Detection and Mitigation of DDOS Attacks in Software Defined Networks: A Survey

Fawad Ahmad[1], Nadir Pervez[2], M. Junaid Arshad[3] and Rizwan Ali[4]

[1]Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, Lahore, Pakistan
[2,3,4]Computer Science and Engineering Department, University of Engineering and Technology, Lahore, Pakistan
[1]fawad.ahmad@kics.edu.pk, [2]ne.nadir@gmail.com, [3]junaidarshad@uet.edu.pk, [4]rizwanhvl@gmail.com

***Abstract*– DDOS Attacks have been a serious threat to the Network Security for the last few years. With the introduction of SDN, Although the problem of DDOS Attacks can easily be resolved in traditional networks as it provides a central control plane paradigm for such attacks in the network but due to the separate control plane it also increases the SDN-self DDOS threats. In this paper, we first look at the algorithms supported by SDN for the detection of DDOS in traditional network. Then we describe the different types of algorithms for the detection and mitigation of SDN-self DDOS threats. Also, we classify the different algorithms based on selected approaches and compare them.**

***Keywords*– Distributed Denial of Service, SDN, Detection Algorithm, Mitigation Techniques, SDN Security and SDN-Self DDOS Attacks**

I. INTRODUCTION

Internet Security has been facing serious threating challenges by Distributed Denial of Service (DDOS) Attacks from the past few years. A DDOS attack is an attempt to make the network resources like CPU, Memory and Network Bandwidth unavailable for intended users. The attacker relies on sending a large number of attack packets to the victim with the aim to make the resources of the network unavailable for the benign users. Several user friendly tools are available to facilitate this attack such as Stacheldraht [1]. The detection for this kind of attack is very hard task, moreover with the use of Internet protocol (IP) Spoofing technology which is used to hide the flooding source and attacker's location it's even more difficult to trace the attacker. To cope with these issues several detection and mitigation techniques have been proposed [2]-[8]. However due to their deployment complexities and operational costs only few of them have been deployed completely. One of the main reason is that these techniques require placing large connection state tables at network devices like routers and switches which needs high end equipment at each device or node ultimately leads to extra storage and computational costs. Moreover, in traditional networks it is very difficult for the network administrator to manually handle each node and detect the attacks happening. So, there is a need for a lightweight, automated and scalable method to cope with these attacks.

Software defined networking is a new paradigm for the computer networks. Unlike traditional network it has one central control for the whole underlying network and hence the network management for the administrator is very easy task. We can scale, modify and change our network topology and it has not put any burden on the network administrator. The administrator can easily detect the victim node of the DDOS attack in the network and can perform actions accordingly. We have highlighted different SDN supported detection and mitigation techniques that are used in traditional networks. In SDN lightweight switches and routers are used that can only performs forwarding decisions and the control part of these devices are merged to the controller. Changes can be made on the controller by writing scripts or programs to the control logic and instructions are given to the switches that perform only forwarding tasks based on these instructions. The overall costs for deploying such network are very low because switches are made lighter and cheaper as they only perform forwarding.

However instead of advantages of the SDN, it has put new faults and attack points to the network security. As there is only one control plane, the DDOS attack on the SDN controller can cripple the whole network. In this paper we have provided different detection and mitigation techniques against each of the SDN components i.e., the controller, the channel between controller and switch and the switches.

We have divided our paper in to seven sections. Section II discusses the SDN technology and DDOS threats in detail. Section III describes the methods for detection and mitigation of DDOS attacks in traditional networks with the help of SDN technology. In section IV challenges to the SDN-self DDOS attacks are discussed. Section V represents the methods available for detection of SDN-self DDOS attacks. In section VI defence methods for SDN-self DDOS attacks are presented. Section VII concludes the paper.

II. DDOS ATTACKS AND SOFTWARE DEFINED NETWORK TECHNOLOGY

A) Distributed Denial of Service Attacks

Unlike DOS attack where the large number of attack packets are coming from one source computer, In DDOS attack, the attacker attacks by compromising the large number

of computers of the same network called as bots or zombies There are total five major components of DDOS attack: (1) The attacker/ master computer that initiates the attacks, (2) the handlers/controlling computers which issue the commands to the zombies/agents and the reflectors, (3) the zombies which are the infected computers and from which instructions propagates, (4) the reflector/amplifier which amplifies the incoming requests from zombies in numbers and (5) the victim server to which all the requests are sent. Fig. 1 illustrates this scenario.

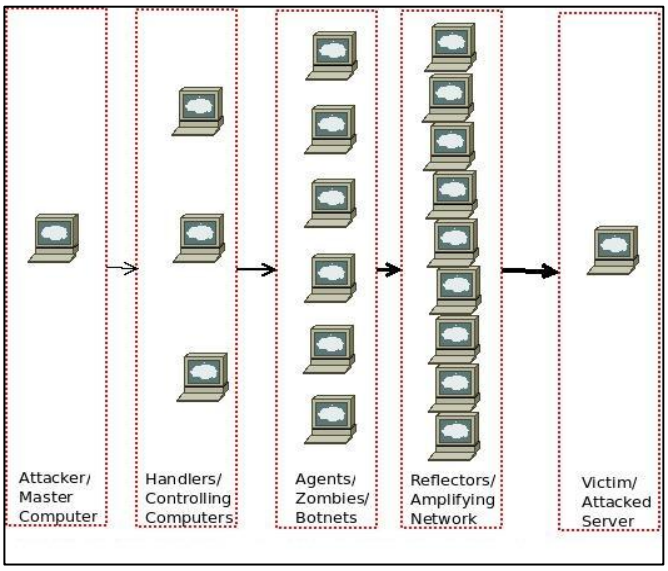

Fig. 1. Architecture of Distributed Denial of Service Attack

There are several types of DDOS attacks as discussed follow:

ICMP Flood Attack:

A large number of Internet Control Message Protocol (ICMP) echo request or ping request are sent to the victim server with Spoofed source IP address to exhaust its resources.

