ISSN: 2834-4928

Research Article

Journal of Electrical Electronics Engineering

Detection and Mitigation of Malicious DDoS Floods in Software Defined Networks

Furqan Ahmad[1]*, Maham Saleem[1], Ubaid ur Rehman[2]

[1]Department of Computer Science, National Textile University, Faisalabad, Pakistan

[2]School of Computing, Engineering and Built Environment (SCEBE), Glasgow Caledonian University, Scotland, United Kingdom

[*]Corresponding Author
Furqan Ahmad, Department of Computer Science, National Textile University, Faisalabad, Pakistan.

Submitted: 2023, Mar 13; **Accepted:** 2023, Apr 25; **Published:** 2023, May 04

Citation: Ahmad, F., Saleem, M., Rehman, U. (2023). Detection and Mitigation of Malicious DDoS Floods in Software Defined Networks. *J Electrical Electron Eng, 2*(2), 120-129.

Abstract
The advent of software-defined networking (SDN) has significantly transformed network management by offering modular control and data plane characteristics, enabling adaptability and flexibility in managing networks. This innovation entails the separation of control and data plane elements to facilitate efficient network administration. Nevertheless, the centralization resulting from control plane separation renders SDN vulnerable to cyber threats, particularly Distributed Denial-of-service (DDoS) attacks that target SDN controllers. Recently, studies have highlighted the relevance of entropy-based attack detection techniques compared to alternative methods. However, relying solely on entropy may overlook detection in specific variables, such as flow specification variations. To address the limitations of entropy-based detection systems, we developed a DDoS attack detection framework within the SDN control plane, integrating the packet flow initiation and specification properties with an entropy-based algorithm to ensure accurate attack detection measures. Our lightweight framework aims to mitigate DDoS attacks by detecting their impact in the early stages, thus preventing SDN controllers from being hijacked due to excessive packet flooding. The simulation is employed in Mininet network simulator to implement, and the testbed is created by focusing UDP flood attacks in widely used data-centric tree topologies. The experimental results demonstrate that our proposed solution effectively detects and mitigates novel parameters of SDN-based DDoS floods within 150 packets while maintaining minimal delay and high accuracy.

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

1. Introduction

Software-defined networking (SDN) was introduced at Stanford University and the University of California at Berkeley to provide flexibility in traditional network infrastructure, which typically relies on hardware components [1]. As a family of technologies, SDNs marked the beginning of revolutionary innovations and changes in the telecommunication sector, as seen in the early ages of internet [2]. The segregation of network planes provided in SDN allowed for adding flexibility, extended quality of service (QOS) management, abstraction, and security in already present traditional components due to having a broader view under centralized control. These features enabled SDN to provide dynamic designs of networks, which has enhanced the network performance to meet new IT infrastructures, making it more like cloud computing than conventional network administration methods [3].

In traditional networks, each component, i.e. (routers & switches) is tightly integrated, making it challenging to develop and deploy novel network applications at a fine-grained level. SDN overcomes these limitations by separating network intelligence from underlying hardware components in the data plane layer. One of the prime goals of SDN is to streamline the network by separating the forwarding and routing of network packets into separate processes [4]. In SDNs, the communication between the network planes is performed by different interfaces under SDN standardization [5]. The northbound interface (NBI) establishes the communication between the application and control layers, while the southbound interface (SBI) communicates control and application layer communications [5]. A typical NBI example is the REST API, while the SBI is operated under the open flow protocol standard to implement on hardware and software environments to implement the SDN [6,7]. The controller in SDN is an essential component that serves as intelligence for the whole network [8]. SDN networks with software switches. Are implemented using a variety of modeling and emulation platforms to virtualize the hardware needs. The generalized architecture of SDN as presented in Figure 1 is composed of three layers. Moreover, SDN exceptionally provides facilities like defining and changing traffic flow rules to lower security issues with an end-to-end visibility to provide connectivity rules with a central control [9,10]. Based on the above discussion, the SDN provides following benefits.

• Programmability: Support to program network according to variable needs.
• Abstraction: provides the necessary functionalities of networking needs by hiding functional details.

• Centralization: Decision management is controlled through centralized control 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 underlying network components i.e., (routers and switches).

