Load Balance in Data Center SDN Networks

Tariq Emad Ali[1], Ameer Hussein Morad[2], Mohammed A. Abdala[3]
[1,2]Department of Information and Communication Engineering, Baghdad University, Iraq
[3]Department of Electronic and Communication Engineering, University of AL Hussain, Iraq

Article Info

Article history:

Received Oct 28, 2017
Revised Jan 31, 2018
Accepted Sep 14, 2018

Keyword:

API
DCN
LD
Open day light
Open flow
SDN

ABSTRACT

In the last two decades, networks had been changed according to the rapid changing in its requirements. The current Data Center Networks have large number of hosts (tens or thousands) with special needs of bandwidth as the cloud network and the multimedia content computing is increased. The conventional Data Center Networks (DCNs) are highlighted by the increased number of users and bandwidth requirements which in turn have many implementation limitations. The current networking devices with its control and forwarding planes coupling result in network architectures are not suitable for dynamic computing and storage needs. Software Defined networking (SDN) is introduced to change this notion of traditional networks by decoupling control and forwarding planes. So, due to the rapid increase in the number of applications, websites, storage space, and some of the network resources are being underutilized due to static routing mechanisms. To overcome these limitations, a Software Defined Network based Openflow Data Center network architecture is used to obtain better performance parameters and implementing traffic load balancing function. The load balancing distributes the traffic requests over the connected servers, to diminish network congestions, and reduce underutilization problem of servers. As a result, SDN is developed to afford more effective configuration, enhanced performance, and more flexibility to deal with huge network designs.

Corresponding Author:

Tariq Emad Ali,
Department of of Information and communication Engineering,
Baghdad University, Al-Jadria, Baghdad, Iraq.
Email: tariqemad@gmail.com

1. INTRODUCTION

Social media, mobile devices, and cloud computing are pushing traditional networks to their limits. Compute and storage have benefited from incredible innovations in virtualization and automation, but those benefits are constrained by limitations in the network. Administrators may spin up new compute and storage instances in minutes, only to be held up for weeks by rigid and oftentimes manual network operations. Software-defined networking has the potential to revolutionize legacy data centers by providing a flexible way to control the network so it can function more like the virtualized versions of compute and storage today. The goal of Software-Defined Networking is to enable cloud and network engineers and administrators to respond quickly to changing business requirements via a centralized control console.

SDN encompasses multiple kinds of network technologies designed to make the network more flexible and agile to support the virtualized server and storage infrastructure of the modern data center and Software defined networking was originally defined an approach to designing, building, and managing networks that separates the network's control (brains) and forwarding (muscle) planes enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services [1]. Software-defined networking providers offer a wide selection of competing architectures, but at its most simple, the Software Defined Networking method centralizes control of the network by separating the control logic to off-device computer resources. All SDN models have some version of an SDN Controller, as well as southbound APIs and northbound APIs. SDN Offering a centralized, programmable network that can dynamically provision so as to address the changing needs of businesses, software-define networking also provides the following benefits: [2]

a. Directly Programmable: Network directly programmable because the control functions are decoupled from forwarding functions. Which enable the network to be programmatically configured by proprietary or open source automation tools
b. Centralized Management: Network intelligence is logically centralized in SDN controller software that maintains a global view of the network, which appears to applications and policy engines as a single, logical switch.
c. Reduce CapEx: Software Defined Networking potentially limits the need to purchase purpose-built, ASIC-based networking hardware, and instead supports pay-as-you-grow models
d. Reduce OpEX: SDN enables algorithmic control of the network of network elements (such as hardware or software switches / routers that are increasingly programmable, making it easier to design, deploy, manage, and scale networks. The ability to automate provisioning and orchestration optimizes service availability and reliability by reducing overall management time and the chance for human error.
e. Deliver Agility and Flexibility: Software Defined Networking helps organizations rapidly deploy new applications, services, and infrastructure to quickly meet changing business goals and objectives.
f. Enable Innovation: SDN enables organizations to create new types of applications, services, and business models that can offer new revenue streams and more value from the network.

