# FLOODING ATTACK DETECTION AND MITIGATION IN SDN WITH MODIFIED ADAPTIVE THRESHOLD ALGORITHM

Nan Haymarn Oo[1], Aris Cahyadi Risdianto[2], Teck Chaw Ling[3] and Aung Htein Maw[4]

[1]University of Computer Studies, Yangon, Myanmar
[2]Gwangju Institute of Science and Technology, Korea
[3]University of Malaya, Malaysia
[4]University of Information Technology, Myanmar

## *ABSTRACT*

*Flooding attack is a network attack that sends a large amount of traffic to the victim networks or services to cause denial-of-service. In Software-Defined Networking (SDN) environment, this attack might not only breach the hosts and services but also the SDN controller. Besides, it will also cause a disconnection of links between the controller and the switches. Thus, an effective detection and mitigation technique of flooding attacks is required. Statistical analysis techniques are widely used for the detection and mitigation of flooding attacks. However, the effectiveness of these techniques strongly depends on the defined threshold. Defining the static threshold is a tedious job and most of the time produces a high false positive alarm .In this paper, we proposed the dynamic threshold which is calculated using modified adaptive threshold algorithm (MATA). The original ATA is based on the Exponential Weighted Moving Average (EWMA) formula which produces the high number of false alarms. To reduce the false alarms, the alarm signal will only be generated after a minimum number of consecutive violations of the threshold. This, however, has increased the false negative rate when the network is under attack. In order to reduce this false negative rate, MATA adapted the baseline traffic info of the network infrastructure. The comparative analysis of MATA and ATA are performed through the measurement of false negative rate, and accuracy of detection rate. Our experimental results show that MATA is able to reduce false negative rates up to 17.74% and increase the detection accuracy of 16.11%over the various types of flooding attacks at the transport layer.*

## *KEYWORDS*

*Adaptive Threshold, Flooding attack, Software-Defined Networking*

## 1. INTRODUCTION

Flooding attack sends an extremely amount of network traffic to overwhelm the targeted network, or a particular victim server in preventing the normal connection requests from the benign users. Thus, it is a common type of Distributed Denial of Service (DDoS) attack. The impact of this flooding attack can bring down the target victim within a very short time. The target can be the network or servers running with traditional or advanced technique, software-defined networking (SDN) [1,2].

The main feature of SDN is decoupling the network control plane from the data forwarding plane and centralized controlling the network by using the controller [3]. This centralized control feature may well be the advantages of monitoring, detecting, and mitigating the attacks. On the other hand, it can also be the weak point of the network. The attackers may launch the flooding attack targeting the controller ,aiming to crash the entire network with a single point of failure [4]. Furthermore, the data forwarding switches, the links between the controller and the switches, and the SDN hosts running a particular service can also be the targets.

There are two different types of flooding attacks: protocol exploited attack, and amplification or reflection attack. The common protocol exploited attacks are SYN flooding, UDP flooding, and ICMP flooding attack. DNS amplification and NTP amplification attacks are belonging to the group of amplification or reflection attacks [5]. This research is focusing on the detection and mitigation of SYN flooding and UDP flooding that exploits the TCP and UDP protocol, respectively.

In the SYN flooding attack, the attackers exploit the TCP's three-way handshaking mechanism of connection establishment process and send a large number of SYN packets continuously. As a result, the server's memory is filled with the connections requests and rejects the normal connection requests from legitimate users eventually [6-8].In the UDP flooding attack, the attackers send a large stream of UDP packets with the specific or random port numbers to the target server using a spoofed source IP address. The victim server responds the ICMP packets for the port which does not listen. As a result, the attack consumes all the network bandwidth and overloads the server to be able to disturb normal operations [9].

The detection and mitigation of SDN-based DDoS attacks have been proposed by using the various mechanisms such as statistical analysis (change point detection, and entropy), machine learning, traffic pattern analysis, connection rate analysis, and integration of traffic monitoring tool and Open Flow [5]. Although each technique has its pros and cons, the widely used technique among them is the statistical analysis technique. The technique is comparing the number of incoming traffic with the threshold and defines it as attack traffic when the threshold is violated by the incoming traffic.

The value of the threshold can be defined statically or dynamically. However, the weakness of the static threshold is raising the high number of false alarms and resulted in tedious job for the network administrators. Thus, the dynamic threshold is commonly used in statistical analysis techniques. The dynamic threshold for the flooding attack can be simply calculated by using Adaptive Threshold Algorithm (ATA) [10]. This algorithm produces high rate of detection but it also raises a high number of false alarms. In order to reduce the false alarms, the alarm signal will only be generated after a minimum number of consecutive violations of the threshold. However, this will increase the false negative rate when the network is under attack. Thus, this algorithm is modified by taking into account the baseline of the network traffic. The main objective of the Modified Adaptive Threshold Algorithm (MATA) is to produce the dynamic threshold value that is adaptable over the incoming traffic based on the baseline.

The rest of the paper is organized as follows. Section 2 describes the related works. Section 3 presents the algorithms for calculating the dynamic threshold. Section 4 describes the processes of the detection and mitigation of flooding attacks. Section 5 presents the experimental testbed. Section 6 describes the detailed implementation of the experiment. Section 7 demonstrates the evaluation results of this system. The final section 8 concludes this paper.

## 2. RELATED WORK

As the various types of DDoS attack targeted the recognized organization's networksbased on either the traditional techniques or the advanced network technologies such as SDN, many different mechanisms for the attack detection and mitigation have been implemented. The authors of[11-13] have been detected the DDoS attacks with entropy. Although entropy has been successfully used in measuring the randomness of the network traffic within a given period of time, it can't consider many different values when calculating the probability of a feature. Moreover, the various types of machine learning techniques including Support Vector Machine (SVM), self-organizing map (SOM), artificial neural networks and fuzzy logic principles and concepts are used in [1][14-19]. These techniques can be effectively used to detect the malicious activities based on the abnormal behavior of the network. However, the performance of these techniques is relying on the training dataset.

Change point detection techniques can be effectively used for the detection of the flooding attacks. These techniques are based on the Exponential Weighted Moving Average (EWMA) formula. The commonly used techniques are the Adaptive Threshold Algorithm (ATA) and the Cumulative Sum (CUSUM) algorithm. Conti et al. [20] detected the DDoS attack with the dynamic threshold calculated by using the non-parameter CUSUM with the adaptive threshold algorithm. Moreover, they also proposed the detection of network reconnaissance attacks by the EWMA control chart in [21]. However, their proposed systems have a little overhead for the collection and manipulation of traffic statistics in the SDN controller. Modified EWMA formula has been used to detect and mitigate the application-specific DDoS attacks in [5].

ATA and CUSUM algorithms have been investigated over the SYN flooding attack in [10]. ATA is not only simple and easy to implement but also effectively detect the flooding attack. However, this algorithm produces a high number of false alarms. Thus, it defines the suspicious event as the real attack after a minimum number of continuous threshold violations for avoiding false alarms. As the consequent result, the avoiding method of false alarms in this algorithm raises the false negative rate when the network is in a real attack. In order to reduce the false negative rate, the existing ATA algorithm is modified by adding a static parameter to the comparison of the current traffic and the average number of previous traffic. The parameter is taken from the measurement of the baseline traffic of the network infrastructure. Moreover, the sFlow-RT analyzer is used for the collection and manipulation of traffic statistics for reducing the overhead of the SDN controller.

