Detection and Prevention of DDoS attacks on Software Defined Networks Controllers for Smart Grid

Zohaib Ahmed
SEECS-NUST
National University of Sciences and Technology
Islamabad, 44000, Pakistan

Naokhaiz Afaqui
SEECS-NUST
National University of Sciences and Technology
Islamabad, 44000, Pakistan

Osama Humayun
SEECS-NUST
National University of Sciences and Technology
Islamabad, 44000, Pakistan

ABSTRACT
With the evolution of smart grid, the operations, planning and maintenance of an electric grid have improved. On the contrary, smart grid totally relies on the computer network so there is a need of complex and efficient network management. Software defined networks (SDN) is a completely new modern architecture that allows the network to be centrally controlled or explicitly programmed using software applications. Traditionally in computer networks, the routing and switching decisions are implemented on a dedicated hardware. This hardware can be a switch or a router. But with the evolution of Software defined networks, the routing and switching function has been separated and is classified in Control and data planes respectively. Generally, in SDN, the control plane is centralized and is responsible to make a decision on what to do with the incoming packet. Once the decision is made, it is saved in the forwarding table of a switch on the data plane. While Software Defined Network (SDN) has its advantages of central management, programmability, agility and vendor neutrality, they carry a high risk of Distributed Denial of Service attack (DDoS). Centralized nature of the control plane in SDN is a huge risk factor because the attacker may bombard the control plane with malicious packets resulting in a single point of failure of the control plane. If the control plane fails, the entire smart grid network will collapse resulting in a massive outage and financial loss to the stakeholders. In this paper, we have devised a distributed approach, using blockchains, to detect and prevent DDoS attacks on the centralized control plane of SDN. We have simulated our approach using AnyLogic simulator and the results show that the proposed approach is more efficient as compared the existing techniques as it substantially reduces the risk of DDoS attacks and SDN controller overhead.

Keywords
SDN, Smart Grid, DDoS

1. INTRODUCTION
Smart Grid is a modern form of an electrical grid that is a combination of different smart devices and sensors such as smart meters, smart appliances, renewable energy efficient resources to improve the operations, planning and maintenance of an electrical grid [1][2]. In Smart Grid, all the components such as Transformers, Electric Power supply lines, Electric home meters and other devices have public IP addresses and these devices are capable for two-way communications. This two-way communication allows the electric power supplier to improve planning and operations by taking decisions efficiently. Traditionally, in a normal grid when there is some fault, customer informs the power supplier by some offline mode which often causes delays. However, in case of smart grid, the electric power supplier automatically knows when there is some fault because the smart meter will not send the reading when it's faulty. This will help the supplier in identifying the fault quickly and to fix it at their earliest [3]. In Smart grids, all the components have public IP addresses and all the communication is packet switched. Since there are millions of electric power consumers, the resulting network design will be very complex and hence there is a need to manage this network efficiently [4].

Software defined networks (SDN) is a completely new modern architecture that allows the network to be centrally controlled or explicitly programmed using software applications. OpenFlow, a Software Defined networking standard, is now largely in use for past many years. SDN is shown diagrammatically by using a physically separated and distributed framework. Still it is connected via a centralized framework for networking. Traditionally in computer networks, the routing and switching decisions are implemented on a dedicated hardware. This hardware can be a switch or a router. But with the evolution of Software defined networks, the routing and switching decision has been separated and is classified in Control and Data planes respectively [5]. Currently running implementations of OpenFlow work on a protocol named "southbound". It actually is comprised of two modes that has rules installed in it namely proactive and reactive. For the first mode, the controller is responsible to make flow rules that are new to the network and then install these flow rules at all of the switches of a network. Whereas for the second state, the controller that is centralized, uses and installs these rules every time a switch individually requests it to do so. The second mode does not need to have long tables for switches rather they have setting to adapt to the changes of a network very quickly. Generally, in SDN, the control plane is centralized and is responsible to make a decision on what to do with the incoming packet. Once the decision is made, this decision is saved in the forwarding table of the switches on the data plane. SDN is also the most efficient approach to create new and large networks once the network design is finalized [6]. If the smart grid is integrated with a Software defined network then the network management will become much more easy, robust and efficient.

