American Academic Scientific Research Journal for Engineering, Technology, and Sciences

ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

http://asrjetsjournal.org/

# Implementation of Entropy-Based DDoS Attack Detection Method in Different SDN Topologies

Mayadah A. Mohsin[a*], Ali H. Hamad[b]

[a]*University of Baghdad, Department of Information and Communication Engineering, Baghdad 10016, Iraq*

[b]*University of Baghdad, Department of Information and Communication Engineering, Baghdad 10070, Iraq*

[a]*Email: mayadahabdalmohsin@gmail.com*

[b]*Email: ahamad@kecbu.uobaghdad.edu.iq*

## Abstract

Software-Defined Network (SDN) brings a lot of advantages to the world of networking through its flexibility and centralized management, but this centralized control makes SDN susceptible to different types of attacks. The Distributed Denial of Service (DDoS) attack is one of the most commonly used attacks since it is relatively easy to deploy and very successful at harming any network, thus researchers are mostly focusing on this type to discover an effective defense mechanism against it. This work investigates the impact of a DDoS attack on an SDN environment and proposes a light and effective method for detecting this attack at an early stage based on calculating the entropy of destination network traffic IP addresses. The proposed method proved its ability to detect the DDoS attack with minimum detection time in three different SDN network topologies which are single, linear, and multi-controller. RYU controller has been used with Mininet emulator and OpenFlow protocol.

***Keywords:*** Software-Defined Network (SDN); Distributed Denial of Service (DDoS); Entropy; RYU controller.

## 1. Introduction

The need for a re-programmability feature in devices has become essential, especially in the networking domain, because of the enormous growth in the number of devices that use the internet even in daily life services. Having redesign flexibility makes any working system a continuous improvement, which is a great feature in the domain of networking offered by the SDN technology through splitting the controlling entity from the forwarding entity, which in turn makes the forwarding devices like switches able to be reprogrammed at any time [1]. The ability to have decisions removed from SDN switches in this structure, and all the management becomes the controller's responsibility. Figure 1 shows the architecture of the SDN system.

------------------------------------------------------------------------

* Corresponding author.

This centralized control gives a great benefit to the SDN network in terms of ease of management. On the other hand, the centralized controller means a single point of failure that can shut down all the network once it fails [2]. However, attackers will find this to be a perfect target.

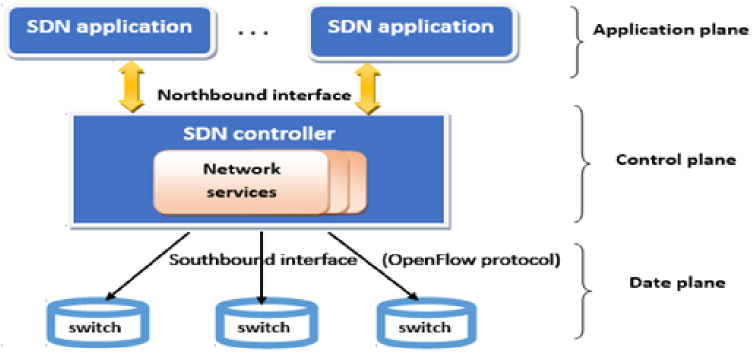


**Figure 1:** SDN architecture concept

There are several types of attacks on SDN networks, like ARP Spoofing Attack, traffic sniffing, brute force, Distributed Denial of Service ("DDoS"), etc. One of the common attacks that threaten any network and has a dangerous effect on SDN is the Distributed Denial of Service (DDoS). It is a DoS attack launched from multiple compromised hosts called zombies or botnets. A DDoS attack is an attempt to consume network or host resources by sending a large number of malicious packets to a single host to make it unavailable to offer its services [3]. DDoS attacks can be divided into three main categories [4] volume-based attacks, protocol-based attacks and application layer-based attacks. Table 1 shows a comparison between these types. DDoS attacks can be divided into three main categories [4]: volume-based attacks, protocol-based attacks, and application layer-based attacks. Table 1 shows a comparison between these types. With the global view offered by the central controller in SDN networks, DDoS attacks can be detected early by a continuous statistics collection from switches using some development in the controller application. Many detection methods are used to detect and mitigate DDoS in SDN, such as machine learning, entropy, intrusion detection systems, and flow statistic monitoring. One of the lightweight DDoS attack detection techniques is entropy. It can detect the attack in its early stages because the small variation between normal and malicious traffic can affect the entropy value [5].

**Table 1:** DDOS attack types

| **DDoS Attack type** | **Target** | **Rate** | **Examples** |
|---|---|---|---|
| Volume-based attacks | consume the bandwidth | bits per second (Bps). | ICMP floods, UDP floods, other spoofed packet attacks |
| Protocol-based attacks | Consume server resources | Packets per second | SYN floods, fragmented packet attacks, Ping of death, Smurf attack |
| Application layer-based attacks | crash the web server | Requests per second (Rps). | Zero-day attack, Slowloris, HTTP flood |

The main contribution of this work can be set in three directions: First is observe the effect of a DDoS attack on the SDN network with single, linear, and multi-controller network topologies using the RYU controller. The second is implement a lightweight method to detect this attack in its early stages before it overwhelms the controller, depending on specific statistical traffic parameter variation, which is entropy. The third is show the effectiveness of the used detection method at different attack rates through simulations.

The rest of this paper organized as follows: section two present some related works, section three discusses DDoS attack in SDN network, section four gives an explanation of the entropy method for DDoS detection, section five shows the implementation of the proposed method, section six present the results and discussion, section seven introduces a comparison with some related work, finally a conclusion is introduced in section eight.