In 2014 Basheer and Bassam [3] presented many components of software programs which were tested by researchers to simplify the selection of the proper controller in Openflow related projects. The tests show that there exists a difference in the controller's performance and applications that support Openflow versions (1.0) and (1.3). In 2015 Mavjeen and Adrashbi [4] implemented different scenarios using Mininet to make load balancing utilization in a data center network. In their implementation the throughput, bandwidth and delay were compared using four different scenarios based on changing the queue time which results in increasing the throughput and bandwidth when the queue time increased while the delay decreases. Also in 2015 E. Silva, G. Pantuza [5] implemented a dynamic load balancing using SDN. In their work a Software Defined Network was implemented to improve efficiency by using POX controller and single switch network consisted of two servers and four clients. The traffic was distributed by a load balancing algorithm which was the Round Robin between the servers and the requesting clients.

In 2016 S. K. Askar [6] proposed a research on Adaptive Load Balancing Scheme for Data Center Networks Using Software Defined Network. He proposes a new mechanism to conduct load balancing for data center networks in order to improve their efficiency. He chooses a specific kind of network topology the one called Fat-tree network topology. the results showed 0% loss rate compared to a loss rate that ranged from 15% to 34% when using the traditional network, whereas in the second scenario, the results showed a loss rate improvement that ranges between 58% and 81% depending on the amount of contending traffic and the additional traffic beyond links' capacity. In term of throughput, hosts utilizing the proposed scheme maintained the same level of throughput without any degradation when new flows joined the network and added additional traffic in the first scenario. On the other hand, hosts that utilizing the traditional scheme underwent from a remarkable reduction in their throughput, the overall reduction in the throughput hits more than 5Mbps, whereas for the second scenario, the proposed scheme outperforms the traditional mechanism, whereas the improvement in throughput recorded amounts that range between 4.2% and 16.6%.

In 2017 S. Wang and J. Luo [7] proposed a research on Randomized Load-balanced Routing for Fat-tree. They present a simple and effective load-balanced routing scheme called DRB for fat-tree networks, which incorporates the randomized load-balancing technique called TTC into the deterministic D-mod-k routing scheme. TTC uses a threshold to effectively reduce traffic redirection operation, which contributes to evenly distributing traffic among uplinks and downlinks. The experimental results show that DRB succeeds to achieve low-levels of path collision in the flow model and low average and tail latency in the packet model. Theoretical results show that the performance of TTC in balanced allocation is similar to that of the two-choice technique.

Also in 2017 J. Saisagar and Prashant Kothari D. [8] proposed a research on 'Sdn enabled packet based load-balancing (plb) technique in data center networks'. They present a strategy to solve the load balancing issue in SDN based Data Center networks. They created the fat-tree topology in a virtual network with the help of Mininet. They implemented packet based load balancing technique to redirect the traffic to different switches by setting a threshold. They found that the packets transferred from one switch to another switch without any packet loss (i.e. 0% packet loss) thereby balancing the load of one core switch with the other. In this paper present the load balance methods (LD) in SDN datacenter (DC) network using Open Daylight controller and test the performance of the DC network.

2. OPENFLOW PROTOCOL

OpenFlow (OF) protocol is considered one of the first software-defined networking (SDN) standards. It originally defined the communication protocol in SDN environments that enables the SDN Controller to directly interact with the forwarding plane of network devices such as switches and routers, both physical and virtual (hypervisor-based), so it can better adapt to changing business requirements. An SDN Controller in SDN is the “brains” of the SDN network, relaying information to switches/routers ‘below’ (via southbound APIs) and the applications and business logic ‘above’ (via northbound APIs). Recently, as organizations deploy more SDN networks, SDN Controllers have been tasked with federating between SDN Controller domains, using common application interfaces, like OpenFlow and open virtual switch database (OVSDB) [9]. To work in an OF environment, any device that wants to communicate to an SDN Controller must support the OpenFlow protocol. Through this interface, the SDN Controller pushes down changes to the switch/router flow-table allowing network administrators to partition traffic, control flows for optimal performance, and start testing new configurations and applications [10]. Figure (1) show the flow-table entries that can be manipulated in an of switch.

