Performance analysis and evaluation of software defined networking distributed controllers in datacentre networks

Worku Muluye*

Wolkite University,
Wolkite, Ethiopia
Email: workumuluye@gmail.com
*Corresponding author

Mesfin Abebe and N. Satheesh Kumar

Adama Science and Technology University,
Adama, Ethiopia
Email: mesfinabha@gmail.com
Email: satish4info@gmail.com

Abstract: Today network works under several problems. To solve these problems OpenFlow-based programmable software-defined network (SDN) was released. Datacenter network is one of the application areas that required successful integration of distributed controllers. However, the need for high-performance SDN controllers' increases with the increasing of network devices in the datacenter network. SDN distributed controller allows to increase or decrease the number of controllers regarding the change of traffics. The aim of this paper is performance analysis of SDN distributed controllers in datacenter networks. Initially, compared distributed controllers and selected ONOS (Open Networking Operating System) controller. Then, created three ONOS controllers in a cluster and designed tree topologies. Following tested and evaluated three ONOS controllers' throughput and latency and presented the results of single, two and three ONOS controllers performance. Finally, based on these results the distributed ONOS controller has better performance than single SDN controllers in datacenter networks.

Keywords: software-defined networking; SDN distributed controller; OpenFlow; open networking operating system; datacentre networks; tree topologies.

Biographical notes: Worku Muluye received his first degree in Information Systems from the University of Wollo, Ethiopia in 2013 and his Master's from the Adama Science and Technology University, Ethiopia in 2017. His fields of interest cover software-defined networking, expert systems, and image and signal processing.

Mesfin Abebe received his Doctoral degree from Korean University in 2015 and serves as an Assistant Professor in Department of Computer Science and Engineering program, Adama Science and Technology University, Ethiopia. He published eight publications in reputed journals.

N. Satheesh Kumar received his first degree in the year 2013 and Master's degree in 2001. He has 18 years of teaching and research experience from various Indian reputed organizations. His areas of interest include mobile ad-hoc networks, software defined networks, data centre networks, and Li-Fi.

---

1 Introduction

Computer network changes the way we do business and the way we live. During this time our day to day activity performs or processes through the network and it makes computers and their users capable of being connected together. This facilitates sharing of resources and information between the users. Networks are a critical element for companies and institutions because it radically increases business activities. Therefore, network administrators are struggling to ensure high network availability. According to Forouzan (2007) communications between remote parties can be achieved through a process

called networking, with the connection of computers, media, and networking devices. When a number of devices connected to the internet grow every day and more and more data are transferred from device to device, scaling up the traditional network gets increasingly difficult. In order to meet the performance, network expansion takes place and much of the efforts go towards configuring each switch and router individually (Humayun Kabir, 2013).

In the traditional networking, most network functionality is implemented in the device; i.e., switch, router, security device, and application delivery controller (Webtorials, 2016). It is hard to configure the network according to predefined rules and to reconfigure it to respond to faults, loads, and changes. Since there is an excessive number of switches and routers and a set of protocols that determine the optimal path from source to destination. Typically, in traditional networking devices, the control planes are developed and maintained by vendors only (Li et al., 2017). This makes the network devices to be vendor dependence or depend on a specific vendor and forces the network operator to configure each network devices individually. In order to increase the performance, user satisfaction scalable the network is upgraded and at this time each individual device is affected in traditional networks. To solve these network limitations a new networking architecture was invented which are called software defined networking (SDN).

SDN is a new network architecture that controls the entire network devices centrally. The concept was introduced by a professor at Stanford University and is based on defining a model where all switches move the capacity of the decision to a central element, to a controller (McKeown, 2017). SDN decouples control and forwarding plane and makes the network directly programmable and the infrastructure to be abstracted for applications and network services (ONF, 2016). In SDN controller, for a single change in the network, the network configurations are changed only at central or some specific controller rather than configuring individual network devices. Because the control plane separated from the data plane enables the control plane to control several devices and data plane to forward the data according to the predefined forwarding table.

