IMPROVING DISTRIBUTED DENIAL OF SERVICE (DDOS) DETECTION USING ENTROPY METHOD IN SOFTWARE DEFINED NETWORK (SDN)

Dani Prasetiawan[1], Maman Abdurohman[2], and Fazmah Arif Yulianto[3]

[1,2,3]School of Computing, Telkom University
Jl. Telekomunikasi No. 01, Bandung 40257, Indonesia
[1]danipras@students.telkomuniversity.ac.id; [2]abdurohman@telkomuniversity.ac.id;
[3]fazmaharif@telkomuniversity.ac.id

Received: 14th August 2017/ **Revised:** 27th October 2017/ **Accepted:** 27th October 2017

Abstract **–** This research proposed a new method to enhance Distributed Denial of Service (DDoS) detection attack on Software Defined Network (SDN) environment. This research utilized the OpenFlow controller of SDN for DDoS attack detection using modified method and regarding entropy value. The new method would check whether the traffic was a normal traffic or DDoS attack by measuring the randomness of the packets. This method consisted of two steps, detecting attack and checking the entropy. The result shows that the new method can reduce false positive when there is a temporary and sudden increase in normal traffic. The new method succeeds in not detecting this as a DDoS attack. Compared to previous methods, this proposed method can enhance DDoS attack detection on SDN environment.

Keywords: Software Defined Network (SDN), Distributed Denial of Service (DDoS), detection, entropy

I. INTRODUCTION

OpenFlow controller is the switch control plane that is implemented in different machines. It is used as a communication channel between the switch and the controller. The switch traffic control system acts according to the flows installed by the controller (Dillon & Berkelaar, 2014).

There are various Denial of Service (DoS) attack schemes that have been used to degrade the availability of targeted attacks (Siregar, 2013; Kandoi & Antikainen, 2015; Oktian, Lee, & Lee, 2014). These attacks can be classified into two levels. There are application level and network level. The objectives of application level attacks are to misuse the software and exhaust resources for processing further requests. These kinds of attacks are more difficult to detect on the network level because of unclear deviation between attack and legitimate traffic. In addition, network level attacks produce huge network traffic that is detectable by bandwidth rate. Amplification and flood attacks are the examples of this type of attack.

There are some methods which have been proposed to detect malicious activity using OpenFlow (Xing *et al*., 2013; Wen *et al*., 2013). One of the Distributed Denial of Service (DDoS) detection methods in SDN that has previously been studied is by using the statistical approach in comparing the threshold from the size of traffic subtracted by mean of traffic and three times of the standard deviation (Yan & Yu, 2015; Yan *et al*., 2016). If the threshold is higher than three times of the standard deviation, the traffic will be detected as DDoS attack. However, the researcher commented that the thresholds used in the previous method were based on simulations. It might have some differences compared to real-world experiment (Dillon & Berkelaar, 2014).

This research proposes a new method to improve the detection accuracy of DDoS attack in SDN network. It is by reducing false positive from existing previous method. This method utilizes entropy value to ensure the detection of DDoS attack.

II. METHODS

Software Defined Network (SDN) is the new paradigm of networking. In the existing traditional network, a switch acts as control and forwarding plane on a single machine. To change the traffic flow, it has to configure a switch and the others. SDN has a concept to separate the control plane from the forwarding plane. Control plane is separated from another machine called as a controller. Moreover, the network architecture is more flexible and has a cost-efficient program (Azodolmolky, 2013). The rule of packet forwarding can be stored in the control plane as a software part. This software rules the forwarding plane through a secure channel.

Figure 1a shows the existing network control system. It is the control plane and the data plane (forwarding plane) on the same machine. Meanwhile, Figure 1b shows the running specification of SDN OpenFlow where the control plane and the data plane are separated in different machines. The separation of control and data plane allows the addition of new protocols and applications independently. Then, the network is reduced to simple forwarding hardware (Azodolmolky, 2013).

In SDN, OpenFlow controller is responsible for maintaining the rules of network and, if it is needed, it distributes appropriate instructions to devices. When using the OpenFlow protocol, the controller is to determine the handling packet flow and to manage the switch flow table (Tiwari, Parekh, & Patel, 2014). Figure 2 shows the role of controller in SDN while the layering concept of SDN is shown in Figure 3.

