International Journal of Engineering & Technology, 7 (2) (2018) 657-662

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i2.9770
Research paper

Dragonfly addressing model for software defined networks based on datacenters

Heba A. Hassan [1] *, Amr A.Al-Awamry [1], M. B. Abdelhalim [2]

[1] Electrical Department, Benha Faculty of Engineering, Benha University
[2] College of Computing and Information Technology, Arab Academy for Science, Technology & Maritime Transport, Cairo, Egypt
**Corresponding author E-mail: ommolham2010@gmail.com*

Abstract

With the advancement of technology, virtualization has become very important for Information Technology (IT) experts. Network Functions Virtualization (NFV) means to address issues resulting from complex hardware-based appliances by developing standard IT virtualization technologies. Software Defined Networking (SDN) solidifies the advantages of datacenter virtualization, increases resource flexibility and utilization, and reduces infrastructure costs and overhead. Datacenter networks should have the ability to guarantee high throughput and resiliency. For such reasons, typical datacenter networks (e.g. Fat Tree) have been evolved to high-radix networks (e.g. Dragonfly). This work aims to investigate how SDN and NFV can improve the advantages of datacenter virtualization by utilizing datacenter topologies such as Dragonfly (DF) topology and Fat Tree (FT) topology in SDN, thus expanding resource flexibility and utilization and diminishing infrastructure costs and overhead. By using Dragonfly topology, the cost is reduced and better scalability is introduced compared to the folded clos networks such as Fat Tree. Here in, a novel addressing scheme is proposed for Dragonfly topology with simulation results included utilizing Mininet, which incorporates MiniEdit that is used to create and run network simulations.

Keywords: *Software Defined Networking; Network Functions Virtualization; Dragonfly topology; Fat Tree topology.*

1. Introduction

With the rise of cloud computing and virtualized infrastructure in the datacenters [9-13], elasticity is thought to be the most critical component for both operational and capital costs for administrators and specialist organizations. There are numerous network challenges when launching a new network service such as expanding costs of energy, and the rarity of skills important to design, integrate and operate increasingly complex hardware-based appliances. Hardware-based appliances aren't efficient as they require a great part of the procure-design-integrate deploy cycle to be reached next to zero income advantage. Hardware lifecycles are also getting to be plainly shorter as innovation and administrations advancement accelerates. Network virtualization offers the solution for meeting these challenges, because it reduces the cost of management and power consumption by enabling the IT to manage the network through an interface without getting to the basic system foundation. Another preferred standpoint of network virtualization is that it reduces the downtime of network and applications making troubleshooting easy. It also makes the system framework more agile and scalable, because logical domains are connected through tunnels. Therefore, IT administrators do not have to physically connect the domains.

Software Defined Networking (SDN) [1], [2] achieves rapid progress in cloud datacenters. SDN technology also gives promising opportunities for high-throughput, and high-volume applications such as big data deployments in the financial and scientific sectors. There is a similarity between Network Functions Virtualization (NFV) [3], [5] and SDN as they quicken advancement by breaking the bond between proprietary hardware and control/application software, so the two architectures are optimized for the dynamic cloud environment at carrier scale. Reducing both OPEX and CAPEX is the principle preferred standpoint of NFV and SDN as they seek to leverage automation and virtualization to achieve greater agility. And hence the two concepts can enhance the benefits of datacenter virtualization by using datacenter topologies such as dragonfly topology and Fat Tree topology in SDN.

The importance of any topology comes from its ability to set performance bounds for the network by establishing the network diameter as well as bisection bandwidth. It also largely decides the cost of the system. The utilization of high-radix routers is vital with the quickened development of technology and increased pin-bandwidth to reduce the diameter, latency, and cost of interconnection networks. Motivation of using high-radix routers is its ability to impact networks used in large-scale systems such as multi computers and datacenters. As a result, the need to use a scalable and a cost-efficient topology is required to properly utilize high-radix routers.For such reasons, typical datacenter networks (e.g. Fat Tree) have been developed to high-radix networks (e.g. Dragonfly).