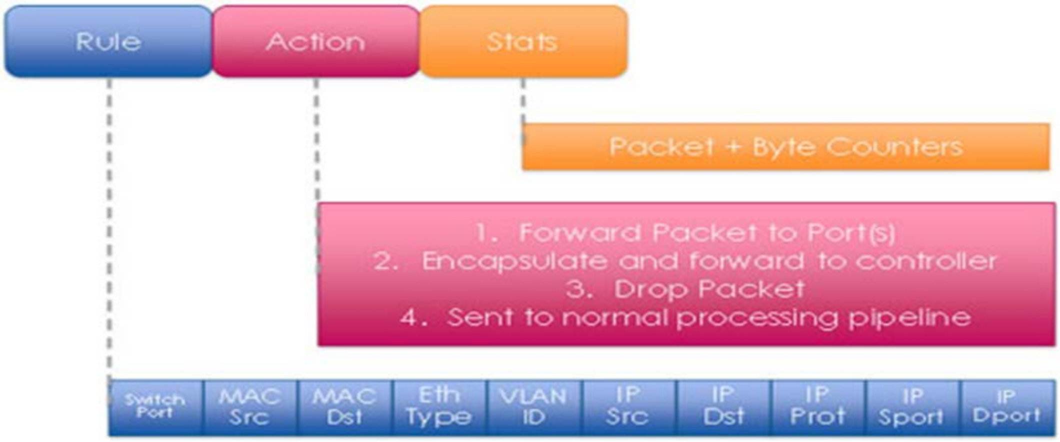

Figure 1. Flow-Table Entries That Can Be Manipulated in an OF Switch

3. SDN FRAMEWORK

Software-defined networking providers offer a wide selection of competing architectures, but at its most simple, the Software Defined Networking method centralizes control of the network by separating the control logic to off-device computer resources [11]. SDN framework consist from Application layer, Control layer and Infrastructure layer as shown in the Figure (2). All SDN models have some version of an SDN Controller, as well as (southbound APIs) and (northbound APIs) [12]:

a. Controllers: The “brains” of the network, SDN Controllers offer a centralized view of the overall network, and enable network administrators to dictate to the underlying systems (like switches and routers) how the forwarding plane should handle network traffic.
b. Southbound APIs: Software-defined networking uses southbound APIs to relay information to the switches and routers “below.” OpenFlow, considered the first standard in SDN, was the original southbound API and remains as one of the most common protocols. Despite some considering OpenFlow and SDN to be one in the same, OpenFlow is merely one piece of the bigger SDN landscape.
c. Northbound APIs: Software Defined Networking uses northbound APIs to communicates with the applications and business logic “above.” These help network administrators to programmatically shape traffic and deploy services.

