Detection and Prevention from DDoS Attack Using Software-Defined Security

Sumit Badotra(✉), Surya Narayan Panda, and Priyanka Datta

Chitkara University Institute of Engineering and Technology, Chitkara University, Raipura, Punjab, India
{sumit.badotra,snpanda,priyanka.datta}@chitkara.edu.in

Abstract. The network which is able to accommodate today's real-time need is growing in a very fast manner. But simultaneously also occurs an increase in the rate of network attacks and threats. Distributed Denial of Service (DDoS) is one of the attacks in which intruder attempts to disrupt normal network traffic by flooding huge traffic into the network and ultimately halt the network services and resources. There are numerous solutions available for the detection and prevention of DDoS attacks in traditional networks but making use of Software-Defined Security (SDS) is a new way of securing the network. The basic principle of separating the intelligence of the network from the infrastructure can be considered as the new hope for securing the network. This chapter aims to provide the need for SDS in networks with related literature survey we have also found out the research gaps from research done till now or going on. A method to prevent a network from DDoS attacks is also proposed using SDS.

Keywords: Traditional networks · DDoS attack · Software-defined security

1 Introduction

In order to build many of the network devices and middleboxes like network switches, routers, load balancers for network, firewalls, Network Address Translation (NAT), etc. used in the network, each and every device needs to be manipulated individually. It is very difficult to make any changes in the traffic with the help of such intermediated. This change in traffic is very complex as compared to simple packet forwarding. Multiple complex network protocols are constituted by these intermediated network devices [1]. These devices are vendor-specific and hence it becomes a tedious job for a network administrator to configure these devices individually. Traditional networks are not only suffering from aforementioned challenges but also suffer from security attacks and threats as well. Although there are many solutions proposed until now to overcome these threats but with the complexity that these networks are comprised of is very difficult to overcome from network attacks. Network attacks can be categorized into two types: Active and Passive. Active Attacks are those types of attacks in which square measure are those within which the hacker makes an attempt to change knowledge or data traveling from sender to receiver within the network. A number of the active attacks square

C. R. Panigrahi et al. (eds.), *Progress in Advanced Computing and Intelligent Engineering*, Advances in Intelligent Systems and Computing 1198, https://doi.org/10.1007/978-981-15-6584-7_21

measure spoofing attack; Spoofing or Hollow attack, Modification, Denial of services (DoS), Sinkhole, and Sybil attack [2]. Passive Attacks are those kinds of attacks within which hackers or unwelcome person don't make changes or modify the information traveling in between the sender and receiver. The intention behind this attack is to browse and analyze the information. A number of passive attacks are traffic analysis, Eavesdropping, and observance [3]. In the chapter, we have taken DDoS attack as a point of study. It is an attempt to disrupt the normal traffic by flooding a huge traffic to the targeted server and ultimately halts the services provided by the server for the legitimate users as well. To overcome the situation in traditional networks Software-Defined Networking (SDN) has come into act [4]. The complexity of today's real-time network is increased at a huge rate. In order to accommodate the alterations making the network programmable is the only solution. This will help in meeting the various requirements of the users. By segregating the intelligence of the network from the proprietary hardware is making it simple to incorporate the various amendments in the network. This segregation is achievable with the help of SDN. The new modified applications and techniques for network management are very easy to implement and use [5]. By making the simplified management and view of the network, new security features can be easily implemented, and hence SDN based network architecture is able to cope with the network attacks.

The main contributions of this chapter are as follows:

- To review current security issues and limitations in networks.
- To find out the research gaps from the previous work done related to securing the network from a DDoS attack.
- Based on these identified gaps, a method or framework is also proposed, which is making use of SDS for detection and prevention of DDoS attacks.

The remainder of the chapter is organized as follows: Sect. 2, related survey regarding the security techniques/approaches used to prevent network from DDoS attack is given. Section 3 is comprised of research findings from the literature studied. In Sect. 4 is comprised of SDN controller whereas in Sect. 5 a discussion is given on the proposed approach to diminish the DDoS attacks by using SDN and finally conclusion is stated in Sect. 6.

