Software-defined Networking-based DDoS Defense Mechanisms

ROCHAK SWAMI, MAYANK DAVE, and VIRENDER RANGA,
National Institute of Technology, India

Distributed Denial of Service attack (DDoS) is recognized to be one of the most catastrophic attacks against various digital communication entities. Software-defined networking (SDN) is an emerging technology for computer networks that uses open protocols for controlling switches and routers placed at the network edges by using specialized open programmable interfaces. In this article, a detailed study on DDoS threats prevalent in SDN is presented. First, SDN features are examined from the perspective of security, and then a discussion on SDN security features is done. Further, two viewpoints on protecting networks against DDoS attacks are presented. In the first view, SDN utilizes its abilities to secure conventional networks. In the second view, SDN may become a victim of the threat itself because of the centralized control mechanism. The main focus of this research work is on discovering critical security implications in SDN while reviewing the current ongoing research studies. By emphasizing the available state-of-the-art techniques, an extensive review of the advancement of SDN security is provided to the research and IT communities.

CCS Concepts: • **Security and privacy → Intrusion detection systems**; **Denial-of-service attacks**; *Security services*;

Additional Key Words and Phrases: Software-defined networking, distributed denial of service, control plane, data plane

ACM Reference format:
Rochak Swami, Mayank Dave, and Virender Ranga. 2019. Software-defined Networking-based DDoS Defense Mechanisms. *ACM Comput. Surv.* 52, 2, Article 28 (April 2019), 36 pages.
https://doi.org/10.1145/3301614

1 INTRODUCTION

Networking has become an essential part of our lives to share information and resources via digital information technology. It is a process of communicating with other devices digitally. However, the current IP networks are not flexible and have a static architecture in which reconfiguration of the new policies and rules are difficult. The reason is that there is a strong coupling of the control and data planes, which means that the controlling and routing policies are embedded in the data forwarding devices/hardware. This property makes it more difficult to manage the network and its protocols dynamically. Whenever there is a need to update any existing policy or to add a new functionality, configuration of all the affected devices are modified. This process is time-consuming and increases the overall cost of the process. To produce more optimized and better results, there

Authors' addresses: R. Swami, M. Dave, and V. Ranga, National Institute of Technology, Department of Computer Engineering, Kurukshetra, Haryana 136119, India; emails: rochakswami123@gmail.com, {mdave, virender.ranga}@nitkkr.ac.in.

0360-0300/2019/04-ART28 $15.00
https://doi.org/10.1145/3301614

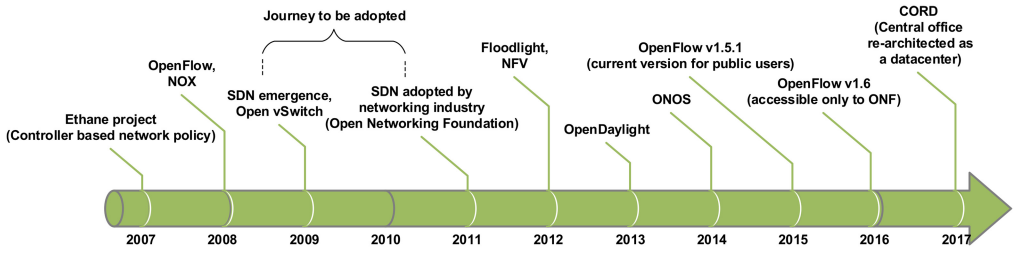

Fig. 1. Development trends in SDN.

is a need to make networks easily adaptable and reconfigurable. Moreover, in the case of cloud computing, service providers need to fulfill the demands of business customers. To achieve this goal, a network needs to be more programmable and agile according to the time-bound requirements for new applications, viz., network virtualization [1]. To overcome the limitations of current networks, a new technology, named Software-defined networking (SDN), has been developed as a revolutionary paradigm in the next-generation architecture for the Future Internet [2]. SDN ensures the desired flexibility for networks. SDN first came into existence by OpenFlow at Stanford University. The Open Networking Foundation (ONF) gave a push toward adoption of SDN by developing the OpenFlow protocol [3, 4]. OpenFlow is designed as one of the first SDN standards. The OpenFlow protocol is responsible for intercommunication between the two planes of SDN (control and data). OpenFlow was proposed in 2008 to provide flexibility and programmability. The ONF has been used by well-known organizations, including Deutsche Telekom, Verizon, and Yahoo!, since 2011 [5]. A development timeline of SDN techniques throughout the years is shown in Figure 1.

SDN has gained attention of researchers for future-generation networks. It has been adopted for its virtualized and flexible behavior by both the academic and industry communities. The main characteristic of SDN is detachment of the control and data planes [6]. The controller manages and controls all the forwarding devices (router, switches) residing in the data plane. This makes forwarding devices no longer smarter. They act as normal forwarding devices. This unique property of SDN makes it different from traditional networking technologies, which combine both planes (control and data plane) tightly with each other. Using this idea, complete functionality is managed by software programming without any modification in the existing network topologies [7]. In SDN, if any rule/policy updates are needed according to user's requirements, then these changes are implemented only in the control plane, thereby reducing the cost of this process. The centralized infrastructure of SDN can provide an efficient use of the resources and improvement of the network performance. SDN makes the network more programmable and innovative. SDN is replacing the traditional networking technologies because of many advantages: It provides a complete view of the network and gives logically centralized control, programmability, simplification of network management, easy reconfiguration, and open programmable interfaces. Advantages of SDN technologies are shown in Figure 3. With the growth of this networking paradigm, various threats against SDN have also grown to disrupt its normal operation.

While discussing network security, the Confidentiality, Integrity, Availability (CIA) triad [8] is designed as one of the most important models for security policies. The CIA triad is a benchmark model for securing networks [9]. There are many threats that attempt to damage a network's services and resources. Denial of Service (DoS) attacks [10] are considered among the most destructive. DDoS attacks exploit one of the components of the CIA model, i.e., availability. DDoS attacks stop legitimate users from availing internet services or resources by making superfluous requests to the systems, i.e., servers and devices [11]. This huge amount of traffic is produced

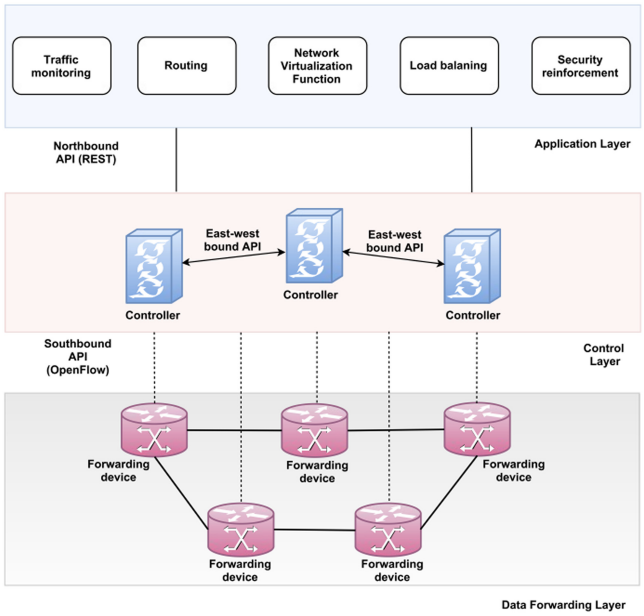

Fig. 2. SDN architecture.

from multiple sources. This makes it very difficult to handle the attack situation. The scenario of DDoS or DoS is similar to a group of customers outside the door of a shop, trying to enter, and making it difficult for benign customers to enter [12]. On March 5, 2018, Github was victimized by one of the largest DDoS attacks, which peaked about 1.7Tbps [13]. Due to the architectural design, SDN is vulnerable to DDoS attacks from many viewpoints. SDN has a central view of the network topology provided by the controller. However, this characteristic makes it vulnerable to several threats. There are many possibilities for an attacker to make modifications to the whole SDN network functionality just by changing the controller. We can analyze the security issues of SDN in two aspects. One aspect is security *by* SDN and another aspect is security *for* SDN. SDN has many capabilities for defending against the DDoS attacks that make it useful for protecting many different types of networking technologies. However, SDN attracts several attacks due to its design features such as detachment of control and data plane.

1.1 SDN Architecture

SDN detaches the control and forwarding (data) logic. SDN architecture is represented in the form of layers, as illustrated in Figure 2. The architecture is separated into three layers: application, control, and data forwarding. The processing starts when a packet arrives at a data forwarding layer that handles the packets and, if required, gives it to the control layer. The control layer may require various applications having different functionalities. These layers of the SDN architecture are described below:

(1) *Data forwarding layer*: This layer contains different switches and routers. They are connected with each other through a wired or wireless channel. An SDN switch is simply used to forward the packets on the basis of the controller's instruction. Each switch has a flow table that contains the entries of packets to make forwarding decisions. Each entry in the flow table has three parts: rule, action, and counter. The rule specifies the field values of the packet header. Whenever the switch receives a new packet, it checks the flow table

to find the rule. If the field values are matched, then the value of the counter increases, and the respective action is taken by the switch. Similarly, if the field values do not match, then the switch informs the controller. The controller takes an appropriate action, i.e., forwarding the packet, dropping or adding new rules to the switches.

(2) *Control layer*: This contains a single or multiple controllers. The complex control logic is implemented in the controller and known as the "brain of SDN." The control layer controls all the switches and manages the whole network. The SDN controller and switches communicate through a standard southbound API (OpenFlow). It provides a full view of the network. When multiple controllers are used, they are connected with each other through an interface known as east–west bound API. This interface makes them share essential information with each other. In the multi-controller environment, each controller handles a group of switches.

(3) *Application layer*: This layer contains different applications required for several business concerns and necessities. An application is a software program that is deployed over the controller. SDN applications communicate with the controller using a northbound interface (REST) according to their network requirements. Some of the required applications are such as traffic monitoring, network virtualization, security reinforcement, load balancing, mobility management, and others. The control layer presents an abstracted view of all the physical elements to the application layer. The applications make logics for decision making used in the control layer. On the basis of this logic, data plane devices perform the forwarding of network packets and take actions for further processing.

Some interfaces are required to provide communication among different layers in SDN. Southbound interfaces allow communication between the data plane and the control plane. Network Configuration Protocol (NetConf) [14] and 'OpenFlow' [3] are two standard southbound interfaces used in SDN implementations. The northbound interface facilitates the communication of controller and the SDN applications. The eastbound interface enables SDN to interconnect with conventional IP networks. The westbound interface allows the necessary information sharing between the controllers of different SDN domains. There are no standard northbound and east–westbound interfaces available yet.

1.2 Security Features of SDN

SDN has several characteristics related to its architecture and design. These design characteristics make it different from conventional network architectures. We will throw light on its design features in the context of security. SDN features are helpful in securing networks with more flexibility in a fast and efficient manner. However, SDN itself may become vulnerable to security threats due to some defects in its design. Therefore, SDN design features can be described in two aspects. One aspect is features that make SDN resilient to DDoS threats and the other aspect is features that can make it vulnerable.

1.2.1 Features Making SDN Resilient to DDoS. SDN provides many advantageous features for dealing with DDoS:

(1) *Centralized monitoring of anomalous traffic*: The controller has the complete information of the network. Therefore, all the anomalous activities going on in the network are observed by the controller.

(2) *Programmable configuration*: One of the important advantages of SDN is its programmability. Whenever any malicious behavior is detected in the network, new programs are configured immediately to deal with the anomalies.

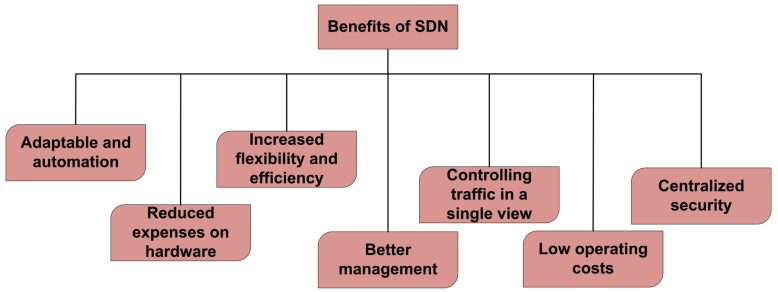

Fig. 3. Benefits of SDN technology.