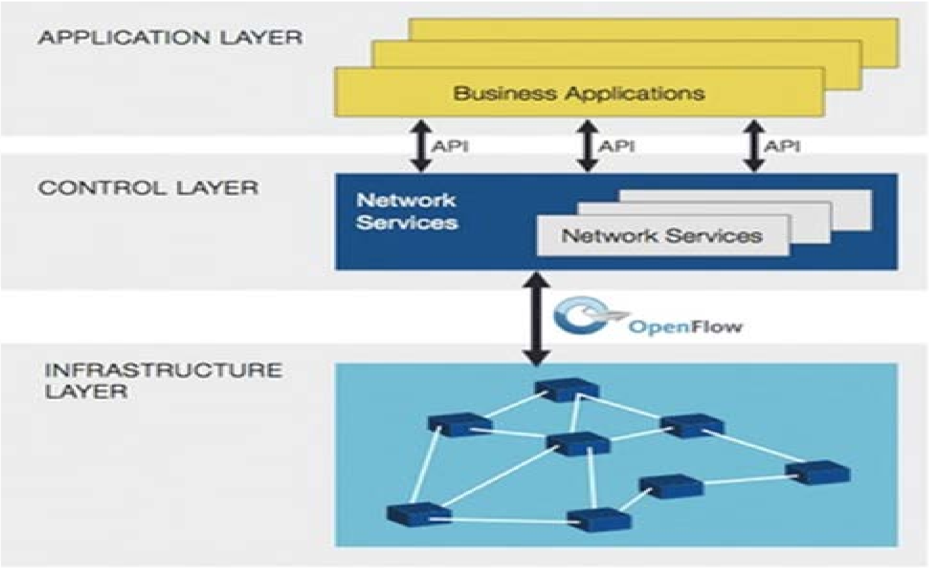

Figure 2. Software Defined Networking (SDN) Framework

4. SDN DATA CENTER DESIGN

SDN DCN construct from number of Openflow switches, hosts and OpenDayLight controllers which is responsible for configuring the switches to forward the traffic as a layer-2 switch and apply the load balance in the data center network, as shown in the Figure (3). This controller allows the traffic forwarding between all of the switches exist in the network that are connected to it [13]. The aim behind using this controller is to offer the control plane decoupling from all the switches and to configure the switches to be forwarding devices by switching the flow tables between them [14].

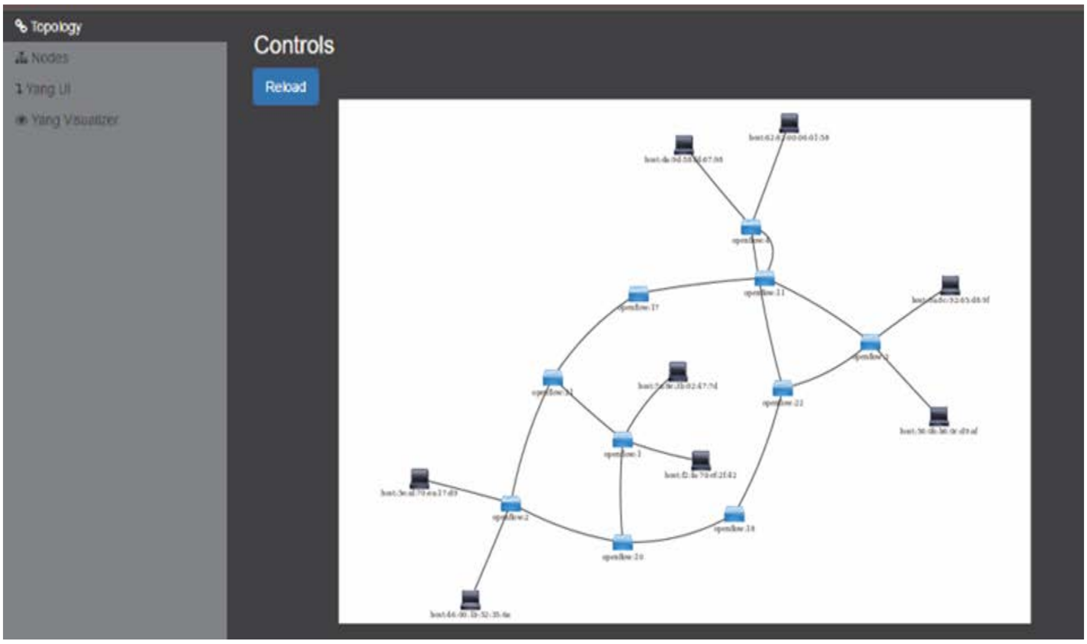

Figure 3. OpenDayLight controllers

5. IMPLEMENTATION SCENARIOS FOR DATA CENTER NETWORK

For implement and understand configurable networks and to achieve the DCN, it is significant to have an environment of numerous specifications. The scenario contains of fat tree topology SDN DCN. The switches OpenFlow-enabled switches are used. These switches were connected to a OpenDaylight controller (C0). Figure 4 show the fat tree topology SDN DCN that used in the scenario