There are some open source controllers that have been developed today (Braun & Menth, 2014). Those are NOX (C++ based), POX (Python based), Beacon (Java-based), Floodlight (Java-based), Maestro (Java-based), NodeFlow (Javascript based), Trema (C and Ruby based), OpenDaylight (Java-based), and Ryu (Python based). The OpenFlow controller uses a new protocol called OpenFlow for communicating with the switch. This protocol is the

main-force of SDN that manages the switches (Mousavi, 2014). Unlike DoS which only has one source of the attacker, DDoS has multiple sources. DDoS attack is a variant of DoS attack. It typically has three main elements, namely the victim, the attacker, and zombies.

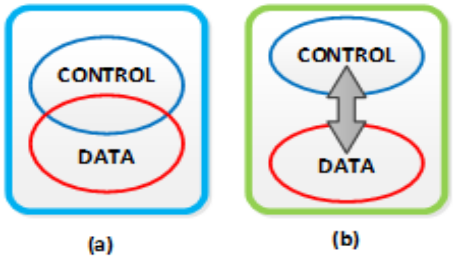

Figure 1 Network Switch Control System
(a) Traditional and (b) SDN
(Source: Azodolmolky, 2013)

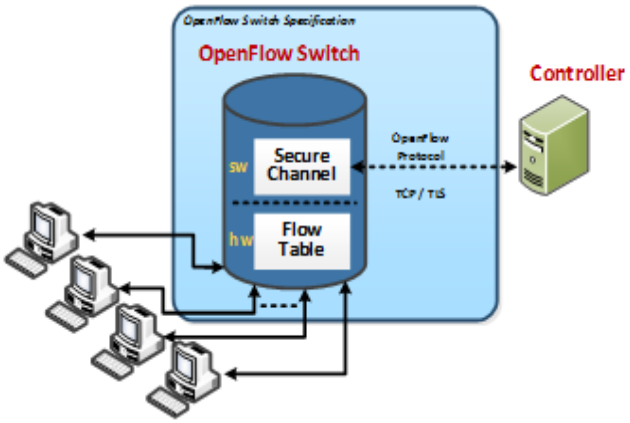

Figure 2 The Specification of SDN OpenFlow
(Source: Mousavi, 2014)

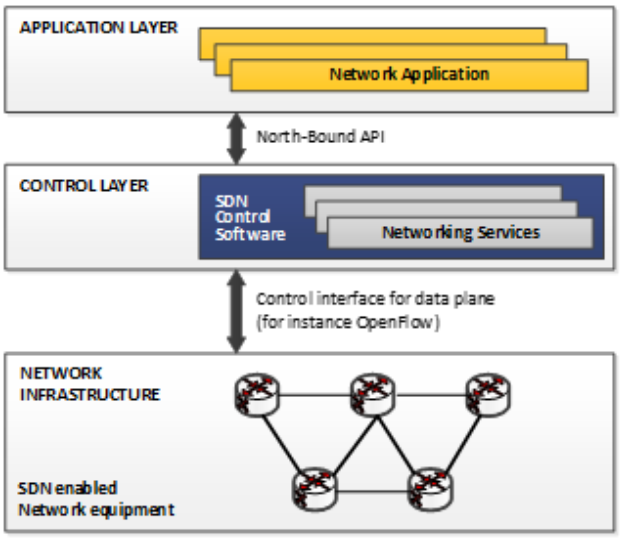

Figure 3 SDN Layers Architecture
(Source: Tiwari *et al.*, 2014)

At the beginning of the attack, the attacker sets a vulnerable system in a set of zombie machines that could be malware, virus, and others. The command that activates the DoS mechanism on zombie's machine is run by the attacker. This multiple source of DoS is called as DDoS.

Some vendors have developed and offered their OpenFlow-enabled switches in the recent years. All switches have tables that show the route of the packet including ingress and egress paths. This table is stored in OpenFlow switch and can be accessed by the controller through a secure channel. The controller can update the stored information in this table. Figure 4 shows a simple network using OpenFlow protocol (Mousavi, 2014). There are different kinds of attacks that can be classified based on the characteristics of the effect on the victim. They are software attack, protocol attack, and bandwidth attack. This research is focused on bandwidth attack. While an attack is happening, the detection mechanism has to recognize whether it is attack traffic or not.