Arins A. proposed firewall as a service in SDN for solving the two main problems of DDoS: distinguishing good packets from bad packets, and dropping bad packets at the closet point to attacker networks [22]. The authors in [23] also drop the detected malicious packets as their mitigation mechanism for protecting the IoT-based DDoS attack in SDN. Lu et al. focused on the source-based defense mechanism against botnet-based DDoS flooding attacks through the combination of the power of SDN and sFlow technology [24]. Conti et al. [21] also mitigated the flooding-based DoS attack by installing the temporarily Drop flow rules. The authors of [12] discussed the anomaly mitigation with the consideration of the centralized control feature of the SDN network for tracing back the source of the attackers and doing the source filtering at the source switch. All their proposed mitigation systems are discarding the detected attack with the installation of a simple drop flow rule according to the advantages of the centralized control feature of SDN. In this system, in order to mitigate the flooding attack effectively, drop flow rules are installed with two options at the ingress switch of the attack. Temporarily drop flow rules are installed for the attacks that come for the first time. Permanent drop flow rules are installed when the same attacks come again after their respective temporarily drop flow rule expires.

# 3. ALGORITHMS FOR DYNAMIC THRESHOLD CALCULATION

One of the effective algorithms for calculating the dynamic threshold is ATA. In this section, both algorithms (i.e. original ATA and modified ATA) are described with their respective false alarm avoiding method.

EWMA is commonly used in finding the dynamic threshold for the network traffic. The calculated threshold value provides not only high detection rate but also high false positive rate.Thus, ATA uses the twice of the EWMA result or more as its threshold value. However, it still raises some false alarms in some cases. In order to reduce false alarms, this algorithm only raises the alarm signalafter a minimum number of consecutive violations of the threshold.

## 3.1. Adaptive Threshold Algorithm (ATA)

Let $CF_t$be the current number of incoming frames at time $t$, $PF_{t-1}$ is the average number of frames estimated from the measurement prior to $t$, and $p$ is the percentage parameter, $p > 0$.

If $CF_t \geq (p+1)PF_{t-1}$ then the alarm signalled at time $t$. (1)

The percentage parameter is used to indicate the anomalous behaviour when the defined percentage of the previous average number of frames is exceeded by the current number of incoming frames.

EWMA formula is used to pre-calculate the average number of incoming frames that will be used in the comparison for the next second as shown in equation (2):

$$PF_t = \alpha PF_{t-1} + (1-\alpha)CF_t \quad (2)$$

$\alpha$ is the factor parameter used in making the decision of the factors of current number of frames and average number of previous frames for calculating the average number of previous frames to be used at the next calculation.

Applying the algorithm directly would yield a high number of false alarms. Thus, a simple modification is made to signal an alarm after a minimum number of consecutive violations of the threshold as shown in equation (3):

If $\sum_{i=n-k+1}^{n} 1_{\{CF_i \geq (\alpha+1)PF_{i-1}\}} \geq k$ then the alarm signalled at time t. (3)

In this equation, $k$ is the parameter that indicates the number of consecutive intervals the threshold must be violated for alarm to be raised,$k > 1$.

## 3.2. Modified Adaptive Threshold Algorithm (MATA)

For reducing the false negative value while avoiding the false alarms, MATA is taken into account the baseline traffic,$b$,in the traffic comparison. Thus, the value for the baseline traffic is needed to define at the initial state.

The baseline of a network can be identified by analysing the monitoring result of the network for a period of time. In this system, the sFlow analyzer is used as the monitoring tool and the

monitoring result (i.e. the event information) is analysed for defining the value of the baseline. The process of defining baseline can be divided into four steps:

Step 1: Collect all event information from the various types of service produced by the analyzer.
Step 2: Categorize the collected information according to their type of service (i.e. Web, FTP, Mail, NTP, DHCP and DNS).
Step 3: Find the maximum number of frames per second for each type of service from the collected event information.
Step 4: Define the maximum number of frames per second as the value of the baseline.

According to the step 3 and 4, the baseline traffic for Web service is defined as the maximum number of SYN frames per second from the Web event information produced by the sFlowanalyzer as shown in equation (4).

$$b_{web} = Max(number\ of\ SYN\ frames\ per\ second\ for\ web\ service) \quad (4)$$

Similar to the Web service, the baseline traffic of other TCP services such as FTP, and mail are identified. For the UDP services, the baseline traffic for the DNS service is identified as the maximum number of frames per second from the DNS event information as shown in equation (5).

$$b_{DNS} = Max(number\ of frames\ per\ second\ for\ DNS\ service) \quad (5)$$

The baseline traffic of the NTP and DHCP services are similarly identified as the baseline definition of DNS service.

Thus, in general, the baseline of a particular service,$b_{SV}$,is identified as the maximum number of frames per secondfor the service, $Max(n_{SV})$,as shown inequation (6).

$$b_{SV} = Max(n_{SV}) \quad (6)$$

In the equation (6), $b$ is the baseline traffic parameter, $SV$ is represented for a particular service, $n$ is the number of frames per second containing in the event information. The calculation of baseline traffic over an emulated SDN network environment is described in detail in section 6.1.2.

After getting the baseline value, the equation (1) of ATA is modified by adding the baseline traffic parameter, $b_{SV}$ ,for indicating the anomalous behaviour when the total number of the defined percentage of the average number of previous frames and the number of frames of the baseline traffic is exceeded by the current number of incoming frames. The modified equation is shown in equation (7):

If $CF_t \geq (p+1)PF_{t-1} + b_{SV}$ then the alarm signalled at time $t$. (7)

As the original ATA, the average number of incoming frames for the next second is pre-calculated by using the EWMA formula as shown in equation (2) of session 3.1.

## 4. FLOODING ATTACK DETECTION AND MITIGATION SYSTEM

The overall architecture of flooding attack detection and mitigation system is composed of two main phases: flooding attack detection, and mitigation of the detected attacks as shown inFigure

1. In the detection phase, the various types of frames from the SDN hosts incoming into the Open vSwitch are collected and detected by sFlow-RT analyzer [25] in order to differentiate the normal frames and the malicious frames of the flooding attack. If the malicious flooding frames are incoming into the switches, then the analyzer produces abnormal event information. The mitigation application running in ONOS controller [26] instantaneously discards the frames when it receives the event information from the analyzer.

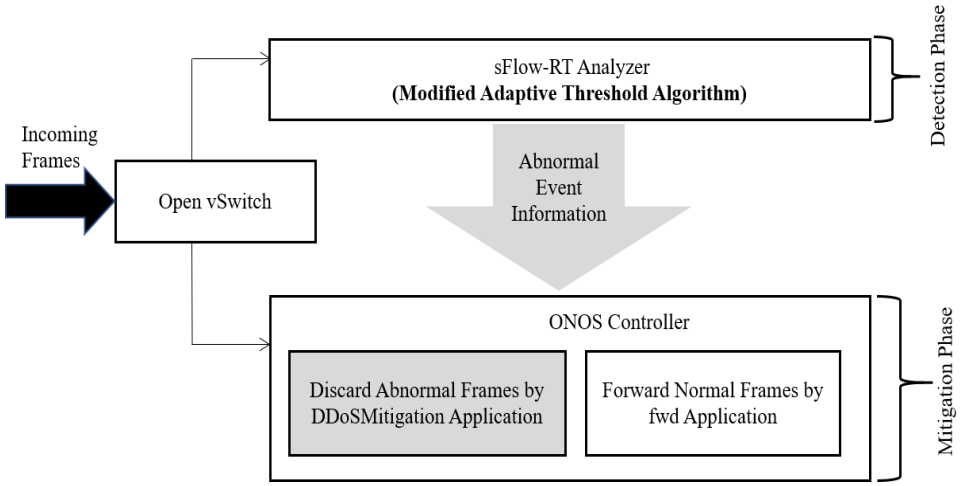


Figure 1. Overall architecture of flooding attack detection and mitigation

## 4.1. Detection Phase

Flooding attack detection phase is implemented by using sFlow-RT analyzer in order to reduce the load of traffic statistic in the SDN controller. It is composed of three parts: flow definition, flow handling, and event handling.

