Detection and Mitigation of Malicious DDoS Floods in Software Defined Networks

Furqan Ahmad (✉ furqanahmad272@gmail.com)
National Textile University

Research Article

Keywords: Software-Defined Networking, DDoS Attacks, Open Flow Protocol, Centralized Control, Network Security

Posted Date: January 4th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2421818/v1

Abstract

Software-defined networking provides modular network management, allowing the flexible quality of services to remove conventional networks' limitations. It implies the concept of separating the control and data plane attributes for flexible network management. Contrary to network flexibilities, the centralized management is exposed to cyber threats i.e., Distributed Denial-of-service (DDoS) attacks which can compromise of SDN controllers. Meanwhile, entropy-based DDoS attack detection methods are most prominent among other detection methods but relying on entropy itself can neglect detection in several parameters i.e., variations in flow specification. In this research, a DDOS attacks detection and mitigation framework inside the SDN control plane is designed to ensure the secure availability of the network. Our approach comprises an entropy-based detection system integrated flow initiation and specification modules to classify the malicious DDoS flows against regular traffic. This lightweight approach is designed to minimize DDoS attacks by detecting its effects in the early stages a perform mitigation before compromising the controller resources. The simulation is performed on Mininet network simulator, for implementing SDN architecture and the testbed is created on various DDOS attacks, i.e., UDP, TCP-SYN, and ICMP ping flood attacks, to validate on commonly used data centric network environments. Based on the results, the proposed solution assures the SDN-based DDoS attack detection and mitigation under 150 packets maintaining significantly low detection time and high accuracy.

1 Introduction

Software-defined networking (SDN) was introduced at Stanford University and the University of California at Berkeley as a new approach to existing network infrastructure [1] As a family of technologies, software-defined networking (SDN) marked the beginning of revolutionary innovation and change in the telecommunications sector, like that seen in those early internet days [2]. The segregation of network planes provided in SDN allows extraordinary features like flexibility, quality of service (QOS) management, abstraction, and security. It enables dynamic network designs to be created, which improves network performance. These features make it more like cloud computing than conventional network administration methods.[3]

Network planes in conventional networking are tightly integrated, making it challenging to develop and deploy novel network applications [7]. SDN overcomes these inflexibilities in conventional networks by separating network intelligence from underlaying hardware components. A primary goal of SDN is to streamline the network by separating the forwarding and routing of network packets into separate processes [4]. In SDN, the communication between the network planes is performed by the north and southbound interface [5]. The northbound interface establishes the communication among the application and control layers, while the southbound interface communicates control and application layer communications [5]. A typical northbound interface example is the REST API, while the other is OpenFlow for the Southbound Interface [1], [6]. OpenFlow protocol is used in hardware and software environments to implement the SDN [7]. The controller is an essential component in an SDN network, which serves as intelligence for whole network communication [8]. SDN networks with software switches

[9] are implemented using a variety of modeling and emulation platforms to virtualize the hardware needs. Generalized architecture of SDN architecture comprising 2 layers is presented in Fig. 1. Moreover, SDN provides facilities like defining and changing traffic flow rules to lower security issues. The centralization in its infrastructure provides flexible network management with end-to-end visibility by providing connectivity rules with a central controller [10]. Based on the above arguments, SDN technologies provide the following benefits to network evolution.

- Programmability: Support to program network according to variable needs.
- Abstraction: Modification in network-wide traffic rules at control plane is possible through separating control and data plane.
- Centralization: Networking decision management uses a centralized control plane known as the SDN controller.
- Dynamicity: SDN provides speedy network design, administration, protection, and optimization for network resources.
- Open standards: remove vendor specific needs from underlaying network components i.e., routers and switches.

1.1. Expansion in SDN Technologies

The rapid expansion of the telecommunications sector, virtualized servers, and cloud computing constrain conventional network topologies [7] Traditional networks organized in congested environments i.e., Data centers, campuses, and mobile networks now have far more dynamic processing and storage needs which require need technological support.[12]. Researchers have increased SDN's energy efficiency by dynamically adjusting the network data plane and leveraging current routing algorithms [13].

The most important factors that necessitate the expansion of networking technologies in the SDN paradigm are the following:

a. Changing Traffic Patterns: Consequently, data centers are investigating a new computing paradigm that might include private cloud computing or a mix of the two.
b. Load on IT Infrastructures: Users' devices, such as smartphones, put IT under increasing pressure to preserve company data and intellectual property while satisfying compliance demands.
c. Cloud Services: Rapid expansion of public and private cloud services results in enterprises' enthusiastic acceptance of these services.
d. Big Data: Parallel processing on hundreds of computers is required for today's "big data" support Energy Consumption in Datacenters

1.2. DDOS Security Threats to SDN

Decoupling control and data plane in SDN also brings various challenges like security, reliability, and load balancing. Denial of services (DOS) is one such common threat which generally compromise the network resources and lower its performance to end nodes. Various techniques like TCP syn, UDP flood and ICMP flood are employed to perform such attacks[14]. In the same way, SDN-based DOS attacks bring flooding to control and data plane resources, downgrading the entire network [15]. To overwhelm the SDN resources, the attacker will continue broadcasting fake packets using the controller's processing resources. The produced flow entries will eventually fill the flow table in the switches, degrading the switches due to the heavy load. On the other hand, the controller will not be able to handle the legitimate packets, rendering the controller's duty ineffectual. Due to this, the controller may get compromised and unavailable for an extended period. SDN-based DDOS attacks are categorized by [16] in Fig. 2 below.

While addressing this problem, a lightweight, scalable solution that handles DOS-related problems at the controller level in the early stages is required for taking care of controller resources. In this research a DOS detection and mitigation approach in SDN controller architecture is designed to ensure the secure availability of network. Our approach comprises a detection system integrated inside an SDN controller that classifies the malicious DDoS traffic against regular network traffic by calculating the variation in their entropy. Based on the detection, it validates the attack using flow initiation and specification rates to reduce false/positive behavior. This lightweight approach minimizes the problem of possible DDoS attacks by detecting its effects in the early phases to mitigate it before the attack can compromise the network.

The remaining work is divided up into the following subsections. The work related to DDOS detection in SDN is presented in Section 2, Section 3 discusses the proposed methodology, Section 4 offers the results and key findings that were reached throughout a project, and finally, in Section 5, a concise summarization of findings is given to draw the conclusion.

2 Literature Review

DDoS attacks are considered the most common threats nowadays, posing a critical danger to network services. Internet is the most common victim of these attacks because they are so simple to carry out and compromise the availability of services in seconds. As analyzed by recent reports, DDoS attacks have grown rapidly over the last several years, which has caused significant financial losses to the business. Moreover, the difficulty in locating the assailant provided a considerable boost to the attack's efficacy. The Mirai botnet attack in 2016 affected network availability over the globe, which brought down a large portion of the Internet [17]. In 2018, GitHub servers were targeted by the largest-ever DDoS attack [18]. In another case, the attack using application layer protocol produced 129 million requests per second and achieved a total traffic level of 1.35 Tbps sent in the attack, which followed the severe attack reported in 2016 [18] During the third quarter of this year, the US and the Netherlands had the largest percentage of DDoS attack-launching botnet devices[19]. As a result of this dilemma, the scientific community has invested a considerable deal of time and effort into developing novel ways to protect IoT systems against

DDoS attacks. Traditional networks are simplified to a large extent by using new network technologies, such as Software-Defined Networking (SDN), which has emerged in the past years.