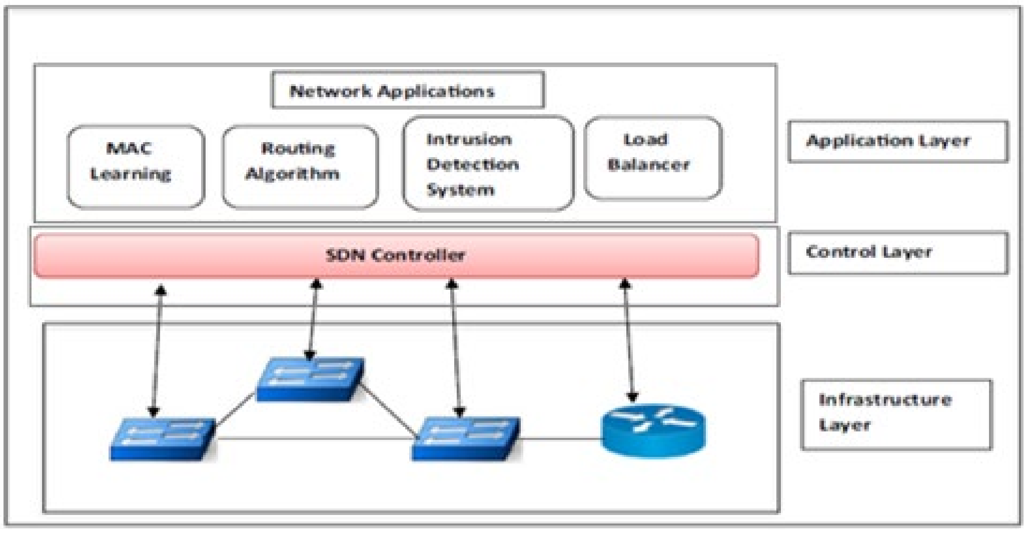

Figure 1: Generalized Architecture of SDNs

1.1 SDN as an Expansion

The rapid expansion in telecommunications sector, virtualized servers, and cloud computing requires quick and up to mark upgrades in network infrastructure [7]. With arrival of SDNs in existing networks i.e., Data centers, and mobile telecom operators now have far more dynamic processing and storage facilities additional intelligence support [11]. Moreover, further research directions have been initiated to achieve energy efficiency by dynamically adjusting the network data plane and leveraging routing protocols [12]. Some of necessary factors to assist the expansion of SDN paradigm are the following:

• *Changing Traffic Patterns:* Cloud data centers are investigating a new way of integrating SDN components that might enhance private cloud computing at fine-grained level.
• *Existing IT Infrastructures:* IT industries are looking forward to integrating the SDNs in their existing networks to handle huge number of Users' devices, such as smartphones, while satisfying compliance demands.
• *Big Data:* Parallel processing on hundreds of computers is required for today's "big data" support Energy Consumption in Datacenters which has open doors for SDN integrations.

The remaining work is divided up into the following subsections. A comprehensive background of related problem of DDoS threats in SDN is presented in Section II, Section III discussed the relevant contributions, while a brief description of methodology is presented in section IV. Finally in section V, a detailed description of results, discussion and conclusion were made to summarization objective.

1.2 Background

The centralized network control plane of an SDN-based network makes it a preferred target for cyberattacks. If the control plane is successfully compromised, the whole network might be affected, resulting in significant disruption. Therefore, sectors employing SDNs must build comprehensive security measures to guard against cyber vulnerabilities to ensure necessary counter measures before such situations may arrive.

1.1.1. SDN Security Threats

Decoupling the control and data plane in SDN brings various security challenges which can leverage its centralizations by compromising the centralized resources. Denial of Services (DoS) attacks are one such common threat which could compromise the network resources to disrupt lower its services to end nodes. SDN-based DoS attacks bring flooding to control and data plane resources to downgrade the entire network [13]. To overwhelm the networks resources, the attackers broadcast the flood of fake network packets which are processed through SDN controller due to lake of policy matching in flow table which will eventually fill the flow table in the switches. On the other hand, the controller will not be able to handle such a heavy load of flow requests which eventually degrade it from its normal operations, Due to this, the controller may get compromised and unavailable for an extended period. Figure 2 depicts some of commonly used SDN-based DoS attacks as categorized by.