Fat Tree (FT) and Dragonfly (DF) topologies are utilized for building a High Performance Computing Cluster (HPC) and for different applications concerned with interconnection network subsystems in a supercomputer. This work presents FT and DF topologies in light of an SDN basis. It also aims to investigate how SDN and NFV can improve the advantages of datacenter virtualization by utilizing datacenter topologies such as DF topology and FT topology in SDN, thus expanding resource flexibility and utilization and diminishing infrastructure costs and overhead. We implemented the topology and tested the proposed model for addressing of dragonfly topology using Mininet network emulator; a GUI called MiniEdit [6]. We simulated a datacenter with FT topology (k=4) [14-17] and DF topology (N=72 hosts) [18-21] and compared the scalability and cost results. The results clearly show that DF topology introduces the best scalability and is less expensive than FT topology.

Whatever is left of this paper is composed as takes after. Section 2 describes the previous works related to SDN, NFV, and datacenter topologies. Section 3 explains the proposed model for addressing of DF topology. Section 4 analyzes the results and compares the performance of both DF and FT topologies. At last, section 5 outlines the work and gives the concluding remarks.

2. Related work

2.1. Software-defined networks

Recently, SDN has turned into another approach for network programmability and management. The fundamental normal for SDN is that it decouples the centralized control plane from the data forwarding plane. SDN has many advantages. One of them is that it characterizes another structure (called controller) that centralizes control intelligence of one or more network elements (basically switches) as discussed in [1, 4]. For communication between control plane and data plane (South Bound interface), different open interfaces have been defined and OpenFlow [7, 8] is thought to be a real standard. On the other hand, applications can be deployed with a network-wide view of data path elements on the north interface of the controller.

2.2. Network functions virtualization

The motivation of utilizing NFV is its ability to utilize IT virtualization related technologies to virtualize the whole classes of network node functions into building blocks that may be associated, or affixed together to create communication services. The NFV offers another approach for arrangement, convey, and oversee organizing administrations by decoupling the network functions such as Domain Name Service (DNS), caching, intrusion detection, Network Address Translation (NAT), firewalling, etc., from proprietary hardware appliances. In this way, they can continue running in software. The NFV likewise expects to reduce equipment costs and decrease time to market while achieving scalability, elasticity, and a strong ecosystem. Another preferred standpoint of network virtualization is that it reduces the downtime of network and applications making troubleshooting easy. It also makes the system framework more agile and scalable, because logical domains are connected through tunnels. So, IT administrators do not have to physically connect the domains. Reducing both OPEX and CAPEX is the principle preferred standpoint of NFV and SDN as they look to leverage automation and virtualization to achieve greater agility. Thus, the two concepts can enhance the benefits of datacenter virtualization by utilizing datacenter topologies such as Dragonfly topology and Fat Tree topology in SDN.

2.3. Datacenter topologies

Datacenter networks [9-13] and associated data flow scheduling represent an importance in large-scale data intensive computing. To satisfy the Quality-of-Service (QoS) requirements for data movement in latency and throughput, various network topologies have been suggested. FT [14-17] and DF [18-21] topologies are thought to be among the advanced network topologies proposed, recently.

2.3.1. FT topology

FT topology is a particular model of a Clos topology [22]. FT topology has some features. One of them is that all switching elements are indistinguishable empowering us to leverage cheap commodity parts for all the switches in the communication architecture. Another feature is that FT systems are non-blocking implying that for subjective correspondence designs, there is some arrangement of ways that will soak all the accessible transmission capacity to the end hosts in the topology.

As discussed in [14], a k-ary FT topology is given, where k port switches are used in the three-layer architecture. The network comprises of k pods, and each has two layers of k/2 switches and in lower layers, every k-ports switch is connected to k/2 hosts, where the remaining k ports are used for aggregation. The total number of core switches is $\left(\frac{k}{2}\right)^2$. Core switches are connected to k pods, which leads to the i^th^ port connected to core switches on (k/2) strides in the aggregation layer of each pod. The FT topology is worked with k-port switches supporting $k^3/4$ hosts.

Addressing for FT Topology

IP addresses are allocated within subnetwork 10.0.0.0/8. Addressing is in standard IP ver.4 form, however pod switches take the form $10.p_n.s_n.1$, where p_n is the pod number [0,k-1], and s_n is the switch number [0,k-1] taking into consideration that numbering order starts from left to right and bottom to top. Similarly, host addresses take the form 10.pod.switch.ID , where ID shows host location in subnet and it lies in the range [2,k-1] starting from left to right. Thus, lower-level switches are connected to /24 subnet of k/2 hosts for k in the range [0,256].