UDP Flood Attack:

A continuous flood of user datagram protocol (UDP) is sent to victim on random or specific ports.

TCP SYN Attack:

A large number of Transmission Control Protocol (TCP) Synchronize (SYN) packets are sent to the victim server. After receiving the SYN messages it replies with SYN +ACK packets and waits for the ACK packets from the source. The attacker doesn't send the ACK packets and so server waits for non-existent ACK packets. The server's limited buffer queue becomes full and legitimate ACK are rejected due to congestion.

Smurf Attack:

This is a reflection and amplification type of ICMP flood attack in which targets are routers and servers. The spoofed ICMP packets whose source IP address is of victim server IP address are sent to routers and other hosts. The number of so that the attack can be distributed all over the network. replies back to the victim spoofed IP address determines the flood traffic. This attack unlike UDP and ICMP Flood is difficult to trace out.

Fraggle Attack:

This attack is similar to smurf attack except instead of sending ICMP packets UDP packets are sent to the victim.

Coremelt Attack:

In this attack, zombies are divided into two groups. The attacker instructs zombies of one group to communicate with other group and this makes ultimately a huge flood of traffic. It is difficult to trace out this attack as flood creates with the help of legitimate users.

HTTP Flood:

It is a volumetric attack in which web server is flooding Hyper Text Transfer Protocol (HTTP) request packets.

B) Software Defined Network

Software defined networking is a new paradigm in computer networks. Unlike traditional networks where data and control plane reside on the same device, in SDN control plane of all the networking devices like switches and routers etc. is decoupled from the data plane and merged into a central node. The components of the SDNs are the switches that are called open flow switches, the controller and the open standard protocol called Open Flow protocol. SDN divided into three layers: 1) Application Layer, 2) Control Layer and 3) Infrastructure Layer. The whole architecture of the SDN is shown in Fig. 2.

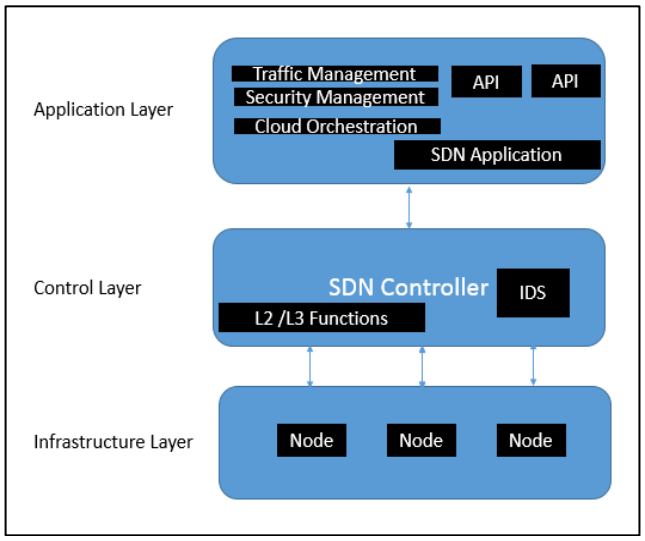

Fig. 2. Architecture of Software Defined Network

The application layer performs the functions traffic management, security management etc. Actually, it contains the different application those are running on the SDN controller. We can make our algorithm and installs it on the controller using Application Program Interface. SDN controller resides on the control layer which performs

different control functions of the network devices. All the routing protocols are running on this layer. The infrastructure layer contains different lightweight and inexpensive network devices like switches and routers etc. that only performs forwarding decision based on the instructions from the controller. These devices contain Tertiary Content Addressable Memory (TCAM) table also called Flow Table in which all the flow entries are stored. In SDN flow of the packet is as follows: when a new packet comes to the ingress of OF-switch it first checks whether there is a flow entry or any flow rule installed in TCAM against this flow. If it founds any flow rule then it forwards the packet to the destination. Otherwise it sends a flow request to the controller and expecting a response from it. The controller checks the validity of the packet and generate a new flow rule for this packet and sends it to instruct the OF switch for it. OF switch installs this new flow entry into its flow table and forwards the packet.

The architecture of an Open Flow switch is shown in Fig. 3.

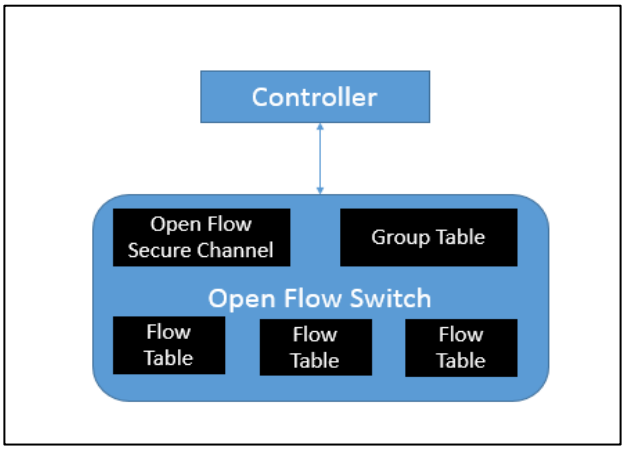

Fig. 3. Architecture of Open Flow Switch

OF switch contains different flow tables and group table that both perform packet lookup and forwarding of the packet. Communication between OF switch and the controller happens in a secure way using Open Flow secure channel. The flow table further contains some fields based on which OF switch differentiate different flows and maps flow rules on different switches. These entries are shown in Fig. 4.

Packet Header Matching Field	Priority	Counter	Cookies	Timeout	Instructions

Fig. 4. Flow Table Fields of OF-Switch

Packet header field contains the Layer 2 and Layer 3 information like VLAN ID, Protocol used, Source IP, Destination IP, TCP information like Source port, destination port and Ingress port etc. The first two fields in this table Packet header field and Priority field both uniquely identifies an entry in the flow table. Counter field counts the no. of packets coming from the same source. To filter the flow statistics cookies field is used by the controller. Timeout field indicates the remaining time of a flow entry in the table. Instructions field contains the flow rule instructions given by the controller to the OF switch.