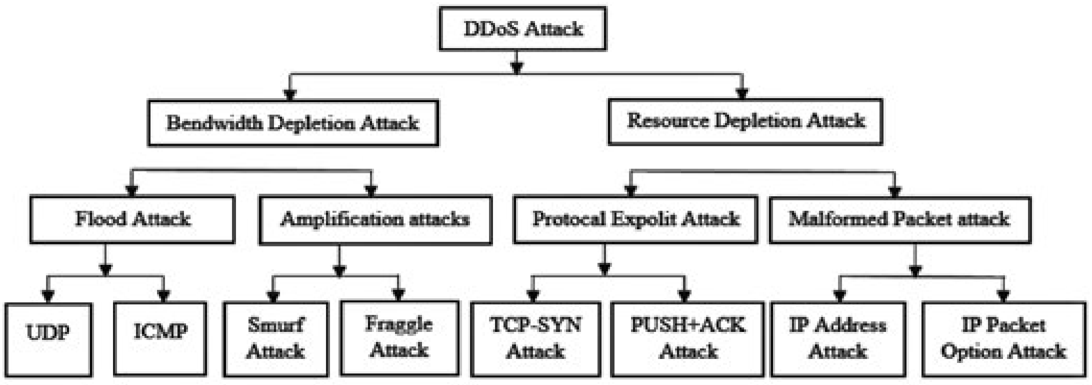

Figure 2: Taxonomy of DDoS Attacks

In this paper we have addressed the problem of DDoS floods in SDN and conveyed its negative effects on SDN centralized architecture. Additionally, we encountered the issue of Entropy based detection for DDoS floods and proposed the integration of flow specification properties with entropy-based algorithm which detects the attack at a far more accuracy. Our approach is lightweight and scalable and handles DoS-related problems at their early stages on controller level.

2. Literature Review

DDoS attacks are considered one of the most common threats nowadays, posing critical danger to network services due to their ease in carrying out and compromising 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. The difficulty in locating the assailant provided a considerable boost to the attack's efficacy. For instance, the mirai botnet attack in 2016 affected network availability over the globe, which brought down a large portion of the Internet [14]. In 2018, GitHub servers were targeted by the largest-ever DDoS attack [15]. 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 [15]. During the third quarter of this year, the US and the Netherlands had the largest percentage of DDoS attack-launching botnet devices [16]. 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.

2.1. Recent Work

SDN brings a flexible quality of service (QoS) for network management which helps to manage the dynamic network architectures through control plan configurations. These configurations, on the other hand are proven robust against cyber-attacks in recent works. Although the policies defined in SDN control architecture support resilience against network-based malicious activities, there have been limitations in handling a wide variety of attacks i.e., DDoS attacks. To address this problem, has given detection and prevention solutions for DDoS attacks using Mininet, a standard emulator for SDN. Their framework measured the flow statistics from switches and parameters like packet rate and bandwidth [17]. If a sudden increase in packet rate was detected, it was considered an ongoing attack, mitigated by producing the forwarding rule for that node, which drops the packets. The research presented by has proposed the solution using an ODL controller by installing [18]. Their solution mitigated the adverse effects of flooded traffic under a negligible controller overhead. The controller collected the flow statistics and estimates the throughput of these flow rules to predict the possibility of an any flood attack. Based on its threshold estimation, it can block traffic that increases by its defined 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 developed an SDN integrated intrusion detection system (IDS) that reacted to attacks near to their source, guaranteeing regular operation. The idea combined an IDS that autonomously identify UDP-based DDoS attacks and created an alert to the SDN controller [19]. In addition, the suggested solution employed 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. The continuous updating of IDS against novel attack patterns remained a concern. In DDoS attack detection is done using the flow collection per unit time [20]. 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 was specified, which blocked the malicious traffic to spread further in the network. This work has shown limitations, i.e., dependency on a specified threshold limits for detection, resulting in a severe downgrade when the attack specifications are changed. On the other hand, this system could fail in case of a low packet forwarding rate.

A dynamic decision-making system is introduced by at SDN switches to filter malicious packets involved in denial-of-service attacks. When determining whether a packet is malicious or not, the system considered a number of characteristics. The packet is discarded if the system detected a packet's score exceeds the danger level [21]. On the other hand, the packet can pass to its regular route if the packet score is below the danger level. In this work, the authors have 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 framework needed to be trained against more datasets for scalable results [22]. In 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. According to their findings, J48 beaten the other ML techniques in case of performance, training and testing time. However, training and testing constraints on these algorithms reveal detection limitations, which may result in false alarms or outright failure to detect.