### 4.1.1. Flow definition

The analyzer collects the incoming flow of each service according to the predefined polling interval. As this system is detecting the flooding attack at the transport layer, it has two types of flow definitions for TCP and UDP protocols. Moreover, this system is especially detecting only the SYN flooding attack for TCP protocol.Thus, the analyzeronly collects SYN frames from the incoming TCP traffic of Web, FTP, and MAIL server by using the flow keys (i.e. source MAC, destination MAC, source IP, destination IP, and destination port) with filtering the TCP's SYN flag and destination port of that frames. For UDP protocol, the analyzer collects all frames from each type of the incoming UDP traffic such as DNS, DHCP, and NTP service with the same flow keys as TCP and only one filtering key (i.e. destination port).

### 4.1.2. Flow handling

The analyzer handles the various types of incoming flows in every second. It also controls the time of handling for each service to handle every flows incoming from the various types of service alternatively. The process of flow handling function for each service can be sub-divided into two parts: frame comparing, and new threshold calculation.

***Frame comparing:*** The sFlow analyzer compares the number of incoming frames with the respective dynamic and adaptive threshold value calculated in the previous second. In the beginning of the frame comparing (i.e. t = 0s), the analyzer compares the number of frames with

the predefined initial threshold. For ATA, the initial value of threshold and the average number of previous frame is 0. For MATA, the two values are initialized by the baseline traffic.

***New threshold calculation:*** The number of incoming SYN frames collected from the flow definition is counted by the analyzer for each TCP service. Similarly, the analyzer counts the number of all incoming frames from the respective UDP flow definition for each UDP service. After counting the number of frames, the average number of previous frames for the next second is calculated by using equation (2) of the ATA, EWMA formula, described in section 3.1. Then the new threshold value is calculated by taking the combination of the twice of the EWMA result and the baseline.

#### 4.1.3. Event handling

According to the frame comparing function, once the threshold is violated by the number of incoming frames, the sFlow analyzer produces the alert messages for indicating that the abnormal event is occurring in the network.

### 4.2. Mitigation Phase

The DDoS Mitigation application running in ONOS controller periodically takes the abnormal event information from the sFlow analyzer via REST API in every second. In this mitigation phase, MATA and ATA operate in a different way over the event information from the sFlow analyzer because the analyzer produced the different abnormal event information in the detection phase.

#### 4.2.1. MATA –based mitigation

As soon as the DDoS Mitigation application receives the information, it firstly extracts the source and destination IP address from the information, and finds the source switch connected with the attacker host by using source IP address. Then, the application installs temporarily drop flow rule for 60 seconds into the source switch of the attack for discarding the flooding packets at the nearest point to the attacker host. If the application receives again the previous event information when the flow rule has been expired, it installs permanent drop flow rules for such event information.

#### 4.2.2. ATA –based mitigation

The ATA-based DDoS Mitigation application does not discard any frames as soon as it receives the event information from the analyzer. It firstly confirms whether the received event information is really signalled the attacks or not because some information might be the false alarms. In order to define the event formation that is not false alarms, the algorithm predefines the number of consecutive threshold violation within a period of time for each service according to the equation (3) of ATA. Thus, the DDoS Mitigation application monitors and counts the number of consecutive event information within a predefined time and compares the number of information with the predefined value. If the predefined value is exceeded by the number of received information, then it finds the source switch connected with the attacker host. Finally, the application installs drop flow rule into the source switch of the attack.

## 5. EXPERIMENTAL TESTBED

The testbed for testing the flooding attack detection and mitigation system is composed of four Open Flow switches, one controller, six servers, and twelve clients as shown in Figure 2.One switch is connected with all servers and the others are connected with the clients. One of the client hosts is treated as an attacker and the remaining clients are benign users. Each server is a target victim alternatively.

The links connected between the switch and the servers and those among the switches are configured with 1000 Mbps and those connections between the switch and the clients are configured with 100 Mbps. The network topology is constructed by using mininet [27] emulator. Two laptops PC are used in setting up this system. ONOS controller and sFlow-RT analyzer are running in one virtual machine on Dell Laptop PC with Intel(R) Core(TM) i7-4500U CPU @ 1.80GHz, 64 bits and 8GiB memory and the based mininet network is running on another Dell Laptop PC with Intel® Core™ i5-4790 CPU @3.60GHz, 64 bits, and 4GiB memory. The two PCs are connected by an Ethernet cable. Since the two PCs for mininet and analyzer are directly connected, the sampling rate and polling interval in the sFlowanalyzer is defined as 1.

Table 1. Testbed information

| Type | Host name | IP address |
|---|---|---|
| Servers | h1 – h6 | 10.0.0.1 – 10.0.0.6 |
| Clients | h7 – h17 | 10.0.0.7 – 10.0.0.17 |
| Attacker | h18 | 10.0.0.18 |
| Switches | s1 – s4 | - |

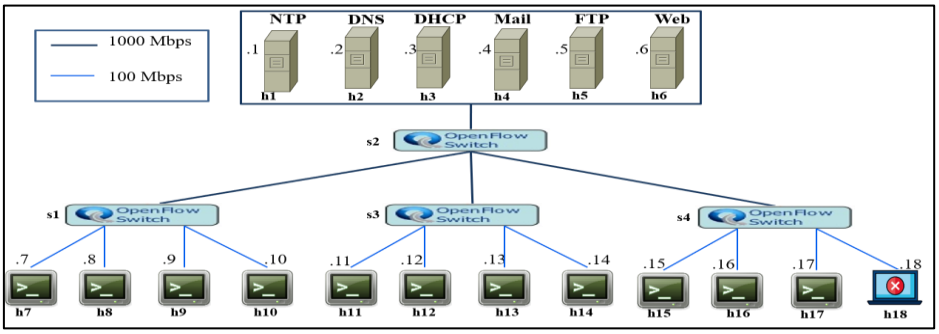


Figure 2. Network topology for the experimental testbed

## 6. EXPERIMENTAL IMPLEMENTATION

The detection and mitigation of flooding attack is implemented with detection phase and mitigation phase based on the original ATA and modified ATA algorithm alternatively in order to evaluate this system. Moreover, an attack scenario is used to test and evaluate this system.

### 6.1. Detection Phase

The sFlow analyzer is mainly used in the detection phase of this system. According to the workflow of the analyzer, this system collects the incoming frames, compares the number of incoming frames with the dynamic and adaptive threshold, raises alert when the threshold is

violated, and calculates the average number of previous frames and the new threshold value for the next second with EWMA formula as described in section 4.1.

In order to take into account the equal amount of current incoming frames and the average number of previous frames when calculating the average number of frames for the next time, the factor parameter $\alpha$ of EWMA formula is identified as 0.5. Moreover, the percentage parameter $p$ inequation (1) is defined as 1 for doubling or taking twice the previous number of frames (i.e. EWMA result)which is to be used as a threshold in the process of frame comparing. As both the dynamic calculation methods (i.e. ATA and MATA) are based on the EWMA formula, they raise a high number of false alarms and must avoid them.

### 6.1.1. False alarms avoidance in ATA-based detection

The ATA raises the alarm signal after a minimum number of consecutive threshold violations for avoiding false alarms. According to the result from the analysis of sFlow event information, the real abnormal event information and false alarms are differentiated by defining the number of same event information occurring continuously within a period of time.

As shown in Figure 3, the sFlow analyzer continuously produced the same three NTP service's abnormal event information during the two seconds. Actually, these events information are false alarms occurring in normal conditions. Thus, more than three same event information that occurred within the two seconds is defined as the real abnormal event information for NTP service. Similarly, the number of event information and its duration are analysed and predefined for each service as shown in Table 2.

