

Motivated by the goals of providing scalable interconnection structures, high bandwidth availability among the interconnected hosts, and avoiding single points of failure, the DCell is proposed in [29] as a recursively-defined structure to interconnect the DCNs' servers. Not like the fat-tree, in DCell, the interconnection between the network entities is mainly built through the servers.

More precisely, high-level DCell are built from many low-level ones, where each server is connected to different pods of DCells via multiple links, in a way that the low-level DCells form a fully-connected graph, see Figure 3.

In terms of structure, DCell employs commodity low-level switches to *scale-out*, instead of the scale-up approaches that requires expensive high-end switches. As a server-centric structure [30], it provides for a double exponential scale with respect to the employed servers' node degree. In practice, a DCell with a small server degree (e.g. say: 4), can support interconnection to as many as several servers without the need of those expensive high-end switches.

Being a structure with no single aggregation points, DCell can be considered fault tolerant with no central points of failure. Moreover, fault tolerance also comes from the rich physical connectivity a DCell has. However, it be considered as a structure that requires high costs for wiring among the servers, since it uses more and longer communication links compared to that in the tree-based models.

Topology: The DCell structural topology is organized as depicted in Figure 3 which comes as a level-based structure. As shown in the figure, n servers are interconnected via $n-port$ switches to build a $DCell_k$. In such recursively-defined topology, a $DCell_{k+1}$ is built from $n + 1$ units of $DCell_k$.

Accordingly, if n_k servers are required to build a $DCell_k$, and $n_k + 1$ units of $DCell_k$ are required to build a $DCell_{k+1}$, then we can generally say that the number of servers n_{k+1} in a $DCell_{k+1}$ is given by $n_k^2 + n_k$.

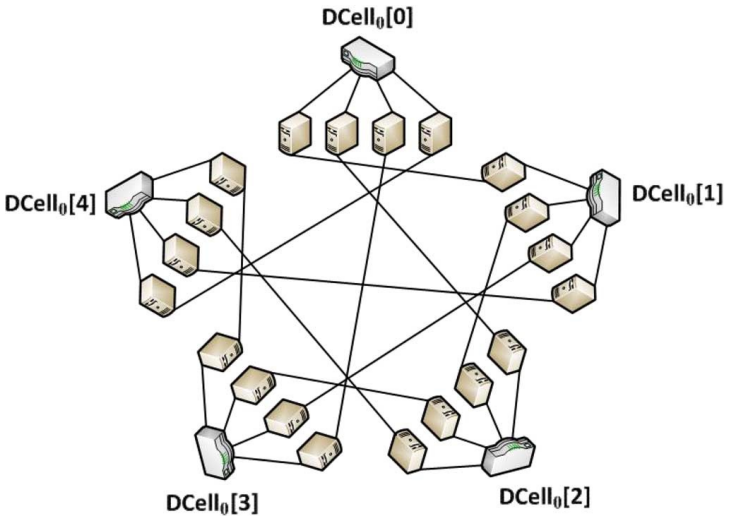

Fig. 3. A DCell Interconnection Structure with $n = 4$ [29]

Switching: In *DCell*, switching is done with the goal of connecting huge number of servers in a way that accommodate for dynamic traffic changes. Accordingly, in DCell interconnection structures, the use of the *global link-state* routing schemes is not a recommended option as they create huge control overhead in the network [29]. To avoid points of failure and bandwidth bottlenecks, the Open Shortest Path First (OSPF) routing protocol is not a routing option too [31], as it imposes huge traffic overhead. Therefore, the authors of [29] proposed a *fault-tolerant* routing protocol that is claimed to provide a decentralized near-optimal routing scheme. Such fault-tolerant protocol claims to effectively handle various failures that may vary between hardware, software, and power issues.

3.4 The BCube Structure

Designed for shipping containers and modular datacenter networks, the BCube structure is proposed in [32] as a *server-centric* network interconnection structure that employs multiple commodity switches in a hierarchal-style to interconnect large numbers of multi-port servers.

In BCube, as shown in Figure 4, commodity four-port switches are employed to create multiple parallel short paths between pairs of servers in a structure that interconnects sixteen different servers. This not only provides for *high one-to-one bandwidth*, but also *improves fault tolerance* and *load balancing*. BCube accelerates one-to-many and one-to-all traffic. Moreover, due to its low diameter, BCube provides *high network capacity* for all-to-all traffic [32].

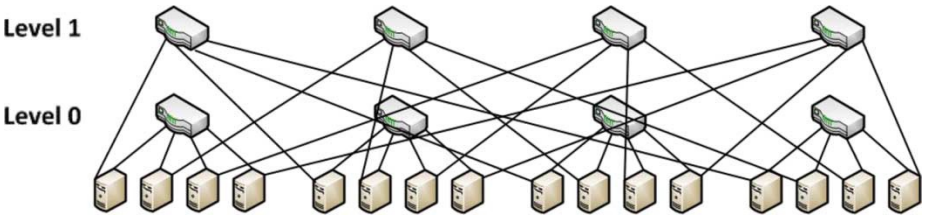

Fig. 4. A BCube Interconnection Structure with $n = 4$ [31]

Compared to the *fat-tree* and *DCell* structures, the authors of [32] claimed that BCube is better than these two structures as it does not have performance bottlenecks, and provides for larger network capacities with direct one-to-x support without minimal upgrade requirements. On the contrary, such direct connection between the network switches and servers *imposes high wiring expenses*.