2.1. Recent Work

SDN brings flexibility in the quality of services for network management which helps to manage its dynamic architecture through control plan configurations. These configurations are also proven beneficial for detecting and mitigating cyber-attacks through policy management. Although these policies support resilience against network-based malicious activities, i.e., DDoS attacks still, they may have limitations in handling a wide variety of attacks. To address this problem, [20] has given detection and prevention solutions for DDoS attacks using Mininet, a standard emulator for SDN. This approach measures flow statistics from switches and parameters like packet rate and bandwidth. If a sudden increase in packet rate is detected, it is considered an ongoing attack mitigated by producing the forwarding rule for that node, which drops the packets.

The research presented by [21] has proposed an SDN-based solution using an ODL controller by installing flow rules in the control plane. This solution mitigates the adverse effects of congested traffic passing through the network by avoiding the controller overhead. The controller collects flow statistics and estimates the throughput of these flow rules to predict the possibility of a denial-of-service attack. Based on this estimation, it can block traffic that increases by its defined threshold limit. This work posed some limitations, like if the attack could bypass the threshold value, the network could be compromised easily.

The work presented in [22] developed an SDN integrated intrusion detection framework that reacts to attacks at their source, guaranteeing regular operation. The idea combined an IDS that autonomously identifies UDP-based DDoS attacks and then creates an alert to the SDN controller. In addition, the suggested solution employs various SDN controller-to-network device traffic forwarding rules. According to the assessment findings, this approach identifies numerous forms of DDoS-related cyber-attacks in real-time, removing the adverse effects on the network's performance. In [23] DDoS attack detection is done using traffic flow collection per unit time. It followed the concept of the sudden increase in flow states which is later represented as a denial-of-service attack. When a sudden increase is detected, the flow rule is specified, which does not allow malicious traffic to spread further in the network. This work has shown limitations, i.e., dependency on a packet/second for detection, resulting in a severe downgrade when the attack specifications are changed. On the other hand, this system could fail in some DDOS attacks, as observed in the case of a low packet forwarding rate.

A dynamic decision-making system is introduced by [24] at SDN switches to filtrate malicious packets involved in denial-of-service (DoS) attacks. When determining whether a packet is malicious or not, the system considers various packet characteristics. The packet is discarded if the system detects a packet's score exceeds the danger level. On the other hand, the packet can pass to its regular route if the packet score is below the danger level. Unless the packets in the dangerous category finished their handshake, they were buffered until the handshake was completed. In his work, he has failed to address the limitations of packet properties against a certain threshold level which can cause the system's failure even when there is no attack. Moreover, the system needs to be trained against more datasets for a scalable result. In [25] researchers have tested multiple machine learning algorithms for detecting and blocking DDoS attacks in an SDN network. The techniques for detection assessment include training, choosing the best relevant model for the network, and implementing the model into threat detection and mitigation strategy. J48 beat the other ML techniques in case of performance, notably training and testing time, as determined by the assessment findings. However, training and testing constraints on these algorithms reveal detection accuracy limitations, which may result in false alarms or outright failure to detect.

In [26], a protocol-independent approach, 'Flood-Defender,' works using the SVM algorithm is introduced. This technique used three approaches, i.e., table miss mechanism, packet filtering, and flow table management. This approach has flaws, such as an increased false-positive rate with an increased attack rate, that need to be addressed. Attacks were occurring at a pace that caused it to discard harmless packets. Attackers can evade detection by sending packets at a lower rate. Work presented in [27] fills the gaps of slow DDOS attacks by using principal component analysis. This work tried to demonstrate the limitations of entropy calculation in mixed DDOS flow with legitimate traffic. The researcher has not demonstrated the validity of work in other DDOS parameters like high flow rates.

In [28] the research implemented entropy to detect DDOS attacks at the controller level. The research used a considerably low window size for the detection of attacks. Moreover, the window is static, which leads to an increment in the error rate due to its small size. The research only focused on high-rate attacks under some critical scenarios, which raises its validity in some critical test cases. In [22] SDN centralized architecture is addressed against DDOS attacks using a lightweight detection approach that could effectively lead under limited resources. The research implemented an entropy-based solution to detect the randomness under 250 packets and an overall detection rate of 96 %. Ths work has some fundamental limitations, which are addressed in our work. Firstly, this research only deals with detecting attacks, and no further measures are taken. Secondly, there is no identification of the path followed by the attack, which could be necessary for the mitigation process later. In [29] entropy-based solution is used for mitigating TCP-syn flood attacks in three steps, i.e., entropy, standard deviation, and weighted moving average. This work has shown the significance of entropy-based solutions due to their lightweight properties in SDN controllers. On the other hand, the research is validated explicitly on a few parameters of DDOS attacks DDOS attack and has not been validated on its varied parameters. To summarize the previous methodologies in DDOS detection, a clear picture of the pros and cons of the recent work has been discussed in Table 1.

Table 1
Pros and Cons of Recent Work.

Work	Pros	Cons
[20]	Appropriate in high-rate flood attacks	Performance downgrade in slow rate DDOS attacks.
[21]	Essential in QoS perameters	The attacker could bypass the threshold value and the network
[23]	Essential QoS in high-flood rates.	Low packet forwarding rate can compromise the detection
[24]	Attack mitigation at the Data plane	Detection is based on packet score
[25]	Detection and mitigation of novel attack features	Novel attack features can fluctuate False/Positive rate.
[26]	Protocol-independence	Packets can be sent at a very low rate to avoid detection
[27]	Efficient for slow rate DDoS attacks	Validity of work is not confirmed in other DDOS parameters
[28]	Lightweight and early detection	Validity in attack parameters is not conformed
[22]	Low overhead	Dependency on IDS

3 Methodology

This section presents a detailed state of the art methodology for DDoS attacks detection and mitigation in SDN centralized architecture.

3.1- Research Framework

SDN provides the facility of programmability inside its architecture which can be used to implement security policies. In this research, an attack detection algorithm based on entropy variation of destination IP addresses is written inside the SDN controller, which can detect malicious DDoS attacks based on randomness in network traffic. The algorithm is based on three key concepts a measure of variation in entropy of destination IP addresses, rate of flow initiation, and flow specification. The Shannon-Wiener index, also known as entropy, is a fundamental notion in information theory. The entropy of random variables, i.e., a destination address, measures uncertainty or unpredictability and is calculated to determine the possible change in network states. The entropy is calculated in the range of [0 log2n^{ip}], where the term "n^{ip}" represents the amount of destination Ip addresses. When all the traffic is headed in the same direction, the entropy value is at its lowest point. Similarly, the entropy is maximized when traffic is evenly divided across all the potential destinations. The collection of packets for entropy analysis is accomplished via a window of a specified length. Using a window size reduces the computational cost of the entropy calculation. In this case the accuracy of the entropy computations may be degraded during periods of low traffic demands, using a fixed-time window is not preferable. This problem is solved by using a window determined by the received packets. The packets' destination IP addresses will be used to split them into groups for each window. Because they are in distinct groups, packets within a group may have various source addresses, but they will all have the same destination address.