1.2.2 Features Making SDN Vulnerable to DDoS. SDN has some design issues that make it vulnerable to various security threats. The security faults in SDN design and its impact are discussed as given below.

(1) *Limited TCAM*: In SDN, OpenFlow switch maintains the flow rules for new incoming packets in its flow table. The switches utilize a "content addressable memory" known as TCAM ('T' for ternary) [15] to reserve flow rules. It is a unique type of memory used for high-speed searching applications. However, SDN switches have limited space of memory in their flow tables. For instance, the Pronto-Pica8 3290 switch can store only 2000 rules [16]. The limitation of the flow table memory can make the SDN sensitive to DDoS attacks.
(2) *Single point of failure and cascading failure of controller*: The SDN controller is a prime target of attackers. It is a centralized entity that may suffer from a single point of failure. Although the controller controls the entire network, its crashing can downgrade the network performance, availability, and integrity of the network. A single SDN controller cannot be efficient in handling large network traffic. In that case, deploying multiple controllers in different network domains can handle the situation. There may be an issue of authenticity, consistency, and scalability of different privacy rules in each domain's controller. This may cause more than one controller to fail in a cascading manner.
(3) *Decoupling of control and data plane*: In SDN architecture, decoupling of these two planes makes it vulnerable to various security threats. These planes communicate using a standard protocol (OpenFlow). An attacker can disturb this communication of information by implementing DoS, saturation attacks, man-in-the-middle attacks, and so on, to choke switch-controller channel bandwidth.
(4) *Dumb switches*: SDN switches are simple forwarding devices and considered as dumb [17]. They rely on the controller for taking an appropriate action to forward packets. This property of OpenFlow switches may reduce the performance of controller and control plane bandwidth because of a large amount of traffic.

2 DISTRIBUTED DENIAL OF SERVICE ATTACKS

DDoS aims to disrupt ongoing operations by overwhelming network devices with connection requests for a certain time period. This flooded traffic of requests forces the target systems to slow down, crash, or shut down. The DDoS attacks keep the systems busy with unusual requests by denying the services to the legitimate customers. The main reason behind DDoS attacks is that most networking architectures have some resource constraints. These attacks mainly deplete resources like bandwidth, memory storage and processing power. All DDoS attacks are not same. They can target victims for different purposes. There are many DDoS attacks that are rapidly growing in the field of internet. Common types of DDoS attacks are illustrated in Figure 4. According

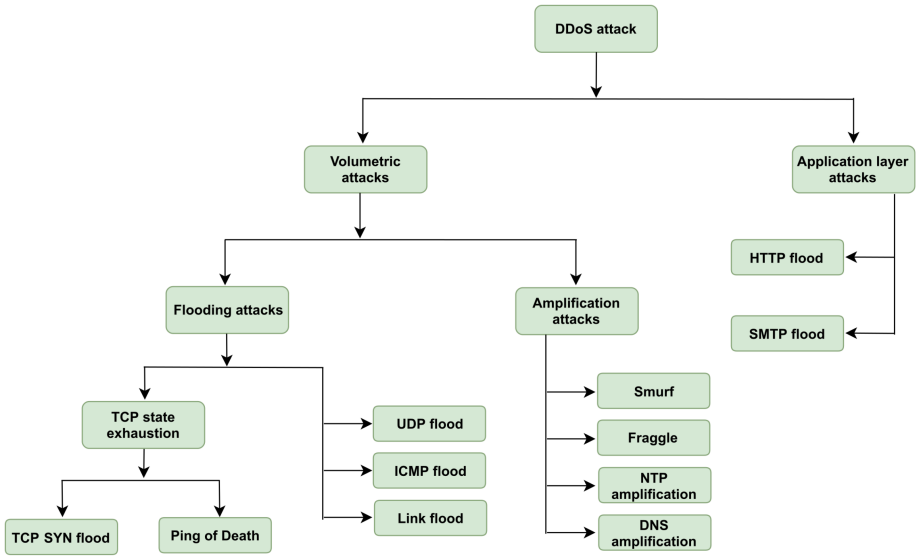

Fig. 4. Taxonomy of DDoS attacks.

to Arbor networks, 65% of the total reported DDoS attacks are volumetric in general [18]. The volumetric attack sends a huge amount of data packets to the target network to overload its bandwidth. The protocol exploitation/state exhaustion attacks exploit the network protocols to attack on the target system, make up about 20% of the reported DDoS attacks in 2014. These attacks exploit the standard application protocols by attacking the online services, e.g., web servers. These are the most challenging attacks that need to be identified and mitigated for efficient running of the network operations. The most common types of application layers attack are HTTP flood and SMTP flood. A detailed discussion on DDoS types is done in Reference [19].

Slow DDoS. One more important type of DoS/DDoS attacks is slow DDoS or low-rate attacks. Most common slow attacks are application layer based, such as HTTP, FTP, SMTP, and IMAP. The slow DDoS attacks are very hard to detect, because traffic generated by these attacks behave as legitimate traffic. These attacks utilize much less bandwidth and resources. With a small consumption of resources, they can create a large destruction. Some HTTP-based slow attacks [20] are as follows:

(1) Slow HTTP header (Slowloris): In this attack, a header is divided into different packets, and these partial headers are sent to server by attacker at a very low rate to make the service unavailable [21].
(2) Slow HTTP POST attack (RUDY): In this attack, the body of the POST message is divided into several packets and sent to the server at a low rate.
(3) Slow read attack: The attacker sends normal HTTP request messages to the server and sees the reply very late from the server.

DDoS attacks target the network and server's resources that are listed below:

(1) *Bandwidth*: Flooding/volume-based attacks consume all the bandwidth in the network. They do not allow the legitimate requests to reach to the server by creating exhaustion on the channel.

(2) *Memory*: Protocol-based attacks such as SYN attack makes TCP connection open all the time that makes the buffer overflow. It consumes the TCP connection table's buffer or memory completely.
(3) *CPU*: Application layer-based attacks target on the web servers. This makes service unavailable for the legitimate users by exhausting processing power of CPU or server and server gets crashed.

Effect of DDoS on SDN Planes. In data plane, switch has a limited size of flow tables. Due to DDoS attacks, a large volume of packets is transferred to the switches. This is called *flow rule flooding*, which exhausts the flow table's memory. The attacker is motivated to send this flood to the controller for saturating the switch-controller bandwidth. One consequence of bandwidth-based attacks is large packet drop. The SDN switch is not able to take any decision in the case of an unmatched entry and therefore sends the packet to the controller. This flooding of unmatched packets at the controller consumes the controller's resources (memory and CPU), degrading the performance of entire system and resulting in an increase in response time and communication overhead.

3 TARGET POINTS OF SECURITY THREATS IN AN SDN NETWORK

The SDN architecture is divided into three layers. All the layers can be targeted by various security threats. However, the controller and control plane bandwidth are the most sensitive target points for DDoS attacks. The possible attacking targets of security threats in SDN networks are shown in Figure 5.

(1) *SDN switch*: SDN switches are used for data forwarding and processing of new incoming packets. They have a very limited size of flow tables. This is a big concern for security.
(2) *Links between SDN switches*: The flow packets are transferred for forwarding from one switch to another switch. Most of the transferred packets are not encoded and may contain sensitive information. These packets can be intercepted by the attackers easily, especially when the links between switches are wireless.
(3) *SDN controller*: As the controller ("brain of SDN network") performs crucial activities for SDN, any abnormality in it can paralyze the whole network. The complete functionality of a network depends on the controller. With this fact, it is the most attractive target for the attackers. It may suffer from a single point of failure in case the network has only one controller.
(4) *Link between controller and switch*: In the case where a packet arrives at a switch and a switch is unable to handle it, then the packet is forwarded to the controller for further processing. Consequently, new packet forwarding rules are appended to the flow table of the switches. The rules contained in a packet are sent through the southbound interface to the switch. These data packets can be interfered with by an attacker on the southbound interface, which results in addition of some malicious rules or alteration of the existing rules. Placement of these fake rules in the switch table leads to misdirection of the packets.
(5) *Links between two controllers*: In the multi-controller-based scenario, communication is shared between the controllers through east–west bound APIs. The packets between the controllers can be obstructed by an attacker to gain essential information for compromising the controllers. Thus, the communication between controllers should be secure and authentic. The distributed controllers may also suffer from cascading failures because of flooded requests.
(6) *Applications*: The applications such as routing, traffic monitoring, and virtualization are implemented on the SDN control layer. Most of the applications are established by third

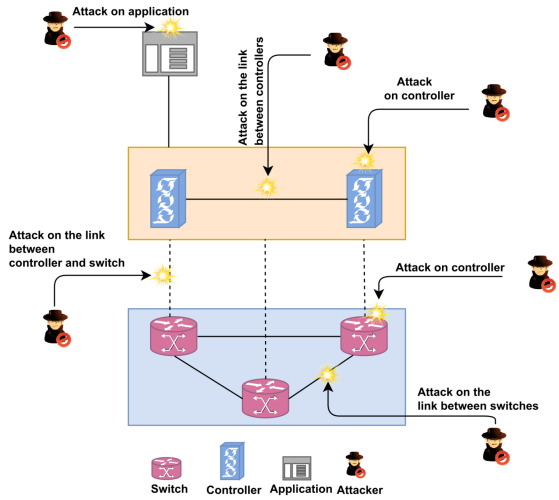

Fig. 5. Attacking points.

parties that do not care for the security requirements. The applications may suffer from unauthorized access. When the functions of the controller are called via northbound API, the malicious entities can be injected into the controllers. Hence, the SDN-based applications can become the easiest target point for blocking service of controllers.

3.1 Potential SDN-specific DDoS Threats

Controller overloading and data-to-control bandwidth congestion are the most concerning DDoS/DoS threats in SDN. Some major threats are discussed here.

(1) *Switch overloading and flow table overflow*: DoS/DDoS attacks generate a large amount of malicious packets that are flooded into a switch. For these malicious packets, the switch will not find a corresponding entry in the flow table. It makes entries for all the unmatched requests in the buffer and sends them to the controller continuously because of unmatched rules. However, a switch has a limited size of TCAM, so it may not be possible to process all the incoming packets. The continued flow of requests will overflow the flow table memory, and, therefore, legitimate requests will have to suffer.
(2) *Controller resource saturation*: The controller is the heart of the SDN network that controls and manages the complete functionality of the network. Hence, the compromised controller hinders the overall network performance. Controller resources such as CPU and memory will be exhausted by processing the flooded requests of the DDoS attacks. When a controller is overloaded, it cannot process the new incoming flows. It degrades the performance of the complete network, because legitimate requests are not handled in a timely manner.
(3) *Switch-to-controller bandwidth congestion*: On table-miss events due to new incoming packets, two actions are performed. First, incoming packets are buffered in the flow table of the switch. Second, an OpenFlow request is created that contains an ID and partial information of the packet header. When this buffer gets full, the complete packet is forwarded to the controller that leads to the OpenFlow channel congestion. Consequently, many packets collide with the southbound interface. In such situations, normal users face the unavailability of the services.

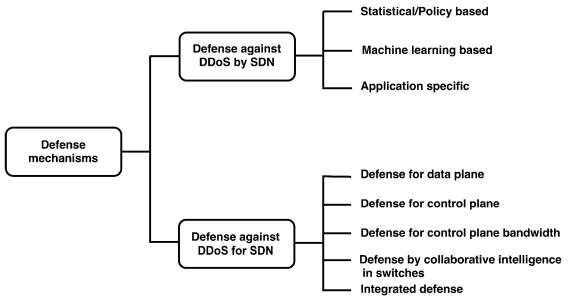

Fig. 6. Defense mechanisms for detection and mitigation of DDoS.