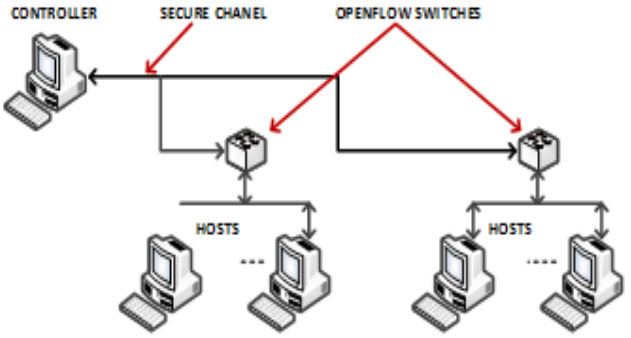

Figure 4 Simple Network Using OpenFlow Protocol
(Source: Mousavi, 2014)

In an attack situation, normal traffic must flow without being disrupted and misclassified. False positive happens when some traffic is misunderstood as attack traffic when it is not. On the other hand, false negatives are considered as legitimate traffic, but they are actually attacks (Carl, Kesidis, Brooks, & Rai, 2006). The completed outcome is shown in Table 1.

In traditional network, researchers have to add a device such as IPS or IDS. Unlike that, some researchers utilize SDN OpenFlow controller to detect DDoS attack in SDN. It is because SDN controller has a feature to record a flow of traffic in OpenFlow programmable table entry. An OpenFlow table entry consists of three main parts, such as rule, action, and status as illustrated in Figure 5.

Although there are only a few researches done, there are some methods that have been developed to detect DDoS attack in SDN. One of them is by using statistical method, and another one is Artificial Intelligence (AI) or Self Organizing Maps (SOM) used in more complicated network. SOM is one of the adapted methods for classifying network traffic in any condition (Braga *et al*., 2010). In traditional network, the light detection in small-scale network, which is used as the limitation of this research, is a statistical approach to the detection. This is because SOM needs higher controller machine specification as it needs to learn before it can detect the attack.

The previous method from Dillon and Berkelaar (2014) proposes a DDoS detection mechanism based on utilizing SDN controller and using statistical approach in the DDoS attack detection. The detection in this method compares the traffic size subscribed with mean to three times of the standard deviations on every 60 sliding windows of traffic (Dillon & Berkelaar, 2014). If the threshold is higher than three times of the standard deviations, the DDoS detection will be triggered by the controller.

Table 1 Outcome of DDoS Detection

	True	False
Positive	True-Positive (Rule matched and attack present)	False-positive (Rule matched and no attack present)
Negative	True-Negative (No rule matched and no attack present)	False-Negative (No rule matched and at-tack present)

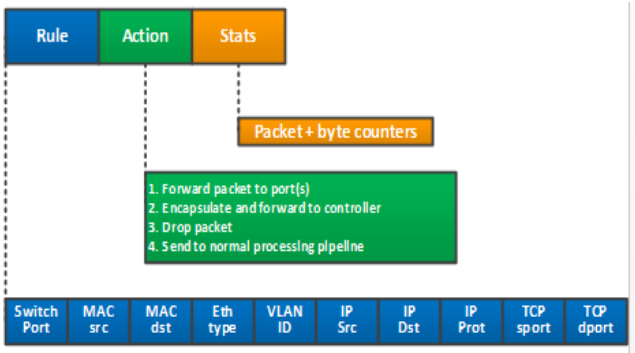

Figure 5 OpenFlow Table Entry
(Source: Dillon & Berkelaar, 2014)

The shortage of this method is when there is a sudden increase in traffic. It can be detected as a DDoS attack although it is a normal traffic. Then, it will increase the false positive because the normal traffic is detected as DDoS attack. According to Dillon and Berkelaar (2014), the first phase is calculating the standard deviation of every 60 sliding windows of traffic. Besides calculating the standard deviation of the packets, the traffic mean of the 60 sliding windows is also calculated. The packet Q_n will then be subtracted by the mean μ(Q) to get the deviation (D). The standard deviation formula can be calculated using equation as follows.

$$\sigma(Q) = \sqrt{\frac{1}{60}\sum_{i=1}^{60}(Q_i - \mu)^2} \quad (1)$$

The next phase is triggering the detection mechanism. There are three steps in this phase, namely Deviation (D) with D > 0; Minimum deviation (M) which is shown in equation 2; and preventing false positives at low bandwidth rates. Moreover, if the three conditions are fulfilled such as shown in equation 3, the traffic will be detected as a DDoS attack.

$$M=(Q_i-\mu(Q))>(3\ .\sigma(Q)) \quad (2)$$

$$D \wedge M \wedge V \rightarrow Z \quad (3)$$

However, as previously mentioned, Dillon and Berkelaar (2014) stated that the threshold used in the research are based on simulations only. It was not the real network system.