Distributed Denial of Service (DDoS) attack is a kind of cyber-attack, in which the attacker tries to consume all the resources of the server so that the server is not able to respond to legitimate requests [7]. Now a days, the DDoS attacks usually target well defined services, causing only the target application to be disabled and halted whereas the other links and switches remain unaffected. These kinds of target-based attacks are easily invisible from other types of traffic. Though it is a well-known fact that SDN brings a lot of flexibility and scalability by separating the control and data plane, the centralized nature of the control plane in SDN can be a huge risk factor. The attacker may bombard the control plane with malicious packets resulting in a single point of failure of the

control plane due to consumption of all of the control plane resources. If an attacker has the capability to create a very large amount of new flows in a limited span of time and also has the intention to overwhelm the controller from its side, it can eventually cause collapse of the centralized network. If the network is down, it leads to the failure of the smart grid [8][9]. The attack process is showed in figure 1.

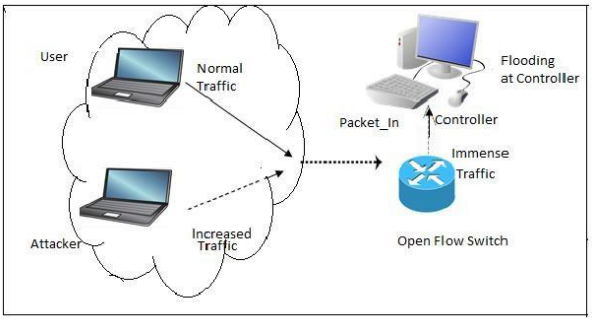

Figure 1: Attack Process

Different techniques have been proposed to mitigate DDoS attack but most of them are centralized. The disadvantages in the centralized approaches are; i) need of dedicated infrastructure ii) huge cost of third-party DDoS protection service providers iii) lack of trust on third party entities iv) single point of failure v) less scalability etc. To address these issues there is a need to employ distributed techniques. Evolution of Blockchain technologies enables us to resolve these issues since it provides decentralized infrastructure with low cost, more trust and better reliability. In this paper, we have proposed a distributed scheme to mitigate the (DDoS) attack on the SDN controller using blockchains. Followings are the contributions of this work:

- We propose a distributed DDoS detection and prevention approach for SDN controllers using blockchains. The SDN controllers are connected to the Ethereum blockchain.
- This approach monitors source IPs, samples and threshold on incoming requests from SDN switches. The threshold value is calculated on the basis of the incoming requests. Based on threshold value, the controller makes efficient decision on classification of sources as legitimate or malicious (attackers). Once the controller has classified the host, this can be shared with other SDN controllers using blockchains.
- We have also performed comprehensive simulation to analyze the efficiency of proposed approach. It has proved to be highly efficient in reducing controller overhead and detecting both slow and fast DDoS attacks on SDN controllers.

The remainder of this paper is organized as follows: The related work has been discussed in Section-II, then we discuss the proposed model for DDoS mitigation in Section-III. Further, in Section-IV we have shared details of our experiment, along with the simulations results and analysis. In Section-V, we have given conclusion of our research and possible future work.

2. RELATED WORK

SDN has potential risk of DDoS attack and there is a dire need for efficient DDoS preventing techniques. These techniques must have following properties:

- Record public IP addresses on which service can be redirected.
- Network administrator must be notified to enhance security in case of an attack.
- In case of reactive approach, the network administrator can deallocate resources after DDoS attack.

During recent years, a lot of researchers have made significant efforts to find some efficient mechanism; some of which are mention in figure 2.