2 Literature Survey

Previous literature lays a foundation to formulate the objective of the research. The existing literature shows state of art technologies used to prevent network against DDoS attack connected works show that each of DDoS attack and countermeasures is kept evolving and growing. It can be observed that traditional methods of mitigating DDoS attacks are mostly on the basis of IP traceback, anomaly detection, filtering (ingress/egress), ISP defense, and network self-similarity as shown in Fig. 1.

Braga et al. [6] have presented a method in order to detect the DDoS attack which is a lightweight. This method is based on network traffic flow features, in which taking out or withdrawal of such information about the attack is made with very low overhead compared to old approaches used in the respective domain. They have made use of

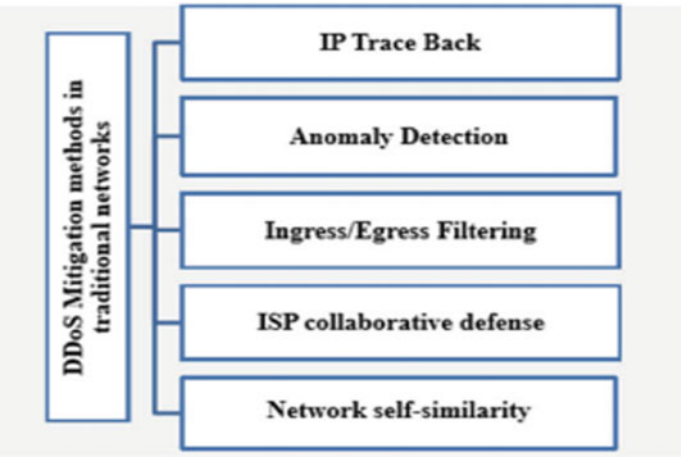

Fig. 1. Traditional methods for militating against DDoS attack

NOX SDN controller, which was providing an interface to fetch such information from network switches. Another major contribution they have made is to include the high rate of detection and very low rate of false alarms obtained by analyzing the network flow using Self Organizing Maps (SOM). The drawback of their work is that they have made use of NOX controller, which is just a reference SDN controller. This work is not executed with a practical based SDN controller such as RYU, Floodlight, etc.

Jun et al. [7] have proposed another method to mitigate the DDoS attack using the flow entropy- and packet sampling-based mechanism which is used to detect DDoS attack. To differentiate between normal traffic and network traffic generated by a DDoS attack, they had used OPNET simulation results. The limitation of their work was that they had only proposed a model using a simulator, the results of using a simulator may vary when implemented in the real world.

Jyothi et al. [8] designed a framework to detect DDoS attack called Behavior-based Adaptive Intrusion detection in Networks (BRAIN). According to the proposed method, various multiple applications behavior is making use of low-level network hardware events. The approach had added advantages for already available hardware performance counters. The combination of previous network traffic statistics and modeled network application behavior for detecting DDoS attacks by making use of machine learning is achieved in this case.

Li et al. [9] have proposed a new system called Drawbridge, which is used to address and manage the network traffic. They have made use of SDN in their research and ultimately, they have investigated a solution through which enabling end hosts to make use of their knowledge of network traffic which they desire in order to improve traffic flow during DDoS attacks. These approaches [9] had shown many limitations to mitigate DDoS attacks and therefore provide a viable solution for that using SDN. In order to differentiate between authorized users from intruders the Trust Management Helmet (TMH) model achieves this by the difference in registering four styles of trusts accustomed and transferred as a part of a license at the multiple users for session reference