The other statistical approach that is usually used in traditional network to detect DDoS attack by measuring the randomness is entropy and Chi-Square statistic (Feinstein, Schnackenberg, Balupari, & Kindred, 2003). The entropy H is defined as follows.

$$H = -\sum_{i=1}^{n} p_i log_2 p_i \quad (4)$$

Subsequently, a mechanism to detect the changing of its randomness is by comparing the value for entropy. Meanwhile, Chi-Square is a model that needs information of intrusion and packet header's type. Chi-Square calculation can be seen in equation 5. N_i is the number of packets and n_i is the expected number of packets in normal condition (Feinstein *et al.*, 2003).

$$X^2 = \sum_{i=1}^{B} \frac{(N_i - n_i)(N_i - n_i)^2}{n_i} \quad (5)$$

In traditional network, mostly the method to detect DDoS attack is using entropy to measure the randomness of traffic. The other detection method that uses AI scheme is SOM as illustrated by Figure 6 (Braga, Mota, & Passito, 2010). The five steps of SOM learning process tasks can be summarized. There are initialization, sampling, and competition based on this equation 6 Where *l* is a neuron's number. Then, there is Synaptic adaptation in equation 7, where t is a current instant, $\theta j(t)$ is the function of neighborhood, and $\eta(t)$ is the learning rate. Last step is the repetition of steps 2 to 4.

$$i(x) = arg_j \ min|| x\text{-}W_j || \ ,j = 1,2,...,l \quad (6)$$

$$W_j\ (t + 1) = W_j(t) + \eta(t)\ \theta_j\ (t)(x(t) - W_j(t)) \quad (7)$$

This research enhances the previous method by adding some process. The design of improved method framework is shown in Figure 7. The difference between the previous method and the proposed method, as seen in Figure 7, is the mean and the counting of entropy in every 60 sliding windows. Then, the entropy will be used as the second mechanism to detect an attack.

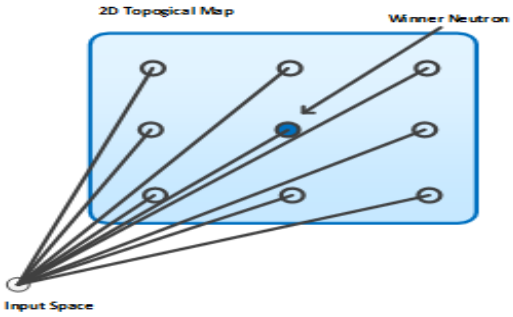

Figure 6 Kohonen's SOM Map Example
(Source: Braga *et al.*, 2010)

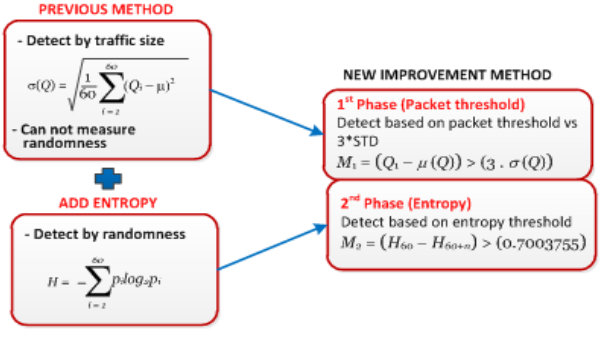

Figure 7 Improved Method

The details of the proposed method have fives principle steps. First, It is reading port statistics. The flow statistics are recorded in the controller from the port of the switch. Second, it is sending statistics to sliding windows. Every port statistic from the switch will be inserted to sliding windows. If it is exceeded, the first entry value of the sliding windows will be deleted. Third, it is calculating the packet entropy. Every packet entropy will be calculated from 60 values of sliding windows. Fourth, it is calculating three times of the standard deviation. The standard deviation is gained from 60 values of sliding windows. Last, it is recognizing an attack. If the new entry (Q1) subtracted with the mean of sliding windows is larger than three times of the standard deviation of 60 sliding windows, the state of the current packet entropy is the normal entropy parameter. It calculates the entropy threshold by subtracting the normal entropy with the current entropy. If the result is higher than 0,7003755, it will be detected as an attack. The steps of proposed method are illustrated in Figure 8.