Topology: As mentioned before, in BCube, the structure is mainly composed of commodity low-level switches with limited number of ports, and multi-port servers that are interconnected to the switches of the upper layer**s**. Being a recursively-defined structure, higher-level BCubes are built form lower level ones. Given a $BCube_k$ that interconnects n servers through n *−port* switches, $BCube_{k+1}$ can be built from n units of $BCube_k$, using n n*−port* switches. $BCube_k$ can interconnect n^{k+1} servers, each comes with $k + 1$ ports connected to $k + 1$ levels of switches, each *level* consists of n^k n *−port* switches. It is *worth to note* that in the BCube topology, switches are only connected to servers and not to other switches. Consequently, switches in BCube are considered as dummy crossbars that provide only the interconnection among the underlying servers.

Switching: In BCube, the DCN servers provide multiple ports. Servers are interconnected to multiple layers of commodity switches providing multiple short paths between the interconnected servers. Such richness of parallel paths can provide higher aggregate bandwidth capacities, along with improved *fault-tolerance*. Taking the advantage of this multi-path property, BCube runs a source routing protocol that is installed over the network servers to balance the traffic and handles the link failures. In the case of server or switch failures, such protocol allows for graceful degradation in the bandwidth capacities of the network. The authors of [32] proposed a new BCube routing protocol suite. In their work they claimed to provide a fast packet forwarding protocol that can decide the next hop of any received packet through one table look-up process. This proposed protocol can be implemented in both software and hardware.

3.5 The FiConn Structure

FiConn is also a server-centric structure, however, its new contribution comes in the utilization of the servers' Ethernet *backup ports*. More precisely, the idea came form the observation that the commodity server machines used in today's DCNs usually have two built-in Ethernet ports, one for connection with the switch and other for

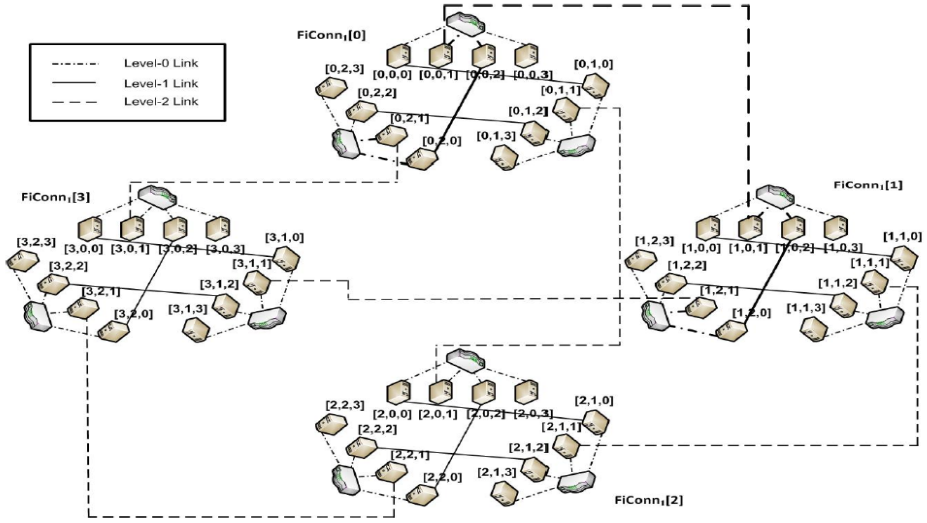

Fig. 5. A FiConn Interconnection Structure with n = 4 [28]

backup reasons. Accordingly, the authors of [28] proposed that activating the servers' backup ports for network connections can represent an opportunity for building lower cost interconnection structures. Indeed, having more ports at the servers level can provide direct interconnection sessions between the different servers without the need to go through switching machines. In this way, low-level commodity switches can handle the aggregation issues to form a scalable effective structures.

As depicted in Figure 5, a high-level FiConn is constructed by many low-level FiConns. When constructing a higher-level FiConn, the lower-level FiConns use *half* of their available backup ports to get interconnected to other servers, and form a mesh. In this way, the number of provided servers grows rapidly with the FiConn level. Not like the Fat-Tree, where the scale is limited by the number ports at the switches, and neither like the DCell that requires higher number of server ports to scale. However, FiConn works only with servers that have a node-degree of two. Although FiConn can provide scalable low-price structures, still, it adds higher control overhead if compared with those of tree-based structures. Moreover, employing low-cost commodity switches can reduce the network cost, but on the other hand, such server-centric structure adds more wiring costs besides the *higher CPU overhead* for resource forwarding at the servers side. In terms of bisectional aggregate bandwidth capacities, Fat-Tree and DCell proves to provide better capacities. Hence, we can easily recognize that lower switching costs in such a structure comes with a *tradeoff* with the provided bandwidth capacities.

Topology: Mainly, as shown in Figure 5, the FiConn topology consists of multiple servers of node-degree two, with one level of commodity low-price switches. In a recursively defined structure, high level FiConns are constructed form low level ones. Compared to the Fat-Tree, if we assume having N servers, then the number of *n−port*

switches needed to interconnect the FiConn structure is given by N/n, while it is $5N/n$ for the Fat-Tree.