III. SDN SUPPORTED DDOS ATTACKS

In this section, we present the up-to-date SDN supported techniques for DDOS detection [9], [10], [11] and DDOS Defence [12], [13], [14] in traditional networks.

A) DDOS Detection

At present there are many research on DDOS attacks detection and mitigation [2]-[8]. In [2] an effective and efficient defence approach against spoofed DDOS traffic is proposed in which they have considered the hop count parameter in the IP packet header which is presented by Time to Live (TTL) value field. They proposed that although the attacker modified the whole IP packet header using spoofed IP technology but one field called hop count cannot be changed by the attacker. Using the mapping between IP addresses and hop counts, the server can be able to distinguish between legitimate traffic and the spoofed IP packets. In [3] defence mechanisms against the two types of DDOS flooding attacks have been proposed: Application level DDOS attacks and Network/Transport level DDOS attacks. These mechanisms are classified based on two criteria. A first criterion is the deployment location where the defence is applied. Based on this criteria defence mechanism against the network/transport level DDOS is categorized into four types source based, destination based, network based and hybrid based and defence criteria for application level DDOS is destination based and hybrid based. Second criteria is the point of time on which certain DDOS attacks happens and based on this criteria defence mechanisms against both application and network/transport level DDOS attacks are categorized into three types: Before the attack(Prevention), During the attack (Detection) and After the attack (attack source identification). In [5] a detection strategy for the DDOS attack has been proposed in which flow similarity and super-points measurements are combined. By discussing the behaviour of DDOS attacks with super-points, apprehensive flows are located in a better detection scheme. In [6] a more sophisticated computer vision based anomaly detection mechanism for DDOS attacks is proposed. By using correlation analysis a mechanism for DDOS detection against data centres is proposed in [8]. Only few of these proposed mechanisms can be hardly applied in traditional networks. With the advantages of SDN, several SDN based flexible DDOS detection techniques have been proposed. In [9] a DDOS detection method called Cloud-Watcher is proposed for the security of Dynamic Cloud Networks. This method is actually an application running on NOX operating system which is the application part of the SDN controller. Two main challenges to the dynamic cloud networks are considered while designing the said algorithm. First challenge is that cloud network can be infected not only by outside network devices but by the internal consumers as VMs are shifted from one host to another in multi-tenant situation. Secondly

deployment of network security devices should depend on the dynamism of the cloud network. As in SDN we can monitor and control the network flows as we want. By considering this fact, the algorithm changes the routing paths of these flows and makes them pass through the network devices where security is implemented. Moreover, it provides a simple policy script language so that people can use their provided services easily. Also in [15] the method provides a fascinating script language to monitor the network traffic effectively and efficiently.

In [10] a new SDN based algorithm called FlowTrApp is proposed for the detection and mitigation of DDOS attacks in data centres. For detection and mitigation this algorithm uses s-Flow which is a tool to gather flow statistics from switches and Open-Flow technology. The algorithm works by using some bounds on two per traffic flow-based parameters that are flow duration and flow rate. It first checks the incoming traffic flow with the legitimate sample and if this flow not matches with the sampled legitimate traffic then it installs the mitigation actions.

Centralized SDN control architecture provides many new opportunities. Among the network management opportunity, measurement of the traffic flows is an important one.

Using these capabilities of SDN in [11] a DDOS attacks detection method is proposed which addresses these two mentioned challenges: 1) for detection accuracy, how to capture traffic rate feature and traffic rate asymmetry/deviation feature, 2) How to efficiently and collaboratively utilize the limited Tertiary Content Addressable Memory (TCAM) tables of the switches to monitor the traffic flow statistics of the whole network. Furthermore, a Sequential and Concurrent method is developed that quickly locate the potential victims and suspicious attackers by efficiently and adaptively changing the monitoring granularities on all the switches.

B) DDOS Defence

The separation of control plane from the data plane in SDN architecture allows network operators to dynamically steer the individual flows using control plane programmability. Network operators can easily enforce the security by using simple policy scripts. So SDN provides an efficient security policy implementation and recovers the overall security of the network.

Hence various defence schemes against DDOS attacks are proposed.

In [12] a flexible and elastic method for DDOS defence method named as Bohatei is implemented. While designing key challenges like scalability, responsiveness and resilience against the adversaries are considered. As an ideal defence mechanism should be flexible in sense that it should be placed anywhere in the network for attacks mitigation. And also, it should be elastic in sense that it should be applied based on type and scale of the attacks. So, this algorithm uses the capabilities of SDN for the flexibility function and Network Functions Virtualization (NFV) features for elastic behaviour. Moreover in [13] a similar type of work is presented. By using the advantages of SDN and NFV, the authors of this paper discussed how flexibility and elasticity is incorporated into the tradition network. Using the benefit of these technologies, they illustrate three design patterns: controller centric, VNF centric and hybrid approach by taking an example of state-full firewalling. By taking comparisons of these approaches they give a guideline for deploying SDN and NFV based appliances in traditional network.

In [14] a SDN based distributed and automate framework for DDOS mitigation is implemented. The proposed framework is distributed between an ISP network and consumer network and provides on demand DDOS mitigation services to the customers. The implementation of this algorithm considers three assumptions: 1) on demand DDOS mitigation service is provided to the customers and to achieve that ISP and customers have to cooperate with each other on an agreement. 2) This framework considers that two SDN controllers are running on each network (i.e. on customer and ISP) and ensures that on local SDN controller (customer end) DDOS detection module is running so that customers privacy leakage is avoided. 3) Both SDN controllers are running and communicating with each other in a secure way. Algorithm works in a systematic way, both controllers monitor the traffic coming from the customer's network. Also, detection module on customer's end is detecting against DDOS attacks. If customer's network detects the attacks, it requests for mitigation service from ISP and accordingly actions from ISP are performed to resolve the issues.