In a protocol-independent approach, 'Flood-Defender,' using the SVM algorithm is introduced [23]. This technique used three approaches, i.e., table miss mechanism, packet filtering, and flow table management. This approach has shown some limitations i.e., increased false-positive rate with an increased attack rate, that need to be addressed. Moreover, the attacker can evade detection by sending packets at a lower rate. The problem of low-rate flood attacks is solved in the work presented by using principal component analysis (PCA) [24]. This work tried to demonstrate the limitations of entropy calculation in case the DDoS flood is mixed in normal traffic [25]. In the authors 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 research only focused on high-rate attacks under some critical scenarios, which raised questions on its validity in some critical test cases.

In SDN centralized architecture is addressed against DDoS attacks using a lightweight entropy detection solution to detect the attack. The work also showed some fundamental limitations as it only deal 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 [26]. In 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 flood attacks. To summarize the previous contributions, a concise list of pros and cons has been listed in table 1.

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

Table 1: Pros and Cons of Recent Work

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, 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 and is used to calculate the entropy of random variables, i.e., a destination address, measures uncertainty or unpredictability to determine the possible change in network states. The entropy is calculated in the range of [O $\log 2n^{ip}$], 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 sent across random destinations. The collection of packets for entropy analysis is accomplished via a window of a specified length. Using a window calculated in time, reduces the computational cost of the entropy calculation. On the other side, the accuracy of the entropy computations may be degraded during periods of low traffic demands, using a fixed-time window. This problem is solved by using a window size 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 n_i shows the total packets with destination IP address
The entropy is computed using the following formula.

$H = -\sum_{i=1}^{m} Fi \log 2\ Fi$
Where $0 \leq Fi \leq 1 \mapsto H \geq 0$

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 = window size and t = time of 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 Figure 3: Research Framework, 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 Specifications. Given below. Moreover, the validity of the proposed detection system is tested on well-known network arrangements shown in further sections.

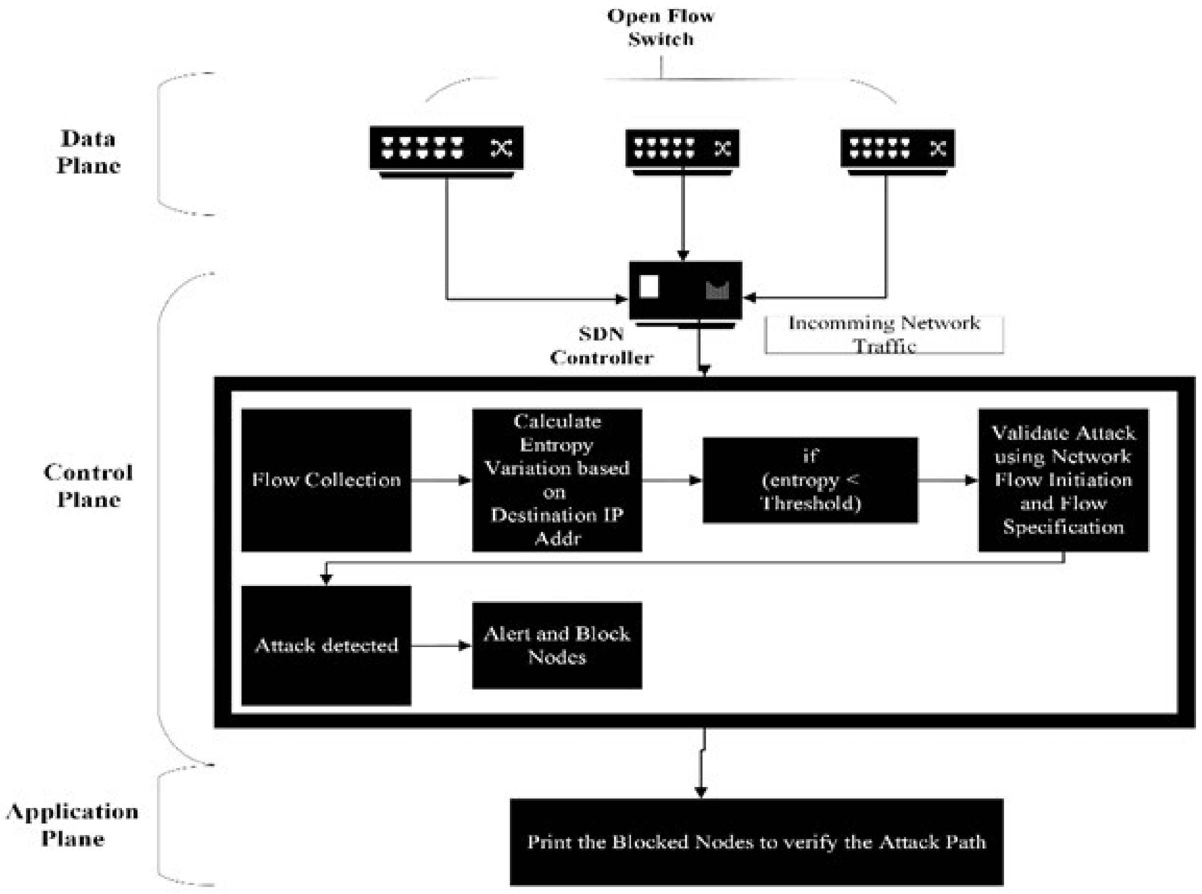
Open Flow Switch
Data Plane
SDN Controller
Incomming Network Traffic
Control Plane
Flow Collection
Calculate Entropy Variation based on Destination IP Addr
if (entropy < Threshold)
Validate Attack using Network Flow Initiation and Flow Specification
Attack detected
Alert and Block Nodes
Application Plane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