Defense Mechanisms against DDoS. SDN security has become a hot topic for researchers in recent years. SDN helps to mitigate attacks including DDoS, IP spoofing, and malware in conventional networks. However, an attacker may successfully carry out DDoS attacks on the SDN itself. Accordingly, the defense solutions are classified on the basis of design features of SDN as shown in Figure 6. The first category is the set of solutions against DDoS attacks offered by SDN. The second category is the defense solutions for securing the SDN from DDoS attacks.

4 DEFENSE MECHANISMS OFFERED BY SDN AGAINST DDOS

SDN has attracted the interest of researchers worldwide due to its effective characteristics for solving and providing new security mechanisms. With the recent upgrade of SDN, it has been a beneficial aspect for the security perspective in the traditional networks. The global view and programmability are the key features to control the impact of DDoS attacks. Various detection and mitigation mechanisms have been described in this section. As per the utilized detection algorithms, all the existing defense mechanisms can be classified into statistical based, machine-learning (ML) based, and application-specific mechanisms.

4.1 Statistical/Policy-based Defense Mechanisms

In this section, statistical and policy-based detection techniques are discussed. The analysis is formulated on the basis of the behavior and properties of the network flow. A statistical analysis involves collecting and exploring the data samples to identify malicious traffic. A comparison of defense mechanisms is shown in Table 1. In this type of analysis, statistical inference test is applied on the network traffic, and if the data cannot be fitted on some statistical models, then they are classified as malicious data. In Reference [22], the authors showed that statistical-based algorithms, such as entropy and chi-square, could be used to detect the DDoS attacks accurately. Some commonly used statistical techniques in SDN are based on adaptive correlation analysis, standard deviation, probability, and entropy measurements. For detecting and mitigating DDoS attacks, policies/rules can also be defined. Many defense techniques have adopted the policy-based solutions in SDN. These policy-based detection techniques involve the implementation of some defined policies or rules on network traffic flows. Thus, if the traffic flows follow these policies, then they are considered as legitimate flows and otherwise declared as malicious flows. Policy-based solutions play the role of firewall and allow only authentic packet flows to pass through it.

Sattar et al. [23] proposed an SDN-based approach, Adaptive Bubble Burst (ABB), to mitigate the DDoS attacks. ABB enhances availability of targeted resources under DDoS attack. ABB replicates the various copies of resource and spreads the attack over these copies. This spreading decreases the effect of the attack on actual resources. The proposed approach does not provide an idea to discover the DDoS attack. One advantage of ABB is that it has no requirement of any software

Table 1. Statistical/Policy-based Defense Mechanisms

Ref.	Defense mechanism	Intrusion type	Targeted resources	Simulation tools	Controller	Advantages	Limitations
[30]	Multislot (controller scheduling method based on time slice allocation)	DDoS (No specific type)	_	POX, Mininet	Single	Provides better protection for internal switches as compared to SingleQ and MultiQ algorithms.	Not applicable for large scale networks (queue maintenance overhead for each switch makes the system complex)
[23]	ABB (Adaptive Bubble Burst)	Service exhaustion attacks (NTP, HTTP, FTP)	CPU/ server	POX controller, Real-time	Single	Completely transparent to client and server ends. Does not require any modifications at both ends, enhance the availability of a specific service during DDoS	Very costly, high per-packet processing Overhead, not a defense solution, more response time
[33]	RADAR (using adaptive correlation analysis)	Link flooding (crossfire), SYN flooding, UDP/DNS amplification attacks	Bandwidth, memory	Flood-light, Mininet	Single	Reduces response time of controller, avoids single point of failure and cascading failure, does not require extra appliances or any modifications	In case of small network topology with few switches incurs more overhead
[24]	Security-centric SDN	DNS amplification attack	Memory	Kinetic, Mininet	Single	Less response time, adaptive, better throughput and less jitter as compared to existing methods [28, 29]	It is not clear if this system works for the NTP amplification attacks
[34]	FlowTrApp (based on flow statistics)	Flooding attacks (UDP, ICMP)	Bandwidth	Flood-light, Mininet	Single	Reduced burden of the OpenFlow controller, reduced false-negative rate compared to a QoS-based mechanism, can detect and mitigate variable rate (low/high) DDoS	Applicable only for flooding attacks

(Continued)

Table 1. Continued

Ref.	Defense mechanism	Intrusion type	Targeted resources	Simulation tools	Controller	Advantages	Limitations
[32]	An anti-spoofing protection mechanism (FP-SYN)	SYN flooding	Memory	Mininet2.2, OVS2.3.1	Single	Reduces cost, reduces detection overhead and bandwidth consumption of the attack traffic	Provides a defense for only TCP-SYN attacks
[44]	C-to-C protocol for secure communication of controllers	ICMP flooding	Bandwidth	POX, Mininet	Multi-controller	Reduced overhead, reasonable resource utilization	examined only for one DDoS type
[43]	ArOMA (SDN-based autonomic mitigation framework)	Flooding attacks (UDP, TCP-SYN, ICMP)	Bandwidth, memory	Ryu, Mininet	Multi-controller	Reduces workload of controller and switch	QoS degraded, Single customer network support
[35]	SDN-Assisted Slow HTTP	Slow HTTP DDoS attacks	CPU	NS3	Single	No computational burden on ISP, quick response, reduces the damage in the network	May have less detection accuracy, communication overhead not discussed
[41]	SDN One-packet DDoS Mitigation (SODM)	Flooding attacks (SYN, ICMP)	Bandwidth, memory	Real time	Single	Quick response to attack	Restricted monitoring window size
[42]	Adaptive policy driven	UDP, TCP-SYN, ICMP flooding, DNS amplification attacks	Bandwidth, memory	Ryu, Mininet	Single	Quick response and fast mitigation, reduces flow entries in the switches, dynamically instantiated and adaptive security policies	Detection mechanism not discussed
[45]	Woodpecker (Policy based approach)	Link flooding (UDP flood)	Bandwidth	POX, Mininet	Single	Reduces bandwidth utilization of congested link	Applicable to a specific type of attack

update for the client–server sides. ABB as a mitigation module is implemented in python-based controllers. The results show that ABB provides similar response time with three servers to the no-DDoS attack cases. Consequently, request completion rate increases suddenly with ABB. A disadvantage is that it is very costly for per-packet processing.

In Reference [24], an approach to defend against DNS amplification attacks via traffic monitoring tool sFlow [25] was proposed by Aizuddin et al. The proposed system utilizes the features of SDN for defending the attack. In the proposed method, flow packets are collected and processed using sFlow. The header field values are checked to identify whether the flow is generated from the DNS server. The suspected flows are delivered to the controller for mitigation purpose. The topology of the scenario contains a controller, a switch, and two hosts. For simulation, an SDN-based emulator Mininet [26] is used with Open vSwitch [27]. For traffic flow analysis, sFlow-RT is used. The results are compared with some existing approaches such as Rossow [28] and Huistra [29].

A "controller scheduling" algorithm named MutliSlot to defend against DDoS attacks was proposed in Reference [30]. The proposed method depends on allocation strategy that is based on time slicing. This method consists of two modules: (1) DDoS detection module and (2) MultiSlot algorithm module. The objective of the method is to segregate the flow requests from various switches. It utilizes different time slice allocation methods for each individual switch. Simulation is performed on Mininet and POX controller [31]. The effectiveness of MultiSlot is compared with two existing techniques, i.e., MultiQ and SingleQ. The proposed scheduling method provides more protection to the internal switches that are affected by the attack indirectly. This advantage is more obvious as the the strength of the attack increases.

Chen et al. [32] proposed a flexible distributed architecture named FlexProtect to provide protection for multi-tenant data centres. FlexProtect utilizes the capabilities of SDN and NFV to defend against DDoS attacks. It works on the network level. The detection and mitigation modules are implemented separately in the FlexProtect system. The detection module is placed near the service provider, and the mitigation module is placed near the edge routers. Both modules are deployed in the form of a virtual network function (VNF). It reduces the detection overhead and the bandwidth consumption of the attack traffic. Further, authors also proposed an anti-spoofing protection mechanism called FP-SYN based on FlexProtect. According to the simulation results, FlexProtect can effectively alleviate the effect of attack and reduce extra length of the routing path. FP-SYN also can identify attackers with high accuracy.

Zheng et al. [33] designed a system to defend against DDoS attacks via adaptive correlation analysis, called Reinforcing Anti-DDoS Actions in Realtime (RADAR). The system is fabricated upon the "commercial off the shelf (COTS)" that is adopted as an SDN switch. It has no requirement of any alteration in the switch or any extra appliances. It is the first system deployed upon COTS switches that can detect different DDoS attacks. The RADAR system is implemented in Floodlight controller. The RADAR comprises three components: collector, detector, and locator. The RADAR can identify different flooding attacks such as crossfire attacks, SYN flood, UDP flood, and DNS amplification attacks in real time. The test is performed on both the Mininet and hardware-based testbeds. Performance is measured in the form of accuracy, delay, and overhead. The results show that the proposed system can discover the DDoS more efficiently with fewer delays and acceptable overhead.

In Reference [34], an SDN-based DDoS defense mechanism (FlowTrApp) for data centers was presented by Buragohain et al. The proposed mechanism works on two parameters: flow rate and flow duration of a flow. These parameters define how much a legitimate user can send and for how long. FlowTrApp tries to detect the DDoS attacks on web-based applications. It relies on the rules set by administrator specific to application layer. One session per IP address is allowed for HTTP requests at a time. A Fattree topology is used to draw a data center scenario. An aggregation

layer is taken in the FlowTrApp architecture in which all the switches are OpenFlow enabled. The performance of FlowTrApp is compared with an existing QoS-based mechanism. Simulation results show that FlowTrApp permits less number of illegitimate packets to be passed through it in comparison to a QoS mechanism. The proposed method is also compared with a load-balancing attack mitigation method. FlowTrapp performs better than a load-balancing method. The results show that the FlowTrapp reduces the burden of the controller.

An SDN-based defense model to defend against Slow HTTP DDoS attacks was proposed in Reference [35]. The detection of slow DDoS attacks is troublesome, because they act as low-rate-benign traffic flows. The proposed defense model comprises a controller running a module called Slow HTTP DDoS Defense Application (SHDA), two OpenFlow devices, and a targeted web server. Further, the clients are classified into three groups of users, i.e., malicious, slow, and normal. SHDA discovers whether a user is malicious or legitimate. SHDA establishes a threshold for HTTP requests to be completed. The requests that exceed the SHDA's threshold value are identified as malicious requests. For mitigation purpose, SHDA instructs the controller to make a fresh rule that blocks the malicious flow at the forwarding device. The results show that the defense model blocks threats efficiently and allows the server to continue its normal operation.

Shtern et al. [36] proposed an architecture to mitigate the effect of application layer-based slow and low DDoS. Their architecture utilizes SDN's adaptive capabilities for defense whenever required. A concept of "shark tank" is also introduced in this scheme where probable malicious users are redirected. Shark tank is a module that analyses the attack activities and guides the system to know about the attack. Recently, in Reference [37], authors presented an approach for detection and mitigation of slow HTTP attacks by utilizing SDN's flexible features. It provides flow-based analysis of the network traffic and detects the attackers and isolates malicious traffic from the network.

In Reference [38], Tripathi et al. analyzed the behavior of slow HTTP-based DDoS attacks on most widely used web servers (Apache, Microsoft IIS, Nginx, and Lighttpd) and proposed a detection mechanism. Authors evaluated the performance of the detection system against two HTTP-based attacks, i.e., slow header and slow message body attacks. The proposed detection approach has two phases: training and testing based on probabilistic distribution of training and testing data. A distance measure named Hellinger is used between the training and testing probability distributions and using that distance measure, attack is identified.