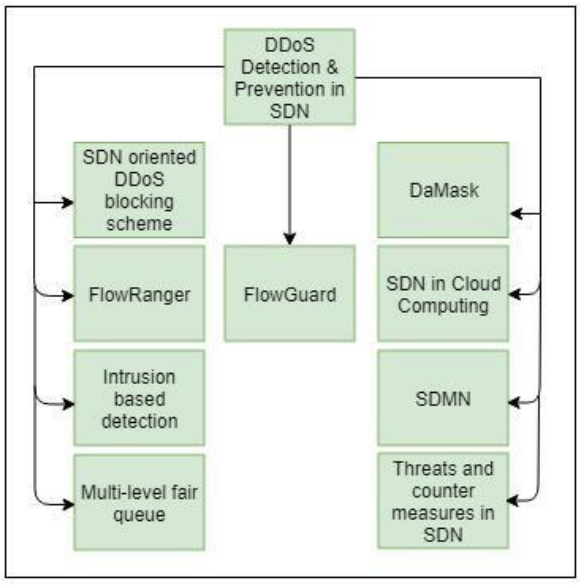

Figure 2: Taxonomy of DDoS Mitigation in SDN

An adaptive preventing technique [10] based on SDN oriented blocking scheme is presented by Lim et.al. in which client requests, that are unknown to any currently running flow, are received at the OpenFlow switch. The controller observes, reports and a flow table entry are created for this new packet. These new reports of flow that are unusual come under observation and then the flows at each switch are examined by network administrator. A threshold value is set and if the value of Threshold exceeds the limit, the network administrator controller notices it. In order to make observations, two groups are made namely legitimate clients and another category of malicious. A legitimate user requests a flow every three seconds whereas Bots issued a request after every one second. Hence, Bots proved to be more active than legitimate users. By identifying the Bot requests and blocking them, all of the connections established at the attacked address were reduced to (zero). Mininet Emulator was used for implementation and simulation. Another approach for secure communication was presented by Wei et.al. [11], in which priority algorithm was used. Priority number of the user request is set by means of a trust label; lower priority request will drop in case of buffer limited capacity. Threshold value is set to observe the number of trusted users and the list of true value users depend on the actual number of users. If the trust value of a certain user is not greater than the threshold value, the user is declared to be unsafe (attacker) and the controller