Here the destination IP addresses are used as a characteristic metric. Randomness is measured by the number of times each IP address appears in the window. The relative frequency of the destination IP address is measured using the following formula.

$F_i = n_i /n$

Where ni shows the total packets with destination IP address

The entropy is computed following the formula.

H=$-\sum_{i=1}^{m} Filog2Fi$

Where 0≤Fi ≤1 ⤇ H ≥ 0

Entropy is maximized when the m IP destination addresses' relative frequencies are equal. (Fi = 1/m for all i). Let us say we have N packets and want to know the likelihood that each packet will arrive at its intended destination. Fi= 1/N and H = −Σ1/N∗ $\log_2$ 1/N. Entropy will be lowered if a small ratio of packets, i.e., 10 out of 30 packets, are sent to a single destination address. This way, DDOS traffic can be identified in a few packets, essentially used to mitigate DDOS attacks in the early phases, leaving enough resources for the controller.

Moreover, entropy itself can be biased if attacker somehow manage to match attack flow according to threshold values. One common case in this regard is to launch DDoS attack on multiple destinations which ultimately lower the attack impact on entropy. To cover this problem, flow initiation and specification properties of network packets are integrated with entropy algorithm to ensure low False/Positive behavior. Flow rate of traffic is calculated by formula as F_R = n/t where n represent the window size and t is the time of that window. Compared to entropy if the initiation rate is less than threshold value the network is in safe position. Moreover, the flow specification parameters are calculated on properties like packets per flow, number of bytes per flow and the duration of flow. In Fig. 3, a complete framework of research is presented in detail. The suggested algorithm can be embedded in any SDN controller until its required modifications are supported. The detection specifications in proposed work is simplified in Table 2 given below. Moreover, the validity of the proposed detection system is tested on well-known network arrangements shown in further sections.

Operations performed in this research can be understood by the given steps.

- Network traffic states are collected on the SDN controller to perform the detection procedure
- The entropy for given network traffic is calculated for normal and attack traffic to look for a suitable threshold value to compare with attack traffic.
- If the value of entropy exceeds specific traffic, then it is considered a DDOS traffic
- Then this traffic is monitored against the rate of flow initiation and specification for further validations to reduce error.
- If the attack is confirmed, a detection alert is printed along with the suspected nodes and attack is mitigated by blocking its path.

Table 2
Detection Specifications.

Specifications	**Parameters**
Procedure	Entropy variation, Flow initiation and Specification rates
Packet Window	30
No. of Repetitions	5
Entropy in Normal Traffic	1.3
Entropy During Attack	0.4
Threshold	0.5

3.2. Algorithm

• Step 1: Initialization

- pkt_count = 0
- win_count = 0

• Step 2: Collect packet_in events

- Select a window size and its count
- Wait for packets to be received

• Step 3: Calculate the entropy

- Count the entropy for normal and attack case
- Set an optimal threshold entropy value

• Step 4: Compare the entropy with the threshold

- if entrpy_cal < threshold

- win_count = window_count + 1
- if win_count = = 5

• Step 5: Detection

• Step 6: Verify Attack

- Apply flow initiation rate by using FR= n/TW where n = size of the window and TW = window duration
- Apply flow specification by analyzing.

a. The number of received Packets /flow.
b. Bytes / flow.
c. Duration of flow.

• Step 7: Print detection alert

- Print Port id of nodes
- Print time of the suspected attack

• Step 8: Mitigation

- Block port, Ip
- Remove/Update Flow rules
- Block requests from that IP

3.3. Experimental Design

The simulation is conducted on Mininet [30] virtual machine with ubuntu 16.04 support in a high-end system having intel i5-8200U CPU with 32 Gb of RAM to avoid resource constraints by carefully analyzing SDN-based DDOS attacks problem in the state of the art. Then the specific tools, i.e., putty SSH client Xming, and Wireshark are interacted with simulation environment to continually monitor the findings. The detailed analysis of whole experimental design is given in subsections below

3.3.1 Network Design

Network topologies may have specific vulnerabilities against DDOS attacks due to their arrangements. To test this impact two different datacentric topologies as shown in Fig. 4 (a) and (b), have been made to test the validity of the solution against DDOS traffic using a centralized SDN controller known as 'POX' [32].

Scapy [31], a widely used packet manipulation tool is used to generate the network traffic with variable properties. In addition to packet generation, sniffing, scanning, and forging, Scapy is also used for creating DDOS attack scripts, i.e., UDP flood and TCP-SYN flood attacks, by manipulating different attack features. Lastly the proposed detection algorithm is also tested on ICMP ping flood attacks under given scenarios.

3.3.2 Testbeds

The testbed is created to validate all possible attack situations under UDP, TCP-SYN, and ICMP ping flood attacks. These DDoS attacks are performed by changing the forwarding properties like bytes sent, duration, and by changing the number of nodes to increase or decrease the flood to make a number of distinct attack patterns. Pattern-1 is designed to have a distinct difference in legitimate and malicious traffic. The attacking hosts generate malicious traffic with variation of 28%, 35%, and 63% of malicious traffic using a number of botnets. Similarly, in the case of multiple destinations attacks, 26%, 42%, and 54% attack load using multiple attacking botnets is generated toward multiple destinations. Attack rate can be calculated by the formula given as:

Attack rate = (DDOS traffic ÷ Total traffic) × 100

In Pattern 2, the normal and malicious traffic characteristics are mixed to have similarities. This test is created to reveal the error rate when the attack behaves like a legitimate flow. Both attack situations, i.e., single destination or multi-destination, include more than 20 simulations to get average response rates regarding delay and accuracy. In a multi-destination attack, a single target is targeted by multiple malicious nodes depending on the type of topology to produce specific loads while splitting the attacks into groups of destinations. Table 2 describes the parameters used for the testbed in detail.

Table 2
Testbed Specifications

Parameters	Specifications
Type of Attacks (DDoS Flood)	UDP, TCP, ICMP
Attack Specifications	Single/multi-destinations
Tool to Formulate the Attacks	Scapy
Simulation Topologies	Simple and Meshed Tree Topology
SDN Architecture	Centralized
Controller	Pox
Simulator	Mininet

4 Results And Discussion

After a complete implication of proposed experimental design on given problem, the detailed results and discussion given under following subsections.