Hirakawa et al. [39] proposed a defense method against HTTP-based slow DDoS attacks (header attack, message body attack, and slow read attack). In this method, the main focus is on the number of connections from an IP address and the duration time. Thus, if the number of connections from a specific IP address exceeds a threshold value, then all the connections from that IP are disconnected. In Reference [20], the authors suggested a solution for detection of HTTP-based slow DDoS attacks. Two parameters, namely window size and delta time of the packet, were used to analyze the traffic patterns generated by an attack and to mitigate it. In Reference [40], Kemp et al. provided an approach to detect slow read attack by using different ML-based classifiers using netflow-based data.

Huang et al. [41] proposed a simple and efficient method named SDN One-packet DDoS Mitigation (SODM) to defend against DDoS attacks. This approach drops all one-packet flows as soon as a DDoS attack is suspected. The network topology consists of an OpenFlow switch, a controller, and a security analyzer. A security analyzer is used to analyze the traffic flow statistics and carry out some malicious indicators. After the traffic is analyzed, attack indicators are sent to the controller to take proper actions. Among all the incoming flows, a single flow may be malicious but acts like a benign flow. SODM drops all one-packet flows if an attack is identified within a given

observation period. This method provides efficient false-positive rate and response rate. Accuracy of the proposed mechanism depends on the monitoring window size.

In Reference [42], a dynamic policy-based mechanism to mitigate the impacts of attacks on the customer networks was presented. Authors attempt to produce a "fine-grained and automated mitigation" system in the Internet Service Provider (ISP) network by using SDN capabilities. It is based on the high-level policies that ISPs have to enforce dynamically. The proposed system is based on user-centric automated response that provides QoS service to the customers. The primary aim is to decrease the effect of attacks on the ISP customers. The global view of SDN helps to achieve this goal. It works on the requirements of customers accordingly. It supports multiple customers that are served by a single ISP. In the proposed approach, a mutual relationship between ISP and its customers is defined to handle the congestion induced by DDoS attacks. Throughput and jitter are utilized for performance measurement of the proposed approach. This mechanism allows policies to be adaptively updated based on customer's requirements. It provides quick response and attack mitigation.

Sahay et al. in 2017 [43] proposed a mitigation framework (autonomic DDoS mitigation framework (ArOMA)) by using dynamic programmability and global view features of SDN. ArOMA provides a collaboration between ISP and its clients to provide an on-demand mitigation of DDoS threats. In this approach, the client side monitors the network traffic and detects the attacks while the ISP side performs DDoS mitigation based on some policies. This method does not bring computational burden to ISP. The clients run their own attack detection module and generate alarms concerned alerts are reported to the ISP. The proposed framework is validated using simulation and testbed experiments. ArOMA supports only a single-client environment. The results conclude that ArOMA provides quick response for recovering benign traffic's performance. Classification of the traffic as benign and malicious is not reported in this work. ArOMA ensures that the video streaming service can maintain its efficiency while it is being attacked by flooding attacks. It also maintains Quality of Experience (QoE) and Quality of Service (QoS) metrics.

In Reference [44], a collaborative mitigation mechanism for DDoS attacks using SDN was proposed. The authors developed a protocol called controller-to-controller (C-to-C) to provide a secure communication between controllers. The developed protocol permits the controllers to share the information with other controllers in different domains and helps to notify them regarding a running attack. This activates an efficient notification for running attacks on the path and filters the network traffic near the origin of the attack. This decreases the processing time and usage of network resources. Authors also created a C-to-C packet that is delivered to the controller by a detection engine. The packet contains three components such as data, certificate, and signature. Different functionality segments are installed on the upper side of controller. The testbed of the proposed mechanism is divided into three networks: source, intermediate, and destination network. Simulation is performed on Mininet. The instances of Mininet emulating different networks are connected by GRE tunneling. In this approach, a node is simulated as a detection engine. It can be placed in any of the networks or above the network. The performance is evaluated in terms of dissemination delay and throughput. It shows the acceptable usage of CPU (35%) and memory (25%) and overhead.

Wang et al. in 2018 [45] proposed a mechanism named Woodpecker to detect and mitigate a new type of DDoS attack: link flooding using SDN capabilities. In the proposed mechanism a number of selected ordinary switches are upgraded to SDN enabled switches. With the help of the global view provided by the controller, Woodpecker locates the congested location and identifies whether the congestion is actually caused by link flooding. Woodpecker imposes traffic engineering as an application on the controller to mitigate the impacts of attack. The results show that the bandwidth

utilization of congested links is reduced up to 50%. The average packet loss rate and jitter are decreased.

4.2 Machine-learning-based Defense Mechanisms

In this section, ML-based mechanisms [46] are analyzed that can detect DDoS attacks. In recent years, ML has gained attention as a promising technique. Various ML algorithms have been adopted for security purposes. Despite traditional networks, they are also being used to detect and mitigate the attacks in SDN. These algorithms are utilized as a classifier to classify the traffic into malicious and benign. The most commonly used algorithms are support vector machine (SVM) [47], neural network, naive bayes, *k*-means clustering, fuzzy logic, genetic algorithm, and self-organizing map (SOM) [48]. These algorithms can be used to both detect and mitigate DDoS attacks. The analysis of ML-based defense mechanisms is shown in Table 2. In Reference [49], the authors have investigated some ML techniques to be used for DDoS defense in SDN. It suggests that all the algorithms have their own positives and negatives, so they can be used according to their requirements.

Quamar et al. [50] presented a Deep learning based multi-vector DDoS detection system in the SDN network. This system is a network application that is deployed on the controller. The proposed intrusion detection system incorporates a stacked autoencoder (SAE)-based deep learning approach to detect multi-vector DDoS attacks. Deep learning has been used for feature reduction. The detection system contains three modules, i.e., Traffic Collector and Flow installer (TCFI), Feature Extractor (FE), and Traffic Classifier (TC). The proposed system relies on each packet for flow computation and attack detection instead of sampling flows, thus minimizing false positives. The features of the dataset produced from collected traffic traces were normalized using max-min normalization. A Hping3 tool is used for launching different kinds of DDoS attacks on the testbed. The proposed system is compared with soft-max and neural network (NN) attack detection models. The SAE model shows better performance as compared to the soft-max and neural network model in terms of accuracy (99.65%). However, SAE suffers from processing capabilities for two reasons: The first is feature extraction from every packet that can be handled by flow sampling and second is TCFI being developed on top of the controller.

In Reference [51], Li et al. introduced a DDoS detection and defense model that is based on DL. Recurrent neural network (RNN), long short-term memory (LSTM), and convolutional neural network (CNN) are used in the detection model. The proposed model is applied to the OpenFlow switches. Performance of the defensive model is verified in a real-time environment by generating traffic through Spirent packet generator. The proposed model provides adaptability for making changes in DDoS detection approach in real time.

Ahmed et al. [52] proposed a mitigation method to defend against DNS query-based DDoS attacks using SDN features. The mitigation model is based on "Dirichlet Process Mixture Model (DPMM)" to differentiate the attack traffic from benign traffic. The proposed system is deployed on control plane. The SDN controller collects all the traffic from switches periodically. The model consists of three modules namely traffic statistics manager, learner component, and network resource manager. The first module captures the features of incoming flows from switches. Learner is responsible for detecting the malicious flows. The third module maintains a record of device's resource utilization. The switches are informed to update a new rule to block the traffic after detecting malicious flows. The proposed DPMM-based model is compared with a Mean-Shift algorithm-based model to evaluate the efficiency. The results indicate that DPMM model performs with higher accuracy than Mean-Shift model.

An approach named FADM for defending against DDoS threats was proposed in SDN environment [53]. FADM achieves a decent efficiency and lightweight properties. FADM incorporates two

Table 2. Machine Learning-based Defense Mechanisms

Ref.	Defense mechanism	Intrusion type	Targeted resources	Simulation tools	Controller	Advantages	Limitations
[62]	DDMF (Big data technologies such as Apache Spark-based traffic analysis)	ICMP flooding	Bandwidth	Real-time	Single	Fast traffic processing	Simulated for one type of DDoS traffic, flow-based detection that may compromise with accuracy
[53]	FADM (entropy-based analysis for feature selection and SVM for classification)	Flooding attacks (SYN, UDP, ICMP)	Bandwidth, memory	POX, Mininet	Single	Less response time, high accuracy, quick recovery	More recovery delay in case of SYN flood than other flooding attacks
[59]	Game theory (GT)–Holt-Winters for Digital Signature (HWDS)	UDP flooding, Portscan	Bandwidth	Real-time	Single	Operable with any SDN configuration for mitigation	Performed with only UDP-based DoS/DDoS
[52]	Dirichlet process mixture model (DPMM)-based clustering	DNS query-based DDoS (DNS amplification, DNS flooding)	CPU	_	Single	Less delays, acceptable overhead, increases accuracy, does not need modification or extra appliances	No mitigation provided
[64]	Machine-learning classification algorithm(sequential minimal optimization)	Flooding attacks (UDP, TCP-SYN, ICMP), misbehavior attack, newflow attack	Bandwidth, memory	POX, Mininet	Single	Supports wired or wireless both infrastructure, useful for cost-benefit analysis for mitigation	Processing overhead not discussed
[54]	SD-Anti-DDoS (neural network-based detection)	Flooding attacks (UDP, TCP-SYN, ICMP)	Bandwidth, memory	Ryu Mininet	Single	Fast response of detection, reduces controller load, no need to add extra hardware.	TFN2k tool is used to produce the traffic that is outdated nowadays.
[51]	Deep learning based	Flooding attacks (UDP, SYN, ARP, Smurf, PingofDeath)	Bandwidth, memory	_	Single	Reduces dependence on the hardware and software, easy to adapt the changes in real-time	Processing overhead

(Continued)

Table 2. Continued

Ref.	Defense mechanism	Intrusion type	Targeted resources	Simulation tools	Controller	Advantages	Limitations
[55]	FlowIDS (Deep learning based)	SMTP flooding attack	Bandwidth	ONOS, Mininet	Single	Quick detection and mitigation (by reducing network bandwidth consumption	Applicable for a specific type of DDoS, unable to simulate on multi-site
[50]	Deep learning based	Flooding attacks (ICMP, UDP, TCP)	Bandwidth, memory	POX, Real-time	Single	High accuracy with low false positives	Per-packet processing overhead for feature extraction, increases burden of the controller
[65]	FL-GUARD	ICMP flooding	Bandwidth	Floodlight, Mininet	Single	High detection accuracy, increases flexibility, other business of the abnormal hosts will not be affected.	Evaluation of all the performance metrics are not shown.
[63]	Clustering-based anomaly detection (density peak clustering with unsupervised cluster-based feature selection)	Information gathering attack, DoS, user to root, remote to local attacks (contained in KDDCup99)	Server, computer's resources (memory, processing power)	MINE, Scikit-learn, Numpy, SciPy, Matplotlib	Single	Quick mitigation, reduces redundant features, can handle high dimensional and unlabeled network data	Clustering algorithms may not give the accurate detection results.
[58]	Athena/DDoS anomaly detection algorithm (k-means, logistic regression)	DDoS flooding attacks	Bandwidth	ONOS, Mininet	Multi-controller	Highly scalable, reduces computational time for detection, reduces programming effort	Flow handling overhead, classification accuracy may be imperfect
[56]	ATLANTIC	TCP-SYN attack, Port Scan	Memory	Floodlight, Mininet	Single	Reduces the overhead of the overall detection mechanism, provides human intervention in case of automated modules not working	k-means can affect the detection accuracy.
[60]	SDN and machine-learning technique (SVM)-based detection	Amplification attacks (DNS, NTP)	Memory, CPU	ONOS, Real-time	Single	Can detect known or unknown attacks, reduces the burden of detection module, high scalability	False-positive/false-negative measures and processing overhead not discussed.

modules, i.e., detection and mitigation. In FADM, the network traffic statistics are analyzed by SDN controller using sFlow method. The proposed approach collects sufficient information for maintaining the desirable accuracy of the system. It cannot collect the information completely for high traffic rates. Current network features are extracted from the collected information. Proposed mitigation module depends on white-list and traffic migration. An entropy-based method is utilized for evaluating the network features and SVM is used for identifying DDoS attacks. The response and performance of the attack detection can be potentially enhanced by combining the proposed approach with the other methods. Experimental evaluation outcomes conclude that FADM can provide accurate detection and effective mitigation of various DDoS threats. In addition, FADM is capable of recovering the network in a very short time.