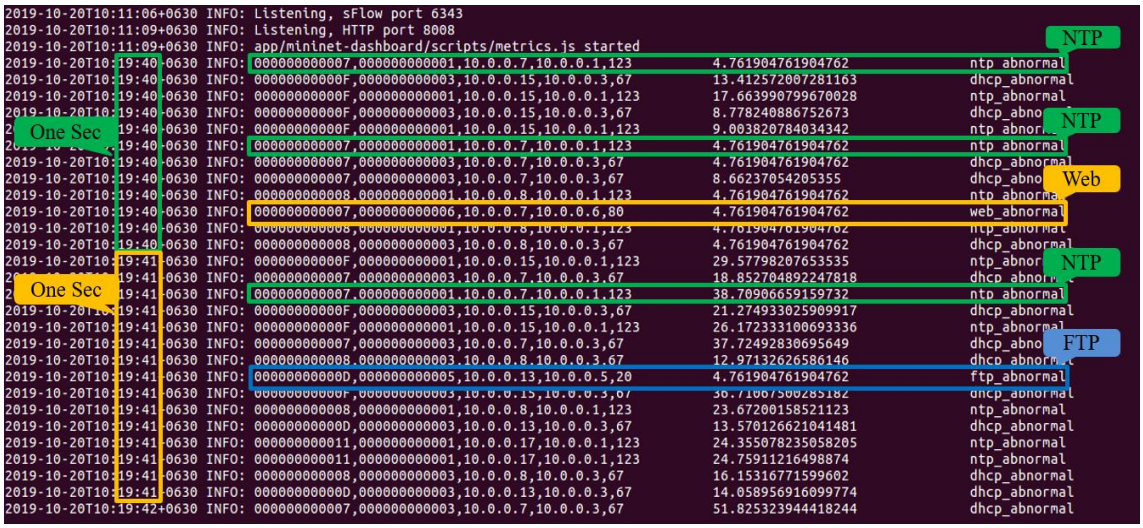


Figure 3. Event information from sFlow analyzer

Table 2. Number of consecutive threshold violation

| Type of service | No. of event information (n) | Time (second) |
|---|---|---|
| Web | n > 1 | 2 |
| FTP | n > 1 | 2 |
| Mail | n > 1 | 2 |
| DNS | n > 6 | 2 |
| NTP | n > 3 | 2 |

### 6.1.2. False alarms avoidance in MATA-based detection

As this algorithm is taken into account the baseline traffic of the network for avoiding the false alarms, sFlow analyzer produces the dynamic threshold which adaptable with the baseline and it can reduce the false alarms significantly.

In order to define the value of baseline traffic for the current network topology, we observed the network for one minute while all normal users are accessing all available network services concurrently. As we are doing the experiment in the virtual mininet network environment, in order to get the baseline similar to the actual baseline traffic of the real network environment, we used D-ITG (Distributed Internet Traffic Generator) tool [28] for generating the network traffic in the virtual network. This tool generates the traffic with Inter Departure Time (IDT) and Packet Size (PS) using stochastic models such as uniform, constant, exponential, pareto, cauchy, normal, poisson, gamma, and weibull distribution. It can also generate the transport layer traffic (i.e. TCP, UDP) and application layer traffic (i.e. DNS, Telnet, VoIP).

The traffic generation model [29] described that poisson distribution can be used to generate the traffic with the number of incoming packets or calls per time unit (i.e. IDT). Moreover, the traffic including the length of each phone call (i.e. PS) can be generated by the exponential distribution. Theoretical traffic model [30] can be summarized as shown in Table 3. The IDT for the telnet traffic and new network transfer protocol (NNTP) traffic can be generated by using poisson distribution and weibull distribution respectively. Moreover, both IDT and PS of VoIP traffic are generated by using the exponential distribution. The PS of NNTP , SMTP, and FTP traffic during the whole session are generated by using $\log_2$-normal distribution. In addition, pareto distribution is used to produce the PS of Web and FTP traffic during a burst session.

Table 3. Theoretical traffic model for generating IDT and PS for each service

| **Service** | **IDT** | **PS** |
|---|---|---|
| Telent | Poisson distribution | - |
| New Network Transfer Protocol (NNTP) | Weibull distribution | $\text{Log}_2$-normal distribution |
| SMTP | - | $\text{Log}_2$-normal distribution |
| FTP | - | $\text{Log}_2$-normal distribution during the whole session<br>Pareto distribution during a burst session |
| Web | - | Pareto distribution |
| VoIP | Exponential distribution | Exponential distribution |

We referenced the combination of the traffic generation model and theoretical traffic model for generating the various types of virtual network traffic similar to the real network traffic as shown in Table 4. The IDT of all traffic is generated by using poisson distribution. $\text{Log}_2$ normal distribution is used for the generation of PS for SMTP and FTP traffic. Moreover, the PS of Web traffic is generated by the pareto distribution. According to the description of the theoretical traffic model, we generate each type of traffic with different percentages of the packet as shown in Table 5.

Table 4. Assumption for generating IDT and PS for each service

| **Service** | **IDT** | **PS** |
|---|---|---|
| NTP, DHCP, DNS | Poisson distribution | - |
| SMTP | Poisson distribution | $Log_2$-normal distribution |
| FTP | Poisson distribution | $Log_2$-normal distribution |
| Web | Poisson distribution | Pareto distribution |

After defining the traffic generation format, firstly we setup the NTP, DHCP, DNS, SMTP, FTP, and Web server on the mininet host h1, h2, h3, h4, h5, and h6, respectively. Then, all client hosts except the attacker host h18 access the server concurrently according to the traffic generation format including specific IDT and PS for each service as shown in Table 4 and the packet generation rate as shown in Table 5.

Table 5. Assumption for packet generation rate

| **Service** | | **Percentage of the packet (%)** |
|---|---|---|
| TCP | Web | 70 |
| | FTP | 10 |
| | SMTP | 5 |
| UDP | NTP | 5 |
| | DNS | 5 |
| | DHCP | 5 |

Table 6. List of baseline for each service

| **Service** | **Baseline** |
|---|---|
| NTP | 64.60285 |
| DHCP | 76.64288 |
| DNS | 8.8075 |
| SMTP | 4.761905 |
| FTP | 4.761905 |
| Web | 4.761905 |

In order to get all possible event information from the sFlow analyzer, the initial baseline and threshold value is defined as zero and the DDoS Mitigation application is not activated in the ONOS controller. The final value for the baseline traffic of each service is defined according to the process and equation which previously described in section 3.2. By applying equation (6), the value for the baseline traffic of the NTP service $b_{NTP}$ is 64.60285 which is the maximum number of frames per second of the service. Similarly, the baseline values for the other services are defined as the NTP service and these values are listed in Table 6.

## 6.2. Mitigation Phase

The mitigation phase is implemented in the DDoS Mitigation application in the ONOS controller. It has two main functions in the application: regular taking event information from the sFlow analyzer, and installation of drop flow rule according to the information in the event information. The MATA-based application installs drop flow rule as soon as it received event information. However, the ATA-based application installs drop flow rule according to the predefined number of events within a period of time as described in Table 2.

### 6.3. Scenario

To evaluate this system with various types of experimental results, a scenario is used for testing the virtualized SDN network with flooding attacks and captures the monitoring results of it. The duration for testing and evaluation time is three minutes. During this time, all client hosts are accessing all available servers in the network concurrently and one attacker host launches the flooding attack for one minute. This scenario includes four steps:

Step 1: We set up NTP, DHCP, DNS, SMTP, FTP, and Web server on the host h1, h2, h3, h4, h5, and h6, respectively.

Step 2: After setting up the servers, all clients (from host h7 to h17) access all the servers concurrently for three minutes (18000 seconds). At the same time, we monitor the victim server by capturing all of its incoming and outgoing traffic with a packet capturing tool (i.e. tcpdump) [31].

Step 3: After one minute from the start of monitoring, attacker host h18 launches the flooding attack to a particular victim server for one minute by using hping3 tool [32].

Step 4: After one minute attack, we check the number of flooding packets that can be able to filter by this system. Since this flooding attack detection and mitigation system installs flow rule in the ingress switch s4 of the attacker hosts h18, we check the number of packet in the drop flow rule at the switch s4.