We summarize all the techniques proposed for DDOS detection and mitigation in Table I.

Table I: SDN Supported Detection & Mitigation Techniques

SDN Supported	Algorithms Used	SDN Capabilities Exploited
DDOS Detection Mechanism	Cloud Watcher [9]	Programmability
	Open-Safe [15]	Programmability
	FlowTrApp [10]	Traffic Analysis
	Sequential &Concurrent Method [11]	Global Monitoring
DDOS Defence Mechanism	SDN/NFV based security policy [12, 13]	Centralized control and programmability
	Distributed & collaborative framework [14]	Centralized control and programmability

IV. SDN-SELF DDOS ATTACKS

We have seen how SDN has solved the problem of DDOS attacks in traditional network by providing centralized control plane and hence central management of the whole network. Due to this feature there are many emergent vendors of this technology like Verizon, AT&T etc. But besides these advantages this technology itself becomes a threat for the

network security. For example, in SDN when a new packet comes to the OF switch it first checks whether there is an entry in the flow table and any flow rule for this packet. If it founds any flow rule then it forwards and routes the packet to the destination. Otherwise it sends the request to the controller requesting a new flow rule from it. The packet sent to the controller by switch is called packet-in-message and controller replies with a flow-mod-message to the switch to install a new flow for this packet. An attacker can make use of these SDN characteristics to attack on the switch, on the data to control channel and finally to the controller. These three types of attacks on the SDN are depicted in Fig. 5.

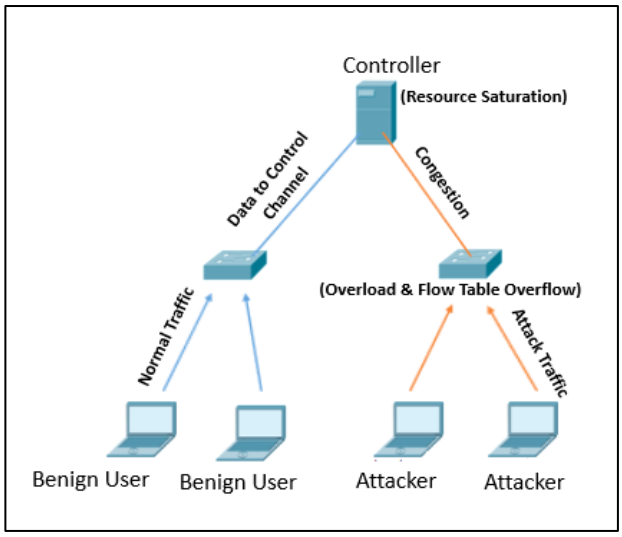

Fig. 5. Classification of DDOS attacks on SDN

A) Switch Overload & Flow Table Overflow

The attacker can send large number of packets to the victim switch. By receiving these packets switch buffer these requests and generates flow requests to the controller. As switch has limited CPU and memory so it can only buffer and requests some packets. So, switch becomes overloaded. Moreover, if the controller executes these requests then all flows must be saved to its TCAM tables and all the previous flow entries must be flushed out. In this way flow table entries are overflown.

B) Congestion in Data to Control Plane

Switch sends the flow request to the controller by sending the packet header information. If the switch's buffer becomes full because of the large number of request it sends the whole packet to the controller instead of only packet header. This result into requiring more bandwidth to pass these requests to the controller hence ultimately leads to congestion in the data to control plane.

C) Controller Resource Saturation

Finally, when all requests arrives at the controller and if no protection mechanism is implemented in the controller, the controller utilizes its resources like CPU memory etc. to execute these requests. As a result, the legitimate requests are dropped by the controller due to these flood requests leading to resource saturation in the controller side.

V. DETECTION MECHANISMS FOR SDN-SELF DDOS ATTACKS

The first and immediate step to tackle with the DDOS attacks is to detect them. Only after the detection we can mitigate it from the network. So, detection of these attacks is the first important countermeasure step. Based on the classification of the DDOS attacks in SDN we can detect them on either of these three points. The main vulnerability in SDN is the controller that is also a single point of failure in the network. Various techniques are proposed to detect the DDOS attacks on the controller. We have further classified these techniques into machine learning based, entropy based and graphic based.

A) Machine Learning Based Techniques

In machine learning based techniques two possible approaches are there as supervised learning and unsupervised learning. In supervised learning we give a whole sequence of traffic features to our algorithm to make differentiation between legitimate traffic and the malicious one. This type of techniques can easily be implemented as these require less processing. However, there is also a possibility that the given dataset of traffic feature may involve hidden malicious contents. So, these techniques become failed in these situations. In second approach that is in unsupervised learning the algorithm itself classify the traffic based on some intelligent mechanisms like Self Organizing Maps (SOM) etc. [16]. There is some processing overhead in these techniques but less false positives. Various techniques are proposed to detect DDOS attack detection we have highlighted some of the popular ones.

In [17] authors have proposed a light weight method based on traffic flow for the detection of DDOS attack in SDNs. The whole procedure is implemented using the NOX network controller where OF switches have all active flow statistics and NOX controller accessed all the needed features from the OF switches in an efficient way using a secure channel between them. Then the flow statistics are passed to an intelligent and efficient detection mechanism. The proposed method mainly consists of three modules: 1) Flow Collector, 2) Feature Extractor and 3) Classifier. Flow collector module collects the flow statistics of traffic from the open flow switch. Then extractor module extracts the features from the flow statistics of switches by using intelligent SOM mechanism and finally classifier classifies whether the traffic is normal or malicious.