to the targeted server firmly [10, 11]. However, this approach also suffers from many disadvantages such as license forgery, replay, deletion attacks, and either by sharing or using the duplicate or previously issued licenses attackers may cheat and ultimately the model fails. To overcome the disadvantages in the previous model on the network routers, approach having synchronization comprised of two-tier network traffic filters from distrustful traffic. The monitoring of the network traffic with a mechanism called as unique RED/Drop tail is also depicted in [11]. However, in this case, also spoofed addresses generated by attackers won't be captured by the routers in the network and therefore it will give an open chance for attackers to launch a DDoS attack. A new approach called a blacklist approach is used later on to overcome the shortcomings. This approach uses a communications protocol. It additionally adds CAPTCHA to differentiate among legitimate users and botnets [12]. Therefore, the protocol to provide the communication between user and server is rejected and the mechanism of CAPTCHA was planned only to provide the mitigation of DDoS attacks specific to application layer botnet. On the other hand managing flash crowd events [13] still remained unaddressed. The use of CAPTCHA may create a hindrance as well to most of the legitimate users and will create a negative impact on various operations which work online. Among all solutions to mitigate DDoS attacks, entropy-based solutions have gathered a lot of attention [14]. However, in spite of various solutions provided and stated the aforementioned related to entropy, these solutions lack to detect low-rate and high-rate DDoS attack. The point of considering entropy to various options from traffic flows was accustomed to kind traditional patterns victimization clump analysis algorithmic rule and determine the deviations from the models that are created [15]. This proposed approach suffers from many cons and it needs an effective algorithmic rule to overcome the issues such as back process time and memory usage in a very high volume and at a very high-speed network. In order to overcome the various limitations in this approach a quick entropy technique that follows the victimization of traffic flow which is based on network analysis was planned [16]. However, this approach is not able to search out the offender and agents responsible for DDoS attack underneath this approach. To make use of cloud computing to defend against the DDoS attacks is not a new technique. Resource distribution of resources dynamically was planned and supported through the queuing theory [17]. Still, servers that are hosted by cloud are vulnerable to the DDoS attacks [18]. Making use of honeypots can be considered as a brand-new effort in providing the mechanism for defense in the network. In the approach of honeypots, on physical servers, a network of virtualized honeypots was deployed and then observation of incoming traffic or malicious activities all together with flooding packets was observed [17]. However, this approach is also vulnerable because the network routers have already been flooded with multiple malicious requests before the honeypots come into play. Another method to mitigate DDoS attacks was ant-based. This technique was victimized by virtual honeypots [18]. Attackers, in this case, may be able to identify the honeypots and it can become a launching pad for attackers to launch DDoS attacks either on the system itself or network and thus worm from one honeypot may spread to other networks as well.

Shin et al. [19] have considered two aspects of one of the most common communication protocols which is used to interact between infrastructure plane and control plane, i.e., OpenFlow. The first aspect is the bottleneck or full memory space of controller

and the second aspect is that of enabling the control plane to expedite both detection of, and responses to, the changing flow dynamics within the data plane. They have provided solutions to both aspects as well but the limitation of their work was that in SDN based network attack between data-to-control plane, sending bogus packet request continuously which increases the network latency and ultimately can lead to DDoS attack.

Shoeb et al. [20] have proposed a method in order to control the network traffic communication flow between the control layer and the infrastructure layer. This enables the key principle to execute the network amendments in a very efficient way. Based on the multiple OpenFlow devices requests sent or received by the underlying networking devices, using the priority method or the traffic flow configuration the compatibility of the network switch are maintained as well. The time out value for network flow between control plane and data plane is considered by increasing the efficiency of both controller and switch by proposing a method that is feasible and efficient.

Wang et al. [21] have presented an architecture to mitigate DDoS attack that facilitates to make the network programmable and flexible in their method they have used graphical model-based attack detection method which can overcome the problem of the dataset. They have used simulation tools to perform the experimentation. The limitation was high latency, low scalability.

Zheng et al. [22] proposed a real-time DDoS Defense using COTS SDN switches via adaptive correlation analysis, it is used to detect DDoS attacks via adaptive correlation analysis on COT SDN switches. The disadvantage of the proposed work was new emerging sophisticated DDoS attacks (e.g., Crossfire) constructed by low rate and short-lived "benign" traffic are even more challenging to capture.