*FiConn*0 is the basic FiConn level, and it is composed of n servers and $n-port$ switches. Usually the number of servers in FiConn structures is even. $FiConn_k$ is build from a set of $FiConn_{k-1}$ entities that are interconnected together using their servers' backup ports. Hence, to build a $FiConn_k$ structure, you need a set of $FiConn_{k-1}$ entities interconnected through their backup ports. If you denote the number of servers that have backup ports in a $FiConn_{k-1}$ structure by S, then you need g_k structures of $FiConn_{k-1}$ to build a $FiConn_k$. This g_k is given by:

$$g_k = S/2 + 1$$

Where only $S/2$ servers of each $FiConn_{k-1}$ use their backup ports to interconnect to another $S/2$ structures of $FiConn_{k-1}$ through their backup ports as well. These selected $S/2$ servers are call *level* $-k$ servers.

Switching: In FiConn, servers are configured with two-ports, where these servers are connected to commodity low-level switches. The servers are configured to use half of their available back-up ports for interconnection with other servers in other FiConns to form a kind of mesh. Routing in FiConn structures is claimed to balance the usage over the different network links, and at the same time improve the resource utilization according to the dynamic traffic changes. Deploying the *traffic-oblivious* routing scheme in FiConn shows good performance in balancing the traffic loads over different levels of the network links, but on the contrary, such scheme has the following *limitations*: (1) For a pair of communicating servers, it is only allowed to use two of the available backup ports. Using more ports is not allowed, even if this was motivated by improving the resulting end-to-end throughput. (2) Due to such rigid settings, it cannot dynamically cope with the real-time changes in the traffic demands in the network.

Therefore, to overcome these limitation, the authors of [28] proposed the *traffic-aware* routing scheme. Briefly, this traffic-aware scheme does not rely on doing the traffic scheduling of the network on central server entities, but instead, it distributes that over the whole network servers. Accordingly, each server will be responsible for balancing its outgoing traffic over its outgoing ports, where ports with the higher bandwidth availability are always selected to hold the new outgoing traffic.

3.6 Summary

This section of the survey discussed four main structures proposed in the literature for the DCNs, i.e., Fat-Tree, DCell, BCube, and FiConn. For which, the survey reviewed their topological, cost, and switching characteristics. Through such presentation, we observed that in DCNs, the objectives of scalability, fault tolerance, and bandwidth capacities get higher priorities than other metrics. In this regard, Table 2 provides an analysis that summarizes the behavior of the aforementioned four structures. In this context, it is worth to note that although those server-centric proposals provide for

scalable structures, still they impose *huge cpu overhead* at the servers' side. This can be considered as a point of limitation, since if we refer to Table 1, we can clearly notice that when it comes to cost, servers take the lead of all other assets a DCN requires.

Table 2. Structural Proposals for Data Center Networks (DCNs)

Proposal	Structure	Switches Cost	Cables Cost	Routing Overhead	Scalability	Bandwidth	Fault-Tolerance
Fat-Tree	Switch-centric	many commodity units	low	on switches	limited by switch degree	best	High Tolerance
DCell	Server-centric	few commodity units	high	on servers	limited by server degree	Less than BCube	Average Tolerance
BCube	Server-centric	more commodity units	high	on servers	Less than DCell	high	High Tolerance
FiConn	Server-centric	few commodity units	average	on servers	limited by backup ports	Less than DCell	Below-Average Tolerance

4 Routing and Traffic Engineering in Data Center Networks

As shown in Section 3, different interconnection structures are proposed in the literature for the DCNs. Challenges are many, and so do the objectives to be achieved from these different structures. Such challenges and objectives vary between cost, reliability, scalability, and bandwidth capacities. An important player in the efficiency of such interconnection structures is the *routing protocols*. These protocols that help in exploring the bandwidth capacities that would be available between the interconnected machines in a DCN. Providing significant bisectional bandwidth capacities is a fundamental aspect for DCNs. Accordingly, intensive efforts and research works are spent to deliver efficient interconnection structures that allow for scalable, and non-blocking topologies.

General speaking, DCNs interconnection structures can be categorized into a set of two main schemes, a *server-centric* and a *switch-centric*. Each scheme has certain characteristics that distinguish it from the other. Different from that proposed for the Internet, researchers have developed a set of routing protocol schemes that are specially designed to suite the DCNs' topologies.

4.1 Routing Schemes

To review the routing schemes of the proposed interconnection structures like Fat-Tree, DCell, BCube, FiConn and others, in this section, we are categorizing them as follows:

Server-Centric Schemes: Recognized from the name, in server-centric schemes, the interconnection responsibilities in a DCN are mainly placed onto servers. Consequently, the servers play a double role, where in addition of being end-hosts, they are relay nodes for other communication paths in the network. FiConn [28], DCell [29], and BCube [32] are all structures that fall into this category.