Figure 1   Vertically integrated control plane and data plane (see online version for colours)
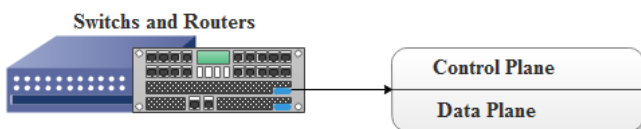

Source: Kreutz et al. (2015)

According to ONF (SDX, 2016) which is responsible for standardising and promoting SDN and defines it as in networking devices, there exist three planes. Data plane is the hardware where packet forwarding takes place. Control plane is software where all network logics and intelligence performed. Management plane typically it is a portion of

control plane that used for network monitoring and control purposes. In general, SDN has three layers such as application layer, control layer, and infrastructure layer (Xie et al., 2015). As ONF report today more than 502 network device companies are members of ONF (SDX, 2016).

As ONF (2016), SDN is an emerging architecture that is the manageable, dynamic, adaptable, cost-effective and scalable network. SDN makes the network devices simpler while giving network administrators better control over the network and increasing network performance. SDN uses OpenFlow protocol (OFP) method to enable the control plane to communicate with the data plane (Feamster, 2013; Xie et al., 2015). In SDN the controller estimates a routing path for that flow on the basis of the global view and then sends the forwarding rule of that routing path to all switches through a secure channel. When those SDN switches receive the forwarding rules, they update their flow tables. Then forward the received flow according to the corresponding flow rules resulting from the controller (Xie et al., 2015). SDN distributed controllers are controllers that enable concurrent and robust network policy implementation. Distributed controller consisting of a collection of controllers that controllers the entire network and provide high network availability (Couto et al., 2014).

Figure 2   Separated control and data plane (see online version for colours)
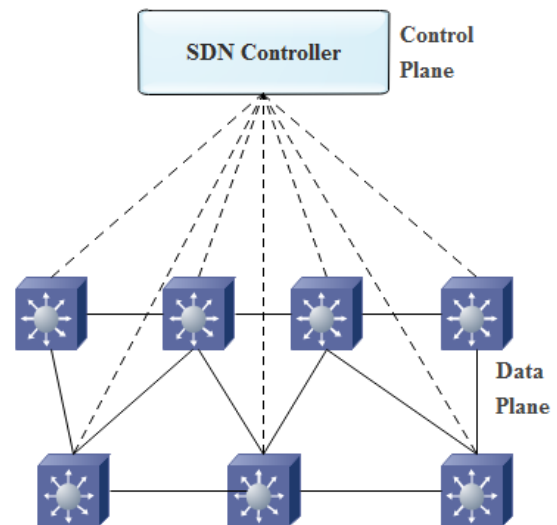
Source: Li et al. (2017)

1.1   Statement of the problem

SDN can be the future of networking, there is no question about it. SDN has emerged as a popular architecture for managing large-scale networks including service providers, enterprises, datacentres and cloud networks. SDN plays a great role to solve traditional network problems (Wang et al., 2016; Techopedia, 2016; Kreutz et al., 2015; Khondoker et al., 2014). The key principle of SDN is the centralised control plane architecture, which allows the network to be programmed by applications running centrally, enabling easier management and faster innovation. However, many of large-scale networks consist

of several interconnected switches and routers in network topology, which cannot easily be controlled by a single centralised controller (Berde et al., 2014; Khondoker et al., 2014; Rowshanrad et al., 2016). This SDN-based centralised network control has performance, scalability, availability, and single point failure problems of the controllers. Since in SDN, when the number of devices is difficult to use a single controller to control the entire forwarding devices. In large networks, this can build to an unacceptable level of delay, degrading network performance and introducing single points of failure. In general, the centralised single controller has scalability, availability, single point of failure, performance, and load balance problems. The researcher motivated to do performance analysis and evaluation of distributed ONOS controller has not investigated. The researchers answer the following questions. How to create and run SDN distributed controllers? How to measure and analyse the performance of SDN distributed controllers?