Tseng et al. [23] have described a protocol PATMOS which was proposed to mitigate against DDoS attack in multi-controller environment using clustering. The main advantage of their work is that they have eliminated overloaded dependency on a single controller which ultimately reducing the CPU usage rate and hence increasing throughput. The proposed work suffers from some cons which is an analysis of network traffic and an increase in the computational cost of the network. Badotra et al. [24] have implemented an SDN based firewall using RYU controller which works on both the transport layer and application layer. In [25] SDN based Collaborative Scheme for Mitigation of DDoS attack is proposed. This work has made use of RYU and POX controller which is not being used in industries. It is just used for experimentation purposes only.

Therefore, from the literature survey discussed above, we can conclude that although there are a number of solutions available to mitigate the DDoS attack in traditional networks, these methods are inadequate because nowadays attackers are making use of dynamic methods of DDoS attack. Therefore, we need a practical and intelligent solution to implement security into the networks. By making use of SDN with its open-source controllers and the basic principle of separating the intelligence and data plane of the network many customized APIs can be built which are open-source and can be used for providing security. We no longer need any middlebox and dedicated hardware which is vendor-specific and non-configurable.

3 Research Gaps

SDN can be considered as an active and vast research area in the field of networking. A large number of researchers are working in various domains of SDN but very few researchers have tried to unfold the security feature of SDN and making it to the most of use. Providing security to the network by using SDN can overcome the various challenges faced by traditional networks. For implementing the security, until now researchers have used only those SDN controllers which are not being implemented in the real world such as NOX, POX, RYU, and Floodlight, etc. These controllers are only used for experimentation. Based on the literature studied following major research gaps have been identified:

- Currently, in order to mitigate DDoS attack various SDN based collaborative schemes and solutions are only making use of such SDN based controllers which are not being used in industries [6, 9, 24, 25].
- Early detection of each low-rate and high-rate DDoS attack remains to be self-addressed [14].
- No Graphical User Interface (GUI) feature and platform support for windows and MAC are supported by currently used SDN controllers. Most of these controllers (POX, NOX, RYU, and Floodlight) are based on Linux based platform only and possess a traditional DDoS mitigation method which has more network computational cost [15, 23].
- Development of such framework which can handle and overcome DDoS attack, make use of open-source API's and can support multiple vendors is still lacking [25, 26]. Summarization of various identified gaps is shown in Table 1.

4 SDN Controllers

Almost every network activity in SDN based network revolves around the centralized controller. It is located at the control layer and hence acts as the intermediate between the underlying infrastructure layer and application layer. Through the bare-metal switches, the controller sends the specific instructions on how to send the data and also on which path to select [27, 28]. Being the vital and important component of SDN based network, SDN controller needs to have reliability and security for a better SDN based environment. The use of multiple controllers must be used for critical application missions. In this case, if one controller is targeted by the attackers (leader controller), other follower controllers come into play to maintain the proper functionality of the entire network. Aside from path selection, other different policies like security, Quality of Service (QoS), network traffic engineering continued by SDN controllers [29, 30]. All the correspondence is possible with the assistance of Southbound and Northbound APIs. Controllers provide the intelligence, cost-efficient mechanism, automation to the network. The SDN architecture heart are controllers, Nicira Networks made the first SDN controller in 2009 and named it as Nox which was developed also with the first version of OpenFlow's [29]. Further, its revised version was developed along with Python support and was called POX controller [31–34]. After that ONIX platform was developed, a distributed platform for