Figure 3: Research Framework

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
• If the value of entropy exceeds the threshold limit, then it is considered a DDOS traffic.
• 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.

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

Table 2: Detection Specifications

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: 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 6: Detection
• Print block status of Ip and port id of nodes.
• Print time of the suspected attack
• Remove/Update Flow rules.
• Block requests from that IP

3.3. Experimental Design

The simulation is conducted on Mininet 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 [27]. Then the specific tools, i.e., putty, 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. 4. 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 Figure 4 have been made to test the validity of the solution against DDOS traffic using a centralized SDN controller known as 'POX' .

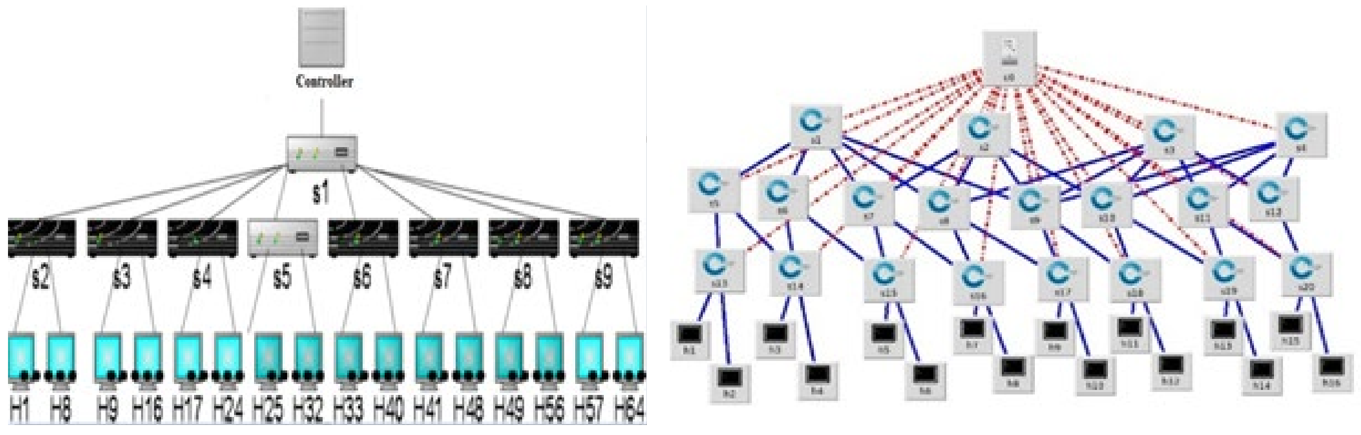

Figure 4: Network Topologies

Scapy 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 foods of variable types under UDP, by manipulating different attack features. Lastly the proposed detection algorithm is also tested on ICMP ping flood attacks under given scenarios [28].

3.5. Testbeds

The testbed is created to validate all possible attack situations under UDP, and ICMP ping flood attacks. In experimentation, these DDoS attacks are performed by changing their patterns in the forwarding properties like bytes sent, duration, and by changing the number of nodes to increase or decrease the flood to make several 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, multiple victims are targeted by multiple malicious nodes depending on the type of topology to produce specific loads while splitting the attacks into groups of destinations. Table 3 describes the parameters used for the testbed in detail [29].

Parameters	Specifications
Type of DDoS Flood	UDP, ICMP
Attack generation	Single and multiple destinations
Tool to Formulate the Attacks	Scapy
Simulation Topologies	Tree Topology
SDN Architecture	Centralized
Controller	Pox
Simulator	Mininet

Table 3: Testbed Specifications

4. Results

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

4.1. Detection Accuracy

In DDoS attacks, numerous attacking hosts transmit the malicious traffic to a single location which 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.