2.3.2. DF topology

The quickening innovation and expanding pin-bandwidth motivate the utilization of high-radix routers to decrease the diameter, latency, and cost of interconnection networks. High-radix routers have a great impact on networks used in large-scale systems such as multi-computers and datacenters. In this way, an adaptable and a cost-productive topology is required to appropriately utilize high-radix routers. The importance of any topology comes from its ability to set performance bounds for the network by establishing the network diameter as well as bisection bandwidth. It likewise largely decides the cost of the system. Existing topologies such as folded-Clos or FT are not efficient, since they consume bandwidth to load balance traffic. So, they pay too high a penalty on load-balanced traffic (e.g. uniform random) to provide good performance on adversarial traffic patterns. On the other hand, a conventional butterfly network is better than a folded-Clos, because it gives significantly lower cost (approximately half) than a folded-Clos on balanced traffic. However, a conventional butterfly has a disadvantage, because it has no path diversity. Thus, its performance is seriously constrained on adversarial traffic patterns. DF topology overcomes this limitation by using a virtual router or a collection of routers, and hence the radix is increased where the scalability of the flattened butterfly [23] is limited by the radix of a single router.

To achieve the benefits of a very high radix, the paper proposes the utilization of a group of routers connected into a sub-network to represent a high-radix virtual router as presented in [18]. The usage of a very highly effective radix enables us to manufacture a system in which all minimal routes traverse at most one global channel and enables the DF topology to provide high scalability with radix 64. Along these lines, the topology can scale to more than 256k nodes with a network diameter of only three hops. The physical length of the global channel is expanded with the utilization of high-radix topology and specifically with the utilization of DF topology, yet the effect of long global channel lengths can be diminished with the utilization of optical signalling technology.

The hierarchical network of the DF topology includes 3 levels; router, group, and system. A group includes a routers associated by means of an intra-group interconnection network formed from local channels. Each router has 3 different types of connections; connections to p terminals, a-1 local channels to other routers in the same group, and h global channels to routers in other groups. Thus, the DF parameters a (number of routers in each group) , p (number of terminals connected to each router) , and h (number of channels within each router used to connect to other groups) can have any values, but the network should have a = 2p = 2h to achieve balance in channel load on load-balanced traffic. Therfore, the radix (or degree) of every router is d = p + a + h – 1. Each group has a.p connections to terminals and a.h connections to global channels, and all routers in a group all in all go about as a virtual router with radix d'=a (p+h). Since g is the number of groups in the system where g = a.h + 1 groups, and N is the number of network terminals, where N = a.p (a.h + 1) terminals, the number of switches = g.a switches.

3. Proposed model

There is no addressing for the DF topology, and hence we assume that we have some groups of switches, and put one of them as a core switch, but the other switches act as pod switches and we make addressing for them similar to FT topology.
Addressing for DF Topology
Figure 1 shows an example of the DF topology with 72 hosts, where p = h = 2, and a = 4. By using virtual routers, the effective radix is increased from d = 7 to d'= 16.

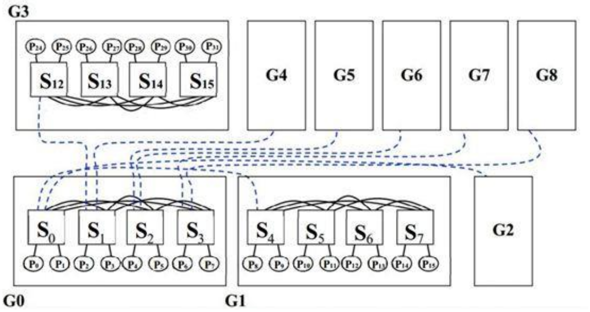


Fig. 1: Block Diagram of DF Topology with N=72 Hosts.

Since g=a.h+1=9 groups and number of switches=g.a=36 switches, we put one of the groups to act as a core switches and the other groups act as pod switches. Pod switches, core switches, and hosts are given the same addresses as in FT topology and we consider K=8. Figure 2 shows the previous example with addresses for pod switches and core switches.