## 2. Related works

Due to its high impact on traditional and SDN networks, DDoS attack detection techniques has attracted many researchers. Dhaliwal and his colleagues [6] developed a statistical algorithm to detect two types of DDoS attack which are SYN flood and HTTP flood using POX SDN controller. A fixed size window has been used and depend on source IP address filtering to mitigate the attacker in both types. Ahalawat and his colleagues [7] proposed a detection and mitigation of DDoS UDP flooding attacks in an SDN environment using the RYU controller. An entropy-based solution was suggested in this work. Pandikumar and his colleagues [8] proposed SDN solution for detecting a DDoS attack in its early stages in multi-controller system using POX controller. The approach is based on the entropy variation of the destination IP address and can detect the attack within the first 250 packets of malicious traffic attacking a specific host in the SDN. Dehkordi and his colleagues [9] combined machine learning and statistical methods to improve the DDoS attacks detection in SDN networks using only one controller. The proposed method was implemented in floodlight controller and no mitigation technique mentioned. Dennis and Li [10] proposed statistical and machine learning approaches to detect and mitigate DDoS attack deployed against POX controller. It uses, destination entropy, flow and packets rates and flow duration for the statistical detection method and Random Forest algorithm as the ML detection method. Also, a drop flow mitigation technique suggested in this work. Hong and his colleagues [11] proposed a detection mechanism-based entropy that applied dynamic threshold depending on network state to detect DDoS attack in SDN network environment using OpenDaylight. Abdullah and his colleagues [12] proposed entropy-based DDoS detection technique using POX controller in different network scenarios, the work focus on the effect of increasing the number of controllers on the attack detection. Swami and his colleagues [13] propose a DDoS detection method to defend POX controller by calculating the entropy of incoming packets, the method can detect attack within 125 first packets. Sahoo and his colleagues [14] proposed General Entropy (GE) based DDoS attack detection mechanism, the work also investigate various security issues of SDN and then focus on DDoS threat to the control layer. Bavani and his colleagues [15] proposed an early-stage DDoS attack detection based on the mean entropy, also the work identify the variations in network traffic terms of percentage drop during attack.

## 3. DDoS attack in SDN

The DDoS attack is defined as a denial-of-service attack (DoS) that is launched from a group of compromised hosts called zombies or botnet. The legitimate owner of the attacking host is usually unaware of the malicious script that is running on his device which is pre-installed by the attacker through exploiting the SDN vulnerabilities. [16]. The attacker attempts to transmit a large number of spoofed packets to the controller to be processed, therefore exhausting the controller's resources because with a spoofed packet, each request that arrives at the switch will be forwarded to the controller to make a decision about it since there is no matching rule in the switch flow table for that IP. The controller then creates a new flow rule for this packet as shown in figure 2.

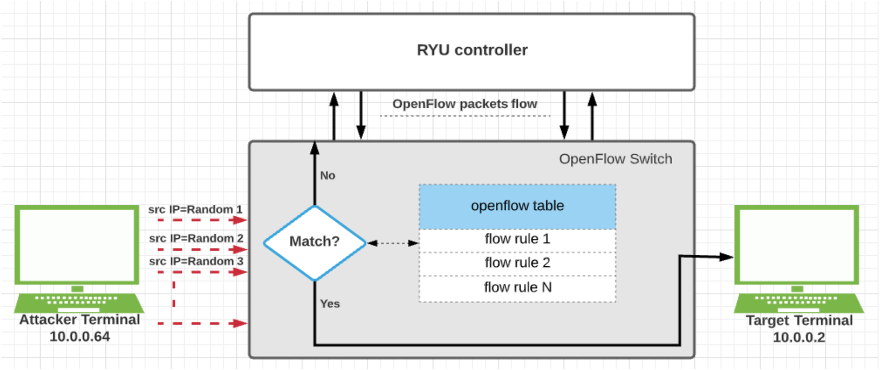


**Figure 2:** Processing the spoofed IP

In this work, a Python script was developed using the Scapy library to create and send several UDP packets per second with spoofed source IP addresses. Which means each packet will have a random source IP address that doesn’t have a matching rule in the switch flow table. If this attack is deployed from various compromised hosts, the load on the controller will increase and probably exceed the overload limit, and the controller will become unavailable for normal traffic processing and eventually fall down, which is a major objective for DDoS attackers.

Several detection techniques have been used to defend the SDN network against DDoS attacks. The Intrusion detection system (IDS) has been widely deployed in SDN as an effective and real-time network traffic monitoring and abnormality detection system with two types: signature-based and anomaly-based IDS. One of the most famous IDS used is Snort [17]. Entropy is another technique to detect traffic abnormalities depending on the randomness of some network features (like source or destination IP address) [18]. Machine learning with several algorithms (like artificial neural networks, decision trees, naive Bayes, SVM, etc.) is also a widely used method that depends on collecting normal traffic features in a data set and learning the algorithm on them to make a decision about the new incoming data stream [19]. sFlow-RT is a real-time monitoring tool used to monitor switches (sFlow agent) traffic statistics of the SDN network through the northbound interface (Restful API) to detect any abnormal behavior [20]. Next section will be a detailed explanation of the entropy-based method since it was the chosen method in this work.