### 6.4 Performance Parameter

The value of each performance parameter for network security is defined according to the result from the scenario.

- True positive (TP): The number of packets passing through the drop flow rule is defined as the value of TP.
- False Negative (FN): The number of packets from the filtering traffic (i.e. from the attacker host h18 to victim server host) including in the results of packet capturing tool is considered as the value of FN.
- True Negative (TN): The number of packets getting from the subtraction of the number of the packet of all capturing traffic from the captured result to the value of FN.
- False Positive (FP): The value of FP is zero because this system is implemented with the avoidance of false alarms mechanism.

After defining the value of each performance parameter, the rate of false negative rate (FNR), detection rate (DR), and accuracy(ACC) is calculated by using the formula as shown in the equations (8), (9), and (10), respectively.

$$FNR(\%) = \frac{FN}{FN + TP} * 100 \tag{8}$$

$$DR(\%) = \frac{TP}{TP + FN} * 100 \tag{9}$$

$$ACC(\%) = \frac{TP + TN}{TP + TN + FP + FN} * 100 \tag{10}$$

# 7. EXPERIMENTAL RESULT

Since this system is implemented with the detection phase and mitigation phase, the experimental results for these two phases are separately described in two sub-sections: detection results and mitigation results.

## 7.1. Detection Results

The detection results consist of three parts. The first part presents the dynamic threshold values adaptable with the incoming traffic produced by each algorithm. The second part shows the comparative results of various performance parameters (i.e. detection rate, false negative rate, and accuracy) over MATA and ATA algorithms to prove why the modified algorithm is chosen to use in detecting the flooding attacks. The evaluation results of the MATA with various rates of the attack are described in the third part of this section for indicating how the MATA can detect the various types of attacks.

### 7.1.1. Comparative results of adaptive threshold over incoming traffic

Figures4 and 5 show the DNS traffic's adaptive threshold dynamically produced by the sFlow analyzer based on the ATA and MATA, respectively. The main difference between the results of the two algorithms is that the false alarms can be seen at the initial state and normal state of the result produced by ATA while the MATA does not produce any of them. The reason is that the MATA reduces the occurrence of false alarms in the detection phase by using its modified technique. However, ATA raises some false alarms in the detection phase and avoids them in the mitigation phase. Each comparative result is divided into three states: initial state (before 10s), attack state (60s - 120s), and normal state ((11s - 59s) and (121s – 180s)).

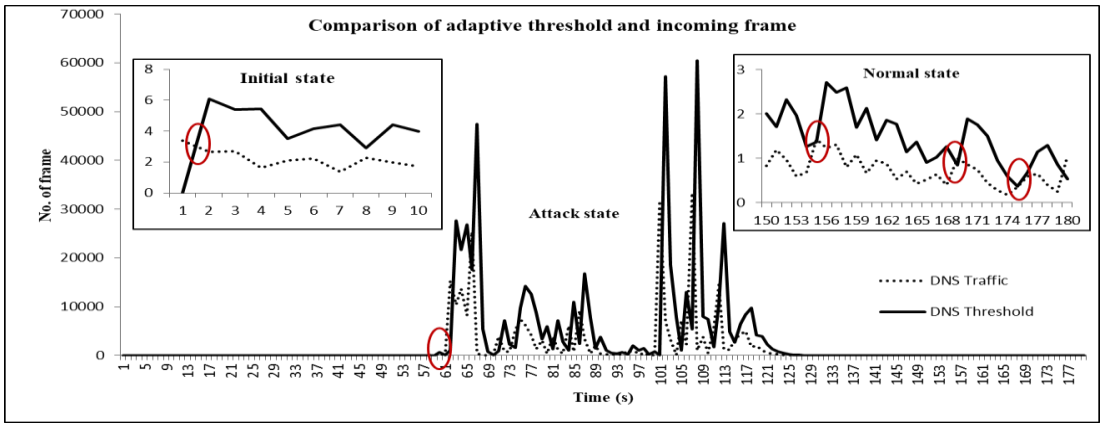


Figure 4. Adaptive threshold produced by ATA

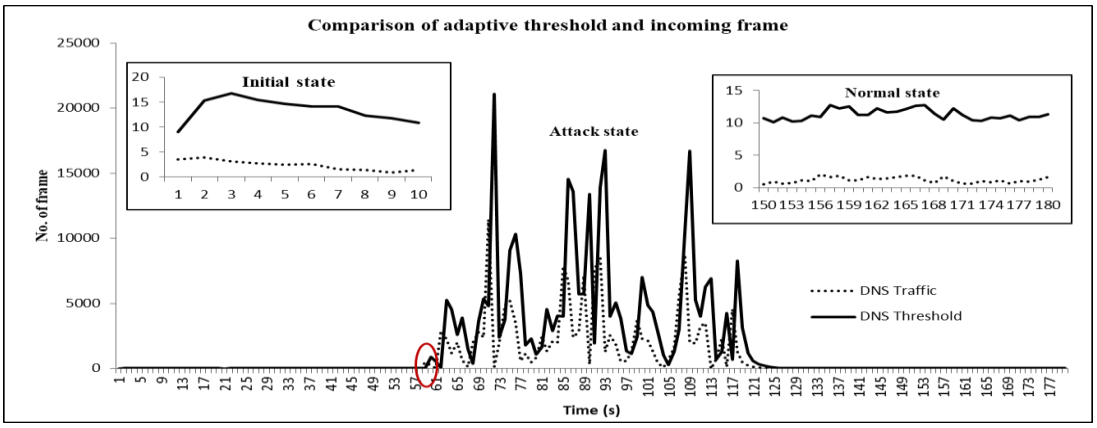


Figure 5. Adaptive threshold produced by MATA

***Initial state:*** In the ATA result, the threshold value is initialized as zero because of no traffic at the beginning. Thus, as soon as the traffic is incoming into the network, the threshold is violated by the incoming traffic as shown in the initial state of Figure 4.In the MATA result, the initial threshold value is defined as the value of the baseline. As a result, there is no false alarm at the initial state as shown in the initial state of Figure 5.

***Attack state :***According to the result of the attack state of each figure, both algorithms could detect the flooding packets immediately when the attack is launching by the attacker.

***Normal state :***In the ATA result ,the false alarms might be raised when the current rate of the incoming frames is slightly stronger than the previous rate as shown in the normal state of Figure 4. In the MATA result, there is no false alarm in this state because the minimum threshold value itself is the same as the baseline traffic and then the threshold values are adaptable with the incoming frames.

### 7.1.2. Comparative results of performance parameters

In general, the performance parameters used in the evaluation of the network security include detection rate, false positive rate, false negative rate, and accuracy. Thus, this system also evaluates its performance by using general performance parameters. However, it does not describe the false positive rate because this system reduces them in its implementation (i.e. the false positive rate is zero). The results are produced by calculating the performance parameters as described in section 6.4. Moreover, the average percentage of each performance parameter is obtained by calculating the average value of ten runs.

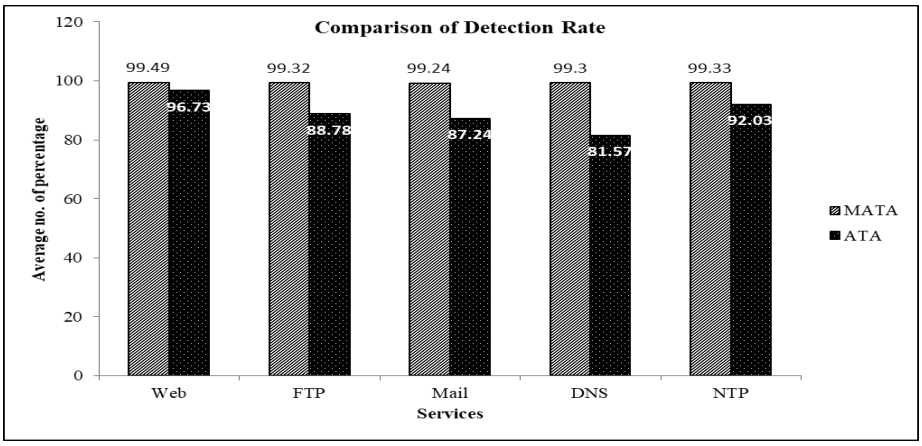


Figure 6. Comparisons of detection rate over various types of services

The comparison of detection rate produced by MATA and ATA over various types of services is described in Figure 6. The detection rate of MATA is slightly higher than the rate of ATA and the rate of the two algorithms is not extremely different because this system modified the original ATA especially in reducing the false negative rate while avoiding false alarms.

