Bulletin of Electrical Engineering and Informatics
Vol. 9, No. 6, December 2020, pp. 2588~2594
ISSN: 2302-9285, DOI: 10.11591/eei.v9i6.2581

Optimal software-defined network topology for distributed denial of service attack mitigation

Branislav Mladenov, Georgi Iliev
Department of Telecommunications, Technical University of Sofia, Sofia, Bulgaria

Article Info

Article history:

Received Jan 31, 2020
Revised Mar 25, 2020
Accepted Apr 15, 2020

Keywords:

DDoS attack
Mininet
Openflow
Software-defined networking

ABSTRACT

Distributed denial of service (DDoS) attacks are a major threat to all internet services. The main goal is to disrupt normal traffic and overwhelms the target. Software-defined networking (SDN) is a new type of network architecture where control and data plane are separated. A successful attack may block the SDN controller which may stop processing the new request and will lead to a total disruption of the whole network. The main goal of this paper is to find the optimal network topology and size which can handle Distributed denial of service attack without management channel bandwidth exhaustion or run out of SDN controller CPU and memory. Through simulations, it is shown that mesh topologies with more connections between switches are more resistant to DDoS attacks than liner type network topologies.

Corresponding Author:

Branislav Mladenov,
Faculty of Telecommunications,
Technical University of Sofia,
8 Kl. Ohridski Blvd, Sofia 1000 Bulgaria.
Email: branislav.mladenov@gmail.com

1. INTRODUCTION

Software-defined networking (SDN) introduce more flexible and scalable services [1, 2]. Security is one of the main topics behind SDN [3], unfortunately, with separate control and data place, the centralized architecture is vulnerable to attacks that target to destroy or disrupt the central SDN unit [4, 5]. SDN controller is the brain of the network but once it is down or inaccessible all new requests will not be processed. A successful distributed denial of service (DDoS) type of attack may successfully exhaust the CPU and memory of the controller. Once the goal is achieved, all new legitime and non-legitim requests will be blocked by the SDN controller.

SDN architecture consists of three main layers and they are as follows: Infrastructure, control and application layer as shown in Figure 1 [2]. The application layer is responsible for applications and communicates with the SDN controller via REST API. The SDN controller translates the requirements provided from the application layer to instructions and configurations on the infrastructure layer. There are several types of DoS attacks that can target each of these three layers. This article is focused on the control layer DDoS mitigation. OpenFlow [6] protocol instructs the switches to send *Packet_in* events to the controller when the requests to the destination are not part of their local flow tables as shown in Figure 2.

The controller decides on how to process the request. If the decision is to approve and process the request, the controller instructs the switch which creates a flow entry in its flow table. Each unique request creates two flow entries in each OpenFlow switch on the way of the chosen path. Software-defined networks have less latency and jitter than traditional networks [7] so during DDoS attack, all requests

are from spoofed IP addresses and each of them is asking the controller to take a proper decision. Due to the huge number of unique requests, all OpenFlow enabled switches may reach the limit of their flow tables fast. OpenFlow switches have a feature to send to the controller the whole packet instead of the header only which may lead to overload the management interfaces between the switch and SDN controller.