Table 1. Gaps identification

Gaps identified	Description
Various SDN based collaborative schemes and solutions are only making use of such SDN based controllers which are not being used in industries for mitigating DDoS attack [6, 9, 24, 25]	Though firewall with SDN solves many traditional firewall drawback but still it lacks behind as it don't have open API's which can be combined with multiple applications and hence can be used by different enterprises having heterogeneous vendors
Every high-rate as well as low-rate DDoS attack which is detected early still needs to be self-addressed [14]	It had been observed that the use of appropriate data helps to magnify the spacing between attack traffic and legitimate for both high-rate and low-rate. This advantage can only take if it is detected in early-stage only
Recently used SDN controllers, does not support Graphical User Interface (GUI) feature for MAC and windows. Most of these controllers (POX, RYU, Floodlight and NOX) are based on Linux based platform only and possess a traditional DDoS mitigation method which have more network computational cost [15, 23]	POX, NOX, RYU, and Floodlight still now are used for experiments only. There is no evidence of using such controllers in industries. These controllers are based on LINUX, as well as they don't support GUI features of windows and MAC. So, experiments on those controllers are not required
Development of such framework which can handle and overcome DDoS attack, make use of open-source API's and can support multiple vendors is still lacking [25, 26]	The absence of non-commercial API's which can be used by any enterprise and then reconfigured accordingly by adding security rules in the network accordingly

the data center with vast scale networks and Google had developed it, a few years later, NTT and Nicira, become the foundation of VMware's SDN controller which is the most used and famous SDN controller in the commercial industry [27]. Some of the popular and most used SDN controllers are defined below:

- **OpenDayLight (ODL)**—It is the most and widely used SDN controller [35] which is an open-source controller project. It is controlled by the Linux Foundation. It is comprised of a huge number of vendors/ enterprises in its group. ODL has successfully made a big change in the commercialization of the SDN sector [36].
- **Open Networking Operating System (ONOS)**—It is the SDN controller platform by Linux Foundation which has the ability to transit from traditional "brown field" networks to new SDN based "green field" networks which help in faster deployment and lowering the cost of deployment [35].
- **Floodlight**—It is developed by open community of developers mostly from Big Switch Networks and used OpenFlow protocol. It was initially offered by Big Switch Networks as part of ODL project. Big Switch then stepped out of this project because of some conflicts with Cisco Systems and now Floodlight is not a part of ODL project [37, 38].

- **Ryu**—Ryu Controller is an open standard SDN controller that is specifically designed to strengthen network agility with easy manageability. This controller is used by NTT in their cloud-based data centers. Ryu bring well-defined APIs along with various software components. Ryu source code is on Github and is managed and maintained by Ryu developer community. It is written in Python and the source code is available under Apache 2.0 license [32].
- **POX**—It is an SDN controller written entirely in Python [31]. It was created after the Nox and become much more popular than Nox. It supports the same graphical user interface as Nox and performs better than Nox in the real world.

5 Proposed Approach and Discussion

As mentioned before as well that SDN acts as a brain of the network, centralized SDN controller is the one who is managing the whole network and has a global view. In the proposed scenario as shown in Fig. 2 SDN architecture is implemented in a network and in this, a control layer constitutes the controller, for example,. ODL, controller is having communication with an application such as a firewall through an API (Application Programming Interface). As the controller is the single point of failure so, to overcome this, we can also add another SDN controller at the edge of the network, for example, ONOS. Whenever there is flooding of traffic from multiple botnets and DDoS attack is launched on a targeted server, at that time edge controller will be handling DDoS attack and another controller will be able to maintain the functionality and working of the network. Both controllers will work simultaneously. Open-source API can be created to work with any security-based application and this API can be reconfigured easily by any enterprise as per their need.

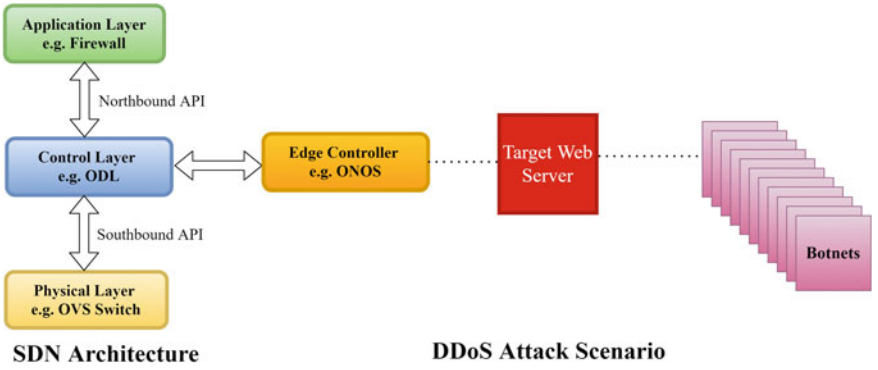

Fig. 2. Proposed framework to defend DDoS attack using SDN based architecture

6 Conclusion

IoT (Internet of Things) is a big buzz nowadays and number of devices connected to the internet is growing at an exponential rate and ultimately increases the number of