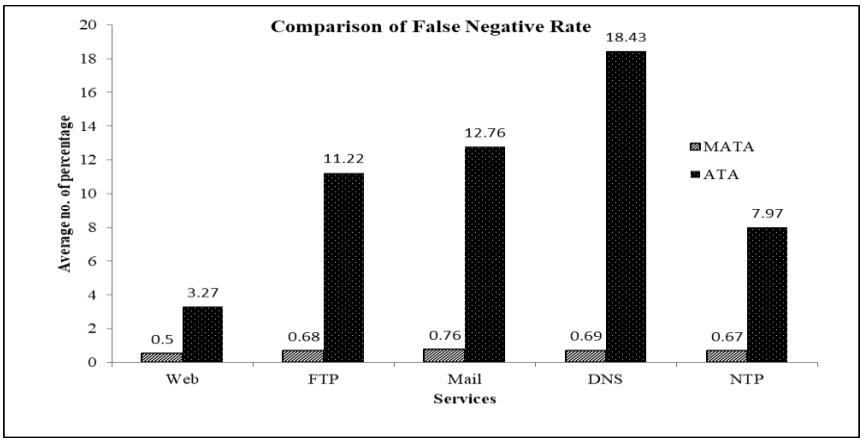


Figure 7. Comparisons of false negative rate over various types of services

The comparative result of the false negative rate for each service isshown in Figure 7. ATA produces the different false negative rates for the services because it distinguishes the normal and malicious packets by considering the number of continuous incoming packets within a time andthe incoming rate of continuous packets is different depending on the type of services.

In this figure, ATA produces a high value of FNR in DNS service because the incoming normal DNS packet rate is higher than the other services and the number of continuous incoming packets during the two seconds is about 6. Thus, ATA defines the incoming packet as the abnormal one when the number of the continuous incoming packet within the two seconds is more than 6 packets as listed in Table 2. As a result, the number of the attack reaches the victim DNS server is high when the network is under attack. In contrast, the number of continuous incoming SYN packets within two seconds is 1 for the Web service. Since the normal incoming packet rate itself is very low, the attack can be detected early as soon as the number of the packet is greater than 1. However, MATA produces similar false negative rates for all services because it uses the same definition based on their baseline to differentiate the normal and malicious packets.

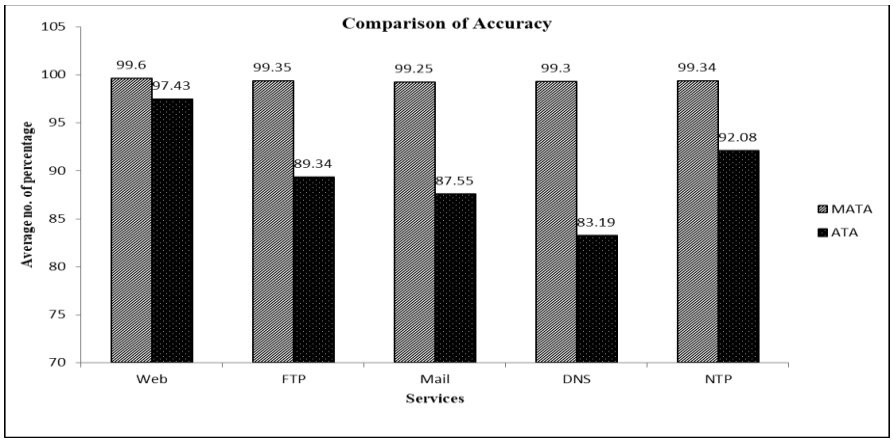


Figure 8. Comparisons of accuracy over various types of services

Similarly, ATA produces a different percentage of accuracy for the services because the calculation of accuracy is also depending on its false negative rate. It provides about 97.4% as its maximum percentage that can be seen in Web service. The minimum percentage provided by ATA is around 83.2%. However, MATA produces an accuracy above 99% for each service as shown in Figure 8.

By reviewing the comparative average number of percentage for detection rate, false negative rate and accuracy produced by MATA and ATA, MATA is an appropriate algorithm for the detection and mitigation of the flooding attacks because it provides a higher percentage of detection rate and accuracy, and a lower percentage of false negative rate for all services than ATA.

### 7.1.3. Comparisons of performance over the various attack rates

Depending on the rate of the attack, the percentage of the detection rate and false negative rate is different. As this system is implemented the flooding attack at the transport layer, the performance comparison of attack rate is described separately for TCP and UDP protocols. Web and DNS service is used to represent the TCP and UDP protocol, respectively. Five different rates of attack (i.e. 10 packets per second, 100 packets per second, 1000 packets per second, 10000 packets per second, and 10000 packets per second) are used to test the performance of the system. For testing each rate of attack, hping3 command is used with u100000, u10000, u1000, u100, u10 and u1 to send the attack packet with 10 packets per second, 100 packets second, 1,000 packets per second, 10,000 packets per second, and 100,000 packets per second, respectively. These results are produced by averaging the results of ten runs.

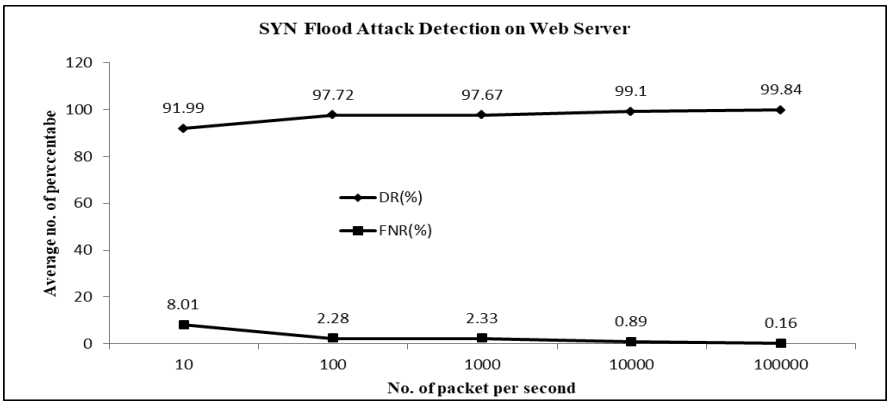


Figure 9. Performance comparisons of various attack rates over Web service

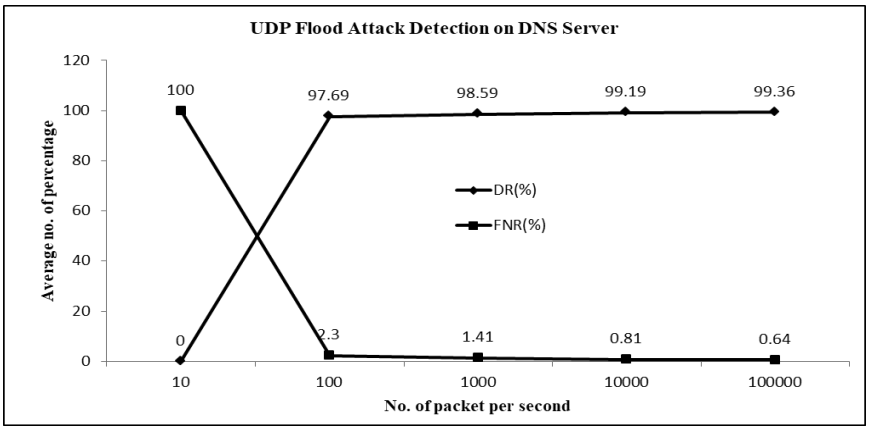


Figure 10. Performance comparisons of various attack rates over DNS service

According to the results of the detection rate and false negative rate as shown in Figures9 and 10, the detection rate of the MATA algorithm can be determined as the higher the attack rate, the higher the detection rate. In contrast, the false negative rate can be defined as the higher the attack rate, the lower the false negative rate. Although the detection mechanism using this algorithm can detect all attack rates for Web traffic, it is not capable to detect the lowest rate of attack (i.e. 10 packets per second) for DNS traffic because the rate of the normal packet of the UDP protocol itself is high. However, it can be starting to detect the attack with the rate of 100 packets per second.