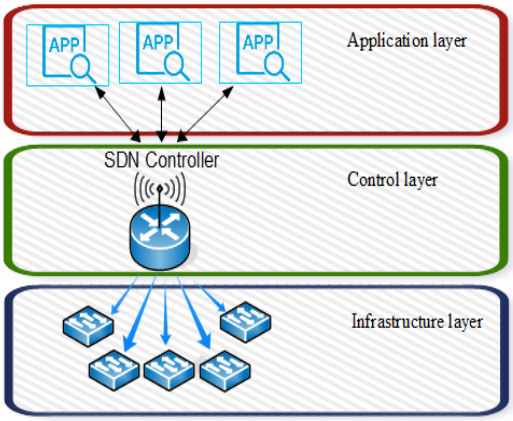

Figure 1. SDN controller layers model

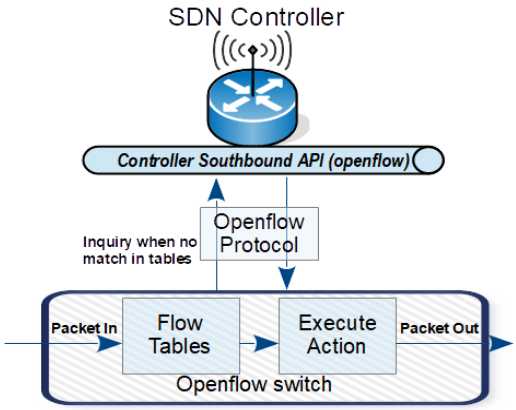

Figure 2. OpenFlow process

There are many proposed solutions for DDoS mitigation [8] and detection [9] but if a DDoS attack is not mitigated before it reaches the SDN controller, the control layer might be affected. Successful DDoS protection is a solution provided by the Internet Service Provider and all non-legitim packets are filtered before they hit the client's network. Mostly used DDoS attacks are volumetric UDP/ICMP amplifications and flood type of attacks. They overload the internet traffic channels and interrupt legitimate traffic [10]. The target of such attacks is corporate networks, data centers with web hosting services, blockchain networks [11] and even campus networks [12]. State exclusion TCP and application-layer attacks may mostly affect the control layer. Some controllers have their protection mechanisms and they stop processing traffic when a threshold is reached. This is good for the controller, but the real effect is that the newly arrived legitim request is not processed as well.

This article is organized as follows. SDN architecture and DDoS effect over it are reviewed in section 2. Related works are discussed in section 3. Section 4 introduces the experimental setup, an explanation about how the DDoS attack is a simulated and different type of topologies. A comparison between them is reviewed and analyzed. The article is concluded in section 5.

2. RELATED WORKS

Our previous work was focused on studying the DDoS attack effect of bandwidth utilization over the SDN controller southbound channel and CPU utilization on SDN controller [13]. The main goal of this paper is to measure the effect when multiple requests for different sources are entering the networks and how the southbound channel is affected. We did two experiments with single switch topology and tree topology with multiple switches but the same number of hosts and then we compared the results for these two topologies. The hosts represent the number of unique sources that are generating new connections that trigger packet_in events. The third experiment was focused on how fast the controller's threshold will be reached by increasing the count of hosts. In this work, the same lab environment is used, but this time we were looking for optimal topology and count of switches part of the network. Increasing the number of switches with the same number of hosts the delay of the controller's response is increasing as well. With switch count increase the network becomes more reliable and high availability with more clients/users that can communicate faster with each other. With the growing number of Openflow switches within the networks, the control plane is a potential bottleneck. The problem with one centralized controller can be easily solved with logically centralized but physically distributed architecture with several controllers [14]. Such architecture can increase the control plane scalability of the networks. Unfortunately, such architecture has drawbacks if the switches are statically configured as the network flow and traffic patterns are unpredictable and at some time of the day or week, one of the controllers can be fully loaded while the rest may have a much smaller load or unutilized. With statically mapped switches the DDoS attack will affect only the controllers that are facing the incoming requests so this would not solve the problem.

There are several solutions for dynamically rebalancing architecture with multiple controllers which can help with scalability on the control plan level. The JGroups [15] proposal is splitting the network into several logical groups and each group is controlled by one active/standby controller cluster. With switch migration, the load can be shifted from one cluster to another to even load balancing to be achieved. This approach avoids a single point of failure because of the supports controller's failover without disruptions. ElastiCon [16] is a solution that presents dynamically increase or decrease of SDN controllers based on load estimation of control plane usage. ElastiCon architecture manipulates the control plane without any interruption as well. Both solutions use a role-request message of OpenFlow protocol for their purposes. In the case of DDoS attack, JP Groupe architecture can be used for mitigation but it will work only if the attack is not so big. Most of the enterprise customers have at least two active/active or active/standby Internet service providers circuits so during the attack the ingress traffic will hit only one or two switches and respectively one or two controllers.