A method (SDN-Anti-DDoS) [54] was proposed and demonstrated to detect DDoS attacks in a fast and efficient manner by Cui et al. The proposed method consists of four modules each serving a special purpose. An attack trigger detection module is implemented to give response against an attack quickly as well as decreases load on controllers and switches. Neural networks are used for the detection of the attacks. Furthermore, authors also proposed a traceback method utilizing capabilities of SDN to track down the attack route. A mechanism is designed and deployed on RYU controller to obstruct the attacks and perform flow table cleaning. The results show that SD-Anti-DDoS can quickly detect the malicious activity within one second and discover the source of the attack. The proposed detection trigger mechanism can respond more quickly against the attack than other existing periodic trigger methods. It also decreases burden of the CPU and the controller. Most importantly, SDN-Anti-DDoS supports different variants of OpenFlow protocol.

An approach was presented to detect and mitigate SMTP flood attacks in SDN [55]. A framework named FlowIDS is used as a detection module to identify anomalies in SMTP flows. Decision tree (DT) and deep learning (DL) algorithms are used to classify malicious and benign traffic accurately. It is combined with Suricata NIDS for controlling and monitoring the traffic flows. The testbed used for the evaluation of proposed technique consists of systems having 8 Core Xeon CPU with 16GB RAM and 80GB storage each. The simulations of the FlowIDS framework were conducted for DT and DL in a single site. As per the results, DL provides better bandwidth utilization and faster network recovery than DT. Simulation of FlowIDS with DL algorithm in multi-site may be considered as future work.

da Silva et al. [56] designed a framework titled ATLANTIC for defending against DDoS was designed by utilizing the SDN features. ATLANTIC combines the functionalities of detection, classification, and mitigation. This framework attempts to block the malicious flows from external networks. It consists of two phases: a lightweight processing phase and a heavyweight processing phase for monitoring and defending against attacks, respectively. For first phase, entropy-based analysis is used for fast detection. The classification is based on supervised and unsupervised ML algorithms. Anomaly detection framework comprises of a classification layer, statistical layer, and a network layer. The statistical layer collects traffic flow statistics and delivers it to the classification layer. An entropy-based analysis is used for detecting variations in traffic features. SVM is used as a classifier and k-means is used for clustering. The experiment is performed considering two different attacks: port scanning and DDoS attack. ATLANTIC is deployed on the controller. SVM is shown to achieve accuracy of 88.7% with 82.3% precision. ATLANTIC minimizes the overhead of the proposed scheme on the controller. Ye et al. [57] also proposed a DDoS detection approach using SVM classifier. The proposed detection framework consists of flow status collection, features extraction, and classification of the extracted feature values. Authors implemented a feature extraction module to extract the features related to DDoS to train the classifier. They have used flooding-based attack traffic (TCP, UDP, and ICMP) to demonstrate the proposed detection approach. The results show that the approach gives an average accuracy of 95.24% and lower false alarm rate.

Lee et al. [58] examined the problem of integrating an "anomaly detection" development framework into current SDN deployments. They proposed a fully distributed application hosting architecture called Athena. The proposed framework facilitates good scalability as compared to previously developed SDN security frameworks. Athena is a software solution that provides an interface and some useful APIs for prototyping and generalizing different anomaly detection methods with minimal efforts required for programming. Authors have shown the effectiveness of proposed framework by considering three scenarios, a large-scale DDoS attack detector, Link Flooding Attacks (LFA) Mitigation, and Network Application Effectiveness (NAE). Only a DDoS scenario is considered in this work. In this scenario first, the DDoS detection model is created followed by feature validation phase then the testing is done at last. A clustering algorithm *k*-means is used for DDoS detection. The main advantage of using Athena is that it requires fewer lines for coding a DDoS detection algorithm, scalable and incurs less processing overhead as compared to Spark and Hama.

Assis et al. [59] introduced Game Theory based on Holt-Winters and Digital Signature (GT-HWDS) against DoS/DDoS threats. The objective of the proposed system is to defend against the attacks on SDN controller. This protection system works on two methods, i.e., HWDS system to identify the malicious activities, and GT-based solution facilitating the selection of an optimal defense strategy against an attack. The GT concept can be used in a more automatic way for mitigating the DDoS and other threats. In GT method, the issue is transformed into a game scenario of different players, i.e., attackers and the protection mechanism. It has the capability of quick decision making for the DDoS attacks. GT-HWDS system contains three interactive modules such as Detection, Information, and Mitigation modules. The detection module analyses seven IP flow dimensions simultaneously for characterizing the traffic behavior. Primary benefit of the system is its operability in any SDN configuration. It does not require any specific configurations for mitigation purpose. The proposed mitigation module can be used as an autonomous approach with other detection modules. Fuzzy-Genetic Algorithm and Digital Signature (GADS) is combined with game theory method for performance evaluation. The Fuzzy-GADS is analyzed with six dimensional traffic flows. The genetic algorithm performs characterization of the traffic flows. Fuzzy logic is used in detection of anomalies. The results show that GT-HWDS approach is more efficient and stronger than fuzzy logic-based approach. It efficiently reduces saturation on SDN controller.

Chen et al. [60] designed a detection method based on SDN for Distributed Reflection DoS (DRDoS) attacks based on amplification. A detection module is attached to SDN controller for detecting DRDDoS packets in the proposed architecture. This module consists of a traffic monitoring tool and a ML-based classifier. An open source software tool, i.e., Netmate is used to capture the network traffic. SVM is used to classify the traffic, it is trained to analyze DNS attack. In addition, it performs well with NTP attacks. It provides high accuracy. The detection module informs the controller about the malicious packets and instructs to block the attack. For experimental design, VMware ESXi is used to launch virtual machines with Open Network Operating System (ONOS) controller [61]. The method can detect and block both the attacks with less response time and blocking time. It is advantageous that controller reduces the burden of the detection module inspecting selected packets. This method can also detect unknown attacks.

Yan et al. [62] proposed a DDoS Detection and Mitigation Framework (DDMF) using features of SDN and Apache Spark. The objective of proposed framework is to discover and reduce the effects of DDoS in time. Apache Spark analyzes the network traffic more quickly. Capture Server, Detection Server (Cluster), and SDN Router Application, are three main components of DDMF. SDN Router Application is responsible for setting up the logic for flow rules of the packets. It controls the network and blocks the malicious flow entries. Capture Server captures the traffic flows and maintains a log file. A DDMF takes the advantage of SCP protocol to transfer the log

file to Detection Server that makes sure the integrity of the transferred files. Detection module analyzes the log file and notifies the router application to block the malicious flows. For detection of the attacks, several methods such as neural network, entropy based, counting based can be used. The simulation of DDMF is performed on the testbed. DDMF blocks simulative DDoS flows automatically on the basis of analysis of the traffic.

He et al. [63] presented two filtered algorithms against DDoS threats to handle large network traffic. The algorithms utilize SDN capabilities to detect anomalies in the network pattern. The proposed algorithms are as follows: "unsupervised cluster-based feature selection" and "density peak-based clustering with sampling adaption." The feature selection algorithm discards the redundant features in the dataset. It is suitable for continuous and discrete both features. Clustering is used for classifying the traffic into benign and anomalous data. The clustering algorithm provides better results in terms of runtime and memory efficiency. Evaluation shows better accuracy results.

Alshamrani et al. [64] proposed a defense system for defending a large variety of DDoS attacks in SDN. The defense system captures the traffic information from switches regularly and uses ML algorithm for classification. The DDoS detection module is based on the appropriate features selection of network traffic. Three algorithms such as ranker, genetic, and greedy algorithm are used for selecting a proper features subset. This system offers a good detection accuracy rate. Further, authors attempt to design other defense modules to mitigate two new attacks including misbehavior attack and newflow attack. The defense modules are executed over the controller. As per evaluated results, it reduces the attacker's capacity and maintains the services for normal users. It is more effective in terms of cost.

Liu et al. [65] presented a defense system named Floodlight Guard (FL-GUARD) to defend against DDoS in SDN. An anti-spoofing module of source IP is integrated with the Floodlight controller and the sFlow-RT collector component in the control layer. The attack detection and attack blocking modules are implemented in the application layer. The attack detection module uses C-SVM algorithm as a classifier to differentiate between normal and malicious flows. To block the attacks at the source port, the flow tables are assigned by utilizing the features of the SDN central control. According to simulation results, the proposed defense system provides a good accuracy by detecting the DDoS attacks effectively.

4.3 Application Specific/Collaborative Defense Mechanisms

This section is available as online supplementary material.

5 DEFENSE MECHANISMS FOR SDN AGAINST DDOS

Despite SDN's centralized control of the entire network to detect and mitigate the DDoS attacks, it is still open to many types of DDoS attacks, which must be addressed. Due to the separation of control logic out of the forwarding devices, it makes SDN vulnerable to several security threats. These threats need to have a point of focus of the researchers and many commercial vendors to secure the networks. Many detection and mitigation mechanisms have been provided to mitigate the DDoS attacks for securing SDN. Some defense solutions proposed in the literature are discussed in this section. Table 3 shows various defense mechanisms of application-specific types. These solutions are classified into different categories such as data plane solutions, control plane solutions, switch-controller-based collaborative solutions, collaborative intelligent switches, and integrated solutions on the basis of possible causes of the threats on the SDN planes.

5.1 Defense Against Attack on Data Plane

Data plane contains dumb switches that can help the attackers to be targeted. The OpenFlow switches in SDN are not capable to resolve the threat issues by their own. DoS/DDoS are the most

Table 3. Defense Mechanisms Against DDoS for SDN

Ref.	Defense mechanism	Intrusion type	Targeted resources	Simulation tools	Controller	Advantages	Limitations
[66]	Using token bucket algorithm	Table overflow attack	Memory	OpenDaylight, Real-time	Single	Efficient in accuracy	_
[68]	Statistical-based detection	Table-overflow attack	Memory	OMNeT++	_	More comprehensive approach as it covers all the header fields, efficient detection	Cannot detect the attacks that involve altering all the headers simultaneously
[69]	Entropy-based DDoS detection	Flooding attacks (UDP, TCP, SYN)	Bandwidth, memory	Pox, Mininet	Single	Reduces communication overhead on southbound interface	No prevention countermeasure considered, supported only for a single controller architecture additional overhead on the controller
[71]	SDNManager (based on bandwidth prediction)	DoS attacks (valid for all types)	Bandwidth, memory, CPU	Floodlight, Real-time	Multi-controller environment	Reduces workload of controller and control channel, smart switches, reduces overhead of controller, high detection rate	Not realized the attacks from multiple attacking machines (only DoS)
[72]	PATMOS	DDoS (No specific type mentioned)	_	POX, Mininet	Multi-controller	Provides better results using more controllers in the cluster	Can lead to security (interception) problems between controllers in cluster
[82]	FMD (based on flow migration)	DoS (No specific type mentioned)	_	Ryu, Mininet	Multi-controller	Reduces control plane & controller congestion	May cause problems between controllers
[74]	SLICOTS (Rule-based DDoS mitigation)	TCP-SYN flooding attacks	Memory	OpenDaylight, Mininet	Single	Smart Switches, switch can take decisions	Functioning only when controller behaves in reactive manner, may increase control plane bandwidth consumption and overload

(Continued)