4.1. Detection Accuracy

DDoS attacks have the feature of "many to one," which implies that numerous attacking hosts transmit the malicious traffic to a single location. It increases the attack's amplitude to disrupt the target swiftly. The single and multi-destination attacks by changing the attack rates are examined to validate the detection process. A multi-destination attack is performed to validate conditions where every network node can be targeted. Figure 5 represents the load on the network before and after the DDoS attack is detected and mitigated. Further analysis of DDoS attack detection and mitigation is discussed in the below subsections

4.1.1 Single destination Attacks

The detection analysis performed in single destination attacks uses distinct patterns, as discussed above in section 3.3.2. A noticeable difference between normal and attack traffic patterns has been set in the first scenario. Here denial of service attacks is performed 20 times in each 28, 35, and 63 percent load cases. In this case, the algorithm quickly detects malicious traffic with 100% accuracy, analyzing a sudden change in entropy values without involving other validation parameters like flow initiation and specification to validate the attack flow further. In the second pattern, the attack traffic is mixed with regular traffic characteristics to check how far the algorithm can detect attacks under a bearable error rate. On careful observation of obtained results, it is observed that the algorithm is working with a maximum detection rate of 98.33% by involving the other two parameters of detection algorithm. Lastly in pattern three the properties of attack and normal traffic are completely mixed to verify the detection. Here, an increasing error rate in term of false positive (FP) and False negative (FN) behavior is observed due to variation of traffic pattern with short and long flows where the shorter flows resemble the attack traffic. Here, the algorithm has achieved a detection rate of up to 91.66% by using flow initiation and specification parameters. Detection analyses of single destination attacks are given in Fig. 6 (a).

4.1.2 Multi-destinations attacks

In this simulation, 20 distinct runs are carried out for each pattern to have an average effect of detection algorithm on malicious flow. Using the scapy packet manipulation the to implement different rations of attack and normal traffic with variable number of normal and attacking nodes. One host node creates attack traffic, while the other 19 hosts run regular traffic for a 26 percent attack ratio in the first scenario. Secondly, two attack nodes deliver the attack toward eight destinations, while 18 hosts generate legal traffic, resulting in 42 percent of the attack traffic sent by two hosts. After three hosts have sent attack traffic,17 hosts create legal traffic while attacking 12 destinations produces 54% of the traffic load. In Pattern1 a similar characteristic as in pattren1 of single destination attacks are detected with a 100% detection rate in all applied attacks is observed because the algorithm can successfully differentiate attacks based on entropy. In pattern 2, attack characteristics are mixed with legitimate traffic, as done in a single destination (pattern 2) attack case. Because the attack traffic is split over several locations, the effect of the attack falls below the threshold level. The algorithm achieved 95% accuracy under 5% error rate by applying flow initiation and specification rate. In pattern3, a considerable decrease in detection rate is observed due to the addition of short and long flow. Due to the addition of these flows in traffic pattern 3 with attack performing toward multiple destinations, determine that choosing a suitable threshold is complex, and it is challenging to establish a margin that fits both flow types. Detection trends in multi-destination attacks are given in Fig. 6 (b).

4.2. Detection Time

The intended goal of this investigation is to identify DDoS traffic as early as possible. Detection time lowers as the attack load rises since the more traffic there is, the faster the packet sample window will be captured. At an average of 11.83 seconds in all proposed test scenarios, the DDoS traffic is detected and mitigated by controller in single-destination attacks. In multi-destination attacks, the average detection time is recorded to 18.9 seconds as well. The average delay of 7.07 seconds as compared to single destination attacks case is due to mixing the properties of attack traffic in normal traffic to behave like normal flows which ultimately reduced the average efficiency and caused a considerable delay. Figure 7 (a) and (b) illustrate the average detection delay in single and multi-destination attacks.

Based on the given study, the following limitations can be counted as (a) The analysis is based on centralized SDN architecture and (b) short and long flow characteristics can be analyzed using machine learning. Based on the current work, a detailed comparison with the previous solution is presented with state of the art in Table 4.

Table 4
Comparison of Current Work with recent solutions

Parameters	**[23]**	**[27]**	**[28]**	**[32]**	**[33]**	**Proposed work**
Techniques	Flow Rate	PCA	Entropy	Entropy	Entropy	Entropy, Flow Initiation & Specification
Centralized Architecture	√	√	√	√	√	√
Attack Traffic	UDP	UDP	UDP	UDP	UDP	UDP, TCP, ICMP
Network Topology	Linear	Ring	Tree	Tree	Tree	Simple and Meshed Tree Topology
Attack Sources	3 to 4-	2 to 3	2 only	1 to 5	1 to 7	1 to 12
Controller	RYU	POX	POX	POX	POX	Pox
Find the Attack Path	-	√	-	-	√	√
Error Rate (Low)	High	Directly proportional	√	√	√	√
Controller Overhead	High	Low	moderate	moderate	Moderate	Low

5 Conclusion

DDoS attacks overwhelm networks by consuming their resources, i.e., bandwidth or computational power. Traditionally, an attacker compromises a network service by flooding it huge amount of fake traffic using botnets. It is not necessary in SDN cases for the attacks to be successful. The attack flow might be spread out as much as possible to avoid detection measures and yet hit the controller and switches. To ensure this, SDN's based DDoS detection method must be scalable enough because attackers continually look for new ways of attacks to avoid detection. Moreover, the detection latency must be extremely short to provide adequate time to build a mitigation process. The prime focus of this research was to apply an authentic and lightweight solution for detecting a variety of DDoS attacks in their early. In this research we have built resilience into the SDN controller, which detects and mitigate the malicious DDoS activity based on variation in network traffic and achieved a high detection rate within range of 91.66% -100% with an average delay of 11.83 seconds for mitigation in all single-destination attacks. Similarly, to extend the adverse effects of DDoS attacks, multi-destination attacks are performed where the algorithm with little decline performance achieved an overall accuracy between 88.9% – 100% with an average delay of 18.9 seconds for mitigation. The surge decline in algorithm performance is noticed due to the attack capability having similar properties to regular network traffic. Early detection is verified by the fact that the approach only uses 150 packets, which shows that an attack can be detected before it can start compromising the network. In the future, we are looking forward to implementing this work in low-rate attack detection in IoT scenario.

Declarations

Funding No funding has been acquired from any source for the current work

Data Availability All data generated/analyzed and or modified during this study are taken from published article [33] (A. Mishra, N. Gupta, and B. B. Gupta, “Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX controller,” *Telecommun Syst*, vol. 77, no. 1, pp. 47–62, May 2021, doi: 10.1007/s11235-020-00747-w) as a a base paper.

Acknowledgments The Author acknowledges the support and interest of the supervisory committee and all the stake holders of National Textile University, Faisalabad, Pakistan in pursuing this research.

Ethical Approval This article does not contain any studies with human participants or animals performed by the author.

References

1. N. Anerousis, P. Chemouil, A. A. Lazar, N. Mihai, and S. B. Weinstein, “The Origin and Evolution of Open Programmable Networks and SDN,” *IEEE Communications Surveys and Tutorials*, vol. 23, no. 3. Institute of Electrical and Electronics Engineers Inc., pp. 1956–1971, Jul. 01, 2021. doi: 10.1109/COMST.2021.3060582.
2. R. Wazirali, R. Ahmad, and S. Alhiyari, “Sdn-openflow topology discovery: An overview of performance issues,” *Applied Sciences (Switzerland)*, vol. 11, no. 15. MDPI AG, Aug. 01, 2021. doi: 10.3390/app11156999.
3. K. Benzekki, A. el Fergougui, and A. Elbelrhiti Elalaoui, “Software-defined networking (SDN): a survey,” *Security and Communication Networks*, vol. 9, no. 18, pp. 5803–5833, Dec. 2016, doi: 10.1002/sec.1737.
4. A. Montazerolghaem, “Software-defined load-balanced data center: design, implementation and performance analysis,” *Cluster Comput*, vol. 24, no. 2, pp. 591–610, Jun. 2021, doi: 10.1007/s10586-020-03134-x.
5. E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, “Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey.”
6. M. Alsaeedi, M. M. Mohamad, and A. A. Al-Roubaiey, “Toward Adaptive and Scalable OpenFlow-SDN Flow Control: A Survey,” *IEEE Access*, vol. 7, pp. 107346–107379, 2019, doi: 10.1109/ACCESS.2019.2932422.
7. “OpenFlow - Open Networking Foundation.” https://opennetworking.org/sdn-resources/customer-case-studies/openflow/ (accessed Mar. 29, 2022).