3. RESEARCH METHOD

The experimental setup is presented in Figure 3. Two virtual machines (VMs) with the following characteristics CPU: Intel i5 6300 2.4 GHz with 4 GB of RAM are connected via a virtual switch. On the first VM of the most popular SDN controllers, the Floodlight OpenFlow controller is installed. On the second VM, Mininet OpenFlow [17, 18] switch emulator is installed. OpenFlow 1.3 version is used for a communication protocol between the controller and Mininet switch. The connection between VMs is via the VMware virtual switch. With a Mininet simulator, a different number of hosts have been created. With the additional script, these hosts are forced to generate traffic targeting other hosts from the network.

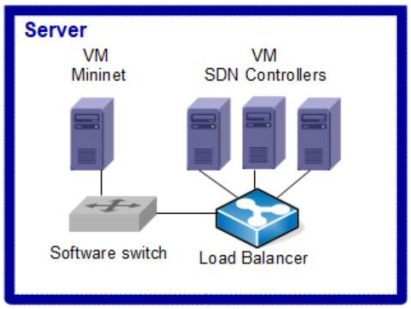

Figure 3. OpenFlow process

These unique requests generate unique connections that must be processed by the SDN controller, so this simulates the effect of a DDoS attack. Depends on the number of hosts the unique requests can be up to 100 requests per second. To load balance the traffic between Mininet simulated switches and SDN controllers a load balancer is configured to load share the southbound connections. The load balancer uses a round-robin algorithm to load-balance all connections equally between controllers. Each switch should communicate with the same controller whenever a new request is triggered so source address persistence is used for the persistence algorithm.

Tests were done on two types of topologies. The first one Figure 4 is the liner and one switch is connected to the other two only. Each switch has at least one host connected as well. The path from the first switch to the last one is passing through all switches of the network, so the controller must configure all switches for each traffic flow request. If we compare the first topology with existing productive topologies it is closer to small office (SOHO) topology where only few end hosts are connected to the switches and each switch is connected to the other one.

The second type Figure 5 of topology is more mesh than linear. Depends of the settings each switch is connected to several others so the traffic flow has more redundancy paths and if we try to compare the traffic flow of this topology with the first one, when the first host needs to communicate with the last host the packets will not go through all switches of the network, but the controller will choose the shortest path based on SPF algorithm. This means that the controller needs more CPU and resources for first type topology comparing with the second one. Second type of topology is more mesh it is looks closer to spine-leaf topologies where each leaf switch is connected to every spine switch [19, 20]. The main advantage is that each switch has several links connected to the other switches which provide more redundant paths. Spine-leaf topologies are more cost effective and provide better performance comparing with other topologies [19].