## 4. DDoS detection technique based-entropy variation

It is essential to employ a technique that protects the SDN and prevents the system service from failing. One of the lightweight statistical techniques to secure SDN is entropy. It is a measure of uncertainty or randomness associated with a random variable, which in this work is the destination IP address. Higher randomness will result in higher entropy. Entropy can be calculated using the following equations:

$$P(x) = \frac{Number\ of\ packets\ with\ X\ dest.IP\ address}{Total\ No.of\ packets} \quad (1)$$

$$H(x) = \mathrm{p(x)}.\log_2 \frac{1}{p(x)} \quad (2)$$

Where P(x) is the probability of a destination IP address (x) occurring and H(x) is the entropy of that IP address. In SDN network traffic classification, entropy is used for detecting different kinds of DDoS attacks based on calculating the randomness of IP addresses in a specified period and comparing this value with a pre-defined threshold value to make a decision by the controller if the flow is legitimate or malicious. A DDoS attack could be detected by checking a sample of the flows (window size of W) and calculating their randomness based on the number of IP occurrences. The entropy value is at its minimum when all the traffic is heading to the same destination (which is attack traffic), and it is at its maximum when the traffic is equally distributed to all the possible destinations (which is normal traffic) [21].

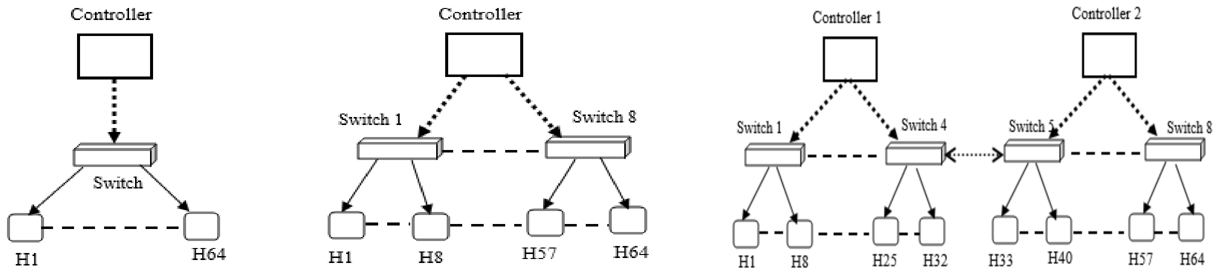
## 5. Proposed system implementation

Entropy-based DDoS detection in SDN networks is proposed in this work with different scenarios, also an evaluation is done through simulation tests for each scenario. The proposed algorithm implementation has been designed and tested under the Linux operating system using an open-source Mininet network emulator and a python-based Ryu controller. The OpenFlow protocol has been used for the SDN system. The monitoring metrics during the evaluation are the variation of the entropy value under different attack rates, the total number of received packets on the victim’s side and controller, and the attack detection time. By measuring these values, the effectiveness, and efficiency of the proposed method has been evaluated, and also the time of the detection has been computed.

### *5.1 Scenarios*

The proposed method has been tested on three types of SDN network topologies to observe the different detection results. First, a single topology was created with one controller, one switch, and 64 hosts (Figure 3-A). The second topology is a linear topology with one controller, eight switches, and 64 hosts (8 hosts per switch) was created (Figure 3-B). Finally, a multi-controller linear topology with 2 controllers, 8 switches, and 64 hosts

(8 hosts per switch) was created (Figure 3-C). These scenarios used RYU as the SDN controller.



(a) Single topology (b)Linear topology (c) multi-controller topology

**Figure 3:** Simulation network scenarios

### *5.2 Traffic generation*

The network traffic is generated using Scapy, which is a packet generator tool that can create any kind of packet using Python code. In all tests, two Scapy programs are running. One is generating normal traffic and the other is generating an attack that sends packets faster than normal traffic. Normal traffic is launched from host one, and attack traffic is started from host 64 with spoofed IP addresses toward host two table 2 illustrates normal and attack traffic specifications. Figure 4 shows the difference between normal and attack traffic.

**Table 2:** Normal and attack traffic specification

| **Traffic parameters** | **Normal** | **Attack** |
|---|---|---|
| Packet type | UDP | UDP |
| Source address | Spoofed Random IP | Spoofed Random IP |
| Destination address | (10.0.0.2) …. (10.0.0.64) | 10.0.0.2 |
| Traffic interval (sec.) | 0.1 | 0.025 |
| Traffic rate (packets/1 sec.) | 3-7 | 9-14 |

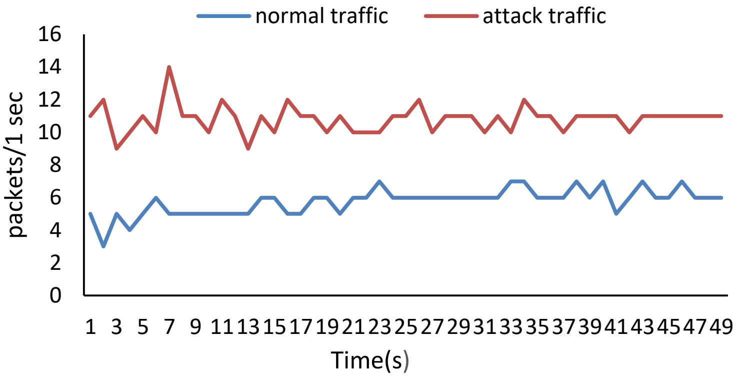


**Figure 4:** Normal vs. Attack packets rate

### *5.3 Method specification*

In this work, a 50-packet window size has been taken for the calculation of the entropy value and the threshold has been selected according to the topology type. The randomness has been measured depending on the destination IP addresses. Algorithm 1 illustrate the detection process using entropy. If the network is running in normal traffic, the randomness in these 50 packets is high, so the entropy is high. While in an attack traffic situation, the randomness drops since the majority of packets are directed to a single host, which means the entropy becomes very low, and if it exceeds the threshold, a DDoS attack detection alarm appears in the Ryu terminal.