8. Y. Hande and A. Muddana, “A survey on intrusion detection system for software defined networks (SDN),” *International Journal of Business Data Communications and Networking*, vol. 16, no. 1. IGI Global, pp. 28–47, Jan. 01, 2020. doi: 10.4018/IJBDCN.2020010103.
9. “Open vSwitch.” https://www.openvswitch.org/ (accessed Apr. 02, 2022).
10. J. Son and R. Buyya, “A taxonomy of software-defined networking (SDN)-enabled cloud computing,” *ACM Computing Surveys*, vol. 51, no. 3. Association for Computing Machinery, Apr. 01, 2018. doi: 10.1145/3190617.
11. G. P. Xavier and B. Kantarci, “A survey on the communication and network enablers for cloud-based services: state of the art, challenges, and opportunities,” *Annales des Telecommunications/Annals of Telecommunications*, vol. 73, no. 3–4. Springer-Verlag France, pp. 169–192, Apr. 01, 2018. doi: 10.1007/s12243-018-0629-4.
12. A. Montazerolghaem, M. H. Y. Moghaddam, and A. Leon-Garcia, “OpenSIP: Toward software-defined SIP networking,” *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 184–199, Mar. 2018, doi: 10.1109/TNSM.2017.2741258.
13. B. G. Assefa and O. Ozkasap, “RESDN: A Novel Metric and Method for Energy Efficient Routing in Software Defined Networks,” *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 736–749, Jun. 2020, doi: 10.1109/TNSM.2020.2973621.
14. “Kaspersky Enterprise Cybersecurity Protecting your business against financial and reputational losses with Kaspersky DDoS Protection Kaspersky DDoS Protection.” [Online]. Available: www.kaspersky.com#truecybersecurity
15. L. F. Eliyan and R. di Pietro, “DoS and DDoS attacks in Software Defined Networks: A survey of existing solutions and research challenges,” *Future Generation Computer Systems*, vol. 122, pp. 149–171, Sep. 2021, doi: 10.1016/j.future.2021.03.011.
16. S. Dong, K. Abbas, and R. Jain, “A Survey on Distributed Denial of Service (DDoS) Attacks in SDN and Cloud Computing Environments,” *IEEE Access*, vol. 7, pp. 80813–80828, 2019, doi: 10.1109/ACCESS.2019.2922196.
17. C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, “DDoS in the IoT: Mirai and other botnets,” *Computer (Long Beach Calif)*, vol. 50, no. 7, pp. 80–84, 2017, doi: 10.1109/MC.2017.201.
18. “DDoS attack that disrupted internet was largest of its kind in history, experts say | Hacking | The Guardian.” https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet (accessed Jun. 10, 2022).
19. “DDoS report Q3 2019 | Securelist.” https://securelist.com/ddos-report-q3-2019/94958/ (accessed Jun. 10, 2022).
20. *2019 IEEE 9th International Conference on Advanced Computing (IACC)*. IEEE, 2019.
21. C. D. Cajas and D. O. Budanov, “Mitigation of Denial of Service Attacks Using OpenDaylight Application in Software-Defined Networking,” in *Proceedings of the 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, ElConRus 2021*, Jan. 2021, pp. 260–265. doi: 10.1109/ElConRus51938.2021.9396272.

22. S. M. Mousavi and M. St-Hilaire, “Early Detection of DDoS Attacks Against Software Defined Network Controllers,” *Journal of Network and Systems Management*, vol. 26, no. 3, pp. 573–591, Jul. 2018, doi: 10.1007/s10922-017-9432-1.
23. “DDOS Attack Detection & Prevention in SDNusing OpenFlow Statistics,” 2019.
24. S. Aluru, Jaypee Institute of Information Technology University, University of Florida. College of Engineering, IEEE Computer Society, IEEE Computer Society. Technical Committee on Parallel Processing, and Institute of Electrical and Electronics Engineers, *2018 Eleventh International Conference on Contemporary Computing (IC3) : 2-4 August 2018, Jaypee Institute of Information Technology, Noida, India*.
25. O. Rahman, M. A. G. Quraishi, and C. H. Lung, “DDoS attacks detection and mitigation in SDN using machine learning,” in *Proceedings - 2019 IEEE World Congress on Services, SERVICES 2019*, Jul. 2019, pp. 184–189. doi: 10.1109/SERVICES.2019.00051.
26. S. Gao, Z. Peng, B. Xiao, A. Hu, Y. Song, and K. Ren, “Detection and mitigation of DoS attacks in software defined networks,” *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1419–1433, Jun. 2020, doi: 10.1109/TNET.2020.2983976.
27. Institute of Electrical and Electronics Engineers., *2018 IEEE International Conference on Communications (ICC) : proceedings : Kansas City, MO, USA, 20 -24 May 2018.*
28. Institute of Electrical and Electronics Engineers, *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU).*
29. S. Batool *et al.*, “Lightweight Statistical Approach towards TCP SYN Flood DDoS Attack Detection and Mitigation in SDN Environment,” *Security and Communication Networks*, vol. 2022, 2022, doi: 10.1155/2022/2593672.
30. “Mininet: An Instant Virtual Network on Your Laptop (or Other PC) - Mininet.” http://mininet.org/ (accessed Mar. 31, 2022).
31. “Scapy.” https://scapy.net/ (accessed Apr. 02, 2022).
32. S. M. Mousavi and M. St-Hilaire, “Early Detection of DDoS Attacks Against Software Defined Network Controllers,” *Journal of Network and Systems Management*, vol. 26, no. 3, pp. 573–591, Jul. 2018, doi: 10.1007/s10922-017-9432-1.
33. A. Mishra, N. Gupta, and B. B. Gupta, “Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX controller,” *Telecommun Syst*, vol. 77, no. 1, pp. 47–62, May 2021, doi: 10.1007/s11235-020-00747-w.