1.2 Methodology

This research is experimental research because we used two parameters such as throughput and latency and performed the experiment. The researchers used 18 scenarios and each scenario was tested ten times this enabled to get accurate results. The researchers identified tree topology to datacentre network and used 10, 20, and 30 switches and hosts. In addition, in this research used ONOS Controller, Mininet emulator, python, and a cbench benchmarking tool to measure the throughput and latency of distributed ONOS controllers.

2 Related works

In Tootoonchian et al. (2012), the authors evaluate the performance of the floodlight controller on a single server and shown that it can handle over 1.6 million requests per second with an average response time of 2 ms. Also, the authors in Zuccaro et al. (2015) present the evaluation result of SDN performance, where the authors run a single controller and determine that a single controller can control up to 5,000 switches with 46 cores. They were able to achieve a throughput of 14 million flows per second with a latency of 200 μs.

The authors in Khattak et al. (2014) performed performance evaluation on centralised OpenDaylight and floodlight SDN controllers. They used Mininet emulator to emulate the topology that contains 8, 16 and 32 switches and cbench to test throughput and latency. The latency test was run 10 minutes and repeated ten times with eight switches; floodlight produces 1,214 responses per second. When the number of switches are increased slight increase in the number of responses, from 1,214 with eight switches to 1,239 with 16 switches and to 1,335 with 32 switches. The latency of floodlight increases when the number of switches is increased but OpenDaylight shows unexpected performance as the numbers of responses decrease when the

number of switches is increased. OpenDaylight has low average responses and high latency as compared to floodlight. In throughput mode, as increased the number of switches to 16 and 32, the number of responses per second decreased to 9,408 and 4,536, respectively. The OpenDaylight had very low responses as compared to floodlight. The average throughput per switch responses recorded on the OpenDaylight controllers were 270, 33 and 34 for 8, 16 and 32 switches respectively.

This article, Wang (2012), introduces when the number of connected devices increase requires a novel solution to control the incoming SDN network resources. The researchers used OpenDaylight controller, Mininet emulator, and cbench benchmarking tools. The testing environment contains two SDN controllers and 26 switches created using OpenvSwitch. They used three experimental scenarios such as single connection, all connection, and selective connections. The result is when the number of packet-in is greater than 40,000 the all and the selective connections are becoming relevant. At 50,000 packet-in this difference becomes more than 8 seconds. Additionally, when the network load is high the throughput of the selective scenario is 66% greater than the all connection scenario and 100% greater than the single connection scenario. Finally, the authors concluded that the selective connection better than any other configuration.

Authors in Cheng et al. (2015) used Chinanet contains eight switches and 59 links and Cernet contains 36 switches and 53 links real network topologies. They used Beacon controller in an individual machine to simulate single centralised controller. Then compared four controllers with the scenarios of the single controller contains a single controller, static distributed controllers contain four controllers and dynamic controller provisioning problem with greedy knapsack and simulated annealing. A single controller has the lowest scalability due to its resource-limited architecture. The average flow setup time for a single controller and distributed controllers are more than 0.65 s and 0.35 s. The authors measured utilisation ratio single controller has 100% utilisation. Each controller in distributed controllers has more than 90% utilisation. The controller in distributed controllers has less than 80% utilisation in average because of the load imbalance among different controllers.

3 Performance analysis and evaluation of distributed ONOS controller

Performance is a critical issue in computer networking especially in SDN it needed a research and analysis. In order to achieve the objectives of this research the following listed procedures are followed. The necessary processes that are performed in this research show the separated control plane and data plane, created three ONOS controllers and integrated three ONOS controllers with Mininet and measure the performance using cbench and analysed and evaluated the controller's performance.