**Algorithm1**: Entropy-based Detection Algorithm

```
initiate variables:

counter=0            #counting number of windows below threshold value

packet_count=0       #counting number of packets

sum_entropy =0       #sum of 50 packet entropy value

threshold = Th       #selected value for each topology

Start:

collect packets from switch(s) in a list with their destination IP address
new incoming packet
packet_count=packet_count +1
calculate probability for each destination IP address
if packet-count = =50:
    calculate sum-entropy
    if sum_entropy < Th:
        counter=counter+1
    else:
        counter =0
    if counter = =5:

        DDoS attack detected alert message
```

### *5.4 Threshold selection criteria*

If the entropy value is less than the threshold for five consecutive times, it is considered an attack. Then to have accurate attack detection, an optimal threshold value must be chosen after running different tests on the proposed topologies. To determine the optimal threshold, a set of tests have been implemented to examine how an attack affects the entropy for different topology types and with different attack rates. Controlling the rate of attack is done by running the attack from more than one host. In this work, four attacking hosts were launched gradually against a single victim in order to monitor and compare the effect of increasing DDoS attack intensity on the entropy value. If one host generates a UDP flood attack, then only a max of 14 more packets per second are injected, so for more attack traffic rates, more hosts will launch the attack script used and the entropy value is monitored at each rate.

By observing the minimum entropy value in normal traffic and the maximum in attack traffic in each scenario, and with a confidence interval (which is the difference between the threshold and the maximum attack entropy) to eliminate false negatives, a suitable threshold can be chosen that will be the limit for attack detection. Table 3 illustrates threshold selection based on entropy value. It has been found that one is the best threshold value for single topology and two is the best for linear and multi-controller topologies, as the entropy in single topology is less than in linear, and linear is less than multi-controller.

**Table 3:** Threshold selection depending on entropy value

| **Topology** | **Normal min entropy** | **Attack max entropy** | **Chosen threshold** | **Confidence interval** |
|---|---|---|---|---|
| Single | 1.16 | 0.9 | 1 | 0.1 |
| Linear | 2.1 | 1.92 | 2 | 0.08 |
| Multi-controllers | 2.55 | 1.6 | 2 | 0.4 |

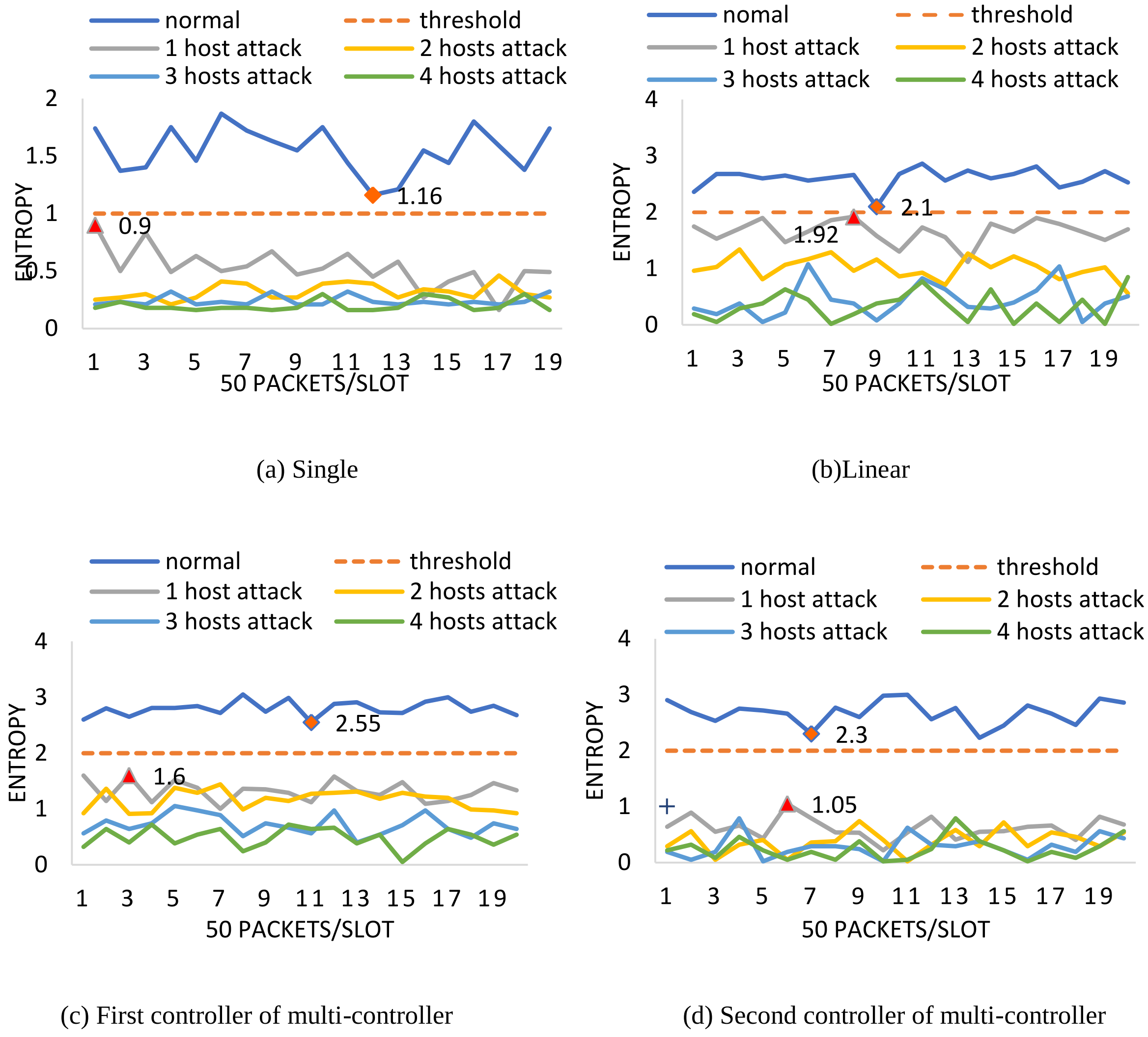
The rate of incoming attack packets could be determined by using the following formula:

$$R = \frac{Pa}{50} \times 100\% \qquad (3)$$

Where Pa is the number of attack packets and 50 is packets window size which is a fixed value. Table 4 shows the attack rate calculation with different topology types. It has been observed that the variation of entropy values requires a change in the threshold value. Also, figure 6 shows the entropy values in the different scenarios and attack rates.

**Table 4:** Attack rate calculation

| Attack rate (pkts/s) | *R* | No. of attackers (zombies) | No. of target |
|---|---|---|---|
| 9-14 | ~25% | 1 | 1 |
| 23-26 | ~50% | 2 | 1 |
| 34-38 | ~75% | 3 | 1 |
| 45-50 | ~100% | 4 | 1 |



(a) Single

(b)Linear

(c) First controller of multi-controller

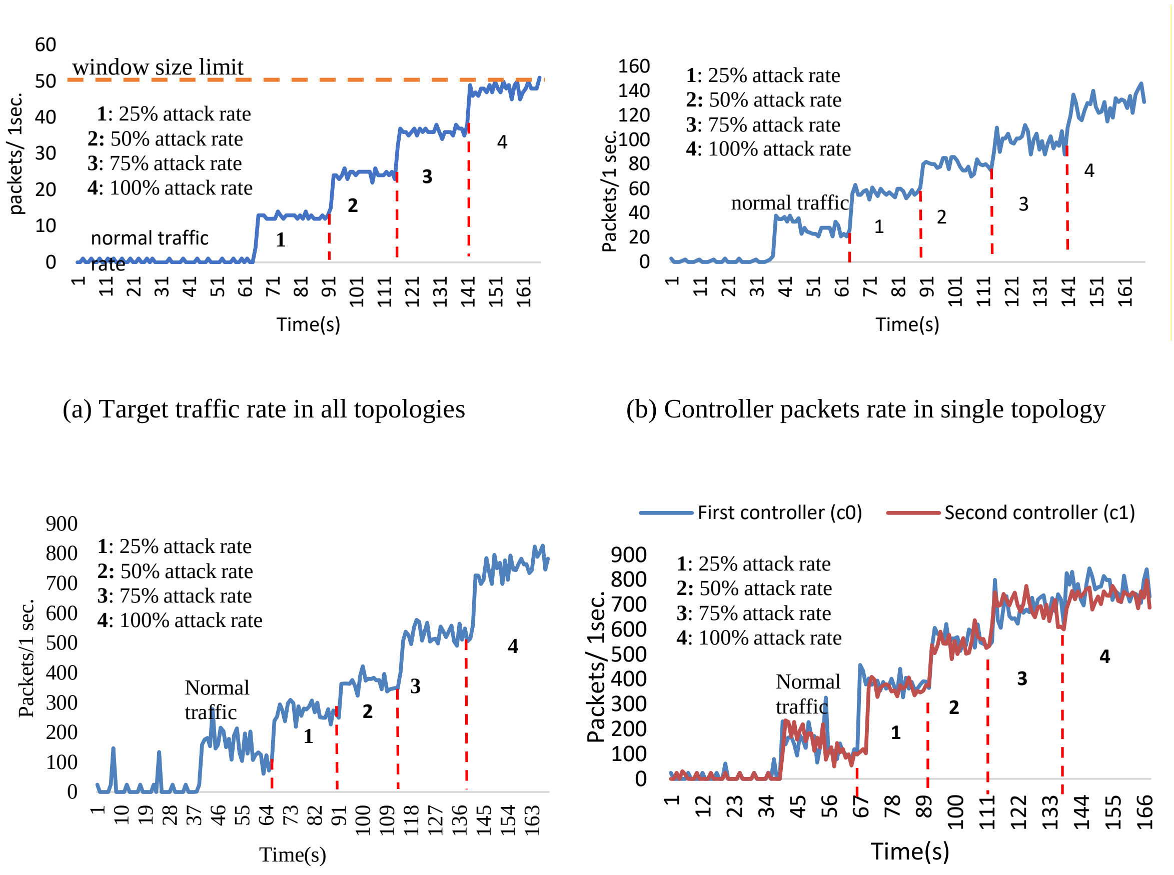
(d) Second controller of multi-controller

**Figure 1:** Entropy variation and threshold selection during normal and attack traffic

## 6. Results and discussion

In this section, the results of different simulation scenarios that have been implemented with entropy method will be evaluated depending on the traffic rate during normal and attack events, and the attack detection time. Measuring these values will evaluate the effectiveness and efficiency of the proposed method. By analyzing the traffic rate variations at the target and the controller, the effect of a DDoS attack on the network can be observed as the total number of received packets per second on the victim and the controller side, shows the spikes of traffic during the attack and its differences according to the topology type.