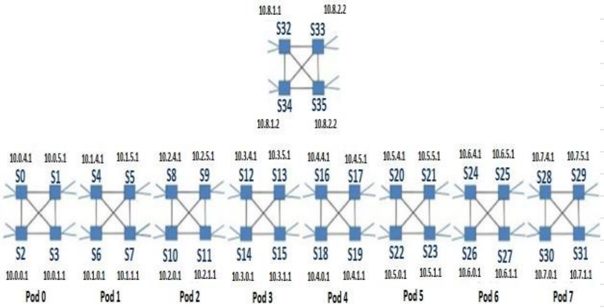


Fig. 2: Addressing for DF Topology.

This paper uses a virtual machine known as Mininet [6] to set up an SDN without hardware. The Mininet network simulator incorporates MiniEdit, a simple GUI editor for Mininet. MiniEdit is an experimental tool made to exhibit how Mininet can be extended. MiniEdit is used to make and run network simulations.
Figure 3 shows that we were able to create a custom network topology graphically using MiniEdit, and then we can extract the python code.

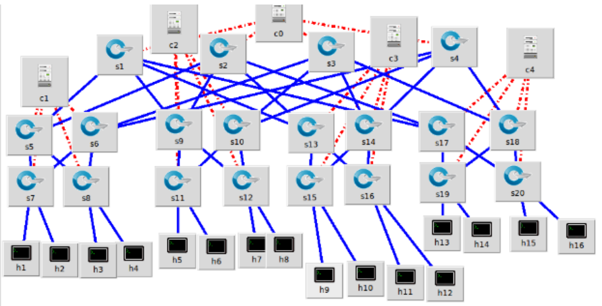


Fig. 3: Custom Network in MiniEdit (FT Topology).

Similarly, after using MiniEdit for the DF topology, the custom network will be as shown in figure 4, and hence the python code can be extracted.

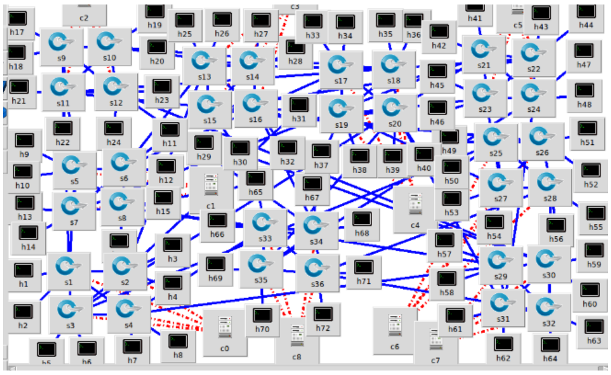


Fig. 4: Custom Network in MiniEdit (DF Topology).

4. Analysis

The cost and performance of a scalable multiprocessor are considered to be the key elements in the interconnection networks. For low-radix routers i.e. routers with few ports, interconnection networks utilize low-radix topology such as 2-D or 3-D mesh, and torus networks or Clos (FT) topologies. Cray T3D, T3E, and XT3 are examples of machines that employ such networks. Because of the low pin bandwidth available in the past [24], [25], low-radix networks provide optimal latency for a given cost. As the pin bandwidth of router chips has expanded because of the expansion in the signalling rate and increase in the number of signals, high-radix routers utilize the advantage of dividing the bandwidth into a larger number of narrow ports [26], where low-radix routers divide the bandwidth into a smaller number of wide ports.
This part introduces an analytic comparison between DF and FT topologies which is summarized in Table 1. As showed up in Table 1, for FT topology, radix implies the number of ports of a switch, and the dimension (n) is concerned with the number of stages in the network, but in DF topology radix infers the number of nodes within a dimension, and the network size can be further increased by adding multiple dimensions [29]. The comparison demonstrates the scalability, i.e. the maximal number of hosts, cost represented in the number of hosts per router, and locality (full bisection group size represented in the number of hosts in the group) [31].

4.1. Scalability

As mentioned earlier for FT topology, the maximal number of hosts $N_{ft} = \frac{d^3}{4}$ hosts where d is the degree or radix. On the other hand, for DF topology, we use a= number of local routers, h= number of global links, and c=concentration. So, the number of hosts N_{df}=g.a.c= (a.h+1) a.c= (2c.c+1) ≈ $4c^4$ assuming balancing conditions of a=2c=2h and router radix d=c+a+h-1. Hence, the maximal number of hosts $N_{df} \approx \frac{d^4}{64} + \frac{d^2}{8}$.