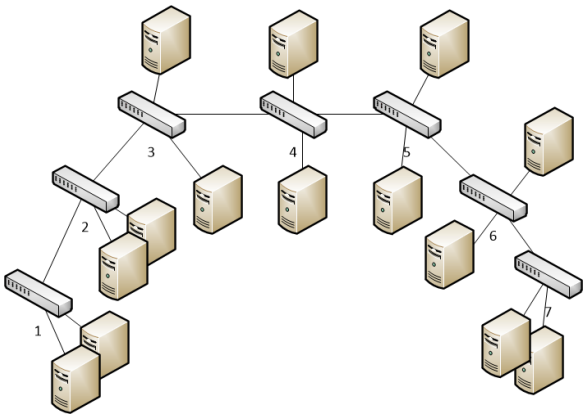

Figure 4. First type topology-linear

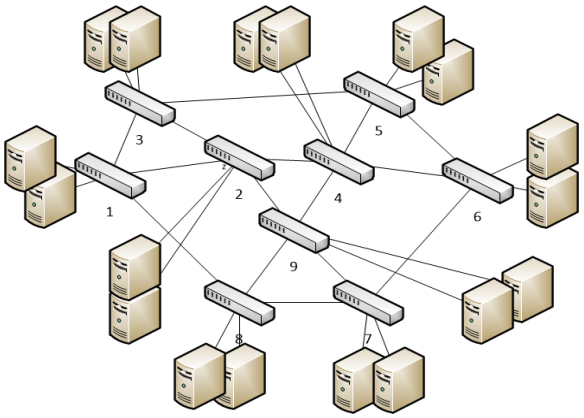

Figure 5. Second type topology-mesh

4. RESULTS AND DISCUSSION

We did several experiments with both topologies and a different number of controllers. Both topologies are tested under the same type of attack and all controllers have the same CPU and memory resources. We have reviewed similar researches [21, 22] with similar tests with Onos controllers [23], but our results are little bit different because of the size of the network and type of the controller-Floodlight [24, 25]. In these researches the networks have only 3 switches and one controller which is not so close to real productive networks. In our experiments we simulate networks starting from 3 up to 80 switches and the simulation of volumetric DDoS attack is closer to real productive networks which explain the difference in results. When the controller takes decisions in bigger networks it should calculate and instruct more switches which takes allocates more CPU/memory resources.

4.1. The first type of topology experiment

In Figure 6 results for the first type topology can be reviewed. The same types of tests are done with one, two and three controllers. We can see that with one controller the optimal topology contains only 22 switches. If more switches are added and similar DDoS attack is triggered the SDN controller win start rejecting new requests so neither the legitim nor malicious traffic will be processed, and the attack will be successful. If the second active SDN controller is added the results show that the optimal topology is 48 switches. Once 3 controllers are activated the maximum count of switches that can handle such attack is 70. The result shows that that first topology reacts similarly during the attack. With increasing the active controllers, the total cost of network infrastructure is increasing as well. It is recommended behind each active controller one standby to be added in a cluster mode for high availability. In case that 3 active controllers are needed some low budget, designs recommend only 2 standby controllers to be deployed to save some cost.

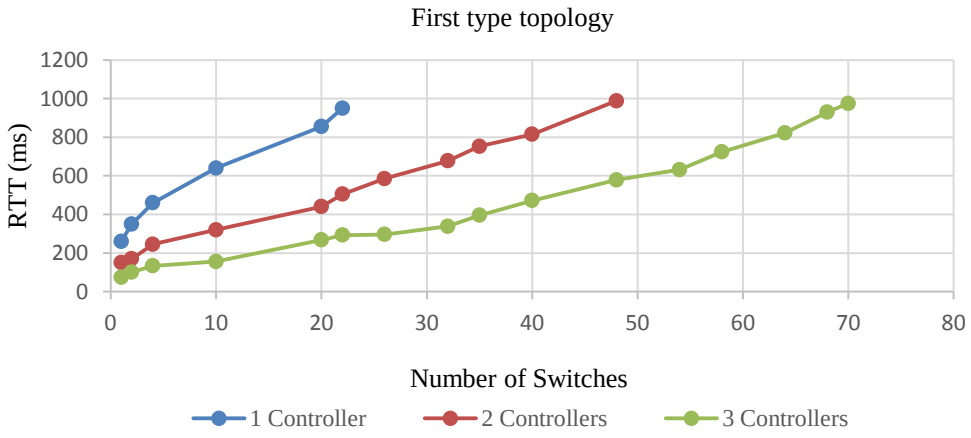

Figure 6. First type topology

4.2. The second type of topology experiment

The second experiment is similar to the first one, but the topology is different. In Figure 7 the results of mesh topology can be seen. The second topology looks more stable and does not consume such resources comparing with the first one. If only one controller is active and same attack is triggered to the network, the topology shows that 26 switches can be active until the SDN controller decides to block all new requests. Once the second active controller is added to the networks it can manage up to 50 switches during a simulated attack. When three controllers are active the number of switches is increased up to 78.

The second topology looks more stable and does not consume such resources comparing with the first one. If only one controller is active and same attack is triggered to the network, the topology shows that 26 switches can be active until the SDN controller decides to block all new requests. Once the second active controller is added to the networks it can manage up to 50 switches during a simulated attack. When three controllers are active the number of switches is increased up to 78.