Table 3. Continued

Ref.	Defense mechanism	Intrusion type	Targeted resources	Simulation tools	Controller	Advantages	Limitations
[75]	Avant-guard	TCP-SYN flooding	Memory	POX	Single	Improves resilience against TCP SYN flood and network scanning attacks	Restricted to the number of proxy ports, not easily deployable in practice, invalid to non-TCP protocols-based attacks
[76]	OverWatch (autoencoder-based detection)	Flooding attacks (SYN, UDP, ICMP)	Bandwidth, memory	Ryu, Real time	Single	Rapid reaction to threats reduces southbound interface overhead	_
[77]	SDNScore (statistical and packet-based defense mechanism)	TCP-SYN, DNS, SQL Slammer, NTP attacks	Memory, CPU	Discrete event simulator (name not mentioned)	Single	Can detect also new unknown attacks	Per-packet processing overhead
[78]	StateSec (entropy-based algorithm with such monitoring features (IP and port addresses))	DoS/DDoS flooding and port scan	Bandwidth	Ryu, OVS, Mininet	Single	Improves the reaction time to threats, reduces burden of the controller and control channel, also extended to detect slow DDoS [84]	Per-packet processing overhead
[80]	SDN-Guard	DoS flooding (TCP-SYN, UDP, ICMP)	Bandwidth, memory	Floodlight, Mininet	Single	Less deployment time, reduces switch to IDS traffic	Complex and processing overhead
[83]	Flood-guard (based on proactive flow rule analyzer and packet migration for mitigation)	DoS flooding (UDP, TCP, ICMP)	Bandwidth, memory	POX, Mininet (Software and hardware both types of evaluation)	Single	Modifiactions are not required in the existing SDN infrastructure	Lacks portability in network deployment, may not ensure fairness of non-attack traffic
[73]	Multi-layer fair queuing-based approach	UDP flooding	Memory, CPU	Floodlight, OVS, Mininet (testbed)	Single	Modifications and extra appliances are not required at data plane	Applicable for a specific type DDoS

convenient threats nowadays to damage the network. They can overload the SDN switches by flooding-based attacks. The SDN switches also suffer from limited size of TCAM. These issues should be addressed to mitigate the DDoS attacks on the data plane. In this section, some existing detection and mitigation mechanisms are presented. The defense mechanisms should be quick enough to react against the attacks. The data plane defense requires additional appliances or modifications in the OpenFlow switches that may be costly. Thus, defense mechanisms should also be cost-effective.

In Reference [66], Xu et al. proposed a model to defend against SDN-based table-overflow attacks. For solving the issue, a mathematical mechanism is designed according to SDN topology. Probable victim switches are defined based on mathematical formulation. A switch with very less vacancies is selected as a target switch (hot switch) for the attack. Attackers use a switch placed at midway in place of endpoints. A monitoring mechanism with three traffic flow features are defined to discover the attack and reveal the attackers. After monitoring the attack traffic, a token bucket algorithm-based defense model is implemented in the controller. This defense model ensures stable transmission rate of normal clients and limits the rate of attackers. Both monitoring and defense models are implemented into the actual routing applications of SDN. The effectiveness of monitoring mechanism is evaluated WAN, LAN, and data center frameworks. For simulation of mitigation model, Open vSwitch as SDN switches and OpenDaylight controller [67] are used. The results show that proposed method performs effectively by reducing the attack rate.

Durner et al. [68] introduced DoS attacks against data plane and their effects in SDN. Authors proposed a statistical approach to detect attacks and a lightweight mitigation method to stop the malicious flows. It focuses on table overflow attacks in switches due to flooding attacks. The detection approach is based on analysis of header fields in the flow tables. It maintains a table of headers of suspected attackers using hashing. Based on that table, new rules are defined to handle or block the attack. Simulations and testbed experiments have been used to analyze the performance of proposed approach. The detection method is performed using OMNeT++ tool on control plane level. The results show that attacks are detected efficiently and reliably with less false positives. The performance can be increased with a proper feature selection. Main limitation of this approach is that it is unable to identify the attackers who alter all header fields simultaneously.

5.2 Defense against Attack on Control Plane

The control plane provides complete visibility of the SDN network. The complete functionality of the network may be disturbed, in the case of control plane breaks down. Because of having a centralized policy, an SDN controller is the most convenient to be targeted by the DDoS threats. Therefore, it is necessary to protect the controller by utilizing some appropriate detection and mitigation mechanisms. All the proposed defense mechanisms try to minimize the controller resource saturation produced by the DoS/DDoS attacks. In this section, some defense mechanisms have been discussed. The detection mechanisms should use quick response methods for identifying the attacks.

Mousavi et al. [69] proposed a system to detect DDoS attacks against controllers in its early stages. The proposed system uses controller's functionalities to protect the SDN network. It aims to quickly detect the attack and to provide a proper mitigation solution before the controller goes off. An entropy method based on destination address is used for detection of the attacks. Entropy is calculated using two factors, i.e., defined window size and a threshold value in the proposed method. A functionality of collecting the destination IP addresses is integrated in the controller. An intrusion is reported if a certain threshold of computed entropy is crossed and vice versa. This method provides lightweight and fast detection of the malicious activities. The number of hosts can be dynamically changed for the proposed solution. The performance is tested on UDP

and TCP traffic. However, the proposed solution also supports ICMP traffic. Major benefit of this solution is adaptability. The parameters used in the proposed algorithm can be modified according to targeted results in real time. This is the first solution that is based on entropy for detecting the DDoS attacks in the SDN controller. A limitation is that the attacks cannot be identified against the whole network. It was designed only for a single controller architecture.

In Reference [70], another entropy-based approach was proposed by Sahoo et al. to detect low-rate attacks at the controller. Authors used generalized entropy and information distance between different probability distributions as detection metrics. With the help of extracted statistical features from switch flow tables an alarm is indicated that shows probable DDoS attack in early stages. The proposed approach gives fast and accurate detection rate when compared to Mousavi's approach [69].

Wang et al. [71] proposed a lightweight and quick DoS defense mechanism called "SDN-Manager." The proposed system consists of five modules, i.e., monitor, forecast engine, checker, updater, and storage service. The system analyzes the flow statistics, forecasts flow bandwidth changes based on these statistics, and updates the network accordingly. SDNManager applies a dynamic-time-series model to improve bandwidth prediction accuracy. SDNManager is implemented on the control plane. It is valid for all types of DoS attacks. Authors also proposed a dynamic controller scheduling (DCS) strategy in a multi-controller environment. The DCS strategy confirms the global network state optimization and defense efficiency. It assigns the controllers to switches dynamically according to the controller load. The average response time of the controller can be reduced by balancing an optimal mapping between controllers and switches. The defense scheme attempts to avoid both single point and cascading failure of controllers. For implementation purpose, a topology is used that consists of eight physical servers and four pica8 switches. The results show that forecast engine performs better than another forecasting model (ARCH). The effects of SDNManager is compared with the defense systems named SGuard and FloodGuard. SDNManager performs better in terms of bandwidth usage and CPU utilization. The DCS efficiency is evaluated using a data center topology containing 720 switches and 3,456 host users with 30 Floodlight controllers. It is observed by the results that response time using DCS is very less than without DCS strategy. It includes some overhead that is in a minor range.

In Reference [72], an approach (PATMOS) to mitigate DDoS attacks in multi-controller SDN environment using clustering of controllers was proposed. PATMOS involves three primary functions, i.e., searching for bottlenecks, leader (controller) election, and composition. First, overwhelmed controllers are searched. Second, a controller is selected as the leader to coordinate among the controllers. Last, controllers are clustered for mitigating DDoS collaboratively. A genetic algorithm is utilized that finds the highest count of controllers in each cluster for handling the DDoS traffic. It helps to optimize the resource being used, thereby increasing network uptime. Five scenarios are considered for experiment and validation of the proposed approach. The first scenario evaluates the controller's normal behavior. In other scenarios, different number of controllers are used to identify the clusters with PATMOS. The effectiveness of the approach is computed based on metrics like CPU usage, latency, throughput, and total received packets.

Zhang et al. [73] proposed a dynamic queue-based method, i.e., multi-layer fair queuing (MLFQ) to mitigate controller's resource saturation attack. This queue management system encourages to fairly share the controller's resources. These dynamic queues can be expanded in case for attack traffic and can be aggregated for benign message requests. Suggested work (2016) is added here.

5.3 Defense Against Control Plane Bandwidth Saturation

The control plane bandwidth is the most concerned target of the DDoS attacker after SDN controller.

In Reference [74], a defense model named SLICOTS to mitigate TCP-SYN flooding attacks by utilizing the SDN capabilities was presented. The mentioned DDoS attack reduces controller's performance. SLICOTS is employed in the control plane. It observes the running flows of requests and prevents the malicious requests in an effective manner. It is efficient for functioning only when controller behaves in a reactive manner. It informs the switches to block the malicious packets after identified an abnormal request. The effectiveness of SLICOTS is compared with a security model OPERETTA and ordinary SDN. It provides better results than OPERETTA based on metrics like response time, detection time, and CPU utilization. SLICOTS does not allow dropping of benign requests.

A framework named "AVANT-GAURD" that provides more defensive power to control plane was proposed in Reference [75]. This framework expands the forwarding plane for completing the TCP handshake process with the TCP source and communicates. In addition to this forwarding plane is only allowed to communicate with the controller. Thus, if the handshake process is successfully completed, forwarding plane permits TCP connection establishment after informing the destination. A significant and unavoidable delay is introduced because of TCP connections. This framework requires reprogramming of SDN switches to add custom features to the forwarding plane.

5.4 Defense by Collaborative Intelligence in Switches

In SDN, the controller is responsible for controlling all the switches and taking routing decisions. Thus, if switches do not find any matching flow entries for incoming packets in its flow table, all the packets are forwarded to controller. The switches are just simple forwarding devices. These switches are called as dumb switches as they cannot take decisions on their own. This characteristic introduces a large communication overhead and delay until the attack detection. This makes the controller overloaded and control channel congested. To overcome this issue, some researchers have proposed collaborative intelligence between switches and controller. Therefore, the switches also can take appropriate actions in detecting the malicious activities. This intelligence in simple forwarding switches may reduce the burden of the controller and control plane bandwidth. Some of these defense solutions are discussed in this section.

Han et al. [76] proposed a cross-plane DDoS attack defense framework named OverWatch. It accomplishes collaborative intelligence between forwarding devices and controller. This proposed framework includes two key methods, i.e., attack detection and reaction. The detection system consists of a coarse-grained sensor and an actuator for flow monitoring on the data plane and a fine-grained- ML based classifier on the control plane. The defense functionalities are split across forwarding and control planes to discover and mitigate the DDoS attacks on different levels. The performance of proposed framework is evaluated using a modified FPGA-based (Altera EP4SGX180) OpenFlow switch and a modified Ryu controller. Testbed of the experiment is performed using up to eight laptop hosts representing DDoS attackers, victims, and normal traffic generators, respectively. Experimental results show the efficiency of the defense system with high detection accuracy and real-time DDoS attack reaction. This method reduces communication overhead on SDN southbound interface.

Kalkan et al. [77] proposed a statistical and packet-based approach called SDNScore to provide defense against DDoS attacks in SDN infrastructure. In the proposed approach, switches are embedded with some intellectual features with the packet forwarding rules to take a decisive action. SDNScore is a hybrid mechanism that works on collaboration between switches and controller. The architecture of SDNScore comprises modules, i.e., profile, actuator, comparator, scorer, and pair-profiler. First four modules are situated on the switch and pair-profiler is implemented on the controller. All the modules coordinate with each other for detecting DDoS attacks. The

proposed approach is motivated by a statistical filtering approach (PacketScore). A score value is calculated using attributes of packets and compared with a threshold. Based on this score, packets are dropped or forwarded. The proposed method can also identify new unknown DDoS attacks. It refines the malicious packets with the help of packet-based analysis instead of blocking all the packets of a flow. The simulation outcomes prove the effectiveness of the proposed approach over entropy-based DDoS detection method.