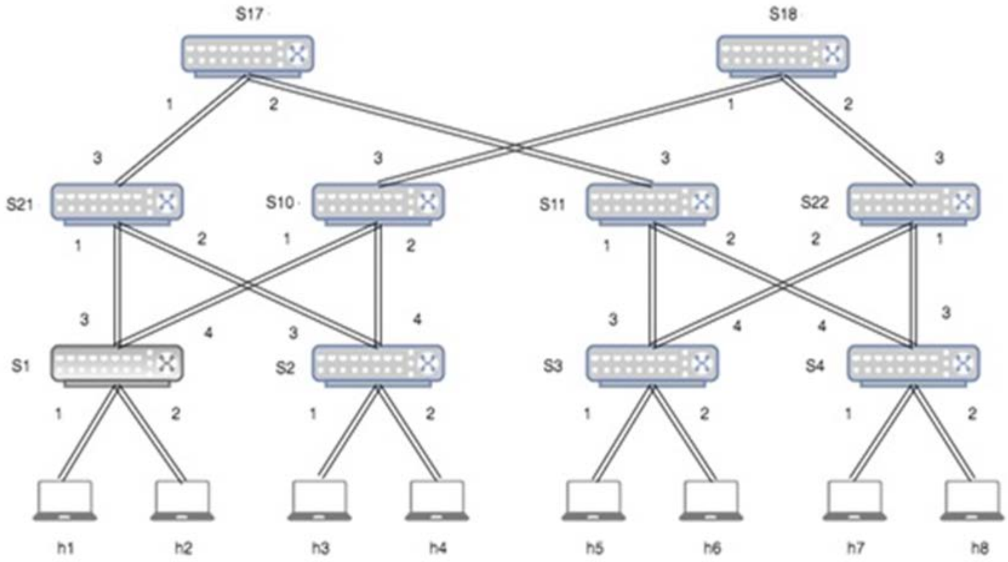

Figure 4. Fat tree topology SDN DCN

The Specifications of OpenDayLight Controller is shown in the Table 1

Table 1. Specifications of Controller

Parameter	Controller
IP address	192.168.1.100
Port Number	6633
CLI	Enabled
OpenFlow version	1.3
Protocol	TCP
Connected Switch	OVS

Finally, the switch by default runs in the root. So, running a command on the “switch” is similar for running it from a regular terminal. In addition, each host has its own IP address which was defined before the emulation start [15].

6. SIMULATION RESULTS

The goal from this paper is to perform the load balancing, but at the same time ensure that the latency is minimum. Dijkstra's algorithm is used to find multiple paths of same length in the fat tree topology. It is also important to note that OpenDaylight controller by default forwards traffic to all ports. So specific rules might need to be pushed to get a proper load balancing output. Currently the program simply finds the path with least load and forwards traffic on that path, then the static flows are pushed into each switch in the current best path such as In-Port, Out-Port, Source IP, Destination IP, Source MAC, Destination MAC is fed to the flows. The REST APIs is used to collect operational information of the topology and its devices. We are performing load balancing between h1, h3 and h4 at the moment. The best path for both is via Switch 1 Port 4. This is the best path selected by OpenFlow protocol. So, after the load balance apply in the SDN DCN using the OpenDayLight controller, we measured the throughput and the average delay before the load balance (BLD) and after the load balance (ALD) to check the performance of the load balance using the OpenDayLight controller on the SDN DCN.

Throughput = maximum receiver bandwidth / round-trip

Figures 5, 6 shows the throughput between H1 to H3 and between H1 to H4 before and after the load balance at tree topology SDN DCN respectively. zince the throughput is defined as the amount of data transmitted from source node to destination node in a given time period. Throughput is typically measured in bits per second (bps). Figures 7, 8 shows the average delay between H1 to H3 and between H1 to H4 before and after the load balance at tree topology SDN DCN respectively.