Figure 3   Performance test and analysis (see online version
                for colours)
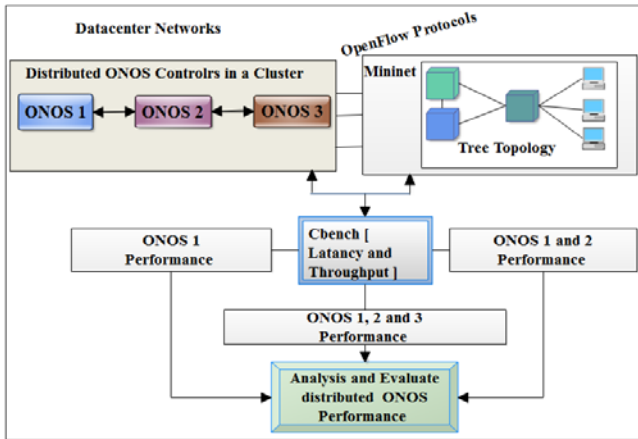
To distribute ONOS controller used clustering network
distributing techniques, because, it is fully supported by
ONOS controller. Clustering is SDN controllers distributing
method, which makes a set of strongly connected SDN
controllers, work together in a logically centralised manner
(Berde et al., 2014; Xiong et al., 2016).

3.1   Measuring and analysis of ONOS controller performance

Different researchers do their research on the performance
of Ryu, floodlight, OpenDaylight, POX, and NOX SDN
centralised controllers. So, we have motivated to do
performance analysis on a distributed ONOS controller in a
datacentre network. The analysis process of the study is
conducted using three ONOS controllers in a cluster.
Distributed ONOS controller in the cluster communicates
each other using gossip and RAFT protocols (Berde et al.,
2014).

1   Single ONOS controller

     In the first scenario, created a tree topology using
     python language. After this, run single ONOS
     controller at remote and run tree topology on Mininet
     SDN emulator. A single ONOS controller at the remote
     connection with Mininet topologies and controlled
     entire network devices. Then using cbench measured
     and analysed the throughput and latency results of
     single ONOS controllers per tree topology. Then
     analysed each topology throughput and latency result
     variations in case of the single ONOS controller.
     Finally, evaluate the throughput and latency results of
     single ONOS controller per tree topologies.

2   Two ONOS controllers

     In the second scenario, created and run two distributed
     ONOS controllers in a cluster and connected with tree
     topologies in Mininet. Then measured the throughput
     and latency of the two distributed ONOS controllers.
     Like the single ONOS controller analysed and
     discussed the throughput and latency results of two
     ONOS controllers in a cluster per tree topologies.

Finally, evaluate the performance of a single ONOS
controller per three tree topologies.

3   Three ONOS controllers

     In this scenario, created and run three distributed
     ONOS controllers in the cluster at the remote. Then
     connected with the designed tree and customised
     topologies in Mininet. Then measured the throughput
     and latency of a three distributed ONOS controllers.
     Like the single and two ONOS controllers analysed the
     throughput and latency results of three ONOS
     controllers. Finally, the researcher took the average
     throughput and latency results of single, two and three
     ONOS controllers per three tree topologies and
     analysed their performance differences.