## 7.2. Mitigation Results

The mitigation results consist of two parts: filtering results, and network performance. The filtering results are produced from two types of comparison: the comparison of network traffic with and without filtering by using DDoS Mitigation application, and the comparison of the percentage of attack packet reaches the victim servers. Moreover, the performance of the network during attack filtering is measured to prove that the source-based Defense mechanism is more effective than the destination-based one.

### 7.2.1. Filtering results

The DDoS Mitigation application drops the abnormal traffic depending on the alert information obtained from the sFlow analyzer. The application takes the information from the analyzer via

REST API every second. Thus, the maximum delay time between the detection and mitigation is one second.

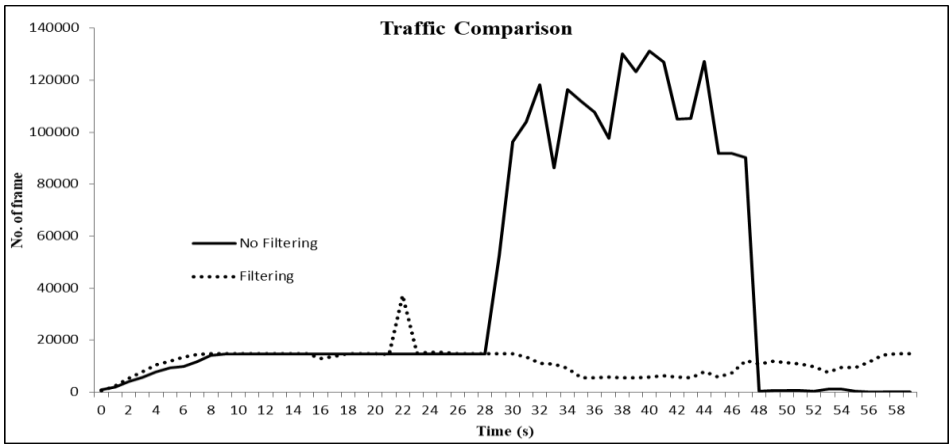


Figure 11. Comparison of network traffic with and without filtering

By filtering the network with the DDoS Mitigation application, the normal users can access a particular service without interrupting even though when the server is under attack. In contrast, without filtering the network with the application, the server can be down as soon as it is under attack, and the service will no longer be available for normal users. The comparative result of filtering the network with and without DDoS Mitigation application is shown in Figure 11. These results are produced from the I/O graphs of the captured results of the network for one minute.

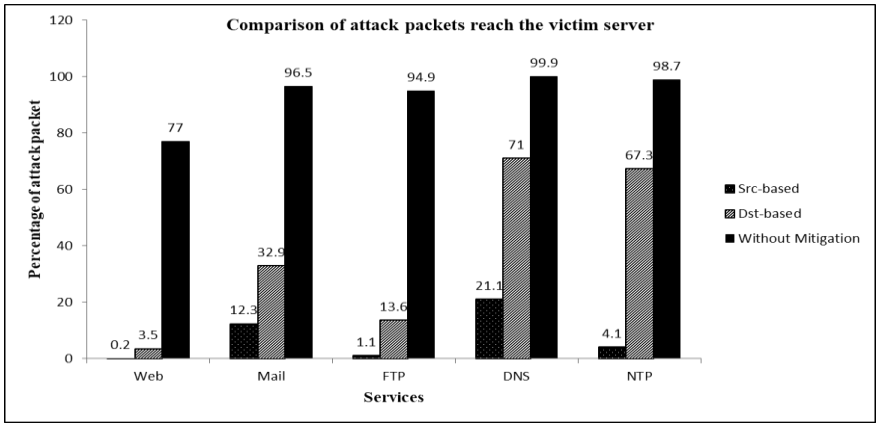


Figure 12. Comparison of attack packets reach the victim servers

Moreover, Figure 12 describes the percentage of the attack packet that reaches the victim server while the network is filtering with the DDoS Mitigation application implemented with two different defense mechanisms (i.e. source-based defense mechanism and destination-based mechanism). To decide how many percentages of the attack packet that can be reduced by each mechanism, the figure also describes the percentage of the attack packet that reaches the victim server when the network is not filtering with the DDoS Mitigation application

.

Table 7. Reduction of attack packets reach the victim servers

| Mechanism | Web | Mail | FTP | DNS | NTP |
|---|---|---|---|---|---|
| Source-based defense | 76.6 | 84.2 | 93.8 | 78.8 | 94.6 |
| Destination-based defense | 73.5 | 63.6 | 81.3 | 28.9 | 31.4 |

By reviewing the results of the percentage of attack packet that reaches the victim server, the source-based defense mechanism could reduce the attack packet up to 94.6% while the destination-based mechanism only reduce them to 31.4% for NTP service as listed in Table 7. Thus, the source-based defense mechanism is more effective than destination-based defense mechanism for the flooding attacks with non-spoofing IP address.

### 7.2.2. Network performance

Figure 13 shows the comparative results ofaverage network performance while the network is being in attack and filtering with source-based and destination-based defense mechanism. The performance is measured by pinging with ten packets from client host h8 to h17. The average performance is also calculated by monitoring the ten times of average time to live (i.ettl) from pinging and averaging the results.

Since the former mechanism is dropping the attack packets nearest to the source of the attack, the network will not be congested with those attack packets.Thus, the source-based defense mechanism maintains higher performance than the destination-based defense mechanism. As shown in Figure 13, the latency of the source-based defense mechanism for each service is about doubling the latency of the destination-based defense mechanism.

However, the source-based defense mechanism can only protect the direct attack because it must know the exact location of the attack so that it can install drop flow rule into the ingress switch of the attack host. If it does not know the location of the attack, the destination-based mechanism is preferred. Although this method can protect the attack with spoof IP addresses, the network might be congested because of attack traffic.

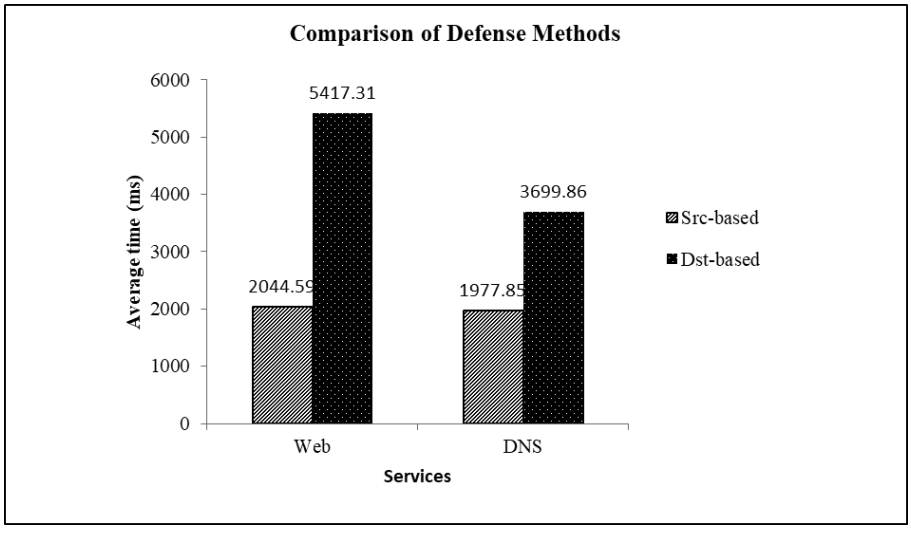


Figure 13. Comparative results for network performance during the network is being in attack

## 8. CONCLUSIONS

Flooding attack might fail the whole SDN network or services during a short period of time. Our proposed flooding attack detection and mitigation using the MATA algorithm could detect and mitigate the attack effectively by modifying the original ATA with the consideration of the baseline info of the network infrastructure.