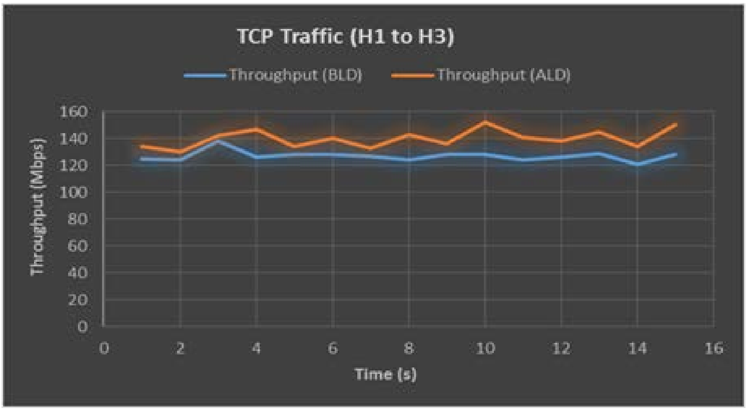

Figure 5. Throughput between H1 to H3 using TCP Traffic

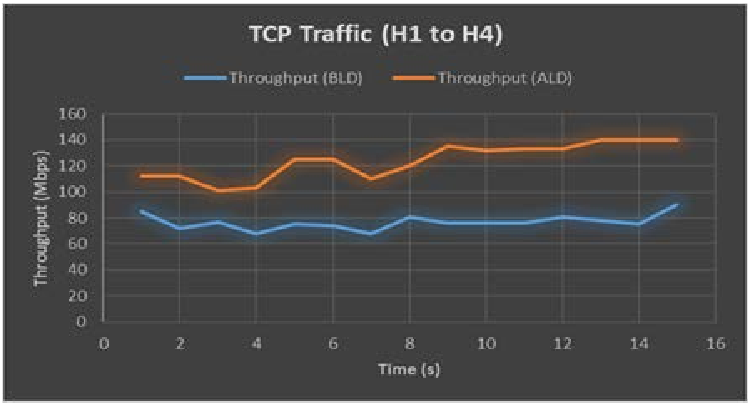

Figure 6. Throughput between H1 to H4 using TCP Traffic

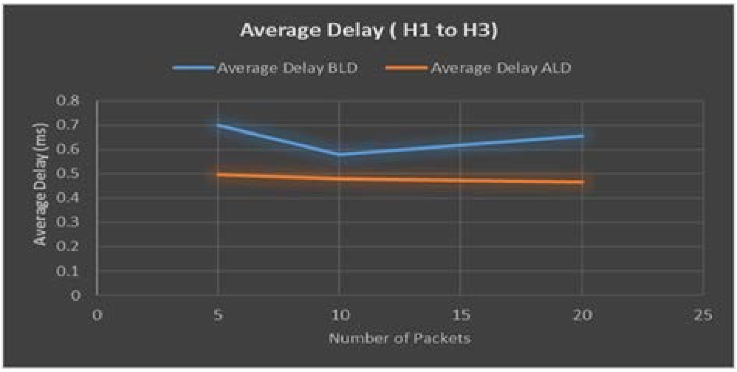

Figure 7. Average delay between H1 to H3 using TCP Traffic

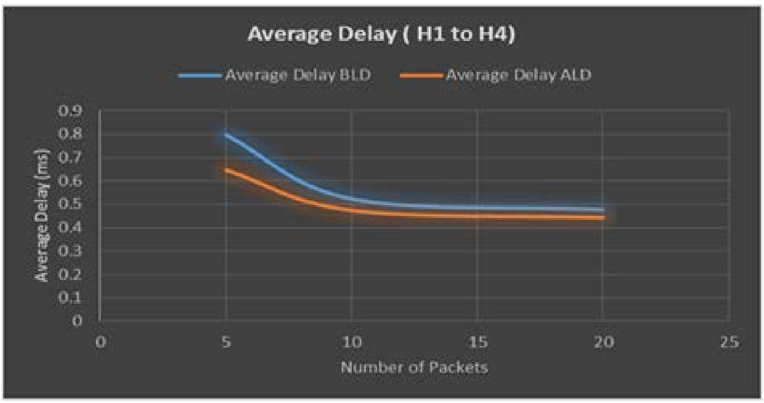

Figure 8. Average delay between H1 to H4 using TCP Traffic

7. CONCLUSION

SDN is a network technology that provide the growing today's network requirements. This can be attained without a need to add new devices or manual configuration for all devices. The only requirement is to have a switch or router with an OpenFlow protocol enabled feature. In this work, load balance technique is designed and implemented to ensure better load distribution on fat tree topology SDN DCN network. For the test condition measured in the work we found that after an application of load balancing the throughput and the average delay is improved.

REFERENCES

[1] N. Feamster, J. Rexford, and E. Zegura, "The Road to SDN: an intellectual history of programmable networks", ACM SIGCOMM Computer Communication Review, New York, USA, April 2014.

[2] Wenfeng Xia, . Yonggang Wen ‘ A Survey on Software-Defined Networking’ IEEE Communications Surveys & Tutorials (Volume: 17, Issue: 1, Firstquarter 2015).