Table 1: Network Characteristics of DF and FT Topologies

	Dragonfly with (*a*=local routers, *h*=global links , *c*=concentration)	Fat Tree with (*k*-ary, *n*-tree)
Total number of nodes (*N*)	$a.c(a.h+1)$	k^n
Degree or radix (d)	$c+a-1+h$	$2k$
Maximal number of hosts	$N_{df} \approx \frac{d^4}{64} + \frac{d^2}{8}$	$N_{ft} = \frac{d^3}{4}$
Number of routers	$R_{df} \approx \frac{4N_{df}}{d}$	$R_{ft} = \frac{5N_{ft}}{d}$
Group size	$G_{df} = \frac{d^2}{8}$	$G_{ft} = \frac{d^2}{4}$
Bisection width (Bc)	$\left(\frac{a.h}{2}\right)^2 \approx 25\%$ of the total number of nodes	$\frac{N}{2}$ = 50% of the total number of nodes
Avg. hop count (H_{avg})	$2a.c/(a.h+1) \approx 2$	$2[n-[1/(k-1)]]$
Diameter (*D*)	3(2 local,1 golabal)	$2n = 2\log_k N$
Number of links	$a(a.h+1)(c+a+h-1)$	$N(\log_k N+1)$

Figure 5 demonstrates that the DF topology introduces better scalability than that of FT topology.

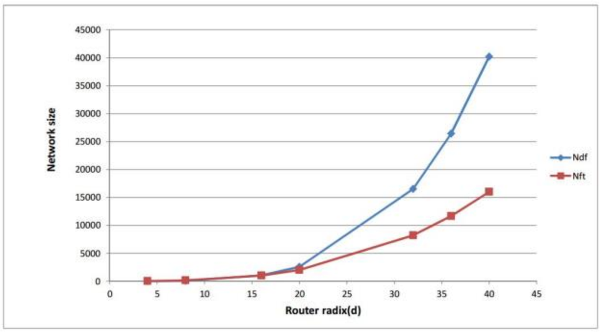


Fig. 5: Maximal Network Size in Number of Hosts vs. Router Radix (D).

4.2. Cost

For FT topology, the number of routers equals $R_{ft} = \frac{5N_{ft}}{d}$, because there are $d^2/2$ edge routers, $d^2/2$ aggregation routers, and $d^2/4$ core routers. Then again, the number of routers in DF topology equals: $R_{df} = g \cdot a \approx \frac{4N_{df}}{d}$. Since cost is determined by the number of hosts per router, we evaluate this ratio for four examples as shown in Table 2, and conclude that DF topology is less expensive than FT topology.

4.3. Locality

For FT topology, the group size equals $G_{ft} = \frac{d^2}{4}$, while the group size for DF topology equals $G_{df} = \frac{d^2}{8}$ assuming balance conditions of a=2c=2h and router radix d=c+a+h-1.

Figure 6 shows how the group size scales with router radix. It is clear from the figure that FT has the best locality.

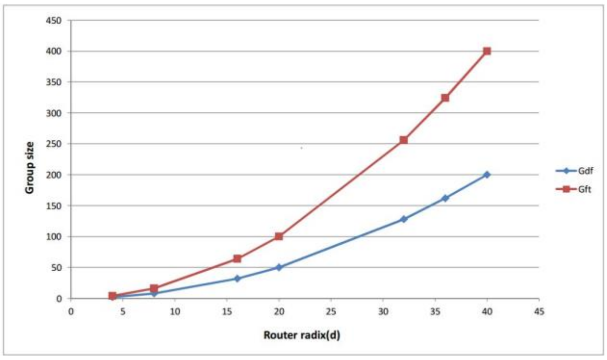


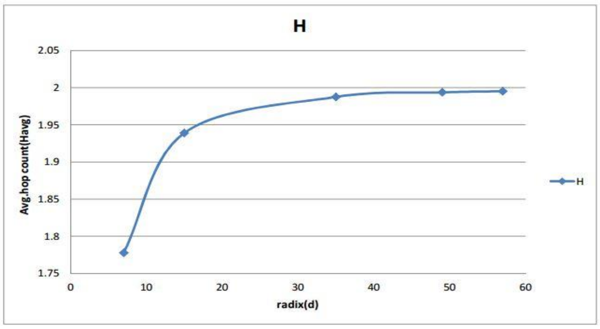
Fig. 6: Group Size in Number of Hosts vs. Router Radix (D).