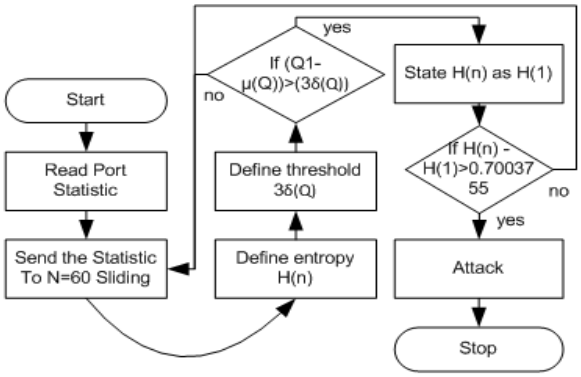

Figure 8 Proposed Method Steps

Moreover, Kumar (2013) analyzed the network traffic of five selected days from the DARPA data-set and plotted the time series. He tuned and filtered the DARPA data-set to test the DDoS attacks detection schemes. The calculated result is seen in Table 2. The researchers have used the packet size entropy threshold from DARPA data-sheet provided by Kumar (2013) in the experiment done for this research.

Table 2 DARPA Calculation from

Data-Set	Packet Rate	Packet Size Entropy	eSD
DARPA/ MIT	871	0,7003755	0,6230978

(Source: Kumar, 2013)

Next, the experiments to prove the concept method are performed using simulation software tool. It is Mininet with the network elements. There are OpenFlow controller (Ryu Controller), OVS Switch (OpenFlow switch), and two hosts (one as an attacker and one with normal traffic), and one host as a victim.

III. RESULTS AND DISCUSSIONS

The topology of the simulation network used in this research is illustrated in Figure 13. The network topology in this scenario uses 3 hosts, Host 1, Host 2, and Host 3 along with one OpenFlow Switch and one controller. Host 2 acts as an attacker, Host 3 is normal user, and Host 1 is victim. Host 3 sends normal packet to Host 1, and Host 2 sends attack packet to Host 1. The researchers choose Ryu Controller in this research because it provides software components with well-defined API making it easy for developers. The scenario of simulation is shown in Figure 9.

In this experiment, the researchers use two scenarios with represented traffic in Telkom network to simulate a normal traffic with sudden change and a DDoS attack traffic. After analyzing the Telkom Mean Router Traffic Gateway (MRTG), the researchers find that there are two types of traffic in Telkom network. There are uniform and non-uniformed.

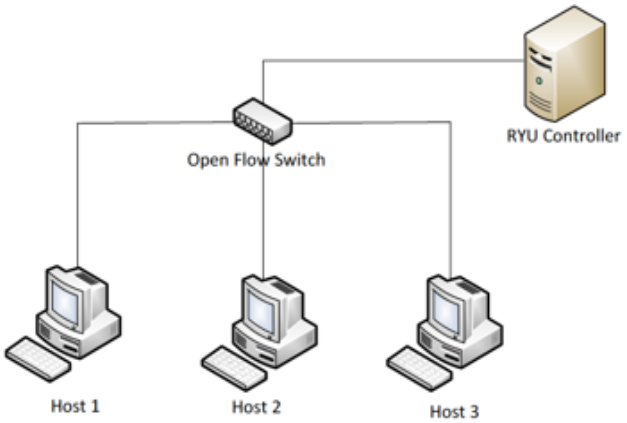

Figure 9 Topology Scenario of Simulation

In Scenario 1, it tests how both methods (previous method and improved method) detect normal traffic with a sudden increase. Host 3 will send normal traffic with large packet to Host 1, while Host 2 will send normal traffic that has various sizes ranging from small to large packet to Host 1. In this scenario, a sudden high traffic will be sent to Host 1 with fluctuation from low to high sizes to simulate a normal traffic. Furthermore, the researchers use traffic pattern from Telkom traffic.

In Scenario 2, the previous method and the improvement method are tested to detect attack traffic with flat condition. Host 2 will send a DDoS attack with large packet to Host 1. Afterwards, Host 2 will send traffic that suddenly changes from small to large packet to Host 1. Meanwhile, Host 3 will send normal packet to Host 1. The high traffic sent by Host 2 can simulate a DDoS attack, and send a flat-sized traffic to Host 1. The researchers also use traffic pattern from Telkom traffic.