Figures

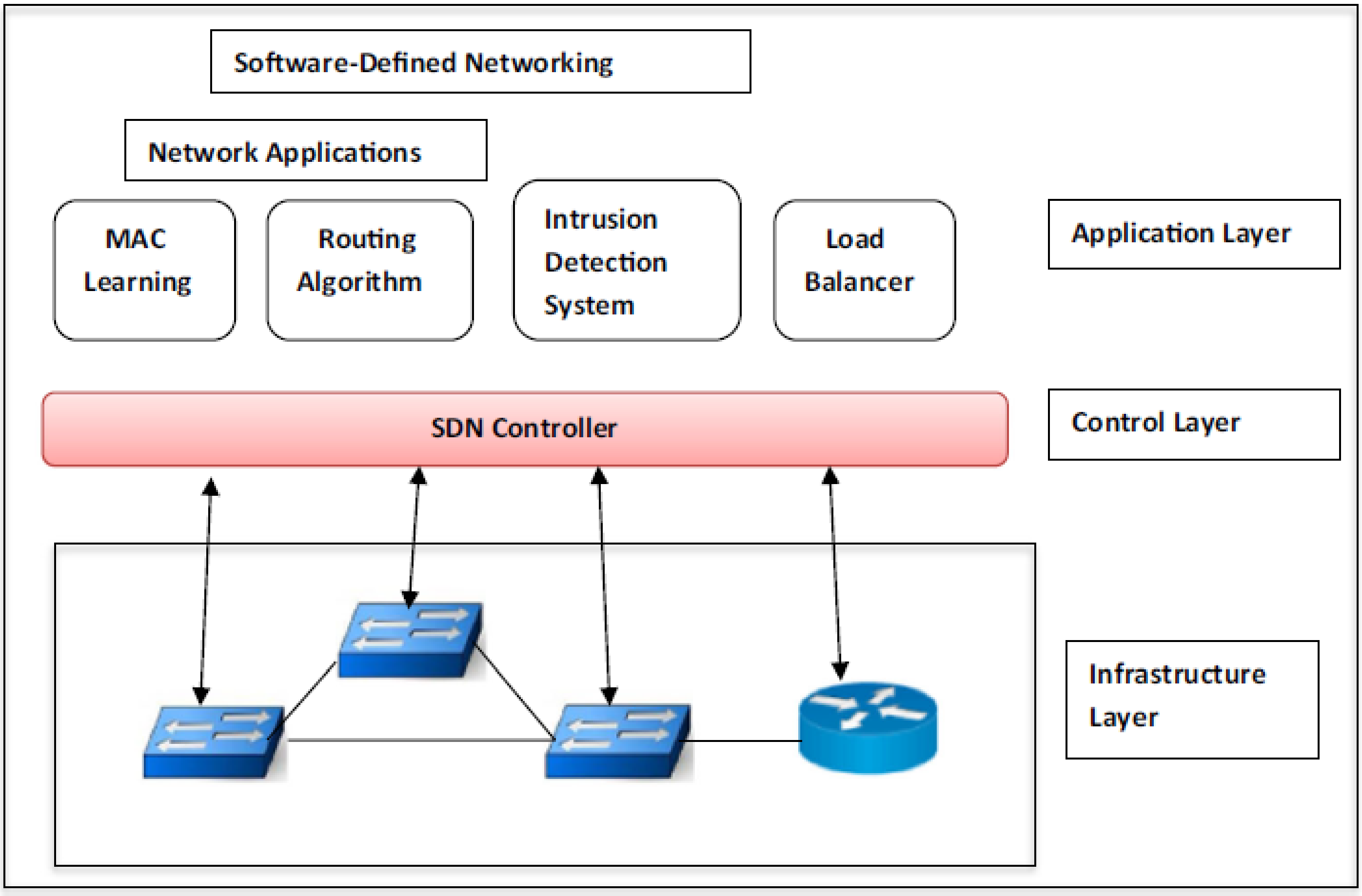

Figure 1

SDN Architecture [11]

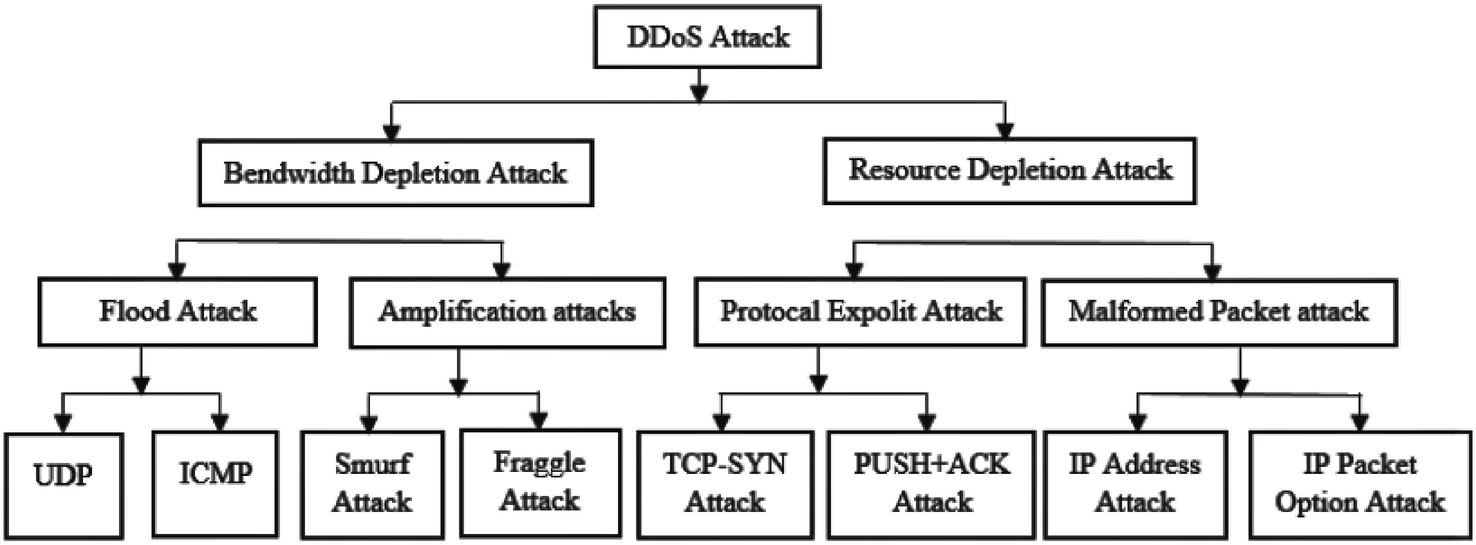

Figure 2

Classification of DDOS attacks [16].