drops the packets of that user. It is in actual a trust-based technique that helps to enhance the serving rate and it serves about 43% more than the conventional FCFS (First come First Serve) policy. In [12], a comprehensive discussion is presented on intrusion base detection, which illustrates detailed behavior of a DDoS attack and its effect on SDN. The re-active mode of SDN has advantages in large networks because the switch does not have to maintain large flow tables. On the contrary the nature of reactive rule installation can cause vulnerability to DDoS attack. In [13] author uses the concept of multi-level fair queue (MLFQ). In case of no attack the scheme followed by MLFQ is to establish a smaller number of queues at the end of controller, and to expand this queue dynamically into several sub queues every time the size of queue exceeds a certain value of threshold. On the contrary, H. Wang et.al. proposed FloodGuard [14], a framework based on proactive approach. In overload condition on control plane, attacker can utilize both data and control plane. FloodGuard is further divided into flow analyzer and packet migration modules and both modules are active one after the other. In the era of cloud computing, a new emergence like SDN can help in providing enterprise IT services. But the facilities provided by the union of the two are diminished due to the increasing risk of network security. The services are made unavailable to users by the version of attack named Distributed Denial-of-Service (DDoS) which drains the network resources. By Wang et.al. in [15] proposed solution for attack, named DaMask which is a highly scalable and much flexible DDoS attack mitigation architecture. DaMask architecture is composed of DaMask-D module. The scheme has been implemented and a simulation-based evaluation has been done by using the Amazon EC2 cloud service. This scheme has been designed to protect the services in the private and public clouds by adapting the change in topologies as per architecture with less overhead and cost. All operations being performed in respective slices are visible to the users. DaMask is capable of adapting to the change in topology and for a malicious entry, no forwarding of any ICMP packet to the network controller has been observed. Security requirements of cloud computing are very crucial, and it has become very necessary. Cloud computing must include confidentiality, integrity and availability. SDN does a software-based traffic analysis on its own to enhance the capabilities of switches to make their decisions. It's forwarding rules help in prompt responses. In [16] [17], few available solutions i.e. FortNox, AVANT-GUARD, VeriCon and Transport Layer Security give solution to the attacks that may occur on application, control or infrastructure layers. AVANT-GUARD provides a solution to the attack made on the infrastructure layer, executed by at-tacking the southbound API and the switch. Content-Oriented Net-working Architecture (CONA) is also one solution to mitigate this attack. There is a proxy node which is located between the user and the content server, so that they may communicate with the controller. SDN is quite flexible architecture which is programmable and is a flow-centric mobile network. In [18] [19] [20] Software Define Mobile Network (SDMN) is presented in which a holistic security approach is utilized where the centralized controller stays tuned continuously for network abnormalities. Destructive traffic generated by the unauthorized user is halted and fails to reach to the core network. Till 2012 most of the mobile operators effected by the DDOS attack experienced that the backhaul portion of the network becomes unreactive for actual traffic. Attackers use a botnet setup to launch an attack on a mobile network. In [21] researchers propose defense framework ArOMA that monitors and mitigates security threats without human intervention. ArOMA also has a collaboration feature and distributes security assistance to ISP, by which they prevent their customers from DDoS attack. Slow action/attack is ignored in this fast-moving world and creates more vulnerability in all aspects. The more difficult part is to identify legitimate and illegitimate users [22] [23]. For this purpose, T. Lukaseder et.al. in [22] proposes a framework to mitigate slow DDoS attack by continuously measuring packet rate and its distance. Researchers elaborate all attack possibilities and their possible solutions in a sophisticated manner. While K.Hong et.al. in [23] proposed SHDA mechanism in which SDN structure becomes simpler by classifying user classes. Regarding critical need of time Qiao et.al propose multi-level DDoS mitigation framework (MLDMF) framework for Industrial Internet of Things (IIoT) as in [16,20] design frameworks to defend cloud computing environments against DDoS attack. In [24] MLDMF framework includes edge, fog, and cloud computing to safeguard IIoT against DDoS.

3. PROPOSED METHODOLOGY

In our proposed approach all the communication between SDN controllers are through Ethereum blockchain. Every controller maintains a lists of IP addresses known as "Blacklist" and "Possible Victim". The SDN controller monitors the rate of incoming traffic from every host. This rate of incoming traffic will be used in computing the threshold for that host. Blacklist IP address list contains the IP addresses of hosts which exceeds the threshold. However, the "Possible Victim" list contains the IP addresses of the server because servers are the victim of DDoS attack. There are two major components of our approach i.e. "Counter" and "Comparator". The counter is responsible to count the packets by using the IP addresses of the senders. The timer will start periodically after every 15 secs. It counts the number of packets from all sender IPs. We use the values of the counter to determine "Average Threshold" value by doing the sampling of incoming packets. The Comparator component is responsible for comparing the packet count per IP address. So, if this packet count is greater than the previously calculated maximum threshold value (Packet count > Max Threshold) an alert message is generated and the controller will then install the rule to drop all the incoming packets for that destination for some time perceiving it to be "malicious". The IP addresses of the attacker and possible victim is shared with the other SDN controllers through Ethereum blockchain. If the value of packet count does not exceed the maximum threshold value (Packet count < Max Threshold) then it means no attack has occurred and it will keep on comparing the packet. During attack, when the other SDN controller receives the list of IP addresses, it checks whether any host in the network is sending packets to the host located in the "possible victim" list. If any host is sending the packets to the server located in the possible victim list then the SDN controller installs the rule to drop the packets of that host. The same process will continue for all the SDN controllers. The flow chart and the pseudocode of the proposed approach is shown in figure 3 and Algorithm 1 respectively. Since our approach is a collaborative approach hence it allows the different SDN controllers to mitigate the DDoS attack collaboratively. In the next section the effectiveness of the proposed approach is evaluated.