Table 2: Overall Results of Two Topologies

	Dragonfly with (*a*=26, *h*=12, *c*=12)	Dragonfly with (*a*=20, *h*=8 , *c*=8)	Fat Tree with (k=18, n=2)	Fat Tree with (k=10, n=4)
Total number of nodes (*N*)	97656	25760	324	1000
Degree or radix (*d*)	50	36	36	20
Maximal number of hosts	97968.75	26406	11664	2000
Number of hosts per router	12.5	9	7.2	4
Group size	312.5	162	324	100
Avg. hop count (H_{avg})	1.9930	1.9875	3.8823	7.8
Diameter (*D*)	3	3	4	8
Number of links	406900	115920	972	50000

Table 2 shows the overall analysis results of two topologies by evaluating degree or radix, maximal number of hosts, number of hosts per router, group size, Avg hop count, diameter, and total number of links. DF topologies with (a=26, c=12, h=12), and (a=20, h=8, c=8) show lower diameter, lower average hop count, and lower bisection width (approximately 25 percent of the total number of nodes) because bisection depends only on global links. As we said before, DF topology utilizes a group of high-radix routers as a virtual router. Thus, the effective radix of the network is expanded, and hence the network diameter, cost, and latency are diminished [30]. This is obvious from figure 7(a) that increasing radix reduces the overall cost of the network. In DF topology, the switch radix is high because it must support the local radix, concentration, and global links. On the other hand, FT topologies with (k=18, n=2), and (k=10, n=4) offer low diameter, high switch counts, and high bisection width (50 percent of the total number of nodes) since the total number of nodes in FT topology N = k^n and bisection width Bc =N/2. Figure 7(b) shows that the Avg.hop count (H_{avg}) is increased as the radix (d) is increased leading to an increase in cost because cost is proportional to H_{avg}. In FT topology, the total cost relies upon the number of levels, which implies that as the number of levels increases, the degree or radix decreases, and the number of links increases and vice versa.

(A) DF Topology



(B) FT Topology

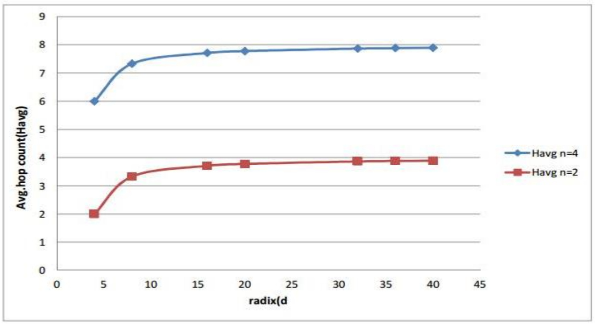


Fig. 7: Avg. Hop Count of A Network Versus Radix For (A) DF Topology and (B) FT Topology.

The main characteristic of high-radix networks compered to low-radix networks is that it requires longer cables. Since cables rule arrange cost, the number of cables, and especially the number of long global cables should be limited to realize an efficient network. As the number of global cables in a network is diminished, while in the same time their lengths are expanded, the DF topology is especially appropriate for implementation using emerging active optical cables which have a high fixed cost but a low cost per unit length compared to electrical cables.

As the high-radix routers become accessible, the modified high-radix folded-Clos topology [28] used in the Cray BlackWidow [27] network exploits this accessibility to reduce the network diameter, but this requires each packet to cross various global and expensive channels. The main feature of the proposed DF topology is that it requires only one global channel to be traversed with the use of minimal routing, and hence the cost is reduced contrasted with that of the folded Clos network (FT). The hierarchical nature of the DF topology is considered to be another significant feature, because it is perfectly matched to the packaging hierarchy of system components.