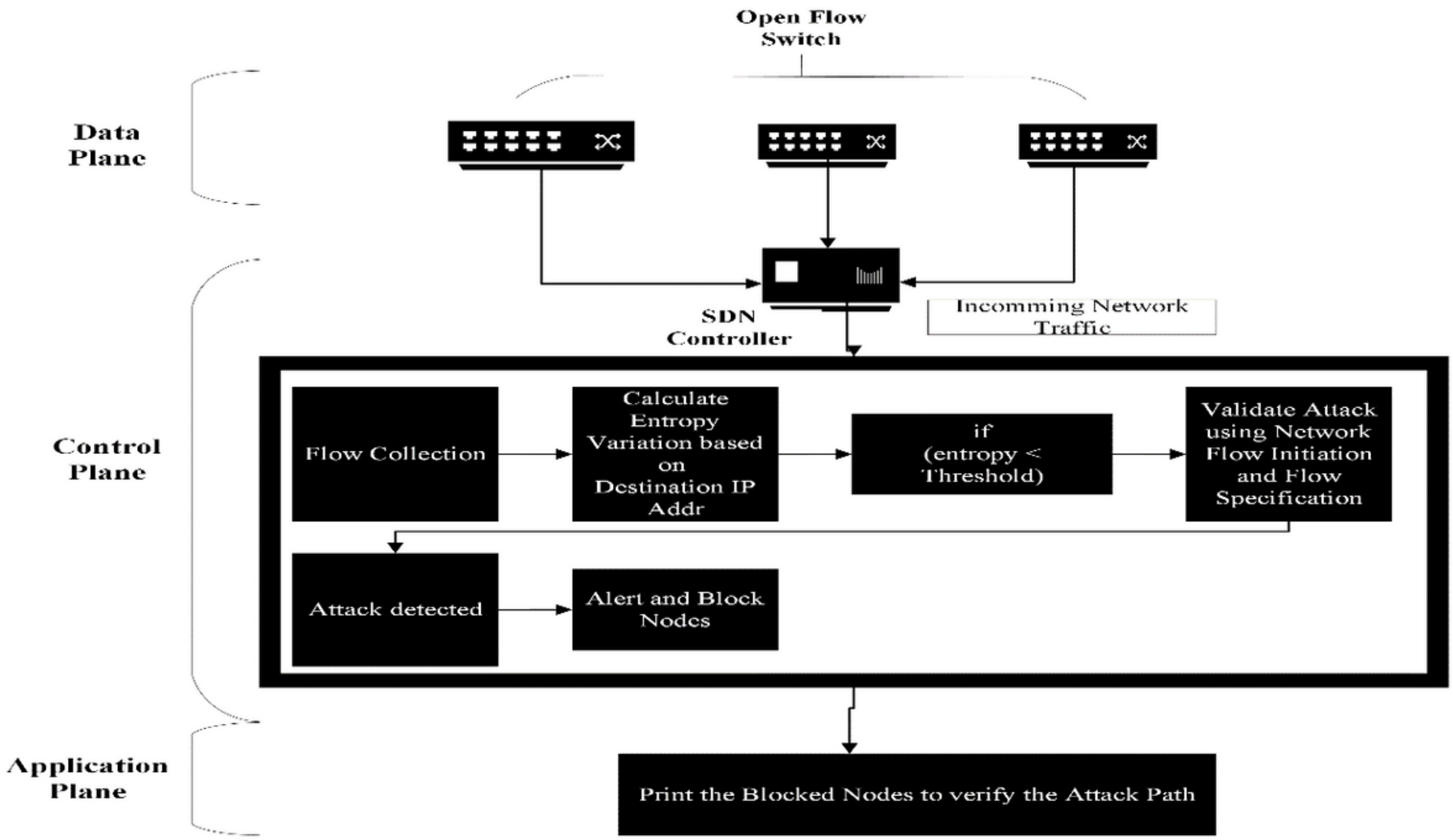
Open Flow Switch
Data Plane
SDN Controller
Incomming Network Traffic
Control Plane
Flow Collection
Calculate Entropy Variation based on Destination IP Addr
if (entropy < Threshold)
Validate Attack using Network Flow Initiation and Flow Specification
Attack detected
Alert and Block Nodes
Application Plane
Print the Blocked Nodes to verify the Attack Path

Figure 3

Research Framework

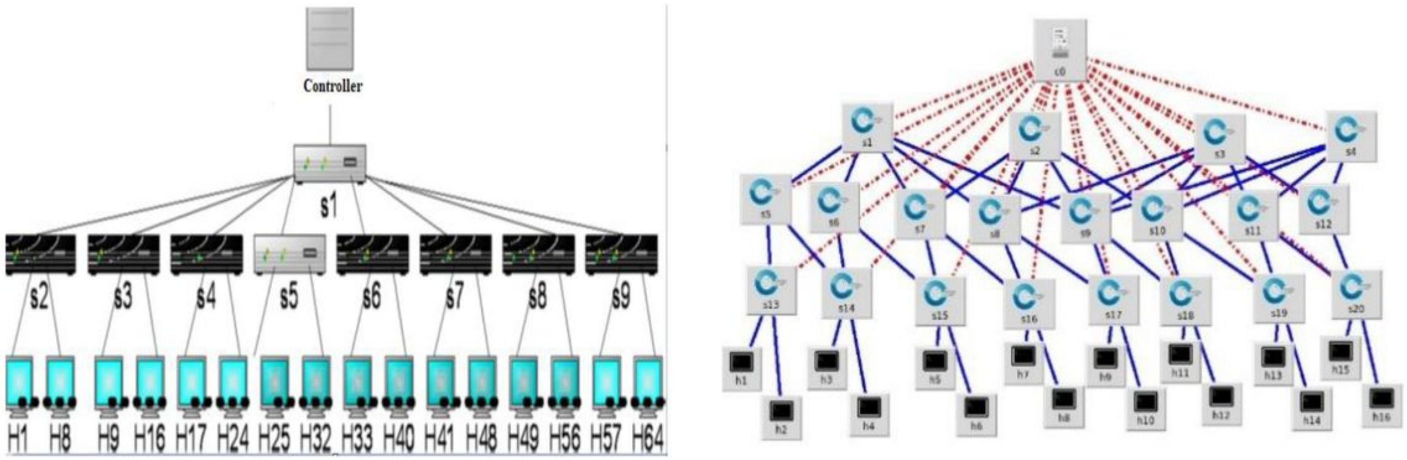

Fig 4 (a):

Fig 4 (b):