Different attack rates are applied to each topology type. Attack from one host with approximately 14 packets/s first applied, then from two hosts it becomes ~25 packets/s and up to 4 hosts it will reach ~50 packets/s. The target packets rate remained the same when changing topology type but increased when increasing the attack rate (figure 7-a), but for the controller packets rate, there is a huge difference between single topology and linear with one and two-controller topologies. In single topology, the maximum attack rate (four hosts attack) can cause the controller to have up to ~160 packets/s (figure 7-b), but in a linear and multi-controller topology, the packets rate reaches a to ~800 packets/s at the four hosts attack rate (figure 7-c, d), which is a huge consuming for the controller processing and resources caused by the DDoS attack. This means that this kind of attack mostly effects the controller resources to make it fail, and this effect is more in linear and multi-controller topologies. Observing traffic is done using a monitoring tool named Wireshark which is used in this work to monitor the packet rate of target and controller in real-time. Accurate detection is provided in the RYU controller terminal depending on the selected threshold after calculating five consecutive windows of entropy value less than the threshold in each topology (figure 8). The minimum attack detection time is one of the goals of this research. The work aims to find a solution to detect an attack at its early stages. Figure 9 illustrates the average attack detection time under each attack traffic load and for different topology types. As shown in the figure, the increase in the attack load causes a reduction in the detection time. This is because when the traffic rate increases, the packet sampling window will be collected faster and, therefore, the detection will also become faster. It should be mentioned that in the multi-controller topology, the target (host 2) is located in the first controller (c0) domain, so it can be seen that the detection time in it is less than in the second controller (c1) because the proposed method depends on the frequency of appearance of the destination IP, which is (10.0.0.2) in this work.



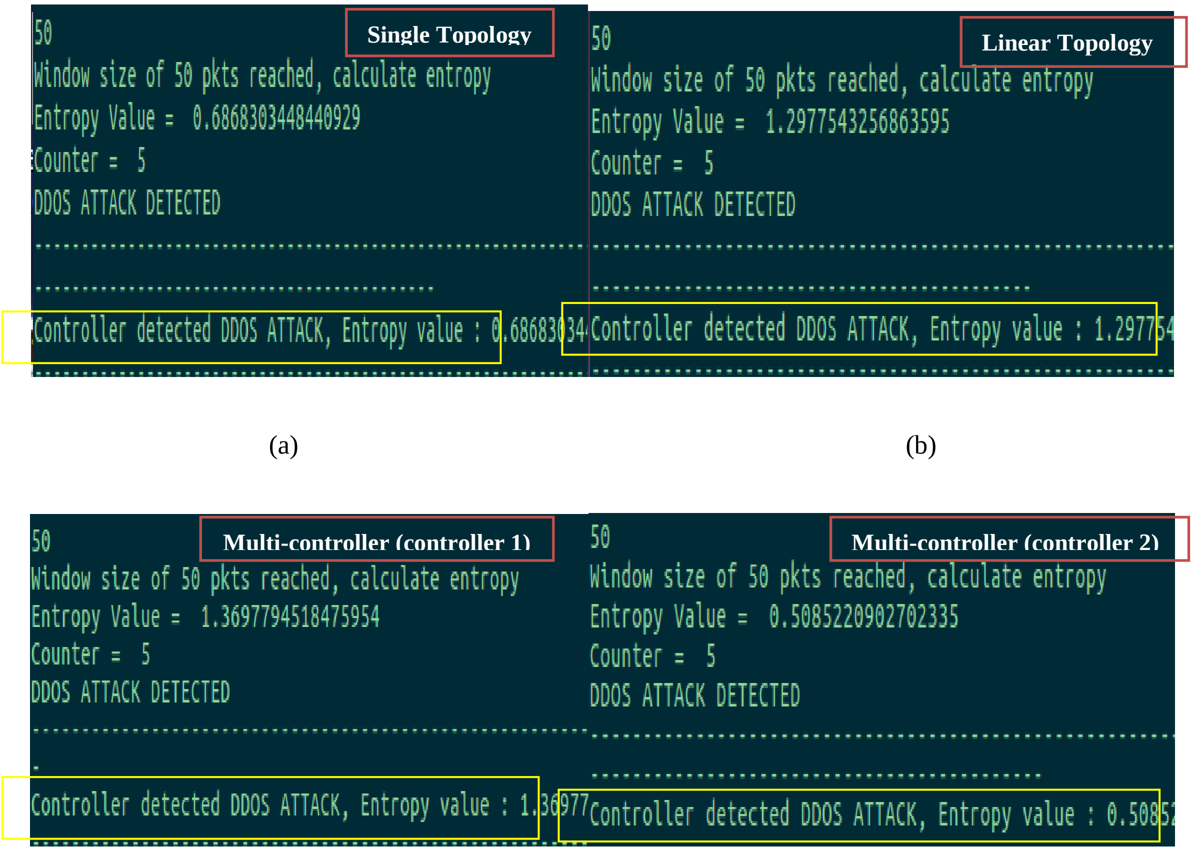
(a) Target traffic rate in all topologies

(b) Controller packets rate in single topology

(c) Controller packets rate in linear topology

(d) Controller packets rate in multi-controller topology

**Figure 7:** Packets rate in target and controller for different network topologies



Single Topology

```
50
Window size of 50 pkts reached, calculate entropy
Entropy Value =  0.6868303448440929
Counter =  5
DDOS ATTACK DETECTED
-----------------------------------------------------------
------------------------------------------
Controller detected DDOS ATTACK, Entropy value : 0.68683034
-----------------------------------------------------------
```