4. RESULTS AND DISCUSSION

Ethereum is one of the most popular blockchain platform and extensive research has been done by the researchers on it. As a result, the performance evaluation of the Ethereum blockchain has been intentionally ignored and only those components which are not the part of the blockchain have been targeted for the evaluation. In addition to this, it is assumed that there is no delay in propagating the IP addresses between different autonomous system over Ethereum blockchain.

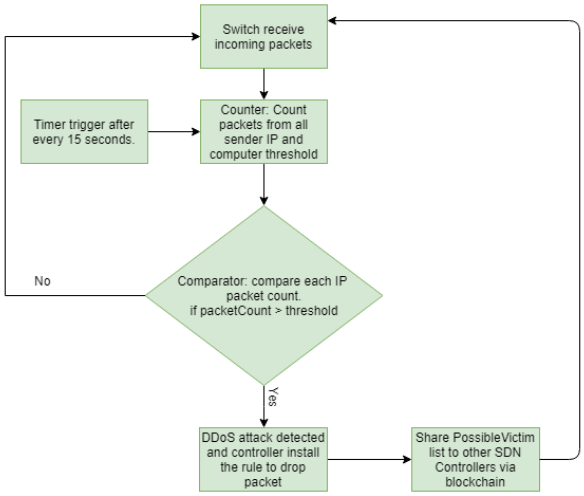

Figure 3: Flowchart of proposed methodology

Algorithm 1 Proposed Algorithm

```
1:  blacklistIPAddresses ← null
2:  possibleVictimIPAddresses ← null
3:  hosts ← null
4:  thresholdPerHost ← null
5:  procedure ADDHOSTS(IPAddress)
6:      hosts ← hosts + IPAddress
7:  end procedure
    ▷ Method calls after every 15 seconds
8:  procedure COUNTER
9:      i ← 1
10:     for host in hosts do
11:         thresholdPerHost[i] ← avg(host.PacketsSent)
12:         i ← i + 1
13:     end for
14: end procedure
15: procedure COMPARATOR
16:     while i > hosts.length() do
17:         packetSent ← hosts[i].PacketsSent
18:         threshold ← thresholdPerHost[i]
19:         IPAddress ← hosts[i].IPAddress
20:         if packetSent > threshold then
21:             blacklistIPAddresses.add(IPAddress)
    ▷ Drop the Packets from blacklisted host and share the blacklist
    IP Addresses with other AS
22:         end if
23:         i ← i + 1
24:     end while
25: end procedure
```

AnyLogic Simulator has been used to observe the behavior of the proposed approach and to determine the results. In AnyLogic, "Agent Based Modeling" is used and the behavior of switches and controllers has been modeled by using "State Charts" as shown in figure 4 and 5 respectively. The behavior of controller and switch has been separately observed. Figure 6 shows the mechanism followed here. Every time when a packet arrives at a switch, it checks that whether the packet is in the forwarding table or not. If the packet is already there in the table, then the switch will take action that has already been saved by the controller. Else, it will send the packet as it is to the controller. Then the controller will process the packet further. This will help the comparator to clearly identify any possibility of attack directly, without processing it any further. Any of illegitimate requests are eventually blocked.