In [18] to find the sets of hosts that have normal or abnormal behaviour, different machine learning algorithms such as Naive Bayes, K-Nearest neighbour and K-means are used in advanced signature- based intrusion detection system (IDS) and best among these are chosen for implementation in future networks.

In [19] the intrusion detection system utilizes SVM classifier to detect DDOS attacks.

B) Entropy Based Techniques

Entropy is the measurement of randomness so it can be used to measure arbitrariness in the packets which are approaching to any network. The lower the randomness the lower is the entropy and vice versa. There are two fundamental components for Detection of DDOS using the entropy technique which is window size and a threshold. Window size can be either based on number of packets or a time period. To measure the uncertainty, Entropy is calculated in this window. To declare an attack a threshold required to be set. If the resultant entropy falls a certain threshold or below then controller considers it a situation under attack.

A simple equation to calculate the entropy (H) by using the number of packets (n) and probability of elements in that window (Pi) as follows [20]:

$$H = -\sum_{i=1}^{n} Pi \; Log \; Pi$$

Packet headers are represented as autonomous information codes with a certain unique chance of occurrence. Initially selecting any random window size for example 5000 and scaling the window further, a pattern can be predicted for every type of packet headers with all probabilities of occurrences. If the bins of any packet header vary from trend or average bin then it can alert the system about abnormality. Entropy is a recognized technique for DDOS detection. Different studies have purposed different parameters and thresholds for effective implementation of this technique.

Qin et al. [21] used window of 0.1 seconds and considered 3 levels of threshold for more accuracy. By using three level thresholds both false positive and negative can be avoided. However, setting such lower time period with more checks is resource hungry and time-consuming process especially in large network where we have millions of flows per second.

A more efficient way was purposed by Ra et al. [22] to use both packet type and total number of packets of that type in the network. Time period window is used as a threshold in this method. The authors used many different datasets to determine a suitable threshold experimental value and concluded that it is multiple of the standard deviation of entropy. But the author has not mentioned any percentage of its accuracy and also there are more false negatives and fewer false positives than other methods under consideration here.

Oshima et al. [23] computed the entropy using a detection of statics in a short-term. The author used a shorter window size with an optimum value of 50 packets for collecting the statics after trying different window sizes. Rather than setting any specific threshold the author purposed a significance test to confirm the DDOS attack situation. In case of attack rate of traffic observed to be increased while normally traffic rate was constant. But this method proved successful only when abnormal traffic is 75% or more of total traffic in the network. In case of lower than 75% attack traffic this method was not proved effective.

SMM and MSH [24] used the assumption that attack would be destined for any IP address within the network. In SDN every new flow is inspected by controller for policy implementation. At this point if any new flow has destination IP address within the infrastructure network is subjected for entropy calculation using a specific window size. SDN controller calculates the entropy by total number of packets destined for any specific host. If all the packets are destined for a single host then entropy will be a maximum and if all the packets are destined for distributed destination then entropy will be the lowest. The window size should be smaller or equal to the total number of hosts in the network. In this study the author used window size of 50. Another reason for selection of smaller window size is that smaller numbers of packets are easy to process in smaller time hence an early detection.

In the study [24] authors also experimented different window sizes and observed memory and CPU utilizations, higher window size does not affect the memory much but increases the CPU utilizations. To detect an attack a hash table of packet is created by adding a function to controller. Entropy is calculated after every 50 packets. If a packet for any destination is new it will be added with a count of 1 and in case of existing destination flow its value will be incremented by 1. Entropy will be maximum if traffic is distributed for different destination and if all the traffic will be destined for a single host only then entropy will be minimum and situation will be considered as attack. By experimental values author set a threshold and supposed that if the entropy is observed lowered than threshold for 5 consecutive calculations then the situation is considered as alarming. Similarly, in real production network thresholds can be set by observing routine trends and alarm triggers can set with approximate variations. So, using this technique with widow size of 50 packets can help admins to detect attack at earliest stage i.e. if 5 connective lower threshold entropies are observed then attack be detected only in 250 packets. The purposed method in this study is compatible with Centralized architecture of SDN and much more convenient than training the SOM [25] and computing the complex matrix for it. As per [24] success rate of attack detection using entropy of destination of IP address with smaller window size 96%. However, this paper does not provide any information if the whole subnet is under attack.

C) Graphic Based Techniques

In a study [25] an advance architecture is purposed for DDOS attack defence in SDN deployed in cloud computing environment named as Damask (DDOS Attack Mitigation Architecture using Software defined networking). In this idea network infrastructure is categorized into three different layers of network i.e. controller Layer, switches layer and application layer. A strong isolation is kept between different layers by designing different policies for each layer which make the core network operations transparent to end cloud users. Demask-D and Demask-M are two different modules used for DDOS attack detection and attack reaction respectively. Demask-D module inspects every new flow and forwards all suspected flows to Demask-M module with relative packet information. Demask-M Module applies the attack mitigation policies. Although proposed solution in this architecture is an effective, efficient and inexpensive architecture which is very easy to deploy in cloud computing but still it has some limitation for example Demask-M module

has very little option to control the malicious flows, it just drops all suspected flows which may cause outages in real production network.

In a work [26] team have evaluated the well-known SDN controllers with Lab generated DDOS attacks and found all of them vulnerable especially to unexpected security threats. To overcome the detected issues team purposed a frame work named as SPHINX which control the abstraction of flow graphs and approximate the real network operations. This approximation is used to real time network properties like traffic flows and also used to detect the unknown security threats both at data plane and at topology and controller levels. The authors have provided design implementations of SPHINX along with its policy engine which allows Network engineers more freedom to apply desired specific policies based on real scenarios and requirements. Team has also provided practical evaluation of their frame work in four different test cases and obtained desired results with acceptable overheads. SPHINX has some limitation for example it can miss some Transit attacks which is major a challenge to it. Flows inconsistence might occur from granularity/rate at which metadata statistics are being collected and updated. In this span of few seconds controller is dependent. Also, some more work is required in flow rule aggregation and implementation of SPHINIX in mixed network.