sources from which DDoS attacks can be launched. Many companies such as CISCO, Juniper, etc. are already making many applications such as Checkpoint, Palo Alto, etc. in order to detect and prevent DDoS attacks but these applications are commercial and one has to pay to get the benefits. Another disadvantage of these applications is non-configurability; one cannot modify it as per their need. In this chapter, need for SDN for securing the network is described. Approaches that are used by traditional networks to secure the networks are discussed with their limitations. A method has also been proposed to detect and prevent the DDoS attack by using SDS with various available SDN controllers' illustration.

References

1. McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Turner J et al (2008) OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Comput Commun Rev 38(2):69–74
2. Sharma K, Khandelwal N, Prabhakar M (2011) An overview of security problems in MANET. In: ISEM international conference
3. O'Brien WJ, Formoso CT, Ruben V, London K (2008) Construction supply chain management handbook. CRC Press
4. Xia W, Wen Y, Foh CH, Niyato D, Xie H (2014) A survey on software-defined networking. IEEE Commun Surv Tutor 17(1):27–51
5. Nunes BAA, Mendonca M, Nguyen XN, Obraczka K, Turletti T (2014) A survey of software-defined networking: past, present, and future of programmable networks. IEEE Commun Surv Tutor 16(3):1617–1634
6. Braga R, de Souza Mota E, Passito A (2010) Lightweight DDoS flooding attack detection using NOX/OpenFlow. In: LCN, vol 10, pp 408–415
7. Jun JH, Lee D, Ahn CW, Kim SH (2014) DDoS attack detection using flow entropy and packet sampling on huge networks of ICN, pp 185–190
8. Jyothi V, Wang X, Addepalli SK, Karri R (2016) Brain: behavior based adaptive intrusion detection in networks: using hardware performance counters to detect ddos attacks. In: 2016 29th international conference on VLSI design and 2016 15th international conference on embedded systems (VLSID). IEEE, pp. 587–588
9. Li J, Berg S, Zhang M, Reiher P, Wei T (2014) Drawbridge: software-defined ddos-resistant traffic engineering. ACM SIGCOMM Comput Commun Rev 44(4):591–592 (ACM)
10. Wang X, Chen M, Xing C (2015) SDSNM: a software-defined security networking mechanism to defend against DDoS attacks. In: 2015 ninth international conference on frontier of computer science and technology. IEEE, pp 115–121
11. Yu J, Fang C, Lu L, Li Z (2010) Mitigating application layer distributed denial of service attacks via effective trust management. IET Commun 4(16):1952–1962
12. Singh KJ, De T (2015) DDOS attack detection and mitigation technique based on Http count and verification using CAPTCHA. In: 2015 international conference on computational intelligence and networks. IEEE, pp 196–197
13. Al-Ali Z, Al-Duwairi B, Al-Hammouri AT (2015) Handling system overload resulting from DDoS attacks and flash crowd events. In: 2015 IEEE 2nd international conference on cyber security and cloud computing. IEEE, pp. 512–512
14. Bhuyan MH, Bhattacharyya DK, Kalita JK (2015) An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection. Pattern Recogn Lett 51:1–7