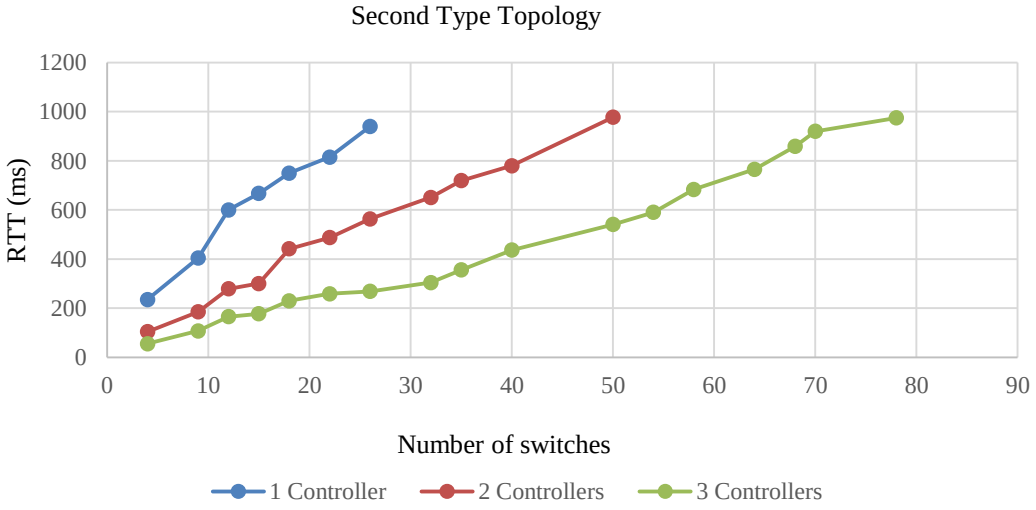

Figure 7. Second type topology-mesh

4.3. Compare both topologies with one and three controllers

The next two figures-Figure 8 and 9 show the comparison between both topologies and based on that we can conclude which of these two is more stable and resistant to DDoS attack.