Patter 1		Pattern 2	
Normal traffic characteristics			
Payload	18 bytes	Payload	18 bytes
Packets sent	7	Packets sent	4
Interval (s)	0.2	Interval (s)	0.1
Rate of traffic (packets/second)	5	Rate of traffic (packets/second)	10
Rate of flow (flow per second)	0.6	Rate of flow (flow per second)	2.5
Single destination attack characteristics			
Payload	-	Payload	-
Number of packets sent	1	Number of packets sent	1
Interval (s)	0.08	Interval (s)	0.08
Rate of traffic (packets/second)	12.5	Rate of traffic (packets/second)	12.5
Rate of flow (flow per second)	12.5	Rate of flow (flow per second)	12.5
Multiple destination attack characteristics			
Payload	-	Payload	-
Number of packets sent	1	Number of packets sent	1
Interval (s)	0.03s	Interval (s)	1
Rate of traffic (packets/second)	33	Rate of traffic (packets/second)	33.3
Rate of flow (flow per second)	33	Rate of flow (flow per second)	33.3

Table 4: Normal and Attack Traffic Patterns

4.1.1. Single-Victim Attacks

The detection analysis performed in single destination attacks uses distinct patterns, as discussed above in section. 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 by attaining maximum accuracy, analyzing a sudden change in entropy values without involving other validation parameters like flow initiation and specification to validate the attack flow. 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 98.75% by involving the other two parameters of detection algorithm. 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.

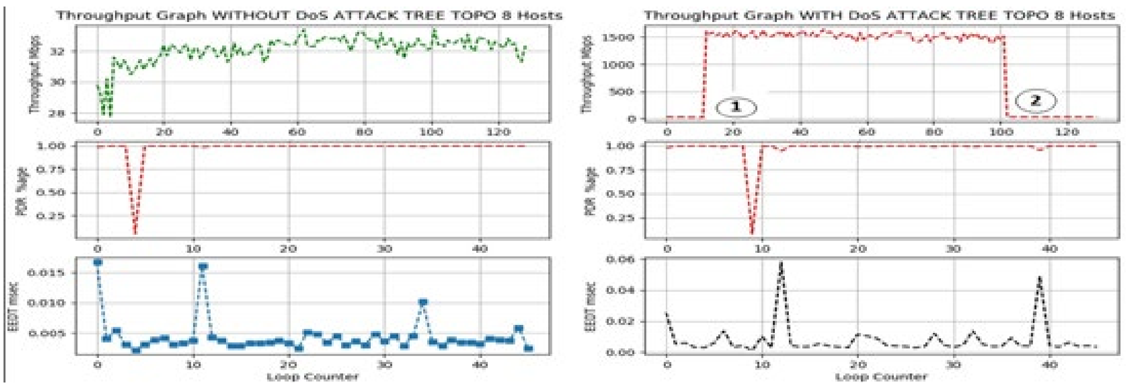

Figure 5: Network Load in Before Detection and After Mitigation of DDoS Attacks.

4.1.2. Multi-Victim 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. In these novel attack conditions, the algorithm still achieved an up to mark 96.25% accuracy by applying flow initiation and specification parameters. This sudden downfall in detection accuracy is happened because choosing a suitable threshold is complex, and it is challenging to establish a margin that fits both flow types (normal and attack flow).

4.2. Detection Time

The intended goal of this investigation is to identify DDoS floods as early as possible. It is detected that the detection delay lowers as the attack load rises since the more traffic there is, the faster the packet sample window will be captured as depicted in Fig 7(a) and (b). It is observed that, the DDoS traffic is detected and mitigated by controller in single-destination attacks under an average of 14.08 seconds in all proposed single victim test scenarios. In multi-destination attacks, the average detection time is recorded to 20.09 seconds as well. The sudden delay of 6.01 seconds as compared to single destination attacks case is due to 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 6 (a) and (b) illustrate the average detection delay in single and multi-destination attacks.