[3] M. Basheer, E. Bassam ‘Survey of software components to emulate OpenFlow protocol as an SDN implementation’ Dept. of Electrical Engineering, Mosul University, Mosul, Iraq, Science Publishing Group, American Journal of Software Engineering and Applications. Vol. 3, No. 6, 2014, pp. 74-82, December 2014..

[4] Mavjeen and Adrashbi ‘ Load Balancing Utilization in Data Center Networks’ Student Symposium 2015, San Jose State University, California, Spring 2015.

[5] E. Silva, G. Pantuza, F. Sampaio, B. Santos, L. Vieira, M. Vieira, and D. Macedo,’ Enforcing Link Utilization with Traffic Engineering on SDN’ University of Federal de Minas Gerais, Portuguese, 2015.

[6] Shavan K. Askar ‘Adaptive Load Balancing Scheme For Data Center Networks Using Software Defined Network’ Electrical and Computer Engineering Department, Collage of Engineering, University of Duhok, Kurdistan Region-Iraq. Vol. 4 , No.2, Pp 275-286, 2016.

[7] S. Wang, J. Luo, ‘Randomized Load-balanced Routing for Fat-tree Networks’ IEEE Aug 30, 2017.

[8] J. Saisagar, Prashant Kothari D., ‘ Sdn Enabled Packet Based Load-Balancing (Plb) Technique In Data Center Networks’ Department Of Computer Science Engineering, Srm University, Vol. 12, No. 16, August 2017.

[9] Chen-xiao, Cui, and X. Ya-bin, "Research on Load Balance Methods in SDN", International Journal of Grid and Distributed Computing, Vol. 9, No. 5 (2016), pp.265-272, Beijing Information Science and Technology University, China, January 2016.

[10] X. Foukas, M. Marina, K. Kontovasilis, "Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture", 1st Edition, WIELY Publisher, Finland, March 2015.

[11] V. Shukla, "Introduction to Software Defined Networking- Openflow & VxLAN", 1st Edition, CreateSpace Independent Publishing Platform, June 18, 2013.

[12] D. Kreutz, F. Ramos, P. Verissimo, C. Esteve, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey", Proceedings of the IEEE. 2015 Jan.

[13] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, “ Software Defined Networking: State of the Art and Research Challenges”, Western University, Canada, 2014.

[14] Mavjeen and Adrashbi, "Load Balancing Utilization in Data Center Networks”, Student Symposium 2015, San Jose State University, California, Spring 2015.

[15] S. Ganesh and Ranjani, "Dynamic Load Balancing using Software Defined Networks", International Conference on Current Trends in Advanced Computing, SRM University, 2015.

BIOGRAPHIES OF AUTHORS

Mr. Tareq Emad Ali has B.Sc and M.Sc. in Electronic and communication Engineering, College of Engineering, Baghdad university. He is Assistant lecturer at Baghdad university, Al-Khwarizmi Collage of Engineering, Information and Communication Engineering Department. He has 5 published scientific & technical papers including IEEE explorer. Mr. Tareq Emad has 10 years of academic & practical's and consulting experience in Networking & Communication. He currently teaches & conducts research programs in the areas of software computer networks, soft computing, intelligent agents, Ad Hoc networks, Wireless Sensor Networks, Routing Protocols and Security of VANETs, Smart Antenna in MANETs and WiMAX Networks, SDN Networks, IOT, IOE, IOV and others.

Dr. Ameer H. Morad is Associate Professor in the Department of Information and Communication, Al Khwarizmi College of Eng., University of Baghdad. Presently he is involved in research work in the areas like Image Processing, Computer Vision, Visual Cryptography and Network security.

Dr. Mohammed A. Abdala is Associate Professor and Head of Electronic and Communication Engineering, University of AL Hussain .He is a Senior member of IEEE, Iraqi Engineers Union and Federation of Arab Engineers. He is active in works that led to the establishment of several engineering departments & colleges in Iraq. He published more than 37 scientific & technical papers. Dr. Abdala has 30 years of academic & consulting experience in Microelectronics & Advanced VLSI. He currently teaches & conducts research programs in the areas of software engineering, image processing, and computer.