Switch-Centric Schemes: Not like server-centric schemes, switches are the only relay nodes in the switch-centric DCNs. In such a scheme, all interconnection session among the network hosts pass through the upper-layer switches (i.e. edge, aggregation, and core). In general, such interconnection structures follow a special instance of the Clos topology [26] (proposed for telephone networks in 1950s), which named the Fat-Tree. In this Fat-Tree structure, commodity Ethernet switches are generally used to interconnect massive number of hosts. In such DCNs, the proposed routing schemes usually follow the topological hierarchy. The structure of Portland [7] and VL2 [33] are also examples for other models that fit in this category.

4.2 Data Forwarding Techniques

A huge portion of the Internet communications and their related computing and storage processes is migrating toward the DCNs. To accommodate this, DCNs must be highly engineered to support scalable and fault-tolerant data center networks. Current routing and data forwarding protocols that are deployed in the DCNs are originally developed for Local Area Networks (LANs). However, such protocols do not show good performance when deployed for networks that interconnect large number of hosts like that of a medium size DCN. Assume a DCN that hosts 100,000 servers, and virtually, each servers runs 20 Virtual Machines (VMs). This comes to approximately 2,000,000 IP and MAC addresses. Not to mention the number of required switches, a network with such size imposes a huge management overhead at the provider's side.

For DCNs, an efficient routing and forwarding protocol should support for scalable and fault-tolerant environment. Such environments that consider [6]:

- Easy migration of any VM to any physical server in the network. This should involve keeping the original IP addresses to avoid breaking old TCP connections and any other application-level sessions.
- Self-learning switches.
- Fault-tolerance scaling.
- No forwarding loops.
- Hosts in the DCN can communicate with each other efficiently over any available path in the network.

Existing layer 2 and layer 3 network protocols face some challenges in satisfying such requirements. However, to some extent, achieving *the first two points* requires deploying a layer 2 fabric through the entire DCN. In a layer 3 fabric, this requires high management overhead for configuring the network switches, each individually, with their sub-network information to distribute the appropriate IP addresses among the network hosts, after being synchronized with DHCP servers. Further, this makes the issue of VMs migration more complicated, since by migrating to another sub-network, VMs should switch their IP addresses to meet the addressing scheme of the new physical location.

Concerning the *scaling requirements*, layer 2 fabrics do not represent an optimal option. Indeed, broadcasting at layer 2 is a challenging issue. Satisfying such scale targets requires special protocols that can quickly propagate the urgent topology up-

dates to their points of interest. Unfortunately, current routing protocols (e.g. OSPF) are broadcast ones. This imposes a kind of configuration overhead, which

contradicts with the second point. Regarding the *forwarding loops problem*, neither layer 2 nor layer 3 protocols can avoid it, since such loops can possibly happen during routing convergence especially in topologies of DCNs which provide for redundant path between the different couples of source-destination servers. Though, it is less of an issue in layer 3 as the Time To Live (TTL) counter limits the packets resource consumption while updating the forwarding tables. For the last point and the *agility* issue, it still seems impractical to build layer 2 forwarding tables with millions of entries. Therefore, really scalable/agile network fabric for DCNs is still not yet achieved, where as presented above there is always a tradeoff between the available network protocols in terms of flexibility, scalability, reliability and performance.

Addressing in layer 3 is done by assigning IP addresses to the network hosts, all in a hierarchical way following the host's directly connected switch. However, layer 3 forwarding has the following limitations [7]:

- Network topology updates (e.g. adding a new switch) can be considered as risky processes, where manual configuration by the network provider is required.
- Improper synchronization and error configurations (e.g. DHCP servers, sub-networks identifiers) among the network components can lead to non-reachable network segments.
- Poor support for scalability and servers' virtualization.

Consequently, to reduce the administrative overhead and avoid any risky configurations, some networks deploy layer 2 forwarding that is performed based on MAC addresses. But still, layer 2 fabrics have the following limitations:

- Relying on the Ethernet bridging techniques limits the scalability properties of the network. Assume a DCN with 100,000 hosts, how to support broadcast through the entire network?
- In topologies that have multiple equal cost paths, how to enhance the performance while relying on a single forwarding tree?

Some propositions suggest a hybrid ground that integrates the positive characteristics of layers 2 and 3, while reducing the problem of broadcasting in layer 2 and providing higher level of scalability. Employing the technology of Virtual LANs (VLANs) in layer 2 fabrics facilitate crossing multiple switch boundaries. However, the VLAN technology itself has some problems like:

- Splitting the network to virtual separate domains requires bandwidth resource reservation for each VLAN at each switch. Such allocations, if not dynamically allocated, provide less flexibility and low bandwidth utilization rates.
- With such broadcast domains, switches must keep a state for each host they connect. This limits the scalability and network agility.
- In VLANs, the use of a single forwarding tables requires large update messages between the network switches which affects the network performance.

4.3 Traffic Engineering in Data Center Networks

Internet routing schemes usually look for routes that connect two network nodes under certain latency constraints, however, in DCNs the scenario is somehow different, where other sophisticated requirements are taken into consideration like high reliability, consumed energy, and specific performance metrics [6]. Satisfying such constraints requires special Traffic Engineering (TE) efforts. In TE, network

providers are adapting the routing decisions of the network traffic according to the network conditions. This can help in optimizing the network performance in accordance to the dynamic traffic status and the behavior of the transmitted data patterns. In DCNs, traffic is divided into two main parts, inter-DCN and intra-DCN traffic. What concerns more is the intra-DCN one, since the performance of a DCN mainly depends on its internal communication patterns [34]. Inter-DCN traffic are routed via the well-known Border Gateway Protocol (BGP) as any other external traffic in the Internet.