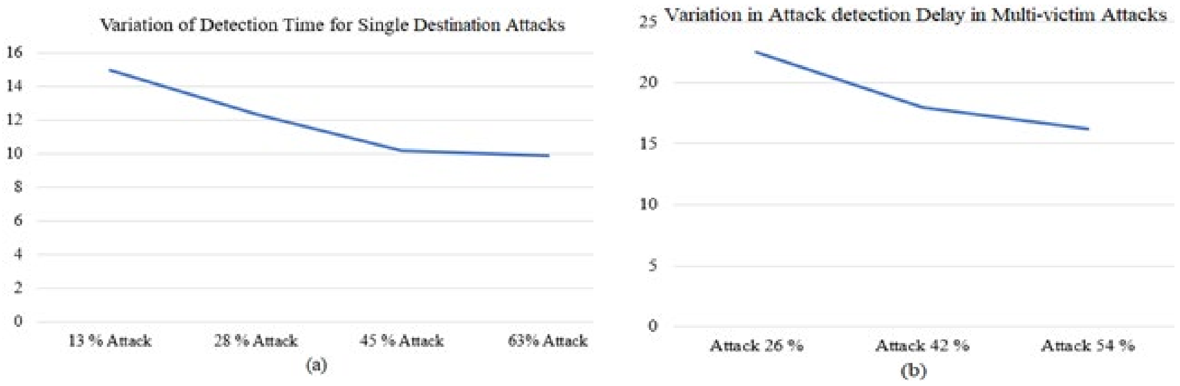

Figure 6: Detection Delay in Single vs Multiple Victim Attacks

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 5.

	[20]	[24]	[25]	[29]	[30]	Proposed work
Techniques	Flow Rate	PCA	Entropy	Entropy	Entropy	Entropy, Flow Initiation & Specification
Centralized Architecture	✓	✓	✓	✓	✓	✓
Attack Traffic	*UDP*	UDP	UDP	UDP	UDP	UDP, 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

Table 5: Comparison of Current Work with Recent Contributions

5. Conclusion

Distributed Denial-of-service (DDoS) attacks are notorious for overwhelming networks by exhausting resources like bandwidth and computational capacity. In most cases, attackers bombard network services with a vast amount of illegitimate traffic, frequently utilizing botnets. Nevertheless, in the context of software-defined networking (SDN), achieving a successful attack does not necessarily require an extensive flow of malicious traffic. Rather, the attack flow can be dispersed to evade detection measures while still impacting the controller and switches. To counteract these attacks, SDN-based DDoS detection methods must be highly scalable, as attackers continuously develop new strategies to circumvent detection. Furthermore, detection latency must be minimal to allow for sufficient time to implement a mitigation process. This study's primary objective was to develop a robust and lightweight solution for early detection of various DDoS attacks. We incorporated resilience into the SDN controller, enabling it to detect and mitigate malicious DDoS activity based on network traffic variations. The proposed solution achieved a high detection rate of 98.75% with an average delay of 14.08 seconds under diverse attack patterns, targeting a single victim within the network. To further examine the impact of DDoS attacks, we conducted multi-victim attacks, where the algorithm maintained overall accuracy between 96.25% and experienced an average delay of 20.9 seconds for mitigation, albeit with a slight performance decline. The reduced performance of the algorithm can be attributed to the attack bearing similarities to regular network traffic. Early detection is confirmed by the approach's utilization of only 150 packets, demonstrating that an attack can be identified before it begins to compromise the network. In future work, we aim to implement this methodology in detecting low-rate attacks in Internet of Things (IoT) contexts.

Author Contribution

Furqan Ahmad[1*] served as the main author, contributing to problem definition, research methodology and results formulation. Maham Saleem1 and Ubaid Ur Rehman2 contributed equally on analysis, proof of concept, Testbed formulation and verification tasks.

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.

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 (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 base paper. The source code and research contribution done in this work is available on request to corresponding author.

Declarations

Conflict of Interest

No conflict of interest found between authors on related contributions

Ethical Approval

The article does not contain any unethical studies with human participants or animals performed by the authors.

References

1. Anerousis, N., Chemouil, P., Lazar, A. A., Mihai, N., & Weinstein, S. B. (2021). The origin and evolution of open programmable networks and SDN. IEEE Communications Surveys & Tutorials, 23(3), 1956-1971.
2. Wazirali, R., Ahmad, R., & Alhiyari, S. (2021). SDN-openflow topology discovery: an overview of performance issues. Applied Sciences, 11(15), 6999.
3. Benzekki, K., El Fergougui, A., & Elbelrhiti Elalaoui, A. (2016). Software-defined networking (SDN): a survey. Security and communication networks, 9(18), 5803-5833.