Conclusion and Future Work

FT and DF topologies are utilized for building a High Performance Computing Cluster (HPC) and for different applications concerned with interconnection network subsystems in a supercomputer. This paper presents FT and DF topologies in light of an SDN basis. It also presents a novel addressing scheme for DF topology with simulation results included utilizing Mininet, which incorporates MiniEdit that is used to create and run network simulations. We simulated a datacenter with FT topology (k=4) and DF topology with (N=72 hosts) and analyzed the scalability and cost results. These results clearly show that DF topology introduces better scalability and less expense than those of FT topology. This work on high-radix routers and networks will be critical after some time in light of fact that the extent of system will keep on increasing. So, interconnection networks will turn out to be more significant to system performance. The research on high-radix networks and DF topology will have a strong effect on the interconnection network used in future large-scale systems, and in SDN specifically.

References

[1] Zhi-jie Han and Wanli Ren "A Novel Wireless Sensor Networks Structure Based on the SDN "International Journal of Distributed Sensor Networks Volume 10 (2014), Article ID 874047, 7 pages. https://doi.org/10.1155/2014/874047.

[2] B. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka and T. Turletti, "A Survey of software-defined networking: past, present, and future of programmable networks," Communications Surveys & Tutorials, IEEE, vol. 16, no. 3, pp. 1617 - 1634, 2014 https://doi.org/10.1109/SURV.2014.012214.00180.

[3] S. Schaller and D. Hood, " Software defined networking architecture standardization", Computer Standards & Interfaces, Volume 54,Part 4, November 2017,Pages 197-202. https://doi.org/10.1016/j.csi.2017.01.005.

[4] L.Bertaux, S.Medjiah, P.Berthou, S.Abdellatif, A.Hakiri, P.Gelard, F.Planchon, and M.Bruyere, "Software defined networking and Virtualization for broadband satellite networks" IEEE Communications Magazine (Volume:53, Issue:3, March 2015) Page(s):54-60

[5] K. Gray, and Thomas D.Nadeau, "Network Function Virtualization", Network Function Virtualization, 2016, Pages 1-18.

[6] B. Lantz, B. Heller, N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks", in: Proceedings of the 9thACM SIGCOMMWorkshop on Hot Topics in Networks, ACM, 2010, p. 19:1-19:6. https://doi.org/10.1145/1868447.1868466.

[7] A. Lara, A. Kolasani, and B.Ramamurthy "Network Innovation using OpenFlow: A Survey "IEEE Communications Surveys & Tutorials, Volume: 16, Issue: 1, First Quarter 2014, 493 – 512. https://doi.org/10.1109/SURV.2013.081313.00105.

[8] McKeown, Nick, T.Anderson, H.Balakrishnan, G.Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. "OpenFlow: enabling innovation in campus networks." ACM SIGCOMM Computer Communication Review 38, no. 2 (2008): 69-74.

[9] Md. Faizul Bari, R. Boutaba, R.Esteves, Lisandro Z.Granville, M. Podlesny, Md Golam Rabbani, Qi Zhang, and Mohamed F.Zhani :" Data Center Network Virtualization: A Survey" IEEE Communications Surveys & Tutorials, Vol. 15, No. 2, Second Quarter 2013, 909-928. https://doi.org/10.1109/SURV.2012.090512.00043.

[10] Caesar Wu, R.Buyya, "Data Center Networks",Cloud Data Centers and Cost Modeling, 2015,Pages 497-576

[11] Yang, T., A., Joshy, N., Rojas, E., Anumula, S. and Moola, J. (2015)."Virtualization and Data Center Design", Global Journal on Technology [Online].09, pp 36-54.Avilable from: http://awer-center.org/git/.

[12] Eun-Sung Jung, Venkatram Vishwanath, Rajkumar Kettimuthu, Mathematics and Computer Science Division Argonne National Laboratory:" Distributed Multipath Routing Algorithmfor Data Center Networks" 2014 International Workshop on Data Intensive Scalable Computing Systems 49-55.

[13] Nataˇsa Maksic and Aleksandra Smiljani: “Improving Utilization of Data Center Networks" IEEE Communications Magazine • November 2013, Volume: 51, Issue: 11, Pages 32-38.

[14] M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” in Proceedings of the ACM SIGCOMM2008 Conference on Data Communication Review 38(4). New York, NY, USA: ACM, 2008, p. 63-74. [Online].Available:http://doi.acm.org/10.1145/1402958.1402967.

[15] Faaiz S. Fizi and S. Askar,"A Novel Load Balancing Algorithm for Software Defined Network Based Datacenters" International Conference on Broadband Communication for Next Generation Networks and Multimedia Applications (CoBCom), 2016, Graz, Austria; ISBN: 978-1-5090-2270-0.