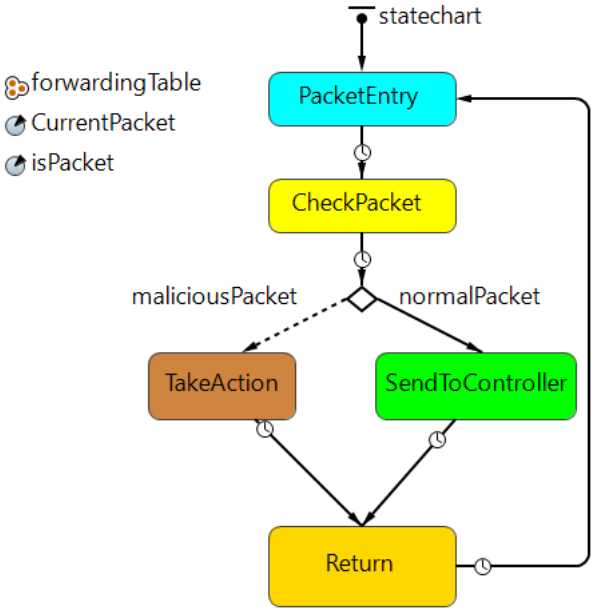

Figure 4: State chart of the switch

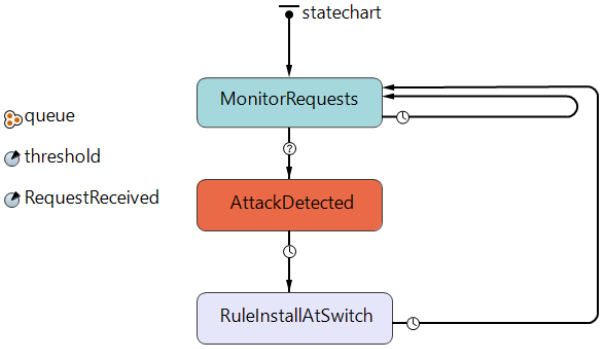

Figure 5: State chart of the controller

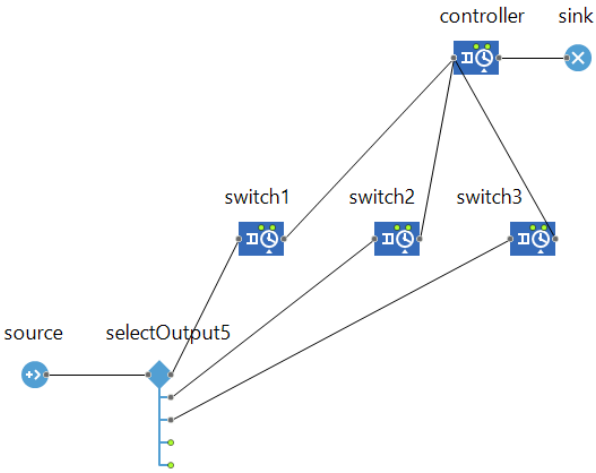

Figure 6: Proposed Scheme Design Topology

We have compared our results with the traditional approach and the results show that we can detect the malicious attacks resulting with a better efficiency i.e. up to 35%. We have reduced the overhead of the controller up to 60%. The figure 7 shows the traffic flow patterns that are coming towards switch 1, for switch 2 and for switch 3. The pattern clearly indicates the traffic for each switch accordingly. The switch 3 receives the maximum number of incoming requests, followed by lower requests for switch 2 and least for switch 1. The traffic pattern for switch 1 is almost 23% of the total traffic, whereas for switch 2 it is 32%, and the largest share of traffic is for switch 3 which is 45% of the total traffic.

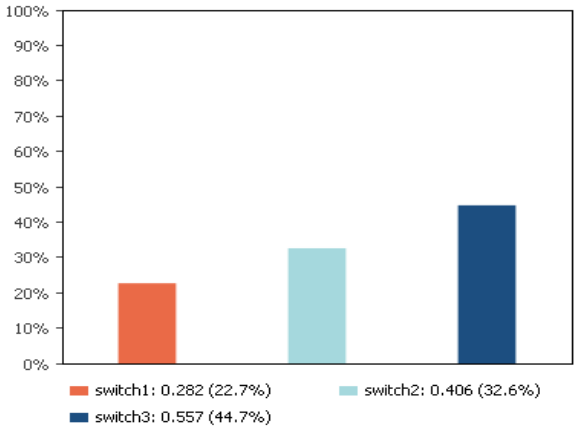

Figure 7: Bar Chart showing the Flow of Traffic

We evaluate the proposed scheme with the traditional approach by determining the overhead on the end of the controller for both of the cases. The results for normal behavior of SDN controller in case of DDoS attack are shown in figure 8. In the traditional case, if the value of overhead goes to 95%, the controller will not be able to serve legitimate requests. However, the evaluation results clearly demonstrate the effectiveness of the proposed approach and show that our system only adds minor overhead. The results of the proposed approach show that the overhead of controller has reduced up to 35% as shown in figure 9.