By using the MATA algorithm, the false negative rate is reduced up to 0.7% and the accuracy is increased around 99% for all network services. Although this method has a little overhead for finding the baseline traffic info of the network infrastructure, it considerably reduces the false alarms by producing the dynamic and adaptive threshold based on the baseline. Consequently, the false negative rate is significantly reduced because the attack might be discarded as soon as the DDoS Mitigation application received the abnormal event information. Moreover, since the application regularly takes the event information from the sFlow analyzer in every second, the maximum delay time between detection and mitigation is one second. Therefore, we can conclude that MATA is an effective algorithm for the various types of flooding attack detection and mitigation

.

## REFERENCES


[1] Braga, R., de Souza Mota, E. and Passito, A., (2010, October). "Lightweight DDoS flooding attack detection using NOX/OpenFlow". In LCN, Vol. 10, pp408-415.

[2] Hu, D., Hong, P., & Chen, Y. (2017, December). "FADM: DDoS flooding attack detection and mitigation system in software-defined networking". In GLOBECOM 2017-2017 IEEE Global Communications Conference pp1-7.

[3] ONF, (2012) "Software-defined networking: The new norm for networks," ONF White Paper, vol. 2, pp2–6.

[4] Dharma, N.G., Muthohar, M.F., Prayuda, J.A., Priagung, K. and Choi, D., (2015, August). "Time-based DDoS detection and mitigation for SDN controller". In 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS), pp550-553. IEEE.

[5] Bawany, N. Z., Shamsi, J. A., & Salah, K. (2017). "DDoS attack detection and mitigation using SDN: methods, practices, and solutions". Arabian Journal for Science and Engineering, 42(2), pp425-441.

[6] Kumar, P., Tripathi, M., Nehra, A., Conti, M., & Lal, C. (2018) "SAFETY: Early detection and mitigation of TCP SYN flood utilizing entropy in SDN". IEEE Transactions on Network and Service Management, 15(4), pp1545-1559.

[7] Ubale, T., & Jain, A. K. (2018, March). "SRL: An TCP SYNFLOOD DDoS Mitigation Approach in Software-Defined Networks". In 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp956-962. IEEE.

[8] Mohammadi, R., Javidan, R., & Conti, M. (2017). "Slicots: An sdn-based lightweight countermeasure for tcpsyn flooding attacks". IEEE Transactions on Network and Service Management, 14(2), pp487-497.

[9] Wei, H. C., Tung, Y. H., & Yu, C. M. (2016, June). "Counteracting UDP flooding attacks in SDN". In 2016 IEEE NetSoft Conference and Workshops (NetSoft) pp367-371. IEEE.

[10] Siris, V. A., &Papagalou, F. (2004, November). "Application of anomaly detection algorithms for detecting SYN flooding attacks". In IEEE Global Telecommunications Conference, 2004. GLOBECOM'04, Vol. 4, pp2050-2054. IEEE.

[11] Giotis, K., Argyropoulos, C., Androulidakis, G., Kalogeras, D., &Maglaris, V. (2014). "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments". Computer Networks, 62, pp122-136.

[12] Wang, R., Jia, Z., & Ju, L. (2015, August). "An entropy-based distributed DDoS detection mechanism in software-defined networking". In 2015 IEEE Trustcom/BigDataSE/ISPA, Vol. 1, pp310-317. IEEE.

[13] Mehdi, S. A., Khalid, J., &Khayam, S. A. (2011, September). "Revisiting traffic anomaly detection using software-defined networking". In International workshop on recent advances in intrusion detection, pp161-180. Springer, Berlin, Heidelberg.
[14] Dotcenko, S., Vladyko, A., &Letenko, I. (2014, February). "A fuzzy logic-based information security management for software-defined networks". In 16th International Conference on Advanced Communication Technology, pp167-171. IEEE.
[15] Priyadarshini, R., & Barik, R. K. (2019). "A deep learning based intelligent framework to mitigate DDoS attack in fog environment", Journal of King Saud University-Computer and Information Sciences.
[16] Phan, T. V., Van Toan, T., Van Tuyen, D., Huong, T. T., & Thanh, N. H. (2016, July). "Openflowsia: An optimized protection scheme for software-defined networks from flooding attacks". In 2016 IEEE Sixth International Conference on Communications and Electronics (ICCE) pp13-18. IEEE.
[17] Nam, T. M., Phong, P. H., Khoa, T. D., Huong, T. T., Nam, P. N., Thanh, N. H., ... &Loi, V. D. (2018, January). "Self-organizing map-based approaches in DDoS flooding detection using SDN". In 2018 International Conference on Information Networking (ICOIN) pp249-254. IEEE.
[18] Kalliola, A., Lee, K., Lee, H., & Aura, T. (2015, October). "Flooding DDoS mitigation and traffic management with software-defined networking". In 2015 IEEE 4th International Conference on Cloud Networking (CloudNet), pp248-254. IEEE.
[19] Latah, M., &Toker, L. (2018). "A novel intelligent approach for detecting DoS flooding attacks in software-defined networks". International Journal of Advances in Intelligent Informatics, 4(1), pp11-20.
[20] Conti, M., Gangwal, A., & Gaur, M. S. (2017, October). "A comprehensive and effective mechanism for DDoS detection in SDN". In 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp1-8. IEEE.
[21] Conti, M., &Gangwal, A. (2017, November). "Blocking intrusions at border using software defined-internet exchange point (sd-ixp)".In 2017 IEEE Conference on Network Function Virtualization and Software-Defined Networks (NFV-SDN), pp1-6. IEEE.
[22] Arins, A. (2015, November). "Firewall as a service in SDN OpenFlow network", In 2015 IEEE 3rd Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE) pp1-5. IEEE.
[23] Özçelik, M., Chalabianloo, N., &Gür, G. (2017, August). "Software-defined edge defense against IoT-based DDoS".In 2017 IEEE International Conference on Computer and Information Technology (CIT), pp308-313. IEEE.
[24] Lu, Y., & Wang, M, (2016, June)"An easy defense mechanism against botnet-based DDoS flooding attack originated in SDN environment using sFlow". In Proceedings of the 11th International Conference on Future Internet Technologies, pp14-20. ACM.
[25] sFlow-RT, May 2014 [Online]. Available from: https://www.inmon.com.
[26] ONOS [Online]. Available from: https://onosproject.org.
[27] Mininet [Online]. Available from: http://mininet.org.
[28] D-ITG Tool [Online]. Available from: http://www.grid.unina.it/software/ITG/
[29] Traffic generation model, https://en.wikipedia.org/wiki/Traffic_generation_model
[30] Avallone, S., Emma, D., Pescapé, A., &Ventre, G. (2005). "Performance evaluation of an open distributed platform for realistic traffic generation", Performance Evaluation, 60(1-4), pp359-392.
[31] Tcpdump [online]. Available from: https://www.tcpdump.org/manpages/tcpdump.1.html.
[32] Hping3 Security Tool [online]. Available from: http://www.hping.org/hping3.html.