(a)

Linear Topology

```
50
Window size of 50 pkts reached, calculate entropy
Entropy Value =  1.2977543256863595
Counter =  5
DDOS ATTACK DETECTED
-------------------------------------------------------------
-----------------------------------------------
Controller detected DDOS ATTACK, Entropy value : 1.297754
-------------------------------------------------------------
```

(b)

Multi-controller (controller 1)

```
50
Window size of 50 pkts reached, calculate entropy
Entropy Value =  1.3697794518475954
Counter =  5
DDOS ATTACK DETECTED
-----------------------------------------------------------
-
Controller detected DDOS ATTACK, Entropy value : 1.36977
-----------------------------------------------------------
```

(c)

Multi-controller (controller 2)

```
50
Window size of 50 pkts reached, calculate entropy
Entropy Value =  0.5085220902702335
Counter =  5
DDOS ATTACK DETECTED
-------------------------------------------------------------
-----------------------------------------------
Controller detected DDOS ATTACK, Entropy value : 0.5085
```

(d)

**Figure 8:** DDoS detection at RYU controller terminal in all topologies

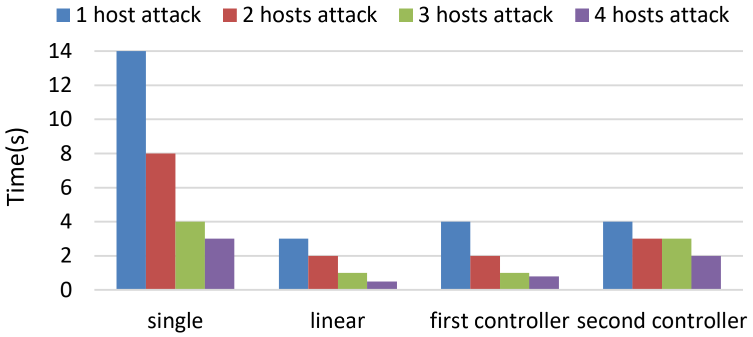


**Figure 9**: Attack detection time in all tests

## 7. Comparison with similar works in literature

Comparing our work with the most similar works in literature, it has been found that:

In [12] authors implemented the entropy method on single-topology and linear with multi-controller's topology using the same threshold for all network types, whereas we use a threshold value for each network topology, which are single, linear, and linear with multi-controllers. Also, the work does not mention the variation of entropy in each network according to changing the attack rate, but in our work, we show in detail the effect of increasing the attack rate on the entropy value in all topology types and choosing the optimal threshold according to the results. In [13] the system uses a linear topology network only to implement the entropy method with just 48 hosts, but in our work, we check the effectiveness of entropy in three types of network topology to prove the method's detection capability in a more scale network. The work chooses a window size of 25, this is half of the chosen window size in our work which is 50 because we found after a deep study of the literature that 50 is the optimum window size. The small window size increases the false positives because it makes the difference in entropy between normal and attack traffic very small, but 50 is a fair value to eliminate the false positives. Moreover, the work only mentions a 25% attack rate, while in our work we implement four attack rates: 25%, 50%, 75%, and 100% to identify the effect of an increasing attack rate on the detection method. In [14] the proposed method has been implemented in linear topology with a single controller, which is POX controller, it uses an attack rate of up to 80% and does not mention the 100% attack rate, also the effect of increasing the attack rate on the controller traffic is not discussed, in our work, we mention the effect of all attack rate on the controller and target traffic and in addition to their network we address the single and multiple controller topologies as we mentioned before. In [15] the work uses one POX controller with 8 switches and 49 hosts as the simulation network, also it addressed just two attack rates 25% and 50% to prove the method's successful detection.

In our proposed system, we implement the method with three different network types consisting of 64 hosts and with four attack rates to prove the effectiveness of the entropy technique in the detection of DDoS attack. Furthermore, all these works use the POX controller as the SDN controller for system implementation, instead, we choose the RYU controller for our work because we found that it has better performance in all topology types of SDN network and also it supports all OpenFlow versions [22].

## 8. Conclusion

Although SDN offers great networking architecture flexibility by providing central management and programmability to the network, it allows attackers to have the perfect target (the central controller) to bring the system down. In this work, an entropy-based DDoS attack detection approach has been built to secure the SDN network, with the primary goal of detecting attacks fast.

It has been found that the method succeeded in detecting the attack with a minimum detection time in three different network topologies by choosing the first 250 packets to count under the threshold value to declare an attack happening. Also, the RYU controller was found to be the most affected by this attack after analyzing the impact of DDoS in each scenario, but adding more controllers minimized the attack detection time. The proposed detection method proved to be able to successfully capture the drop in entropy with fast and accurate detection at 25%, 50%, and 75% attack rates. At a 100% attack rate, the 50-packet window size will sometimes be full of attack traffic that is destined for a single host, which will drop the entropy to zero at some points and lead to a small percentage of detection errors. Increasing attack rate minimized the detection time because the packet sampling window will be collected faster. Since this work concentrates on the detection of attacks in the early stages, it focuses on low packet-rate attacks that do not eliminate the normal traffic completely, so the

100% rate can be considered a high packet rate attack because it fills the window size with only attack traffic at some time intervals.

Finally, it has been found that the proposed method is effective in detecting DDoS attacks of type UDP Flood packets that are launched from one or multiple attack hosts targeting a single victim.