Simulation result and analysis of Scenario 1 and Scenario 2 show the traffic flows throughout the network in Figure 10. The traffic has some sudden increases. To simulate the sudden change, the high traffic sent by Host 2 starts at second of 83. From that, the traffic fluctuates from high to low sizes to simulate normal traffic.

Using the existing method from the 60 pools of traffic, the researchers have the calculation results from the data in Figure 10. From the traffic flows at the 60 sliding windows from second of 23 to 82, it can be calculated that the mean of the traffic is 3,59. Using equation 4, the standard deviation of the traffic is 3,01. Thus, three times of the standard deviation is 9,04.

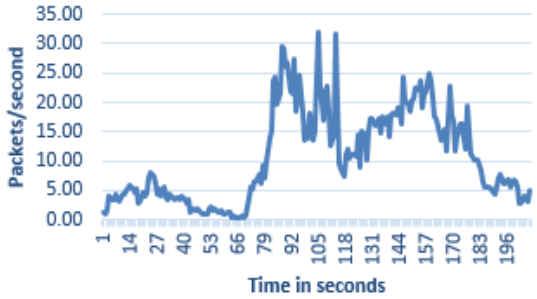

Figure 10 20M Traffic Injection at Second of 83

At second of 83, there is a sudden increase of the traffic to 23M. In this condition, the detection process starts as the value of the traffic subtracted by higher mean than three times of standard deviation. The result of 23M subtracted by 3,59M is 19,41M which is larger than three times of the standard deviation (9,04M). Since this method only uses the threshold of standard deviation, the traffic is detected as an attack. However, it is a normal traffic with a sudden temporary increase.

While using the existing method for traffics as shown in Figure 11, it is detected as DDoS. If it uses the proposed method that adds the second detection and states the entropy of traffic, it is not detected as DDoS. In this experience, the starting entropy at second of 82 is 5,84022. From the proposed equation, the researchers have stated that the traffic is detected as DDoS attack when the starting threshold subtracted by the entropy of traffic is higher than 0,7003755. The threshold will be calculated on 60 sliding windows and stopped when it reaches 0,7003755. In this case, the entropy threshold at second of 82 to 141 is various from 0 to 0,23333. Although the detection traffic size subtracted by mean is larger than three times of the standard deviation, the entropy threshold is still lower than the parameter. Thus, the traffic is not detected as DDoS attack.

Comparisons between the previous method and the improvement method are shown in Table 3. It shows the comparisons between the previous method and the new method that adds entropy checking. With the previous method, detection to the normal traffic with sudden increase is calculated by subtracting the size of traffic with the mean. If the result is higher than three times of the standard deviation, it will be detected as DDoS attack. This method can increase false positive as there are much normal traffic which is detected as attack. By using the new method that combines the previous method with entropy, the DDoS attack traffic entropy can be checked with normal traffic entropy. Therefore, the sudden increase traffic will not be detected as DDoS attack and the false positive can be decreased.

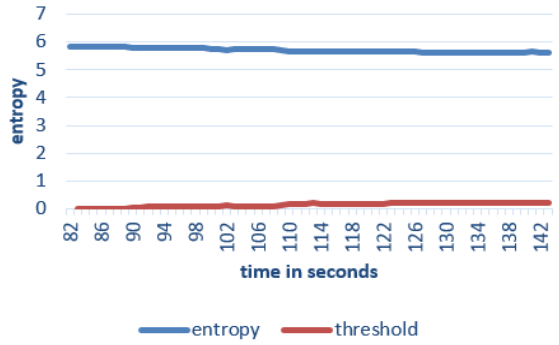

Figure 11 Entropy and Entropy Threshold in Scenario 1

For Scenario 2, traffic flowing through the network is shown in Figure 12. It can be seen that a high traffic is simulated as an attack by sending traffic to make the traffic flows almost flatly. In second of 74, the high traffic is injected from Host 2 to Host 1.