3.2   Throughput analysis

In the throughput mode cbench test, distributed ONOS
controller in a cluster was analysed for how many total
flows it can process in a second. We tested a single, two and
three ONOS controller throughput performance with respect
to three topologies under a 10, 20 and 30 number of
switches and host. We have calculated the average
throughput differences to compare the throughput among
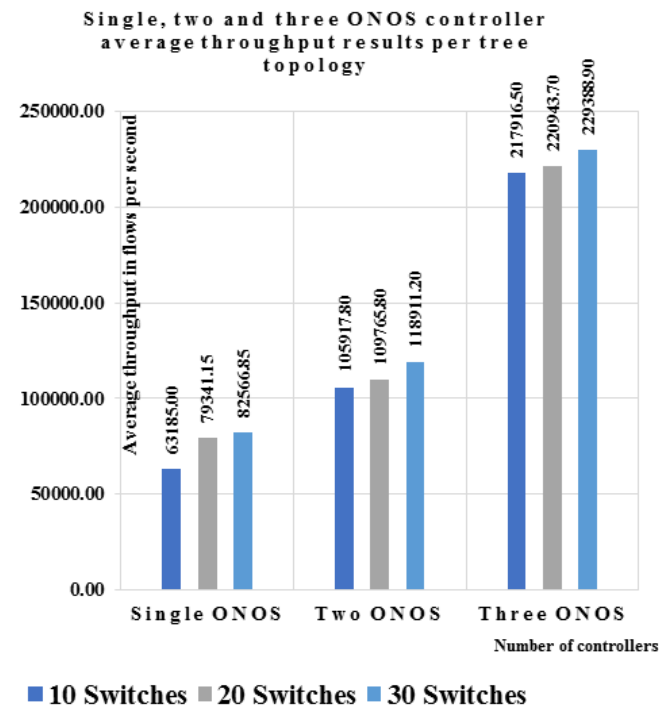three controllers.

Figure 4   Single, two and three ONOS controller average
                throughput results per tree topology (see online version
                for colours)

Figure 4 shows single, two and three ONOS controller
throughput results. The throughput increase when the
number of switches and hosts are increased from 10 to 20
and 30. The single ONOS controller has 63,185.22,
79,341.15 and 82,566.85 flow per second average
throughput results. Two ONOS controllers have 105,917.80,

109,765.80 and 118,911.20 flow per second average throughput results. Three ONOS controllers have 217,916.50, 220,943.70 and 229,388.90 flow per second average throughput results. Finally, three distributed ONOS controllers provide high throughput than single and two ONOS controllers. This enabled to increase the throughput of ONOS controller by increasing the number of ONOS controller in a cluster.

3.3 Latency analysis

The latency of ONOS controller means that how much time ONOS takes to process a given packet. We tested the distributed ONOS controller performance latency. We used the same number of switches, hosts, and links like throughput analysis. This enabled to measure a number of responses that flowed in 1 millisecond (ms). We have calculated the average response per second result differences to compare the latency among three ONOS controllers.

Figure 5 Single, two and three ONOS controller average latency results per tree topologies (see online version for colours)
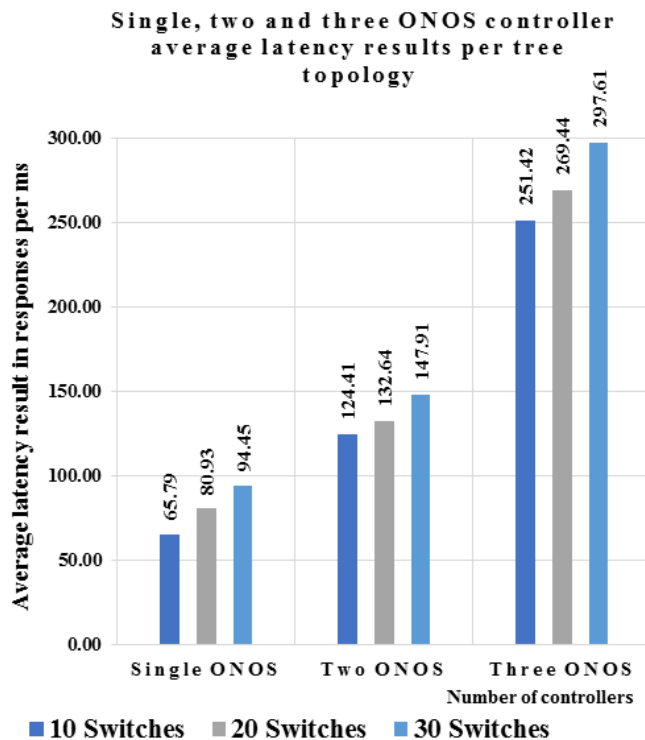

Figure 5 shows single, two and three ONOS controllers average latency results. In latency mode, the cbench measured the latency the number of responses in one millisecond. The single ONOS controller generates 65.79, 80.93 and 94.45 average responses per ms latency. Two ONOS controllers generate 124.41, 132.64 and 147.91 average responses per ms latency. Three ONOS controllers generate 251.42, 269.44 and 297.61 average responses per ms latency. Finally, three distributed ONOS controller provides high responses per ms than single and two ONOS controllers. The single ONOS controller provides minimum

responses per ms and when adding more than 30 devices in the topology it generates less and inconsistence latency results.