VI. DEFENCE MECHANISMS FOR SDN-SELF DDOS ATTACKS

After the DDOS attacks are identified by detection algorithms, a timely efficient and effective defence mechanism is required to reduce the network loss and restore the network functions. Hence several defence mechanisms are proposed for each three types of DDOS attacks on SDN as mentioned in above section. In this section we summarize all those defence algorithms.

A) Defence Against Switch Overload

In [27] a source based IP filtering technique is proposed to detect and prevent DDOS attacks. The proposed algorithm makes a temple table called T table to store the unique source IP addresses of the forwarded IP packets from the switch. This table contains some fields like: 1) Counter field for each unique IP address to identify and track the no. of packets arrived. 2) Minimum no. of packets per connection. 3) The average connections for frequent users. 4) The statistic counter of the IP address.

Whenever traffic from unique IP address comes to the controller then controller firstly considers it as malicious traffic. The controller makes a new entry in the table for this IP address and assigns it two timeouts: idle-timeout and hard-timeout that are smaller than normal timeout. The reason for assigning these short timeouts is that malicious traffic can be quickly removed from the switch TCAM table. After comparing the parameters in the table and knowing the statistics for normal and malicious traffic controller design a policy rule. Based on the values if the traffic user is a legitimate one then controller vanishes the two created timeouts and assigns it a normal timeout. And if the traffic is malicious then assigns a dropped packet policy action. If the duration of malicious traffic becomes huge or the no. of flows coming to the switch are more than its capacity then this framework can't work well.

In [28] a mechanism is proposed to elastically scaling up the capacity of SDN control plane by using v-Switch based overlay. This framework is used to scale up the capacity of control plane so that more flows requests are accommodated by switch. As the capacity of the hardware switches to incorporate the flows request is limited so software switches are used to resolve this problem since they can run on more powerful CPUs and so they handle more flows requests. News flows coming to the hardware switches are redirected to the software switches and there the flows requests are generated. For large throughput, hardware switches still forward the traffic as they have large throughput than software switches. Although this framework improves the capacity of the control plane to accommodate and generate more flows request than the physical switch but it is not enough for the adversaries where an attacker can send the malicious traffic even higher rate than software switch capacity to handle it.

B) Defence Against Data to Control Channel Congestion

In [29] different types of threats occurred on the SDN control channel are studied and a light weight information hiding authentication mechanism is proposed to prevent the DOS attacks. The algorithm uses the IP Identification (IPID) field of the IPv4 to verify the authentication of the switches and the controller on the control channel. The main purpose of the IPID in the IPv4 is to reassemble the packets after the fragmentation of the packets.

In [30] an algorithm called FlowSec is introduced to mitigate the attacks on the bandwidth of the controller. It puts a rate limit on the number of packets that a switch can send to the controller. The algorithm dynamically calculates the controller bandwidth by collecting the switch statistics. If an attack is identified the algorithm uses Floodlight module to instructs the switch port to slow down. Although the algorithm mitigates the DDOS flood attacks against controller bandwidth, but it can also hinders the normal traffic as well.

C) Defence Against Controller Resource Overload

In [31] by focusing on controller resource saturation an efficient queuing based mitigation mechanism is proposed. In prior study to handle multiple queues of traffic a single layer fair queening mechanism was used by the controller in which all the requests and flood traffic is put into one single queue. Although this mechanism solves some problems, but it ends up with creating more queues to handle for the controller. The proposed mechanism called Multi-Layer Fair Queuing (MLFQ) solves this problem by dynamically expanding and aggregating the queues depending the load on the controller. Using MLFQ mechanism, in normal situation, the controller only maintains a small number of queues and it can effectively separate the flood requests of DDOS attacks from the normal traffic.

In [32] to improve the performance efficiency of SDN and to prevent from DDOS attacks, a request prioritizing

algorithm named as FlowRanger is proposed. Unlike First come First Server (FCFS) queue-based processing the algorithm prioritize the packets based of attacking likelihood of the source. In FCFS if the controller buffer is full then it not only drops the malicious traffic but the legitimate ones too. To avoid such problem the proposed algorithm gives the higher priority to the normal packets and lower priority to the short lived spoofed packets. If even a flood of requests comes to the controller, it doesn't drop the normal traffic packets and hence improves the efficiency of the network. The algorithm consists of three components: trust management, queuing management and requests scheduling. When a new packet request goes to the controller, the controller computes its trust value by using trust module. Based on the trust value that is calculated by trust management module in controller queue management component maps the packet request to a corresponding priory queue. Finally, third component that is request scheduling determines the request's processing order in all the different queues.

In [33] to avoid against data to control plane saturation an algorithm called FLOODGAURD is proposed. Unlike other mechanisms this algorithm provides defence against generic flow requests flood attacks. This algorithm contains two modules as proactive flow analyser and packet migration. These two modules work simultaneously. Before the attacks occur, traffic flow is normal as from OF switch to controller. Whenever an attack is detected by the controller, it redirects the flow towards in the OF switch to the data control plane cache. Packet migration does this job. At the same time other module (Proactive Flow Analyser) tracks the malicious traffic value in the running applications on the controller. The controller generates some rules dynamically by symbolically executing the controller applications and sends these rules to OF switches to supress the incoming and the future traffic. Now all the requests come slowly to the controller from data plane cache instead of Open Flow switch. Although this method resolves the problem of general flood attacks but due to its symbolic execution it may not exhausts all the possible execution paths for the complicated controller applications.