Using the existing method, the researchers can have the traffic calculation from second of 14 to 73 on the 60 sliding windows. The mean of the traffic is 2,78. The standard deviation of the traffic is 0,73 which makes three times of the standard deviation (2,19). With the traffic injection on second of 74, the traffic has a sudden increase that causes the flows to rise to 22M. Starting the detection mechanism, the result of subtracting the value of traffic with the mean is 19,22. It is higher than three times of the standard deviation. Since this method only uses threshold of standard deviation mechanism, the traffic is detected as an attack.

Figure 13 shows how entropy is used to measure the traffic randomness to detect an attack. If it uses the proposed method that adds the second detection when detected as DDoS, at second of 74, the controller states the entropy of traffic at second of 60 as starting entropy. The starting entropy on second of 73 in this experience is 5,90689. From the proposed equation, the researchers have stated that the traffic is detected as DDoS attack when the starting threshold subtracted by the entropy of traffic is higher than 0,7003755. The entropy threshold will be calculated on 60 sliding windows and stopped when it reaches 0,7003755. The entropy at second of 110 is 4,73457, and the threshold is 0,7679005. It is higher than 0,7003755. The entropy and entropy threshold can be seen in Figure 13.

In this case, after all steps are matched, the DDoS detection can be triggered. When DDoS attack happened, the entropy is lower than the normal entropy which is 0,7003755. This shows that the traffic is more flat than normal traffic leading to detected DDoS attack.

Table 3 Comparison between Testing Scenario 1 with Sudden Increase

Methods	Traffic	Mean(60)	Q1-M	3*STD	1st Detect	2nd Detect	Detection
Existing	20M	3,59	19,41	3,01	Attack	N/A	Attack
Add entropy	20M	3,59	19,41	3,01	Attack	Not Attack	Not Attack

The comparisons between the previous method and the improvement method are shown in Table 4. It shows comparisons between the previous method with new method that adds entropy checking. By using both methods, the attack traffic is detected as DDoS attack. The previous method detects the attack at the first mechanism. Meanwhile, the new method detects it after checking the entropy threshold.

As for the summary of findings based on the experiment, it is said that the Ryu controller can be programmed to detect DDoS attack in SDN network. This is one of the advantages of traditional network as the traditional switch is not programmable to do this task. Some vendors have developed several devices to add to the traditional network to do this task such as Intrusion Detection System (IDS). However, it has a few disadvantage such as requiring a large amount of money and increasing the point of failure in the network. Another finding is that in the first detection, both traffics are detected as DDoS attack. This still happens although the traffic in Table 2 is a normal traffic that only experiences a sudden rise of traffic. It usually happens in real traffic. On the other hand, DDoS attack usually generates a large traffic to make the network overload. By adding the second method that detects the randomness of traffic using entropy, Scenario 1 in Table 2 is not detected as DDoS attack while Scenario 2 in Table 3 is detected as DDoS attack.

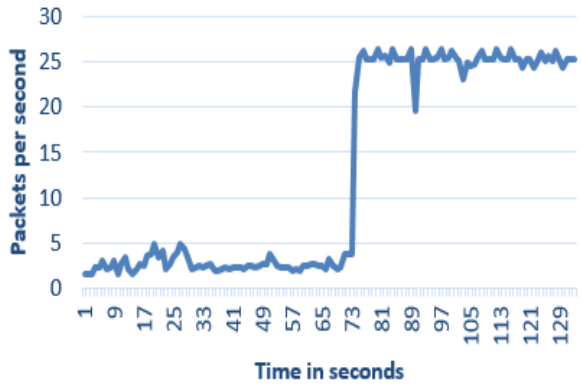

Figure 12 20M Traffic Injection on Second of 74

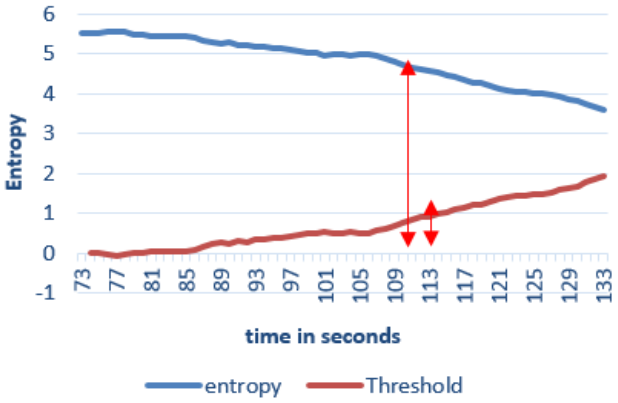

Figure 13 Entropy and Entropy Threshold in Scenario 2

Table 4 Comparison between Testing Scenario 2

Methods	Traffic	Mean(60)	Q1-M	3*STD	1st Detect	2nd Detect	Detection
Existing	20M	2,78	19,22	0,73	Attack	N/A	Attack
Add entropy	20M	2,78	19,22	0,73	Attack	Attack	Attack