15. Qin X, Xu T, Wang C (2015) DDoS attack detection using flow entropy and clustering technique. In: 2015 11th international conference on computational intelligence and security (CIS). IEEE, pp 412–415
16. David J, Thomas C (2015) DDoS attack detection using fast entropy approach on flow-based network traffic. Proc Comput Sci 50:30–36
17. Yu S, Tian Y, Guo S, Wu DO (2013) Can we beat DDoS attacks in clouds? IEEE Trans Parallel Distrib Syst 25(9):2245–2254
18. Alqahtani S, Gamble RF (2015) DDoS attacks in service clouds. In: 2015 48th Hawaii international conference on system sciences. IEEE, pp 5331–5340
19. Shin S, Yegneswaran V, Porras P, Gu G (2013) Avant-guard: scalable and vigilant switch flow management in software-defined networks. In: Proceedings of the 2013 ACM SIGSAC conference on computer & communications security. ACM, pp 413–424
20. Shoeb A, Chithralekha T (2016) Resource management of switches and controller during saturation time to avoid DDoS in SDN. In: 2016 IEEE international conference on engineering and technology (ICETECH). IEEE, pp. 152–157
21. Wang B, Zheng Y, Lou W, Hou YT (2015) DDoS attack protection in the era of cloud computing and software-defined networking. Comput Netw 81:308–319
22. Zheng J, Li Q, Gu G, Cao J, Yau DK, Wu J (2018) Realtime DDoS defense using COTS SDN switches via adaptive correlation analysis. IEEE Trans Inf Forensics Security 13(7):1838–1853
23. Tseng Y, Zhang Z, Naït-Abdesselam F (2016) Controllersepa: a security-enhancing SDN controller plug-in for openflow applications. In: 2016 17th international conference on parallel and distributed computing, applications and technologies (PDCAT). IEEE, pp 268–273
24. Badotra S, Singh J (2019) Creating firewall in transport layer and application layer using software defined networking. In: Innovations in computer science and engineering. Springer, pp 95–103
25. Hameed S, Ahmed Khan H (2018) SDN based collaborative scheme for mitigation of DDoS attacks. Future Internet 10(3):23
26. Pal C, Veena S, Rustagi RP, Murthy KNB (2014) Implementation of simplified custom topology framework in Mininet. In: 2014 Asia-Pacific conference on computer aided system engineering (APCASE). IEEE, pp 48–53
27. Shalimov A, Zuikov D, Zimarina D, Pashkov V, Smeliansky R (2013) Advanced study of SDN/OpenFlow controllers. In: Proceedings of the 9th central & eastern European software engineering conference in Russia. ACM, p 1
28. Chen M, Qian Y, Mao S, Tang W, Yang X (2016) Software-defined mobile networks security. Mob Netw Appl 21(5):729–743
29. Badotra S, Panda SN (2020) SNORT based early DDoS detection system using Open daylight and open networking operating system in software-defined networking. In: Cluster Computing
30. Oktian YE, Lee S, Lee H, Lam J (2017) Distributed SDN controller system: a survey on design choice. Comput Netw 121:100–111
31. Kaur S, Singh J, Ghumman NS (2014) Network programmability using POX controller. In: ICCCS international conference on communication, computing & systems, vol 138, IEEE
32. Ryu SDN Controller https://osrg.github.io/ryu/. Accessed on 29 Apr 2019
33. Shalimov A, Zuikov D, Zimarina D, Pashkov V, Smeliansky R (2013) Advanced study of SDN/OpenFlow controllers. In: Proceedings of the 9th central & eastern European software engineering conference in Russia. ACM, p 1
34. Open Networking Foundation: ONF (2019). https://www.opennetworking.org. Accessed on 01 May 19
35. Badotra S, Panda SN Evaluation and comparison of OpenDayLight and open networking operating system in software-defined networking. Cluster Comput 1–11

36. Badotra S, Singh J (2017) Open daylight as a controller for software defined networking. Int J Adv Res Comput Sci 8(5)
37. Badotra S, Panda SN (2019) A review on software-defined networking enabled iot cloud computing. IIUM Eng J 20(2):105–126
38. Morales LV, Murillo AF, Rueda SJ (2015) Extending the floodlight controller. In: 2015 IEEE 14th international symposium on network computing and applications. IEEE, pp 126–133