In [34] a comprehensive defence mechanism against DDOS attacks called SDNShield is proposed. The previous proposed mechanisms only focus on the control plane saturation or switch flow overload, but this mechanism focuses the both. Using the SDN capability of software switch customization the framework uses the special software boxes to scale up the switches so that they can incorporate the control plane surge workload. Also, it saves the controller from malicious requests. For this purpose, it implements two stage defence algorithms. In first stage it eliminates the malicious traffic and filter out the legitimate one using a statistical differentiation method. As the statistical methods are inevitable to hundred percent filtering of legitimate traffic. Some false positives occur during this filtering so a next stage is required to recover these false positives. Hence in second stage a TCP Verification handshake mechanism is used to deeply inspect the packets and recover it. As most of the traffic on the internet is TCP based. Knowing this fact, the authors used this mechanism, but this mechanism is not recover the traffic other than TCP based. One extra thing is TCP verification mechanism only handles limited number of packet requests per second so in large network where thousands or millions of packets are coming per second it fails.

FortNOX [35] introduces the tunnelling attack and proposes a security enforcement kernel to defend against this attack. Rosemary [36] introduces a sandbox-based framework to safeguard the SDN control layer against malicious or faulty control applications. TopoGuard [37] studies the network topology poisoning attack and proposes an extension to mitigate against the attack.

We have summarized all the defence mechanism in the Table II.

Table II: Mitigation Techniques for SDN Self DDOS Attacks

Defence Techniques	Classification of DDOS Attacks		
	Switch Overload	Channel Congestion	Controller Resource Saturation
IP Filtering [27]	✓		
Scotch [28]	✓		
Lightweight [29]		✓	
FlowSec [30]		✓	
MLFQ [31]			✓
FlowRanger [32]			✓
FloodGuard [33]			✓
SDNShield [34]			✓
FortNOX [25']			✓
TopoGuard [37]			✓
Sphinx [26]			✓
Sandbox-based [36]			✓

VII. RECOMMENDATION AND DISCUSSION

In this paper, we present different proposed techniques of DDOS detection and mitigation for the traditional and SDN network. Different techniques focus on the different attack points but not all the points. Our recommendation is if we use an algorithm which combine the different efficient techniques proposed we can solve the DDOS attack problem easily. We have to think which algorithm applies on what time and how it can be most effective. Similarly, we also highlight three different approaches for the detection of SDN-Self DDOS attacks. Different approaches have certain limitations we have to make a new algorithm which may detect the attacks efficiently and effectively and for mitigation mechanism it not only discards the attacks but also finds the location of the attacker and the source victim. So, collaboration of these different proposed techniques may solve the problem and would results into an effective solution.

VIII. CONCLUSION AND FUTURE WORK

SDN is a new paradigm for networking as it solves many security problems in the traditional network by providing centralized programmable control for all the networking devices. In this paper we actually discussed two different major research domains for the detection and mitigation of the DDOS attacks. In domain one, we summarize how SDN has solved the problem of DDOS attacks in traditional network and then secondly SDN self DDOS attacks problem is discussed. We represent a comprehensive survey of proposed detection and mitigation techniques for the SDN-Self DDOS attacks. We also classify the detection techniques into three main categories: Machine learning based, Entropy based and Graphic based. Machine learning techniques are good choice but they require a huge number of training set and also results in more time overhead. Entropy based techniques relatively have low processing time but these techniques should have to combine with others to make threshold determination. Graphic based approaches are effective but when the network topologies changes very frequently, many learned invariants can be interrupted resulting into false positives. Although many methods for this concern are proposed but there are still many areas to pay attention. Most of the mitigation techniques discussed only the anomaly behaviour of traffic and only focus on the attack detection and mitigation but not the attack source or victim location. This can be a new research point. Moreover, the proposed techniques only focus on the one-controller based network but, in a network, where multiple controllers are working simultaneous how we can utilize different controller to solve the different security problems. And how the algorithms for detection and mitigation can be designed so that it can efficiently and accurately solve the problem without any overhead can be new points for research.

REFERENCES

[1] "The "stacheldraht", distributed denial of service attack tool, December 31, 1999.

[2] C. Jin, H. Wang, and K. G. Shin, "Hop-count filtering: an effective defense against spoofed DDoS traffic," in *Proceedings of the 10th ACM conference on Computer and communications security*, 2003, pp. 30-41.

[3] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *IEEE communications surveys & tutorials*, vol. 15, pp. 2046-2069, 2013.

[4] Q. Liao, D. A. Cieslak, A. D. Striegel, and N. V. Chawla, "Using selective, short- term memory to improve resilience against DDoS exhaustion attacks," *Security and Communication Networks*, vol. 1, pp. 287-299, 2008.

[5] H. Jiang, S. Chen, H. Hu, and M. Zhang, "Superpoint-based detection against distributed denial of service (DDoS) flooding attacks," in *Local and Metropolitan Area Networks (LANMAN), 2015 IEEE International Workshop on*, 2015, pp. 1-6.

[6] Z. Tan, A. Jamdagni, X. He, P. Nanda, R. P. Liu, and J. Hu, "Detection of denial-of-service attacks based on computer vision techniques," *IEEE transactions on computers*, vol. 64, pp. 2519-2533, 2015.

[7] Z. Tan, A. Jamdagni, X. He, P. Nanda, and R. P. Liu, "A system for denial-of-service attack detection based on multivariate correlation analysis," *IEEE transactions on parallel and distributed systems*, vol. 25, pp. 447-456, 2014.

[8] P. Xiao, W. Qu, H. Qi, and Z. Li, "Detecting DDoS attacks against data center with correlation analysis," *Computer Communications*, vol. 67, pp. 66-74, 2015.

[9] S. Shin and G. Gu, "CloudWatcher: Network security monitoring using OpenFlow in dynamic cloud networks (or: How to provide security monitoring as a service in clouds?)," in *Network Protocols (ICNP), 2012 20th IEEE International Conference on*, 2012, pp. 1-6.

[10] C. Buragohain and N. Medhi, "FlowTrApp: An SDN based architecture for DDoS attack detection and mitigation in data centers," in *Signal Processing and Integrated Networks (SPIN), 2016 3rd International Conference on*, 2016, pp. 519-524.