IV. CONCLUSIONS

The findings of this research show that OpenFlow controller in SDN can be programmed to detect DDoS attack. The existing method that only uses three times of the standard deviation threshold as detection mechanism is added with a new mechanism of measuring the randomness of the traffic. As a proof of the experience, the combination of the existing method with entropy can improve the DDoS detection as it has better result in reducing false positive than the existing method. It is because the existing method cannot filter whether it is a normal traffic or attack traffic by its randomness. Consequently, when there is a high rise of traffic, it will be detected as an attack. On the other hand, adding the entropy method computes the randomness. This method can detect the traffic as a temporary increase that usually occurs in normal traffic. It does not detect it as an attack. This fact shows how SDN can be used to detect an attack with statistical approach. The existing method such as three times of the standard deviations threshold mechanism can be combined with entropy to measure the randomness of packets.

REFERENCES

Azodolmolky, S. (2013). *Software defined networking with OpenFlow.* Birmingham, UK: Packt Publishing.

Braga, R., Mota, E., & Passito, A. (2010). Lightweight DDoS flooding attack detection using NOX/OpenFlow. In *Local Computer Networks (LCN), 2010 IEEE 35th Conference*. Denver, USA.

Braun, W. & Menth, M. (2014). Software-defined networking using OpenFlow: Protocols, applications and architectural design choices. *Future Internet, 6*(2), 302-336.

Carl, G., Kesidis, G., Brooks, R. R., & Rai, S. (2006). Denial-of-service attack-detection techniques. *IEEE Internet Computing, 10*(1), 82-89.

Dillon, C., & Berkelaar, M. (2014). *OpenFlow (D)DoS mitigation*. Retrieved from http://www.delaat.net/rp/2013-2014/p42/report.pdf

Feinstein, L., Schnackenberg, D., Balupari, R., & Kindred, D. (2003). Statistical approaches to DDoS attack detection and response. In *DARPA Information Survivability Conference and Exposition, 2003. Proceedings* (Vol. 1, pp. 303-314). IEEE.

Kandoi, R., & Antikainen, M. (2015). Denial-of-service attacks in OpenFlow SDN networks. In 2015 *IFIP/IEEE International Symposium on Integrated Network Management (IM)* (pp. 1322-1326). IEEE.

Kumar, T. (2013). *An improved packet size entropy based DoS attack detection scheme* (Doctoral Dissertation). Rourkela, India: National Institute of Technology Rourkela.

Mousavi, S. M. (2014). *Early detection of DDoS attacks in software defined networks controller* (Master Thesis). Ottawa, Ontario: Carleton University.

Oktian, Y. E., Lee, S., & Lee, H. (2014). Mitigating Denial of Service (DOS) attacks in openflow networks. In 2014 *International Conference on Information and Communication Technology Convergence (ICTC),* (pp. 325-330). IEEE.

Siregar, J. J. (2013). Analisis explotasi keamanan web denial of service attack. *ComTech: Computer, Mathematics and Engineering Applications, 4*(2), 1199-1205.

Tiwari, V., Parekh, R., & Patel, V. (2014). A survey on vulnerabilities of Openflow network and its impact on SDN/Openflow controller. *World Academics Journal of Engineering Sciences, 1*, 1-5.

Xing, T., Huang, D., Xu, L., Chung, C. J., & Khatkar, P. (2013). Snortflow: A openflow-based intrusion prevention system in cloud environment. In *Research and Educational Experiment Workshop (GREE), 2013 Second GENI* (pp. 89-92). IEEE.

Wen, X., Chen, Y., Hu, C., Shi, C., & Wang, Y. (2013). Towards a secure controller platform for openflow applications. In *Proceedings of the second ACM SIGCOMM Workshop on Hot Topics in software Defined Networking* (pp. 171-172). ACM.

Yan, Q., & Yu, F. R. (2015). Distributed denial of service attacks in software-defined networking with cloud computing. *IEEE Communications Magazine, 53*(4), 52-59.