A new stateful approach (StateSec) to protect communication endpoints from DDoS attacks in SDN environment was presented by Boite et al. [78]. The idea behind StateSec is to develop stateful data plane API for SDN with the help of OpenState specification [79]. It attempts to unload the controller and control channel by assigning local decision capabilities to the switches. In this approach, the switches work more smartly than classical SDN switches. Finite state machines are implemented inside the switches to achieve this goal. The complete defense idea depends on monitoring, detection, and mitigation of the attack traffic. Monitoring and detection functions are implemented inside the switches and mitigation is handled by the controller. A tool sFlow is integrated with Open vSwitch to analyze the incoming traffic. An entropy-based algorithm is used for detecting anomalies. The proposed method can detect multiple types of DDoS attack traffic. StateSec gives more efficient results in terms of detection rate and overhead on the control plane.

5.5 Integrated Mechanisms for DDoS Mitigation

Most of the proposed DDoS mitigation approaches address a particular SDN DDoS threat issue, i.e., switch overflow, controller saturation or controller bandwidth congestion. In this section, some mitigation solutions are shown that are able to reduce the impact of the mentioned threats simultaneously. By avoiding all the issues, it can increase the SDN network's performance efficiently.

Dridi et al. [80] proposed an approach named SDN-Guard to protect the SDN network against DoS attacks. It attempts to reduce the effects of DoS attacks on SDN controller, controller-switch bandwidth, and switch memory usage. This approach depends upon an Intrusion Detection System (IDS). SDN-Guard is placed over the controller as a security application. IDS sends a notification to this security module when it finds a malicious behavior. Based on this alert, SDN-Guard module takes suitable decisions to mitigate the attacks. Because IDS handles all the traffic flows of network, it may get overloaded. Therefore, primary aim of this approach is to find an optimal place for IDS, and to reduce the packet flows sent to IDS by switches. The results show that SDN-Guard minimizes the effect of DoS, and minimizes the controller, switch memory overloading and switch-controller channel consumption up to 32%. SDN-Guard uses traffic sampling to minimize packet loss and RTT while the network is going through DoS attack.

SDN and cloud are being used together to provide new ideas to design the network in programmable and portable manner. Security of SDN-based cloud is also a challenge that must be solved. To resolve this issue, Chen et al. utilized a ML-based classifier called extreme gradient boosting (XGBoost) to detect the DDoS attack in Reference [81]. XGBoost is implemented in the SDN controller as a detection module. Authors focused on the control channel congestion and controller resources saturation threats. In the SDN-based cloud scenario, different clouds are connected with each other. One cloud containing switches behaves as malicious cloud and targets to SDN devices in another cloud. Results show that XGBoost gives better accuracy and lower false-positive rate as compared to other classifiers (Random Forest, SVM and Gradient-based decision tree).

A flow migration defense (FMD) approach to protect the SDN network was suggested in Reference [82]. Main idea is based on the migration of flooding requests from a master controller to a slave controller. FMD is implemented in the controller and does not require any change in network. In this mitigation method, controller and switch-controller channel are protected from DoS

threats. In a normal scenario, the traffic requests are normally handled by the master controller. Upon detecting an attack, the suspected flows are migrated to the slave controller for processing. This migration of flows between controllers can protect the switch-to-master controller channel. The migrated requests are transferred to the master controller for further processing at a limited rate. It handles these requests with a dynamic adjustment of the requests depending on its actual workload. The performance of the FMD is evaluated in Mininet using Ryu controller. The authors also proposed an "adaptive rate adjustment (ARA)" method to gain a dynamic adjustment to handle flooding requests having no risk of overloading. The results are compared with existing approaches such as MLFQ and FloodDefender. FMD performs better in terms of response time, interface congestion, mitigation time, and packet loss ratio.

Wang et al. in 2015 [83] proposed a defense mechanism "Flood-Guard" against DoS attacks to avoid overloaded switch, control channel bandwidth, and the congested controller. Authors proposed a module named proactive flow rule analyzer that acts as a controller. This module controls all the new incoming packets in place of the controller during the attack. It works on the idea of dynamically changing of flow rules at runtime. It attempts to reduce the traffic burden on the overloaded controller. The proactive flow rule analyzer is not always efficient in providing accurate derivations. This approach was tested using both simulation and testbed. However, proposed approach may result in enhanced delay in processing the data packets that increases the time of setting up the new rules.

Furthermore, the security mechanisms are analyzed on the basis of threat cause and its impacts on the different planes. This analysis is shown in Table 4.

6 PERFORMANCE EVALUATION

Proposed different DDoS defense frameworks and solutions can be compared by evaluating some standard performance metrics. These performance metrics are as follows:

(1) Classification metrics: Performance of attack/non attack traffic classification approach is measured by some parameters, i.e., recall, precision, F-measure, accuracy, and ROC curve. These parameters are computed with the help of outcomes that are true-positive, false-positive, false-negative, and false-positive rates.
(2) Other performance metrics: The complete network can be analyzed by evaluating some important parameters. These parameters are end to end delay, CPU and memory usage, throughput, communication overhead, packet loss ratio, and network response time.

7 RESEARCH CHALLENGES AND ISSUES

We have identified several research issues from our extensive study. Most importantly, both detection and mitigation modules are essential requirements for securing a network. There are various research challenges and issues that need to be discussed and addressed for complete adoption of SDN technology.

7.1 Security for SDN Switches

SDN switches have very limited memory (TCAM) to store the flow rules for the new incoming traffic. Due to storage constraints, switches gain attention of DDoS flooding attackers. Flooding attacks send a large number of packets aimed to consume all the storage of the flow tables in switches. Exhaustion of SDN switches can also interrupt the functionality of network. Just because controller is the main component of SDN, most of the work has been done for providing security to the controller. However, security of the switches used in the data plane must be studied in the similar manner.

Table 4. Analysis of Defense Mechanisms for SDN

Authors/Ref.	Cause of attack	Defenseapproach	Switch overload	Control plane bandwidth congestion	Controller saturation
Wu et al. [82]	Centralized control	By flow migration between two controllers	_	Yes	Yes
Han et al. [76]	Dumb switches	By using collaborative intelligence between switch & controller	_	Yes	_
Wang et al. [71]	Single point & cascading failure of controllers	By bandwidth prediction & controller dynamic scheduling	_	_	Yes
Xu et al. [66]	Limited TCAM in switches	Using statistical and token bucket approach	Yes	_	_
Durner et al. [68]	Limited TCAM in switches	By using statistical approach and hashing function	Yes	_	_
Mohammadi et al. [74]	Separation of planes	By using dynamic programmability nature of SDN	_	Yes	_
Kalkan et al. [77]	Dumb switches	By making switches smarter to take actions	_	Yes	Yes
Boite et al. [78]	Dumb switches	By delegating local processing to switches	_	Yes	Yes
Dridi et al. [80]	Central control of SDN network	By leveraging an IDS and finding an optimal placement for IDS	Yes	Yes	Yes
Macedo et al. [72]	Single point of failure	Through clustering of the controllers	_	_	Yes
Mousavi et al. [69]	Single point of failure	Variation in entropy of destination IP address	_	_	Yes
Wang et al. [83]	Due to large amount of table-miss messages in switches	By proactive flow rule analyzer and packet migration	_	Yes	_
Shin et al. [75]	Separation of planes	Based on connection migration (inspired by SYN proxy)	_	Yes	_

7.2 Cost for Additional Hardwares

Researchers have reported few security suggestions and countermeasures for data plane security. These data plane defense proposals require modifications in the OpenFlow switches or usage of additional specific appliances. This increases the cost of setting up the SDN network. Therefore, some mechanisms should be proposed to overcome issues related with data plane security, while minimizing the overall network setup cost.

7.3 Tradeoff between Concepts of Actual SDN and Smart Switches

SDN switches are not capable to take any smart decision from their own in an unobvious situation. This property leads to sending huge volume of unknown traffic to the controller that can create communication overhead. Therefore, some researches have suggested providing intelligence to SDN switches for enabling it to take some decisive actions. This feature can reduce the burden on the controller and its chances to get collapsed. However, the main fundamental concept of the SDN having simple forwarding switches should not be compromised. Therefore, the system designer should be careful for this tradeoff between the actual SDN concept and the smarter switches.

7.4 Slow DDoS

It is very tough to discover slow and low-rate DDoS attacks, because traffic flows in slow attacks act just like benign traffic flows. It requires very less resources to get launched and even can make unavailable the services of the web servers using just one host. High-rate DDoS are more easier in a way than slow DDoS. Hence, slow DDoS attack mitigation needs some serious efforts and research work.

7.5 Lack of Standard Communication Protocols and Harmful Applications

For the communication of applications and control plane, there is no standard northbound interface used yet. Northbound interface provides a programmable nature to install the security and other required applications into the control plane. This open and programmable nature can make it vulnerable to malicious applications that can even change the complete network functionality and provide unexpected results. Attackers can implement their own policies and add to the controller to take the control of network in their hands. This insecure application-control channel may be a convenient target for the attackers, hence securing the communication channel becomes an important issue.

The flexibility and programmability are the key features of SDN. These features expose the network to the user applications. This may result in installation of malicious applications with fake rules. The malicious applications can degrade the performance of the controller. Protecting the controller from malicious applications can be considered as an another research area.

7.6 Scalability and Interoperability of Controllers

Controller is the most salient part of SDN. As network size increases, single controller is not capable to handle all the traffic alone. Therefore, backup and additional controllers are deployed to reduce the chance of single point of failure and to handle the traffic. However, different controllers have different policies and routing techniques. Distributed controllers suffer from scalability and interoperability. Interoperability of controllers being used in different networks for facilitating consistent network operation and scalability needs attention. An standard east-west bound interface is recommended for secure communication between controllers.

7.7 Efficient Analysis of Network Traffic

DDoS defense techniques need real-time monitoring and tracing of the network traffic to be analyzed. Use of tools (sFlow, netFlow, etc.) to monitor the traffic may cause additional overhead. The detection mechanisms may utilize packet-based or flow-based traffic analysis. The packet-based analysis imposes large overhead while increasing the accuracy while less overhead with low accuracy is achieved in flow-based analysis. Hence, finding the best trade-off between overhead and accuracy can also be considered as an important research issue.

7.8 A Sole Solution for all DDoS

A defense mechanism should be able to mitigate different kinds of DDoS attacks. Existing DDoS defense mechanisms found in literature can handle only a specific type of DDoS attack. These mechanisms are designed with a restricted hardware appliance and a fix functionality that are incapable of handling different kinds of attacks. These security solutions need to be enhanced to detect more types of attacks with minimum communication overhead. Thus developing a mechanism to defend multiple DDoS attacks is a major research issue.

7.9 Ability to Analyze Real DDoS

In most of the existing defense solutions, a small network scenario consisting of few devices is used to test the performance of the solutions. Hence, the performance of these solutions may not work well in the case of large networks. Such small networking environment may not be able to demonstrate the defense of real DDoS attacks completely. This issue should be considered as a major research challenge. Therefore, there is a need to analyze the real attacks defense by providing scalability in various large network scenarios. The defense mechanisms should be designed in a way to be able to analyze real attack cases.

8 OUR RESEARCH CONTRIBUTION AND ITS SIGNIFICANCE

In this research article, a detailed study on SDN security issues is done. There are few research articles that are available in literature related to DDoS attacks and SDN security. Wang et al. [85] discussed various security issues in the cloud environment in 2015 that can be handled by flexible SDN paradigm. For securing cloud, Yan et al. [86] have discussed the DDoS and SDN security. One existing work [87] has described the DDoS vulnerabilities, detection and mitigation mechanisms by using centralized characteristic of SDN. They divided the discussed mechanisms on the basis of used detection methods such as based of entropy, ML, and traffic analysis. Further, the mitigation methods such as dropping the packets or blocking the port are also studied. However, the authors have not studied DDoS attacks for the SDN security solutions. Kalkan et al. [88] have presented DDoS defense solutions in the SDN environment and classified these solutions on the basis of detection methods. Imran et al. [89] provided various defense mechanisms against DoS attacks and classified them according to their strategies to reduce the impact of attacks. Authors have also presented some limitations of existing defensive mechanisms.