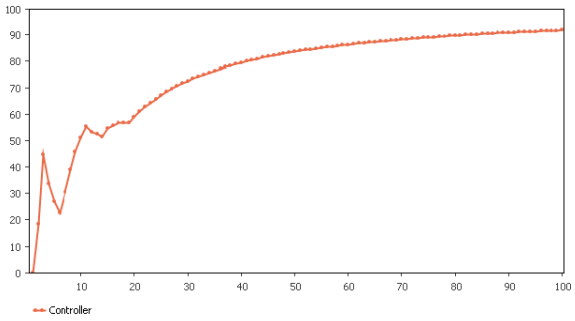

Figure 8: Controller Overhead for Traditional Approach

5. CONCLUSION

We propose an approach to efficiently detect and prevent DDoS attack on an SDN controller for smart grids using "Controller End" monitoring, sampling and thresholding. This approach continuously monitors the behavior of the incoming requests from the SDN switches and makes efficient decision on source IP addresses to classify the requests as legitimate or illegitimate. The simulation and analysis have proved that this scheme is quite efficient in reducing SDN controller overhead and also in detecting slow and fast DDoS attacks. In future work, we are working to introduce an adaptive thresholding technique for further improvement.

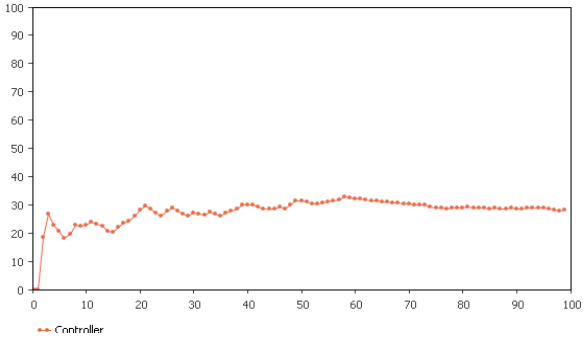

Figure 9: Controller Overhead for Proposed Approach

6. REFERENCES

[1] "Smart grid", En.wikipedia.org, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Smart grid.

[2] D. Brett, "Smart Grid — Student Energy", studentenergy.org, 2018. [Online]. Available: https://www.studentenergy.org/topics/smart-grid.

[3] M. Rouse, "What is smart grid? - Definition from WhatIs.com", WhatIs.com, 2018. [Online]. Available: https://whatis.techtarget.com/definition/smart-grid.

[4] Jianchao Zhang, Boon-Chong Seet, TekTjing Lie and Chuan Heng Foh, "Opportunities for Software-Defined Networking in Smart Grid", 2013 9th International Conference on Information, Communications & Signal Processing, Tainan, 2013, pp. 1-5.

[5] P. Rengaraju, V. R. Ramanan and C. Lung, "Detection and prevention of DoS attacks in Software-Defined Cloud networks", 2017 IEEE Conference on Dependable and Secure Computing, Taipei, 2017, pp. 217-223.

[6] P. Zhang, H. Wang, C. Hu and C. Lin, "On Denial of Service Attacks in Software Defined Networks", in IEEE Network, vol. 30, no. 6, pp. 28-33, November-December 2016.

[7] "Denial-of-service attack", En.wikipedia.org, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Denial-of-service attack.

[8] Y. Yan, Y. Qian, H. Sharif and D. Tipper, "A Survey on Cyber Security for Smart Grid Communications", in IEEE Communications Surveys & Tutorials, vol. 14, no. 4, pp. 998-1010, Fourth Quarter 2012.