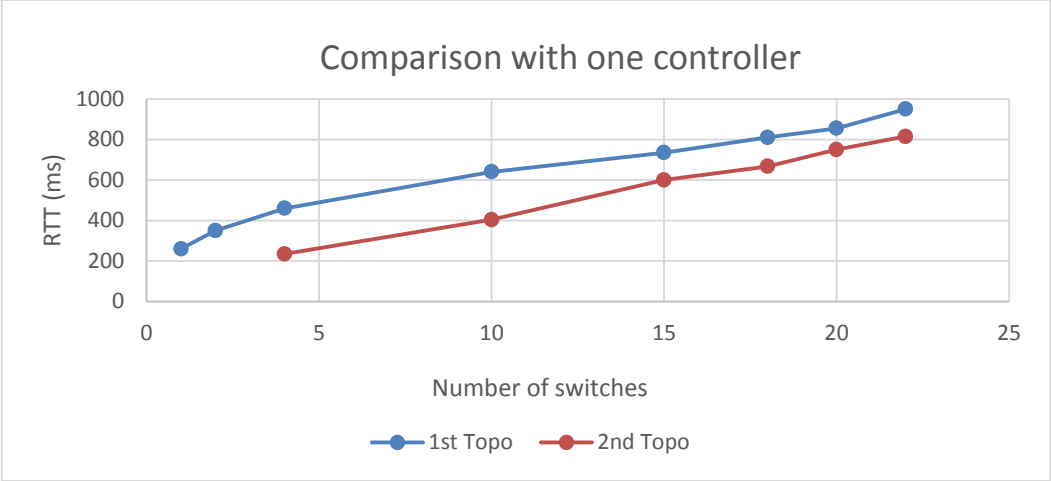

Figure 8. Comparison between both topologies with one controller

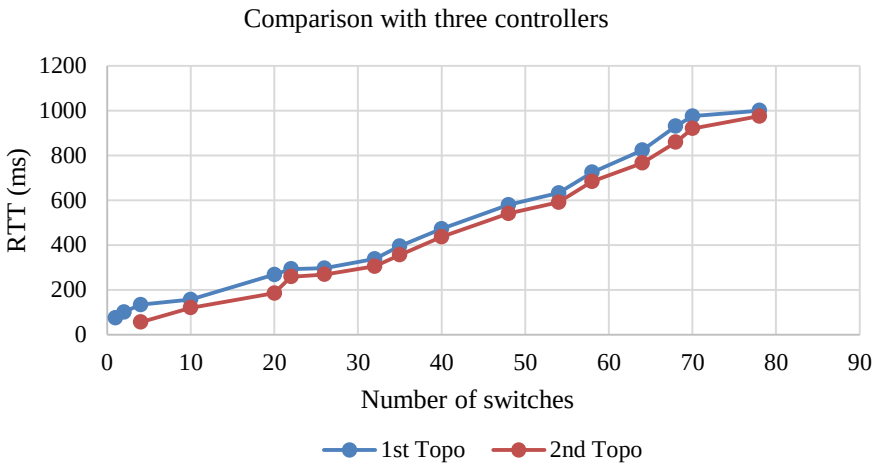

Figure 9. Comparison between both topologies with three controllers

5. CONCLUSION

This paper has presented an optimal network topology and size to handle distributed denial of service attack without management channel bandwidth exhaustion or run out of SDN controller CPU and memory. It can be concluded that the second type of topology is more resistant to DDoS attack. When only one controller is active the second topology looks more stable than the first topology type. If three controllers are activated to process new requests the results show that the difference between both topologies is much smaller but still the second type topology is more stable. We can conclude that mesh topologies with more links between switches are more resistant to DDoS attacks. In the future, we are planning to do researches with different types of controllers-Onos, Floodlight, and Ryu or different OpenFlow versions.

ACKNOWLEDGEMENTS

This work was supported by the Project № ДН 07/22 of the Bulgarian "Scientific Research Fund".

REFERENCES

[1] A. L. Valdivieso Caraguay, L. I. Barona Lopez and L. J. Garcia Villalba, "Evolution and Challenges of Software Defined Networking," *2013 IEEE SDN for Future Networks and Services SDN4FNS*, Trento, pp. 1-7, 2013.
[2] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," in *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14-76, Jan 2015.
[3] N. Dayal, P. Maity, S. Srivastava, "Research trends in security and DDoS in SDN," *Security and Communication Networks*, vol. 9, no. 18, pp. 6386-6411, 2016.
[4] M. Antikainen, T. Aura, M. Särelä, "Spook in your network: attacking an SDN with a compromised OpenFlow switch," 19th *Nordic Conference on Secure IT Systems*, Norway, pp. 229-244, 2014.
[5] D. Kreutz, F. M. Ramos, P. Verissimo. "Towards secure and dependable software defined networks," *HotSDN 2013-Proceedings of the 2013 ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, New York, USA, pp. 55-60, 2013.
[6] A. Lara, A. Kolasani and B. Ramamurthy, "Network Innovation using OpenFlow: A Survey," in *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 493-512, First Quarter 2014.
[7] P. Numan, K Yusof, M Marsono, "On the latency and jitter evaluation of software defined networks," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 8, no. 4, pp.1507-1516, December 2019.
[8] B. Mladenov, "Research and solutions for DDoS detection and mitigation with SDN," *53rd ICEST, Sozopol, Bulgaria*, vol. 28-30, pp.142-146, 2018.
[9] A. Fadil, I. Riadi, S. Aji, "Review of detection DDOS attack detection using naive bayes classifier for network forensics," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 6, no. 2, pp.140-148, June 2017.
[10] N. Rajkumar, "A survey on latest DoS attacks: classification and defense mechanisms," *International Journal of Innovative Research in Computerand Communication Engineering*, vol. 1, no. 8, pp.1847–1860, 2013.
[11] Y. K. Tomov, "Bitcoin: Evolution of Blockchain Technology," *2019 IEEE XXVIII International Scientific Conference Electronics ET*, Sozopol, Bulgaria, pp. 1-4, 2019.