3.4 Evaluation

In this section, the researcher evaluated distributed ONOS controller's throughput and latency results by using previous papers results. The authors in Khattak et al. (2014) have been compared and evaluated single floodlight and single OpenDaylight controller throughput and latency. In their OpenDaylight controller test, obtained 5,500 to 45,000 average response per second and 0.005 to 0.45 average responses per ms. In their floodlight controller test, obtained 1,000 to 65,000 average response per second and 1.2 to 1.45 average responses per ms. And they conclude that floodlight controller has high throughput and low latency than an OpenDaylight controller. According to Cheng et al. (2015) this result, the single ONOS controller has high throughput and high latency than floodlight and OpenDaylight controllers. Also, two and three ONOS distributed controllers generated high amount throughput and high latency than floodlight and OpenDaylight controllers. Finally, distributed ONOS controller is the best controller than floodlight and OpenDaylight controllers.

4 Conclusions

Software-defined networking is a new network architecture that separates the control plane and the data plane. Today SDN network is a hot research area due to this researcher and network organisations highly contributed to this network. The researchers performed, reviewed literature, compared and selected SDN distributed controllers and emulator tools. Created three ONOS controllers in a cluster, designed tree Mininet topologies. Performance analysis and evaluation of distributed ONOS controller by running single, two and three ONOS controllers in a cluster has been made. To analysed and evaluated ONOS controller performance the researchers used throughput and latency key performance parameters.

The researchers presented the throughput and latency results of single, two and three ONOS controllers. The single ONOS controller has 60,665.22 to 82,566.85 flow/s throughput and 62.89 to 94.45 responses/ms latency. Two ONOS controllers have 105,564.30 to 118,911.20 flow/s throughput and 117.53 to 147.91 responses/ms latency. Three ONOS controllers have 216,816.80 to 229,388.90 flows/s throughput and 297.61 to 245.32 responses/ms latency. Single, two and three ONOS controllers per tree topology has high throughput and high response per ms.

Distributed ONOS controller performance analysis and evaluation enable to improve the controller performance. Distributed ONOS controller has high throughput and partially high latency SDN controller. However, when one controller failed in the cluster the devices controlled by the failed controller start to discover another controller in the cluster this takes a time. In addition, when the number of

devices is increasing ONOS graphical user interface not properly show the designed topology and displayed killed devices. Cbench is SDN OpenFlow controller benchmarking tool. It measures the throughput and latency of OpenFlow controllers. However, in latency mode cbench measured a number of responses/ms but not the amount of time that controller takes to provide a response. In addition, cbench have not the graphical user interface.

5   Future work and recommendation

Future work will focus on OpenFlow controllers performance parameters such as packet loss, Jitter, topology discovery time and sojourn time test. Single cluster, like that of the single controller, has a single point failure problem. Also, ONOS controller performance evaluation by using more than one cluster with different topologies. The researcher will focus on improving distributed ONOS controller performance. Furthermore, distributed controller placement problem affects ONOS controller performance. This problem needs study and mathematical model design. To create multiple controllers and distribute the controllers, ONOS uses clustering method, because ONOS is logically centralised SDN controller. The researcher recommended that ONOS should be logically and physically distributed.

References

Berde, P. et al. (2014) 'ONOSS: towards an open, distributed SDN OS', *Computer-Communication Networks*, 22 August, pp.1–6.