4. Montazerolghaem, A. (2021). Software-defined load-balanced data center: design, implementation and performance analysis. Cluster Computing, 24(2), 591-610.
5. Hodo, E., Bellekens, X., Hamilton, A., Tachtatzis, C., & Atkinson, R. (2017). Shallow and deep networks intrusion detection system: A taxonomy and survey.
6. Alsaeedi, M., Mohamad, M. M., & Al-Roubaiey, A. A. (2019). Toward adaptive and scalable OpenFlow-SDN flow control: A survey. IEEE Access, 7, 107346-107379.
7. "Specification, O. S. (2013). Open networking foundation. Version ONF TS-015, 1(3), 1-164."
8. Hande, Y., & Muddana, A. (2021). A survey on intrusion detection system for software defined networks (SDN). In Research Anthology on Artificial Intelligence Applications in Security (pp. 467-489). IGI Global.
9. "Open vSwitch."
10. Son, J., & Buyya, R. (2018). A taxonomy of software-defined networking (SDN)-enabled cloud computing. ACM computing surveys (CSUR), 51(3), 1-36.
11. Montazerolghaem, A., Moghaddam, M. H. Y., & Leon-Garcia, A. (2017). OpenSIP: Toward software-defined SIP networking. IEEE Transactions on Network and Service Management, 15(1), 184-199.
12. Assefa, B. G., & Özkasap, Ö. (2020). RESDN: A novel metric and method for energy efficient routing in software defined networks. IEEE Transactions on Network and Service Management, 17(2), 736-749.
13. Eliyan, L. F., & Di Pietro, R. (2021). DoS and DDoS attacks in Software Defined Networks: A survey of existing solutions and research challenges. Future Generation Computer Systems, 122, 149-171.
14. Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. Computer, 50(7), 80-84.
15. "DDoS attack that disrupted internet was largest of its kind in history, experts say | Hacking the Guardian [2022].
16. "DDoS report Q3 2019 | Securelist."
17. Hathaliya, J., Sharma, P., Tanwar, S., & Gupta, R. (2019, December). Blockchain-based remote patient monitoring in healthcare 4.0. In 2019 IEEE 9th international conference on advanced computing (IACC) (pp. 87-91). IEEE.
18. Cajas, C. D., & Budanov, D. O. (2021, January). Mitigation of Denial of Service Attacks Using OpenDaylight Application in Software-Defined Networking. In 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus) (pp. 260-265). IEEE.
19. Mousavi, S. M., & St-Hilaire, M. (2018). Early detection of DDoS attacks against software defined network controllers. Journal of Network and Systems Management, 26, 573-591.
20. "Ahuja, N., & Singal, G. (2019, December). DDoS attack detection & prevention in SDN using OpenFlow statistics. In 2019 IEEE 9th International Conference on Advanced Computing (IACC) (pp. 147-152). IEEE.
21. Aluru, S. (2018, August). Jaypee Institute of Information Technology University, University of Florida. College of Engineering. In 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) (pp. 2-4).
22. Rahman, O., Quraishi, M. A. G., & Lung, C. H. (2019, July). DDoS attacks detection and mitigation in SDN using machine learning. In 2019 IEEE world congress on services (SERVICES) (Vol. 2642, pp. 184-189). IEEE.
23. Gao, S., Peng, Z., Xiao, B., Hu, A., Song, Y., & Ren, K. (2020). Detection and mitigation of DoS attacks in software defined networks. IEEE/ACM Transactions on Networking, 28(3), 1419-1433.
24. Institute of Electrical and Electronics Engineers. (2018). IEEE International Conference on Communications (ICC): proceedings : Kansas City, MO, USA, 20 -24 May.
25. Institute of Electrical and Electronics Engineers. (2019) 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU).
26. Batool, S., Zeeshan Khan, F., Qaiser Ali Shah, S., Ahmed, M., Alroobaea, R., Baqasah, A. M., & Ahsan Raza, M. (2022). Lightweight Statistical Approach towards TCP SYN Flood DDoS Attack Detection and Mitigation in SDN Environment. Security and Communication Networks, 2022.
27. "Mininet: An Instant Virtual Network on Your Laptop (or Other PC) - Mininet."
28. "Scapy."
29. Mousavi, S. M., & St-Hilaire, M. (2018). Early detection of DDoS attacks against software defined network controllers. Journal of Network and Systems Management, 26, 573-591.
30. Mishra, A., Gupta, N., & Gupta, B. B. (2021). Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX controller. Telecommunication systems, 77, 47-62.