[12] M. Naagas, E. Mique , T. Palaoag, J. Cruz, "Defense-through-deception network security model: Securing university campus network from DOS/DDOS attack," *Bulletin of Electrical Engineering and Informatics*, vol. 7, no. 4, pp. 593-600, December 2018.
[13] B. Mladenov, "Studying the DDoS Attack Effect over SDN Controller Southbound Channel," *2019 X National Conference with International Participation ELECTRONICA*, Sofia, Bulgaria, pp. 1-4, 2019.
[14] M. F. Bari *et al*, "Dynamic Controller Provisioning in Software Defined Networks," *Proceedings of the 9th International Conference on Network and Service Management CNSM 2013*, Zurich, pp. 18-25, 2013.
[15] C. Liang, R. Kawashima and H. Matsuo, "Scalable and Crash-Tolerant Load Balancing Based on Switch Migration for Multiple Open Flow Controllers," *2014 Second International Symposium on Computing and Networking*, Shizuoka, pp. 171-177, 2014.
[16] A. Dixit, F. Hao, R. Kompella, "Towards an elastic distributed sdn controller," *ACM/IEEE Symposium on Architectures for Networking and Communications Systems ANCS*, pp. 7–12, 2014.
[17] I. Bholebawa, R. Jha, U. Dalal, "Performance analysis of proposed OpenFlow-based network architecture using mininet," *Wireless Personal Communications*, vol. 86, no. 2, pp.943-958, 2016.
[18] F. Keti and S. Askar, "Emulation of Software Defined Networks Using Mininet in Different Simulation Environments," *2015 6th International Conference on Intelligent Systems, Modelling and Simulation*, Kuala Lumpur, pp. 205-210, 2015.
[19] M. Alizadeh and T. Edsall, "On the Data Path Performance of Leaf-Spine Datacenter Fabrics," *2013 IEEE 21st Annual Symposium on High-Performance Interconnects*, San Jose, CA, pp. 71-74, 2013.
[20] M. Alizadeh, T. Edsall., "CONGA: distributed congestion-aware load balancing for datacenters," *In Proceedings of the 2014 ACM conference on SIGCOMM* , pp. 503-514, Aug 2014.
[21] R. K. Arbettu, R. Khondoker, K. Bayarou and F. Weber, "Security analysis of OpenDaylight, ONOS, Rosemary and Ryu SDN controllers," *2016 17th International Telecommunications Network Strategy and Planning Symposium Networks*, Montreal, QC, pp. 37-44, 2016.
[22] S. Hamid, N. Zakaria, S. Ahmed, "ReCSDN: Resilient controller for software defined networks," *International Journal of Advanced Computer Science and Applications IJACSA*, vol. 8, no. 8, pp.202-208, 2017.
[23] U. Krishnaswamy et al, "ONOS: an open source distributed SDN OS," *Proc. of the third workshop on Hot topics in software defined networking (HotSDN'14)*, pp.1–6, Aug. 2014.
[24] Floodlight is a Java-based OpenFlow controller, 2012. [Online]. Available: http://floodlight.openflowhub.org/
[25] Nippon Telegraph and Telephone Corporation, "Ryu Network Operating System," 2012. [Online]. Available: https://ryu.readthedocs.io/en/latest/

BIOGRAPHIES OF AUTHORS

Branislav Mladenov received his bachelor degree in telecommunications engineering in 2007, from Technical University of Sofia and master degree in Business Administration in 2011, from University of National and World Economy in Sofia. He is currently pursuing the Doctor of Philosophy Degree with faculty of Telecommunications, Technical University of Sofia. His current research interest includes Software define Network, Computer Networks and Systems.

Georgi Iliev received MEng degree in Telecommunications from Technical University of Sofia (TUS), Bulgaria in 1990 and PhD degree in Adaptive Signal Processing from the same university in 1996. He was with the Department of Telecommunications, Technical University of Sofia from 1993 as an Assistant Professor, and from 2003 as an Associate Professor. In 2011 he was elected as Head of Department of Communications Networks and in 2012 became a full Professor in the same department. His interests are in signal processing, adaptive systems and algorithms, and noise cancellation. He has been working on systems for speech recognition in noisy environments, adaptive equalisation for communication channels, and the development of new computationally-efficient adaptive algorithms.