## 9. Recommendations

For future work, we recommend implementing a mitigation technique to eliminate the attack effect after detection. Also, another detection technique could be used, such as machine learning, Furthermore, adding more SDN controllers in a multi-controller topology could be tested.

## References

[1] J. C. C. Chica, J. C. Imbachi, J. F. B. Vega, “Security in SDN: A comprehensive survey,” Elsevier, Journal of Network and Computer Applications, vol. 159, 2020.

[2] S. Wang, K. Gomez, K. Sithamparanathan, M. R. Asghar, G. Russello, and P. Zanna, “Mitigating DDoS attacks in SDN-BASED IOT Networks Leveraging secure control and data Plane Algorithm,” Applied Sciences, vol. 11, no. 3, p. 929, 2021.

[3] N. Ravi and S. M. Shalinie, "Learning-Driven Detection and Mitigation of DDoS Attack in IoT via SDN-Cloud Architecture," IEEE Internet of Things Journal, vol. 7, no. 4, pp. 3559-3570, April 2020,

[4] S. Bendale, C. Dharmadhikari, S. Kulkarni, S. Temkar, “A Study of DDoS Attacks in Software Defined Networks,” International Research Journal of Engineering and Technology (IRJET), vol. 6, no. 12, Dec 2019.

[5] L. Zhou, M. Liao, C. Yuan, and H. Zhang,” Low-Rate DDoS Attack Detection Using Expectation of Packet Size,” Wiley, Security and Communication Networks, vol. 2017, 2017, pp. 1-15.

[6] S. Dhaliwal, “Detection and mitigation of syn and http flood ddos attacks in software defined networks,” M.S. thesis College of Eng. and Sc Ryerson Univ., Toronto, 2017.

[7] A. Ahalawat, S. S. Dash, A. Panda, and K. S. Babu, “Entropy based ddos detection and mitigation in OpenFlow enabled sdn,” International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), 2019, pp. 1-5.

[8] T. Pandikumar, F. Atkilt, A.K Hassen,” Early Detection of DDoS Attacks in a Multi-Controller Based SDN,” International Journal of Engineering Science and Computing, Vol. 7, no.6 ,2017.

[9] A. B. Dehkordi, M. R. Soltanaghaei & F. Z. Boroujeni,” The DDoS attacks detection through machine learning and statistical methods in SDN,” springer, the journal of supercomputing, vol.77, no.3, 2021.

[10] J. R. Dennis, and X. Li, “Machine-Learning and Statistical Methods for DDoS Attack Detection and Defense System in Software Defined Networks,” M.S. thesis College of Eng. and Sc Ryerson Univ., Toronto, 2018.

[11] G. Hong, C. Lee and M. Lee, "Dynamic Threshold for DDoS Mitigation in SDN Environment," Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2019, pp. 1-7.

[12] M. Z. Abdullah, N. A. Al-awad, F. W. Hussein,” Implementation of entropy-based distributed denial of service attack detection method in multiple pox controllers,” Review of Computer Engineering Studies,2019, Vol. 6, No. 2, pp. 29-38.

[13] R. Swami, M. Dave and V. Ranga, "Defending DDoS against Software Defined Networks using Entropy," 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), 2019, pp. 1-5.

[14] K. S. Sahoo, B. Sahoo, M. Vankayala and R. Dash, "Detection of Control Layer DDoS Attack using Entropy metrics in SDN: An Empirical Investigation," Ninth International Conference on Advanced Computing (ICoAC), 2017, pp. 281-286.

[15] K. Bavani, M. P. Ramkumar and E. Selvan G.S.R., "Statistical Approach Based Detection of Distributed Denial of Service Attack in a Software Defined Network," 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 380-385.

[16] S. Ali, M. K. Alvi, S. Faizullah, M. A. Khan, A. Alshanqiti and I. Khan, "Detecting DDoS Attack on SDN Due to Vulnerabilities in OpenFlow," International Conference on Advances in the Emerging Computing Technologies (AECT), 2020, pp. 1-6.

[17] O. Tayfour, M. Marsono,” Collaborative Detection and Mitigation of Distributed Denial-of-Service Attacks on Software-Defined Network.” springer, Mobile Networks and Applications, vol.25, 2020, pp. 1-10.

[18] M. Aladaileh, M. Anbar, I. H. Hasbullah, Y. K. Sanjalawe and Y. Chong,” Entropy-Based Approach to Detect DDoS Attacks on Software Defined Networking Controller,” Computers, Materials & Continua, vol. 69, no.1, pp. 373–391, 2021.

[19] O. Rahman, M. A. G. Quraishi and C. Lung, "DDoS Attacks Detection and Mitigation in SDN Using Machine Learning," IEEE World Congress on Services (SERVICES), 2019, pp. 184-189.

[20] B. H. Lawal and A. T. Nuray, "Real-time detection and mitigation of distributed denial of service (DDoS) attacks in software defined networking (SDN)," Signal Processing and Communications Applications Conference (SIU), 2018, pp. 1-4.

[21] N. M. Abdel-Azim, S. F. Fahmy, M. A. Sobh, A. M. Bahaa-Eldin, “A hybrid entropy-based DoS attacks detection system for software defined networks (SDN): A proposed trust mechanism,” Elsevier, Egyptian Informatics Journal, vol. 22, no. 1, 2021, pp. 85-90.

[22] A. Jehad, S. Lee and B. Roh, “Performance Analysis of POX and Ryu with Different SDN Topologies,” Proceedings of the 2018 International Conference on Information Science and System, 2018.