Figure 4

4 (a): Simple Tree Topology

4 (b): Tree Meshed Topology

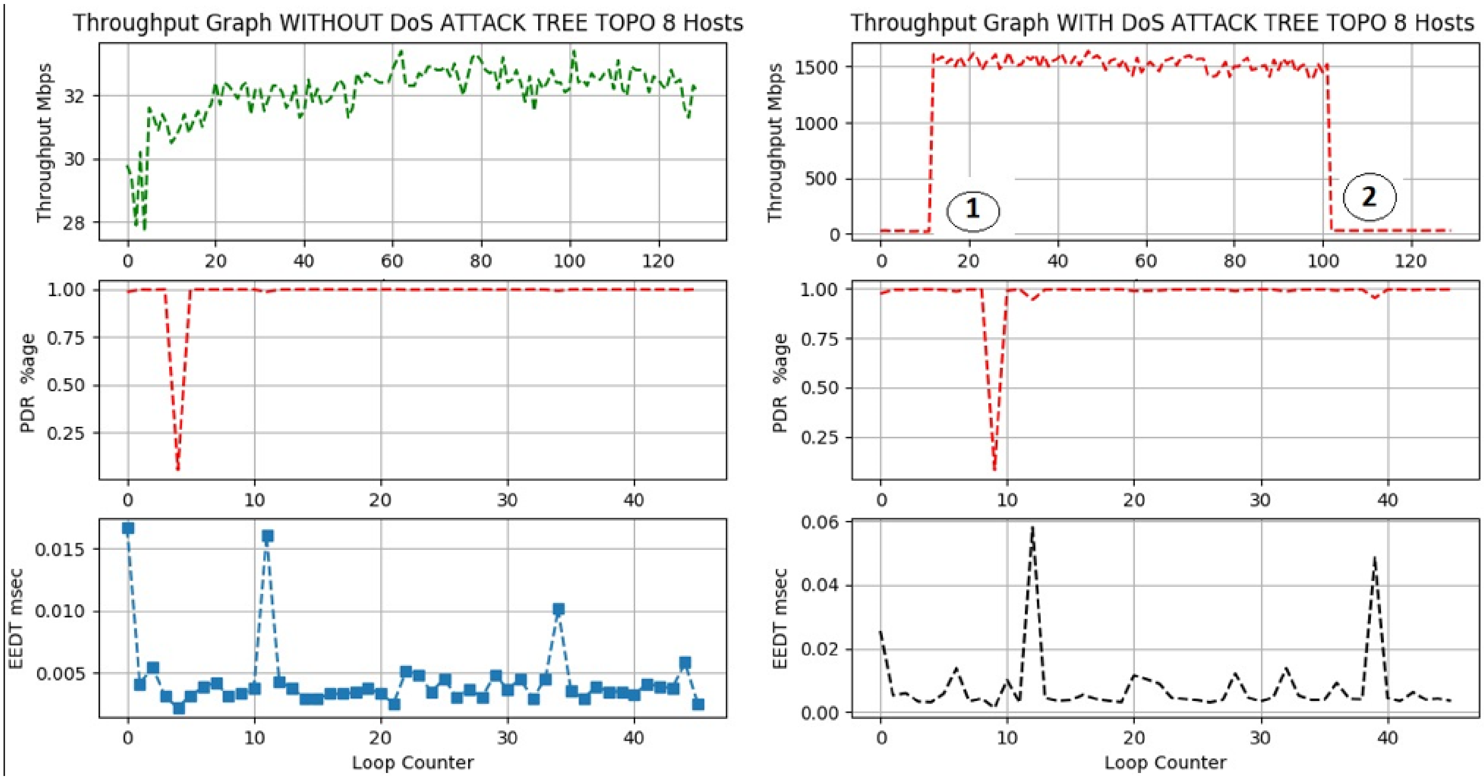

Figure 5

Network Load in Before Detection and After Mitigation of DDoS Attacks.

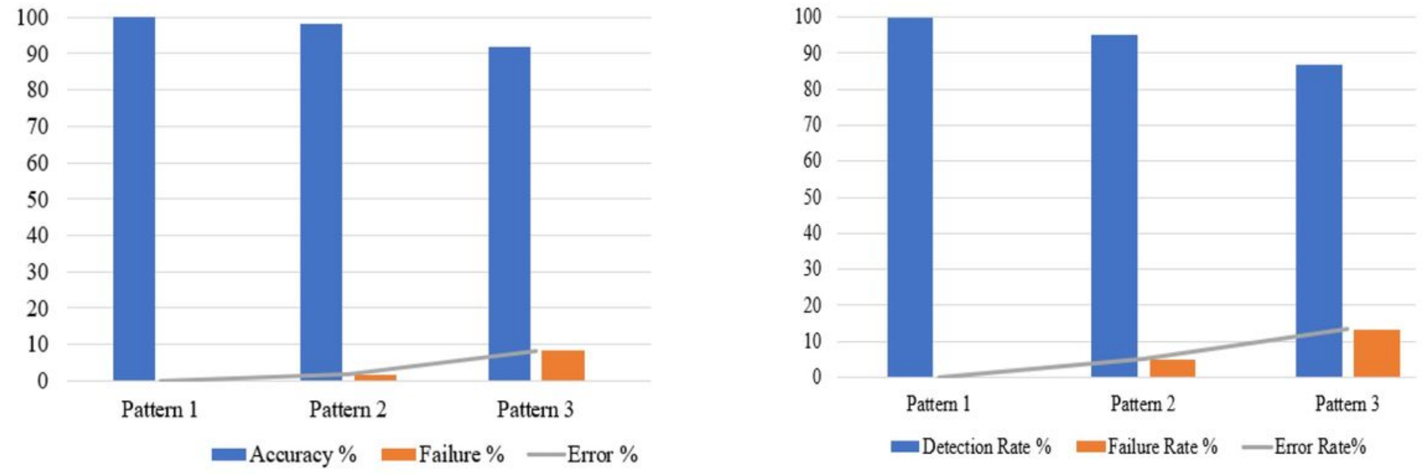

Fig 6 (a):

Fig 6(b):

Figure 6

6 (a): Detection rate in single Destination Attacks

6 (b): Detection rate in multi-destination attacks

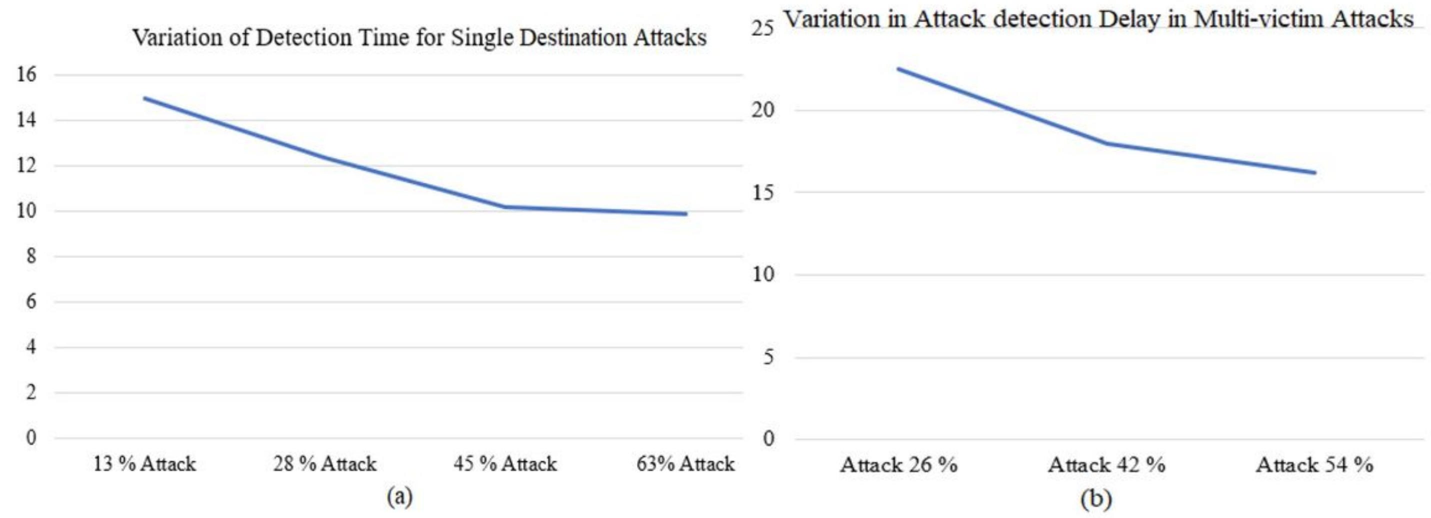

Figure 7

(a), (b): Detection delay in single and multi-destination Attacks