Cheng, G., Chen, H., Hu, H. and Wang, Z. (2015) 'DHA: distributed decisions on the switch migration toward a scalable SDN control plane', *National Digital Switching System Engineering & Technological R&D Center*, 3 December, pp.1–9.

Couto, R.S. et al. (2014) 'A reliability analysis of datacenter topologies', *Communication QoS, Reliability and Modelling Symposium*, 26 September, pp.1890–1895.

Feamster, H.K.N. (2013) 'Improving network management with software defined networking', *IEEE Communications*, 23 February, pp.114–115.

Forouzan, B. (2007) *Data Communication, and Network*, 5th ed., Alan R. Apt, Amsterdam.

Humayun Kabir, M. (2013) 'Software defined networking (SDN): a revolution in computer network', *IOSR Journal of Computer Engineering (IOSR-JCE)*, 8 December, Vol. 15, No. 5, pp.103–106,

Khattak, Z.K., Awais, M. and Iqbal, A. (2014) 'Performance evaluation of OpenDaylight SDN controller', *Software Defined Networking Benchmarking*, 23 December, pp.1–6.

Khondoker, R. et al. (2014) 'Feature-based comparison and selection of software defined networking (SDN) controllers', *Controller Comparison*, Vol. 10, No. 3, pp.1–7.

Kreutz, D. et al. (2015) 'Software-defined networking: a comprehensive survey', *Software-Defined Networking*, 31 January, Vol. 103, No. 1, pp.1–61.

Li, W., Menga, W. and Kwok, L.F. (2017) 'A survey on OpenFlow-based software defined networks: security challenges and countermeasures', *Journal of Network and Computer Applications*, 10 May, pp.3–4.

McKeown, N. (2017) *Stanford*, Stanford University, 15 January 2016 [online] http://yuba.stanford.edu/ (accessed 23 June 2017).

ONF (2016) *ONF, Listing Mambers (Ouer Members)*, November [online] https://www.opennetworking.org/our-members (accessed 17 November 2017).

Rowshanrad, S., Abdi, V. and Keshtgari, M. (2016) 'Performance evaluation of SDN controllers: floodlight and opendaylight', *IIUM Engineering*, 30 November, Vol. 17, No. 2, pp.47–56.

SDX (2016), *sdxcentral*, SDX site, 12 January [online] https://www.sdxcentral.com/res7ources/sdn/who-is-open-networking-foundation-onf/ (accessed 25 June 2017).

Techopedia (2016) *Distributed Network*, Techopedia site, 26 December [online] https://www.techopedia.com/definition/27788/distributed-network (accessed 12 August 2017).

Tootoonchian, A., Gorbunov, S. and Ganjali, Y. (2012) 'On controller performance in software-defined networks', *Cloud, and Enterprise Networks and Services (Hot-ICE)*, pp.1–5.

Wang, A.J. (2012) 'Scalable software-defined network controllers', in *Proceedings of the ACM SIGCOMM 2012 conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*.

Wang, T., Liu, F. and Xu, H. (2016) 'Dynamic SDN controller assignment in data center networks: stable matching with transfers', in *IEEE INFOCOM 2016 – The 35th Annual IEEE International Conference on Computer Communications*, Hong Kong.

Webtorials (2016) *Citrix*, '*webtorials*', 8 January [online] http://www.webtorials.com/content/2014/01/2013-guide-to-network-virtualization-sdn-3.html (accessed 24 May 2017).

Xie, J. et al. (2015) 'Control plane of software-defined networks: a survey', *Computer Communications*, 18 June, pp.1–10.

Xiong, B., Yang, K., Zhao, J., Li, L. and Li, K. (2016) 'Performance evaluation of OpenFlow-based software-defined networks based on queueing model', *Computer Networks*, 4 April, pp.172–184.

Zuccaro, L. et al. (2015) 'Distributed control in virtualized networks', The *10th International Conference on Future Networks and Communications (FNC 2015)*, pp.276–283.