[9] H. Wang, L. Xu and G. Gu, "FloodGuard: A DoS Attack Prevention Extension in Software-Defined Networks", 2015 45th Annual IEEE/IFIP International Conference on Depend-able Systems and Networks, Rio de Janeiro, 2015, pp. 239-250.

[10] S. Lim, J. Ha, H. Kim, Y. Kim and S. Yang, "A SDN-oriented DDoS blocking scheme for botnet-based attacks", 2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN), Shanghai, 2014, pp. 63-68.

[11] L. Wei and C. Fung, "FlowRanger: A request prioritizing algorithm for controller DoS attacks in Software Defined Net-works", 2015 IEEE International Conference on Communications (ICC), London, 2015, pp. 5254-5259.

[12] A. AlEroud and I. Alsmadi, "Identifying cyber-attacks on software defined networks: An inference-based intrusion detection approach", Journal of Network and Computer Applications, vol. 80, pp. 152-164, Feb 15 2017.

[13] P. Zhang, H. Wang, C. Hu and C. Lin, "On Denial of Service Attacks in Software Defined Networks", in IEEE Network, vol. 30, no. 6, pp. 28-33, November-December 2016.

[14] H. Wang, L. Xu and G. Gu, "FloodGuard: A DoS Attack Prevention Extension in Software-Defined Networks", 2015 45th Annual IEEE/IFIP International Conference on Depend-able Systems and Networks, Rio de Janeiro, 2015, pp. 239-250.

[15] B. Wang, Y. Zheng, W. Lou and Y. T. Hou, "DDoS Attack Protection in the Era of Cloud Computing and Software-Defined Networking", 2014 IEEE 22nd International Conference on Net-work Protocols, Raleigh, NC, 2014, pp. 624-629.

[16] Q. Yan and F. R. Yu, "Distributed denial of service attacks in software-defined networking with cloud computing", in IEEE Communications Magazine, vol. 53, no. 4, pp. 52-59, April 2015.

[17] Z. Shu, J. Wan, D. Li, J. Lin, A. Vasilakos and M. Imran, "Security in Software-Defined Networking: Threats and Coun-termeasures", Mobile Networks and Applications, vol. 21, no. 5, pp. 764-776, 2016.

[18] M. Chen, Y. Qian, S. Mao, W. Tang and X. Yang, "Software-Defined Mobile Networks Security", Mobile Networks and Ap-plications, vol. 21, no. 5, pp. 729-743, 2016.

[19] M. Liyanage, A. B. Abro, M. Ylianttila and A. Gurtov, "Opportunities and Challenges of Software-Defined Mobile Net-works in Network Security", in IEEE Security & Privacy, vol. 14, no. 4, pp. 34-44, July-Aug. 2016.

[20] Q. Yan, F. R. Yu, Q. Gong and J. Li, "Software-Defined Net-working (SDN) and Distributed Denial of Service (DDoS) At-tacks in Cloud Computing Environments: A Survey, Some Re-search Issues, and Challenges", in IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 602-622, First quarter 2016.

[21] R. Sahay, G. Blanc, Z. Zhang and H. Debar, "ArOMA : An SDN based autonomic DDoS mitigation framework", Computers & Security, vol. 70, pp. 482-499, 2017.

[22] Lukaseder, Thomas, Lisa Maile, Benjamin Erb, and Frank Kargl. "SDN-Assisted Network-Based Mitigation of Slow DDoS Attacks", arXiv preprint arXiv:1804.06750 (2018).

[23] K. Hong, Y. Kim, H. Choi and J. Park, "SDN-Assisted Slow HTTP DDoS Attack Defense Method", in IEEE Communications Letters, vol. 22, no. 4, pp. 688-691, April 2018.

[24] Q. Yan, W. Huang, X. Luo, Q. Gong and F. R. Yu, "A Multi-Level DDoS Mitigation Framework for the Industrial Internet of Things", in IEEE Communications Magazine, vol. 56, no. 2, pp. 30-36, Feb. 2018.