[11] Y. Xu and Y. Liu, "DDoS attack detection under SDN context," in *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, 2016, pp. 1-9.

[12] S. K. Fayaz, Y. Tobioka, V. Sekar, and M. Bailey, "Bohatei: Flexible and Elastic DDoS Defense," in *USENIX Security Symposium*, 2015, pp. 817-832.

[13] C. Lorenz, D. Hock, J. Scherer, R. Durner, W. Kellerer, S. Gebert, *et al.*, "An SDN/NFV-Enabled Enterprise Network Architecture Offering Fine-Grained Security Policy Enforcement," *IEEE Communications Magazine*, vol. 55, pp. 217-223, 2017.

[14] R. Sahay, G. Blanc, Z. Zhang, and H. Debar, "Towards autonomic DDoS mitigation using software defined networking," in *SENT 2015: NDSS Workshop on Security of Emerging Networking Technologies*, 2015, p. .

[15] J. R. Ballard, I. Rae, and A. Akella, "Extensible and Scalable Network Monitoring Using OpenSAFE," in *INM/WREN*, 2010.

[16] M. Ramadas, S. Ostermann, and B. Tjaden, "Detecting anomalous network traffic with self-organizing maps," in *International Workshop on Recent Advances in Intrusion Detection*, 2003, pp. 36-54.

[17] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *Local Computer Networks (LCN), 2010 IEEE 35th Conference on*, 2010, pp. 408-415.

[18] L. Barki, A. Shidling, N. Meti, D. Narayan, and M. M. Mulla, "Detection of distributed denial of service attacks in software defined networks," in *Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on*, 2016, pp. 2576-2581.

[19] R. Kokila, S. T. Selvi, and K. Govindarajan, "DDoS detection and analysis in SDN-based environment using support vector machine classifier," in *Advanced Computing (ICoAC), 2014 Sixth International Conference on*, 2014, pp. 205-210.

[20] "Shanon Formula for Entropy Calculation " https://en.wikipedia.org/wiki/Entropy_(information_theory).

[21] J. Zhang, Z. Qin, L. Ou, P. Jiang, J. Liu, and A. X. Liu, "An advanced entropy-based DDOS detection scheme," in *Information Networking and Automation (ICINA), 2010 International Conference on*, 2010, pp. V2-67-V2-71.

[22] G. No and I. Ra, "An efficient and reliable DDoS attack detection using a fast entropy computation method," in *Communications and Information Technology, 2009. ISCIT 2009. 9th International Symposium on*, 2009, pp. 1223-1228.

[23] S. Oshima, T. Nakashima, and T. Sueyoshi, "Early DoS/DDoS detection method using short-term statistics," in *Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on*, 2010, pp. 168-173.

[24] S. M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," in *Computing, Networking*

and Communications (ICNC), 2015 International Conference on, 2015, pp. 77-81.
[25] B. Wang, Y. Zheng, W. Lou, and Y. T. Hou, "DDoS attack protection in the era of cloud computing and software-defined networking," *Computer Networks*, vol. 81, pp. 308-319, 2015.
[26] M. Dhawan, R. Poddar, K. Mahajan, and V. Mann, "SPHINX: Detecting Security Attacks in Software-Defined Networks," in *NDSS*, 2015.
[27] N.-N. Dao, J. Park, M. Park, and S. Cho, "A feasible method to combat against DDoS attack in SDN network," in *Information Networking (ICOIN), 2015 International Conference on*, 2015, pp. 309-311.
[28] A. Wang, Y. Guo, F. Hao, T. Lakshman, and S. Chen, "Scotch: Elastically scaling up sdn control-plane using vswitch based overlay," in *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, 2014, pp. 403-414.
[29] O. I. Abdullaziz, Y.-J. Chen, and L.-C. Wang, "Lightweight Authentication Mechanism for Software Defined Network Using Information Hiding," in *Global Communications Conference (GLOBECOM), 2016 IEEE*, 2016, pp. 1-6.
[30] M. Kuerban, Y. Tian, Q. Yang, Y. Jia, B. Huebert, and D. Poss, "FlowSec: DOS Attack Mitigation Strategy on SDN Controller," in *Networking, Architecture and Storage (NAS), 2016 IEEE International Conference on*, 2016, pp. 1-2.
[31] P. Zhang, H. Wang, C. Hu, and C. Lin, "On Denial of Service Attacks in Software Defined Networks," *IEEE Network*, vol. 30, pp. 28-33, 2016.
[32] L. Wei and C. Fung, "FlowRanger: A request prioritizing algorithm for controller DoS attacks in software defined networks," in *Communications (ICC), 2015 IEEE International Conference on*, 2015, pp. 5254-5259.
[33] H. Wang, L. Xu, and G. Gu, "Floodguard: A dos attack prevention extension in software-defined networks," in *Dependable Systems and Networks (DSN), 2015 45th Annual IEEE/IFIP International Conference on*, 2015, pp. 239-250.
[34] K.-y. Chen, A. R. Junuthula, I. K. Siddhrau, Y. Xu, and H. J. Chao, "SDNShield: Towards more comprehensive defense against DDoS attacks on SDN control plane," in *Communications and Network Security (CNS), 2016 IEEE Conference on*, 2016, pp. 28-36.
[35] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, and G. Gu, "A security enforcement kernel for OpenFlow networks," in *Proceedings of the first workshop on Hot topics in software defined networks*, 2012, pp. 121-126.
[36] S. Shin, Y. Song, T. Lee, S. Lee, J. Chung, P. Porras, *et al.*, "Rosemary: A robust, secure, and high-performance network operating system," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, 2014, pp. 78-89.
[37] S. Hong, L. Xu, H. Wang, and G. Gu, "Poisoning Network Visibility in Software-Defined Networks: New Attacks and Countermeasures," in *NDSS*, 2015.