A challenging problem for TE in DCNs is how to expect the traffic patterns. For different applications, such patterns may vary significantly and in some cases the traffic traces can have a kind of confidentiality. Moreover, DCNs are growing rapidly, and the need for scaling is evolving, which adds more complexity and challenge in how to efficiently control and manage such expands and variety of applications.

Design Principles: When proposing a DCN TE model, the following principles should be taken into account:

- Reliability: The first goal when proposing a TE model should be optimizing the routing scheme to provide a reliable and fault-tolerant data forwarding patterns. Mostly, such DCNs carry important information that provides crucial services and important application for different business operations, and end-users. Thus, a successful DCN is that which provides reliable and robust services to its users. Consequently, reliability is considered as a point of concern for both DCN service providers and their subscribers.
- Resource Utilization: Reliability and fault-tolerance highly depend on how the network resources are utilized. Better bandwidth utilization allows for higher throughput, lower blocking, and less latency. Moreover, it greatly affect the both capital and operational expenses of the network. Hence, an efficient TE model should adapt the routing schemes to utilize the network bandwidth capacities in order to serve varying applications, each with different traffic pattern, while providing Quality of Service (QoS) and performance guarantees.
- Power Expenses: To provide efficient services with competing prices, DCNs providers should try to minimize the network expenses the most possible. Operating network machines consume energy, in this context, efficient TE models should direct the routing schemes to use the least possible number of links and switches. This can reduce the energy expenses, and consequently maximize the DCNs profits while allowing them to offer their services with market-competing prices.

How TE in Data Center Networks differ from that in the Internet? Designs of the DCNs are different from that of the Internet [6], where some features in DCNs

requires new design directions. Accordingly, when designing a TE model for a DCN, the following points should be considered:

- Node Location: Traditional TE problems usually deal with fixed source-destination locations, and the traffic is distributed over the Internet links. In DCNs, the scenario is different, where a VM that runs service *x* can dynamically change its location for better performance and agility issues. This can allow for adopting better more efficient routing schemes.
- Topology: Interconnection structures in DCNs are mostly symmetric, having multiple paths between the interconnected network servers. TE engineering models that utilize such redundant paths for performance utilization require special routing schemes different from that in the Internet.
- Centralized TE: Not like the Internet, DCNs represent a convenient network style in which centralized TE and management schemes can be efficiently deployed. Such schemes where a centralized network operator entity can control and collect performance metrics of the whole underlying network components. Although this may impose higher control overhead, but provides for simplified implementation.
- Infrastructure: Driven by the cost-efficiency requirements, DCNs are usually built from commodity layer 2/3 switches with higher link densities. So, compared to the Internet, DCNs nodes are not expected to be as reliable as the high-level routers with more open cost availabilities.
- Multi-rooted Designs: To provide full bisectional bandwidth capacities in commodity DCNs interconnection designs, multi-rooted tree topologies are a necessity, where aggregating bandwidth capacities over such multi-rooted paths may deliver the desired capacities among the network hosts. In the Internet topologies, such redundant paths are not allowed as it creates the undesired forwarding loops.

Moreover, compared to the Internet, routing in DCNs has the following unique characteristics:

- Common Topologies: With the reason of increasing the network performance and scalability, DCNs designs are mostly employing very similar routing protocols.
- Short packet life: Statistics show that most of the traffic patterns in DCNs are of short-life ones, hence this adds some challenges in expecting the dynamic traffic patterns and employ the proper TE design.
- Agility: In DCNs, agility is necessary for load-balancing and availability concerns. In regard to the Internet traffic Internet.

5 Directions for Open Issues and Future Research

The works surveyed throughout this papers shed the light on the issues of structural and routing challenges in the areas of cloud DCNs. Proposals of network structures are many, however, only those who provide for efficient services and high performance metrics prevail. Providers of cloud DCNs will always seek for those proposals that allow for easy scale and agile topologies. Such topologies that

do not require frequent structural updates, while providing for sufficient service levels at competing prices. In this context, and beside the operational behavior of any proposed structure, the providers of the cloud DCNs must tackle the challenge of efficient service provision and service-leasing issues. Indeed, the theme of cloud DCNs is mainly the provision of computing resources in the form of services. Such services vary between software, platform, and infrastructure. Efficient service provision in such environments comes through the following: (1) efficient resource utilization, (2) efficient service allocation schemes. Therefore, we consider the aforementioned provision challenge points as hot research topics and open fields for deep tackle. Briefly, we will discuss them in the following:

5.1 Efficient Resource Allocation

Almost all Cloud DCN structures proposed in the literature provide for high resource capacities, such as those of switching and links bandwidth capacities. Richness in resource capacities is a fundamental aspect for such sort of networks, indeed, as the theme of cloud DCNs is to provide the networks end-user (i.e. service tenants) with their required levels of services with reasonable price units. This type of business necessitates certain levels of guarantees that provide the end-users with the necessary satisfaction rates in regard to the resource availability and service price units. To provide the aforementioned guarantees, cloud DCN resources need to be smartly utilized. In this context, various research works are proposed in the literature. Among the reviewed proposals, many proposed approaches modeled the problems of resource allocation/reservation as auctions where the cloud DCN resources are leased to certain bidders who satisfy predefined conditions. In [35], the network's bandwidth *reservation* process is modeled using a *Vickrey-Clarke-Grove* (VCG) auction [36], a mechanism that is inherited from the *Game-Theory*, through which the cloud provider assigns bandwidth reservations among the cloud service tenants based on: (1) their offered bids, and (2) the affect of their presence in the network on a social welfare value that is calculated by the system. Instead of the VCG, the authors of [37]

proposed using the *Shapley value* [38] for price-unit calculations. So, according to their work, the price of the amount of bandwidth resources *allocated* to a service tenants is calculated by the cloud provider, in accordance to the average marginal charge for the allocated resources. In [39], the work proposes a model that tackles the problem of both bandwidth reservation and allocation through a two-tier approach. In which, the cloud provider runs an *auction* for bandwidth reservation *first*, and *then* after the reservation round ends, remaining bandwidth resources are *auctioned* to be allocated to the service tenants. Price calculation in [39] varies between reservation and allocation processes. In the reservation auction, the price unit is initially set to a *premium price* to encourage high bid offers [40]. In the allocation phase, the model considers a *market clearing price* for allocation. This market price is defined according to the lowest *accepted* bid received in the auction. *Accepted* here refers

to those bids who have sufficient bandwidth resources at the provider's side to satisfy their requests, regardless of their offered bid price. This assumes fair pricing for all bidders [41]. The aforementioned works considers either the *provider's* interest or

the *tenant*, but not both. In [41] and [42], the authors proposed a model that considers the interest of both, i.e. the provider and the tenant. In their work, the authors proposed a resource allocation model based on a *bargaining game*, through which they studied the resource allocation problem for Virtual Machines (VMs) over a set of physical DCN servers. The allocation model presented in this work is formulated with the objective of maximizing both utilities, i.e. of the providers and tenants together.

5.2 The Challenge of DCN Migration

The previous section discussed approaches may provide for optimal allocation/reservation decisions for a service instance or a VM, however, what about an optimal approach for full VDCN allocation over a physical DCN, does this seems feasible? Among the service-themes provided by the cloud DCN is the *infrastructure as a service*, in which, a user can lease a whole Virtual DCN (VDCN) infrastructure

from a cloud physical DCN. For load and scale necessities, a VDCN provider can choose to migrate from its current place (i.e. the physical DCN that currently hosts the VDCN) to other new place (i.e. new physical DCN) that provide for larger resource/scale capacities to suite the dynamic VDCN load requirements. The research in such a problem can be considered as a novel open direction that is still in its early stages, motivated researchers may choose such hot-issue to be tackled by their future research works.

6 Conclusions

Computation is moving into the cloud, and thus into DCNs. Within the DCN, proposed interconnection structures must be aware of the end-to-end system requirements before providing their suggested solutions. Hence, structures should provide for agility, reliability, cost-efficiency, and high resource utilization. In cloud data centers, automation is a necessity for scale, and it is accordingly considered as a fundamental principle of design. The soul of DCN lies in the theme of Virtualization,

which represents a promising aspect for higher performance and maximum reliability, and it can be deployed in both server and storage equipments. Moreover, employing the concept of consolidation beside such virtualization technologies in DCNs can enable the IT organizations to turn computing and storage resources from monolithic systems into a shared pool of resources. Such pools that consists of standardized components which can be dynamically aggregated, tiered, provisioned, and accessed

through an intelligent network. However, virtualizing such networks has some constraints that vary between real time replication and whether the considered application can be clustered or not. But still, the journey to fully virtualized and autonomic DCNs is still in its early stages. Though, we should admit the issue that we no longer design for individual or single server applications, the work now evolves toward the cloud and huge clustered network applications. Finally, we can say that a properly planned DCN is that who protects the application and data integrity, optimizes their

availability and performance, and allows for scale and change according to the market requirements and business priorities.

7 References

[1] L. A. Barroso and U. Hlzle. The datacenter as a computer: An introduction to the design of warehouse-scale machines. *A Publication in the Morgan and Claypool Publishers series*, **2009**.

[2] W. Ni, C. Huang, and J. Wu. Provisioning high-availability datacenter networks for full bandwidth communication. *Computer Networks*, 68, Pages 71-94, **2014** https://doi.org/10.1016/j.comnet.2013.12.006

[3] E. Giesa. Data center virtualization q & a. *F5 White Paper*, **2011**.

[4] A. Singh, M. Korupolu, and D. Mohapatra. Server-storage virtualization: Integration and load balancing in data centers. In *in the Proceedings of the ACM/IEEE conference on SuperComputing, SC'08*, pages 1–12. IEEE, Nov **2008**.