[16] D. Coudert, G. Ducoffe, Data center interconnection networks are not hyperbolic, Theor. Comput. Sci. 639 (2016) 72–90. https://doi.org/10.1016/j.tcs.2016.05.025.

[17] T. Chen, X. Gao, G. Chen, The features, hardware, and architectures of data center networks: asurvey, J. Parallel Distrib. Comput. 96 (2016) 45–74. https://doi.org/10.1016/j.jpdc.2016.05.009.

[18] J. Kim, W. J. Dally, S. Scott and D. Abts :" Technology-Driven, Highly-Scalable Dragonfly Topology"in Proceedings of the 35th Annual International Symposium on Computer Architecture, ser, ISCA' 08.Washington, DC, USA:IEEE Computer Society,2008,p.77-88

[19] Garca Marina, et al.”Efficient routing mechanisms for dragonfly networks.”IEEE International Conference on Parallel Processing, 2013, 582-592.

[20] M. Garc´ıa, E. Vallejo, R. Beivide, M. Odriozola, C. Camarero, M. Valero, G. Rodr´ıguez, J. Labarta, and C. Minkenberg, “On-the-fly

adaptive routing in high-radix hierarchical networks," in The 41st International Conference on Parallel Processing (ICPP), 09 2012.

[21] N. Jiang, J. Kim, and W. J. Dally, "Indirect adaptive routing on large scale interconnection networks," in 36th Intl. Symposium on Computer Architecture (ISCA '09), 2009, pp. 220–231. https://doi.org/10.1145/1555754.1555783.

[22] C. Clos. A Study of Non-Blocking Switching Networks. The Bell System technical Journal, 32(2):406–424, March 1953. https://doi.org/10.1002/j.1538-7305.1953.tb01433.x.

[23] J. Kim, W. J. Dally, and D. Abts. Flattened Butterfly: A Cost-Efficient Topology for High-Radix Networks. In Proc. of the International Symposium on Computer Architecture (ISCA), pages 126–137, San Diego, CA, June 2007. https://doi.org/10.1145/1250662.1250679.

[24] A. Agarwal. Limits on Interconnection Network Performance. IEEE Trans. Parallel Distrib. Syst., 2(4):398–412, 1991. https://doi.org/10.1109/71.97897.

[25] W. J. Dally. Performance Analysis of k-ary n-cube Interconnection Networks. IEEE Transactions on Computers, 39(6):775–785, 1990. https://doi.org/10.1109/12.53599.

[26] J. Kim, W. J. Dally, B. Towles, and A. K. Gupta. "Microarchitecture of a High-Radix Router". In Proc. of the International Symposium onComputer Architecture (ISCA), pages 420–431, Madison, WI, 2005. https://doi.org/10.1109/ISCA.2005.35.

[27] D. Abts, A. Bataineh, S. Scott, G. Faanes, J. Schwarzmeier, E. Lundberg, T. Johnson, M. Bye, and G.Schwoerer. The Cray BlackWidow: A Highly Scalable Vector Multiprocessor. In Proc. of the International Conf. for High-Performance Computing, Network, Storage, and Analysis (SC'07), Reno, NV, November 2007, pages 1-12. https://doi.org/10.1145/1362622.1362646.

[28] S. Scott, D. Abts, J. Kim, and W. J. Dally. The BlackWidow High-radix Clos Network. In Proc. of the International Symposium on Computer Architecture (ISCA), pages 16–28, Boston, MA, June 2006. https://doi.org/10.1109/ISCA.2006.40.

[29] D. Abts, J. Kim-High Performance Datacenter Networks_ Architectures, Algorithms, and Opportunities -Morgan & Claypool (2011)

[30] A. Kodi, B. Neel, and W.Brantley,"Photonic Interconnects for Exascale and Datacenter Architectures." In IEEE Micro Magazine, 2014, pp. 18-30.

[31] A. Shpiner, Z. Haramaty, S. Eliad, V.r Zdornov, B. Gafni and E.Zahavi," Dragonfly+: Low Cost Topology for Scaling Datacenters" Conference: Conference: HiPINEB 2017 The 3rd IEEE International Workshop on High-Performance Interconnection Networks in the Exascale and Big-Data Era, At Austin, TX, USA.