[5] D. Li, C. Guo, H. Wu, K. Tan, Y. Zhang, S. Lu, and J. Wu. Scalable and cost-effective interconnection of data-center servers using dual server ports. *IEEE/ACM TRANSACTIONS ON NETWORKING*, 19(1), Feb **2011**. https://doi.org/10.1109/TNET.2010.2053718

[6] K. Chen, C. Hu, X. Zhang, K. Zheng, Y. Chen, and A. V. Vasilakos. Survey on routing in data centers: Insights and future directions. *IEEE Networks*, 25(4), Aug **2010**.

[7] R. N. Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat. Portland: a scalable fault-tolerant layer 2 data center network fabric. In *in Proceedings of the ACM SIGCOMM '09 conference on Data communication*, pages 17–21. IEEE, Aug **2009**.

[8] A. Wiess. Computing in the cloudes. *NetWorker Magazine: Cloud computing, PC functions move onto the web*, 11(4), Dec **2007**.

[9] S. Long. Taking the enterprise data center into the cloud, achieving a flexible, high-availability cloud computing infrastructure. *A White Paper from the Experts in Business-Critical Continuity*, Dec **2010**.

[10] K. M. Sup, T. Ali, A. Leon-Garcia, and H. J. Won-Ki. Virtual network based autonomic network resource control and management system. In *Proceedings of the IEEE Com '05*. IEEE, **2005**.

[11] G. Khanna, K. Beaty, G. Kar, and A. Kochut. Application performance management in virtualized server environments. In *in the Proceedings of the Network Operation and Management Symposium*, pages 373–381. IEEE, Apr **2006**. https://doi.org/10.1109/NOMS.2006.1687567

[12] M. Cardosa, M. R. Korupolu, and A. Singh. Shares and utilities based power consolidation in virtualized server environments. In *Proceedings of the International Symposium of Integrated Network Management*, pages 327–334, Long Island, NY, USA, IEEE. Jun **2009**. https://doi.org/10.1109/INM.2009.5188832

[13] M. Steinder, I. Whalley, D. Carrera, I. Gaweda, and D. Chess. Server virtualization in autonomic management of heterogeneous workloads. In *in Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management, IM 07*. IEEE, May **2007**.

[14] R.D. Couto, S. Secci, M. E. Camptisa, and L. H. Costa. Reliability and Survivability Analysis of Data Center Network Topologies. In *ACM Journal of Network and Systems Management*, 24 (2), Pages 346-392, April **2016**

[15] W. Enck, T. Moyer, P. McDaniel, S. Sen, P. Sebos, S. Spoerel, A. Greenberg, S. Y.-W. Eric, R. Sanjay, and W. Aiello. Configuration management at massive scale: system design and experience. *IEEE JSAC, Network Infrastructure Configuration*, 27(3), **2009**.

[16] Z. Kerravala. Configuration management delivers business resilience. *The Yankee Group*, Nov **2002**.

[17] M. Isard. Autopilot: Automatic data center management. *Operating Systems Review*, 41(2), **2007**. https://doi.org/10.1145/1243418.1243426

[18] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel. The cost of a cloud: research problems in data center networks. *ACM SIGCOMM '09 Computer Communiction Review*, 39(1), Jan **2009**.

[19] A. Hammadi, M. Mohammad., and T. El-Gorashi. et. al. Resource Provisioning for Cloud PON AwGR-Based Data Center Architecture. In *Proceedings of the 21st European IEEE Conference on Networks and Optical Communication*, Pages 178-182, Lisbon, Portugal, June **2016**. https://doi.org/10.1109/NOC.2016.7507009

[20] A. Curtis, T. Carpenter, M. Elsheikh, A. López-Ortiz, and S. Keshav. REWIRE: an optimization-based framework for unstructured data center network design. In *Proceedings of the IEEE INFOCOM*, pages 1116–1124, **2012**.

[21] N. Boden, D. Cohen, R. Felderman, A. Kulawik, C. Seitz, and J. Seizovic. Myrinet: A gigabit-per-second local area network. *IEEE*, 15(1), **1995**.

[22] M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity data center network architecture. In *in Proceedings of the ACM SIGCOMM '08 conference on Data communication*, pages 63–74. IEEE, Aug **2008**.

[23] C. Kachris and I. Tomkos. A survey on optical interconnects for data centers. In *IEEE Commun. Surv. Tutor*. 14(4), Pages 1021-1036, **2012.** https://doi.org/10.1109/SURV.2011.122111.00069

[24] A. Greenberg, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta. Towards a next generation data center architecture: Scalability and commoditization. In *in the Proceedings of the ACM workshop on Programmable routers for extensible services of tomorrow*, Seattle, Washington, USA, Aug **2008**.

[25] Q. Zhang, M. Zhani, R. Boutaba, and J. Hellerstein. Dynamic heterogeneity-aware resource provisioning in the cloud. In *IEEE Trans., Cloud Computing*, 2(1), Pages 14-28, **2014.**

[26] C. Clos. A study of non-blocking switching networks. *Bell System Technical Journal*, 32(2), **1953**. https://doi.org/10.1002/j.1538-7305.1953.tb01433.x

[27] A. F. Mohammad, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan. Data center tcp (dctcp). In *in the Proceedings of the ACM SIGCOMM 2010 conference*, Seattle, Washington, USA, IEEE, Sep **2010**.

[28] D. Li, C. Guo, H. Wu, K. Tan, Y. Zhang, S. Lu, and J. Wu. Ficonn: Using backup port for server interconnection in data centers. In *in the Proceedings of IEEE INFOComm '09*, pages 2276–2285. IEEE, Apr **2009**.

[29] C. Guo, H.Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu. Dcell: A scalable and fault-tolerant network structure for data centers. In *in the Proceedings of the ACM SIGCOMM conference on Data communication*. IEEE, Oct **2008**. https://doi.org/10.1145/1402958.1402968

[30] A. Hammadi, T. El-Gorashi, and M. Mohammad. et. al. Server-Centric PON Data Center Architecture. In *Proceedings of the 18th International IEEE Conference on Transparent Optical Betworks*, Trento, Italy, July **2016**. https://doi.org/10.1109/ICTON.2016.7550695

[31] J. Moy. OSPF version 2. *RFC 2328*, Apr **1998**.

[32] C. Guo, G. Lu, D. Li, H.Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu. Bcube: A high performance, server-centric network architecture for modular data centers. In *in the*

Proceedings of the ACM SIGCOMM conference on Data communication. IEEE, Oct **2009**. https://doi.org/10.1145/1592568.1592577

[33] A. Greenberg, J. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, and P. Patel. Vl2: A scalable and flexible data center network. In *in the Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, Barcelona, Spain, Aug **2009**. IEEE. https://doi.org/10.1145/1592568.1592576

[34] N. F. et. al. Helios: A hybrid electrical/optical switch architecture for modular data centers. In *in Proceedings of the IEEE SIGComm '10*. IEEE, **2010**.

[35] Yang, G.; Zhenzhe, Z.; Fan, W.; Xiaofeng, G.; Guihai, C. SOAR: Strategy-proof auction mechanisms for distributed cloud bandwidth reservation. Proceedings of the 2014 IEEE International Conference on Communication Systems (ICCS). Macau, China, Nov **2014**; IEEE, https://doi.org/10.1109/ICCS.2014.7024786

[36] Quttoum, A.N.; Otrok, H.; Dzion, Z. ARMM: An Autonomic Resource Management Model for Virtual Private Networks. Proceedings of the 2010 IEEE International Conference on Consumer Communications and Networking Conference (CCNC). Las Vegas, NV, USA; IEEE, , Jan **2010**. https://doi.org/10.1109/CCNC.2010.5421818

[37] Jinwu, G.; Xiangfeng, Y.; Di, L. Uncertain Shapley value of coalitional game with application to supply chain alliance. MDPI, Journal of Sensors, *56*, pp. 551-556, July**2017.** https://doi.org/10.1016/j.asoc.2016.06.018

[38] SHI, W.; Wu, C.; Li, Z. A Shapley-value Mechanism for Bandwidth On Demand between Datacenters. IEEE Transactions on Cloud Computing, **2015**.

[39] Wee, K.T.; Dinil, M.D.; Mohan, G. Uniform Price Auction for Allocation of Dynamic Cloud Bandwidth. Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, NSW, Australia; IEEE, June **2014**.

[40] Baseem, W.; Nancy, S.; Ahmed, K. Modeling and pricing cloud service elasticity for geographically distributed applications. Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). Ottawa, ON, Canada; IEEE, May **2015**. https://doi.org/10.1109/INM.2015.7140337

[41] Jian, G.; Fangming, L.; Haowen, T.; Yingnan, L.; Hai, J.; John C.; Lui S. Falloc: Fair network bandwidth allocation in IaaS datacenters via a bargaining game approach. Proceedings of the 21st Internation IEEE Conference on Network Protocols (ICNP). Goettingen, Germany; IEEE, Oct **2013**. https://doi.org/10.1109/ICNP.2013.6733583

[42] Jian, G.; Fangming, L.; Dan, Z.; John, C.S.L.; Hai, J. A cooperative game based allocation for sharing data center networks. Proceedings of the IEEE 2013 INFOCOM Conference. Turin, Italy; IEEE, April **2013**. https://doi.org/10.1109/INFCOM.2013.6567016

8 Author

Ahmad Nahar Quttoum holds an Assistant Professor position at the Computer Engineering Department in the Hashemite University, Jordan. Prior to that, he worked as a Postdoctoral researcher at the LTIR lab in the Université du Québec à Montréal (UQAM), Montreal, Canada. In that, he worked on the NetVirt project; a project for Ericsson-Canada, where mainly, he was concerned with Cloud-Service Data Center Networks. In Oct 2011, he obtained a Ph.D. degree from the Department of Electrical and Computer Engineering at the University of Quebec, Montreal, Canada. His Ph.D. research topic was about Resource Management for Virtualized Networks; a project for Bell Canada. In late 2007, he obtained a M.Sc. degree in Network Systems from

the Department of Engineering, Computing & Technology at the University of Sunderland, United Kingdom. During his M.Sc. studies, he worked on various research topics on network security ended with a thesis in security attacks, detection and prevention. In early 2006, he obtained a B.Eng. degree in Electrical and Computer Engineering from Jordan University of Science and Technology, Irbid, Jordan. His research interests include cloud computing, data center networks, virtualized networks, autonomic resource management, and network security. He is also a technical reviewer for different journals and specialized magazines.

Article submitted 11 August 2017. Published as resubmitted by the authors 03 October 2017.