The motivation behind this research work is to highlight the present security challenges and their countermeasures in the domain of SDN-based DDoS defense mechanisms. It covers most of the possible issues altogether that must be considered for effective security of SDN-based networks. The SDN and DDoS have an antithetical relationship with each other. On one side, SDN can help in defeating DDoS attacks by utilizing its security features. On the other side, SDN itself becomes a target of attackers because of its inherent design issues. As far as our literature is concerned, this antithetical correspondence of DDoS and SDN security solutions is not studied deeply in earlier research. This work contains SDN-specific DDoS threats and its impact on the SDN architecture. The article covers all the potential DDoS-based vulnerabilities that can cause harm to SDN. It is also observed that SDN can be collaborated with some new technologies such as NFV, Blockchain, Smart contracts, IoT, Honeynet, and so on. This collaboration of technologies can improve the existing mitigation systems. Such approaches can provide the adaptability and dynamic functionality to the cloud vendors and their customers.

9 NEW RESEARCH DIRECTIONS

A significant research direction can also be combining SDN with information-centric networks (ICN). ICN has come to the fore as the traditional networking paradigms are host-centric whereas

the user's main focus is on getting access to the information regardless of from where it comes. To support this concept, ICN [90] extracts the information or the content from the IP packets. ICN [91] offers named-based routing and in-network caching to the modern networks. It provides fine grained control on the information transfer. The integration of ICN with SDN provides better content-centric security due to adaptive nature of SDN and inherent working of ICN. ICN has direct access to content as it is able to do named-based caching of content at intermediate network nodes. SDN can obtain such content available in the packets directly. The transmission of named content instead of full IP packet reduces the network overhead and increases the throughput. The integration of SDN and ICN improves management of the network as well as providing security services to each other [92]. ICN prevents the data plane from different threats by offering self-certifying names and content-based security mechanisms. ICN reduces the probability of spoofing and interception of communication on switch-controller bandwidth by giving some authenticated control messages. One important security benefit of ICN is that it focuses on content rather than location or IP address so that the controller cannot be impersonated by some malicious/fake entity. Therefore, the integration of ICN and SDN should offer better and efficient network services.

A new research direction can be enhancement of security in Bring Your Own Computer (BYOD) policy using SDN. According to BYOD policy, employees are free to use their own mobile devices and gadgets to access the workplace's services, which is going to become a trend soon. Centralized control of SDN may keep an eye at all the activities happening in BYOD environment for the security perspective. SDN makes it possible to quickly deployment of security services to provide more security to BYOD organizations.

It is already known that SDN suffers from various security issues due to centralized and open programmable behavior. Therefore, SDN controller should incorporate network behavior measurement modules for its security. For early detection of anomalies and malicious activities integration of measurement tools and SDN can be considered as a new research direction. Some important network state measuring parameters could be network latency, available bandwidth, and topology discovery. The measurement tools for these parameters can be deployed in the controller to monitor all the states in the network periodically. The analyzed network behavior may provide advance information against imminent threats.

A new research direction is fusion of SDN and traditional/legacy networks. Due to high cost of SDN devices, it may not be economically feasible to set up a pure SDN network. A unified network using SDN and non SDN technologies may offer benefits of SDN for easy deployment of security services. As SDN attracts attacks such as DDoS, code injection, man-in-the middle, and so on, efficient defense mechanisms are needed to secure the network. However, security of such networks that use both SDN and legacy network technologies is an important open research direction.

10 CONCLUSION

This work is significantly centered around the recent advancements and progressions in detection and mitigation procedures for defending SDN security from DDoS. Two prospectives of SDN security are considered. In first, SDN may help to protect the traditional networks while in second SDN may be a victim itself. Various defense mechanisms are classified into two categories, i.e., defense by SDN and defense for SDN that are based on the design characteristics of the SDN architecture. Further, a comparison of these mechanisms has been discussed on the basis of the detection and mitigation algorithms. It is concluded that there has been a significant growth in the research field of providing security by utilizing the SDN features. However, from the viewpoint of the impacts of DDoS threats in SDN, it can be a DDoS target itself because of its centralized nature. SDN is not completely secure hence, there is a need to explore more efficient defense mechanisms for DDoS mitigation.

REFERENCES

[1] Nick Feamster, Jennifer Rexford, and Ellen Zegura. 2014. The road to SDN: An intellectual history of programmable networks. *ACM SIGCOMM Comput. Commun. Rev.* 44, 2 (2014), 87–98.

[2] Akram Hakiri, Aniruddha Gokhale, Pascal Berthou, Douglas C. Schmidt, and Thierry Gayraud. 2014. Software-defined networking: Challenges and research opportunities for future internet. *Comput. Netw.* 75 (2014), 453–471. DOI: http://dx.doi.org/10.1016/j.comnet.2014.10.015

[3] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig. 2015. Software-defined networking: A comprehensive survey. *Proc. IEEE* 103, 1 (Jan. 2015), 14–76. DOI: http://dx.doi.org/10.1109/JPROC.2014.2371999

[4] J. Tourrilhes, P. Sharma, S. Banerjee, and J. Pettit. 2014. SDN and OpenFlow evolution: A standards perspective. *Computer* 47, 11 (Nov. 2014), 22–29. DOI: http://dx.doi.org/10.1109/MC.2014.326

[5] Paul Goransson, Chuck Black, and Timothy Culver. 2016. *Software Defined Networks: A Comprehensive Approach*. Morgan Kaufmann.

[6] Keith Kirkpatrick. 2013. Software-defined networking. *Commun. ACM* 56, 9 (2013), 16–19.

[7] H. Kim and N. Feamster. 2013. Improving network management with software defined networking. *IEEE Commun. Mag.* 51, 2 (Feb. 2013), 114–119. DOI: http://dx.doi.org/10.1109/MCOM.2013.6461195

[8] Per Oscarson. 2003. Information security fundamentals. In *Security Education and Critical Infrastructures*. Springer, Berlin, 95–107.

[9] Muhammad Umar Farooq, Muhammad Waseem, Anjum Khairi, and Sadia Mazhar. 2015. A critical analysis on the security concerns of internet of things (IoT). *Int. J. Comput. Appl.* 111, 7 (2015).

[10] Christos Douligeris and Aikaterini Mitrokotsa. 2004. DDoS attacks and defense mechanisms: Classification and state-of-the-art. *Comput. Netw.* 44, 5 (2004), 643–666.

[11] Stephen M. Specht and Ruby B. Lee. 2004. Distributed denial of service: Taxonomies of attacks, tools, and countermeasures. In *Proceedings of the International Society for Computers and Their Applications and the International Conference on Parallel and Distributed Computing Systems (ISCA PDCS'04)*. 543–550.

[12] A. D. Wood and J. A. Stankovic. 2002. Denial of service in sensor networks. *Computer* 35, 10 (Oct. 2002), 54–62. DOI: http://dx.doi.org/10.1109/MC.2002.1039518

[13] Dan Goodin. 2018. US service provider survives the biggest recorded DDoS in history. Retrieved March 30, 2018 from https://arstechnica.com/information-technology/2018/03/us-service-provider-survives-the-biggest-recorded-ddos-in-history/.

[14] Rob Enns. 2006. NETCONF configuration protocol. Technical Report RFC 4741.

[15] Xianfeng Li and Wencong Xie. 2017. CRAFT: A cache reduction architecture for flow tables in software-defined networks. In *Proceedings of the 2017 IEEE Symposium on Computers and Communications (ISCC'17)*. 967–972. DOI: http://dx.doi.org/10.1109/ISCC.2017.8024651

[16] Naga Katta, Omid Alipourfard, Jennifer Rexford, and David Walker. 2016. CacheFlow: Dependency-aware rule-caching for software-defined networks. In *Proceedings of the Symposium on SDN Research (SOSR'16)*. ACM, New York, NY, Article 6, 12 pages. DOI: http://dx.doi.org/10.1145/2890955.2890969

[17] Giuseppe Bianchi, Marco Bonola, Antonio Capone, and Carmelo Cascone. 2014. OpenState: Programming platform-independent stateful openflow applications inside the switch. *SIGCOMM Comput. Commun. Rev.* 44, 2 (Apr. 2014), 44–51. DOI: http://dx.doi.org/10.1145/2602204.2602211

[18] CALYPTIX. 2015. DDoS Attacks 101: Types, targets, and motivations. Retrieved April 26, 2015 from https://www.calyptix.com/top-threats/ddos-attacks-101-types-targets-motivations/.

[19] B. B. Gupta, R. C. Joshi, and Manoj Misra. 2009. Defending against distributed denial of service attacks: Issues and challenges. *Inf. Secur. J.* 18, 5 (2009), 224–247. DOI: http://dx.doi.org/10.1080/19393550903317070

[20] N. Muraleedharan and B. Janet. 2017. Behaviour analysis of HTTP based slow denial of service attack. In *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET'17)*. IEEE, 1851–1856.

[21] B. B. Gupta and Omkar P. Badve. 2017. Taxonomy of DoS and DDoS attacks and desirable defense mechanism in a cloud computing environment. *Neur. Comput. Appl.* 28, 12 (2017), 3655–3682.

[22] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred. 2003. Statistical approaches to DDoS attack detection and response. In *Proceedings DARPA Information Survivability Conference and Exposition*, Vol. 1. 303–314. DOI: http://dx.doi.org/10.1109/DISCEX.2003.1194894

[23] D. Sattar, A. Matrawy, and O. Adeojo. 2016. Adaptive bubble burst (ABB): Mitigating DDoS attacks in software-defined networks. In *Proceedings of the 2016 17th International Telecommunications Network Strategy and Planning Symposium (Networks'16)*. 50–55. DOI: http://dx.doi.org/10.1109/NETWKS.2016.7751152

[24] Ahmad Ariff Aizuddin, Mohd Atan, Megat Norulazmi, Megat Mohamed Noor, Shadil Akimi, and Zainal Abidin. 2017. DNS amplification attack detection and mitigation via sFlow with security-centric SDN. In *Proceedings of the 11th*

International Conference on Ubiquitous Information Management and Communication (IMCOM'17). ACM, New York, NY, Article 3, 7 pages. DOI: http://dx.doi.org/10.1145/3022227.3022230

[25] Sonia Panchen, Peter Phaal, and Neil McKee. 2001. InMon corporation's sFlow: A method for monitoring traffic in switched and routed networks. Technical Report RFC 3176.

[26] Jiaqi Yan and Dong Jin. 2015. VT-Mininet: Virtual-time-enabled mininet for scalable and accurate software-define network emulation. In *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research (SOSR'15)*. ACM, New York, NY, Article 27, 7 pages. DOI: http://dx.doi.org/10.1145/2774993.2775012

[27] Ben Pfaff, Justin Pettit, Teemu Koponen, Ethan J. Jackson, Andy Zhou, Jarno Rajahalme, Jesse Gross, Alex Wang, Joe Stringer, Pravin Shelar, et al. 2015. The design and implementation of open vSwitch. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI'15)*. 117–130.

[28] Christian Rossow. 2014. Amplification hell: Revisiting network protocols for DDoS abuse. In *Proceedings of the Network and Distributed System Security Symposium (NDSS'14)*.

[29] David Huistra. 2013. Detecting reflection attacks in DNS flows. In *19th Twente Student Conference on IT*. https://pdfs.semanticscholar.org/4ad8/24537f212f70e25e4cbab55498f5a8e43942.pdf.

[30] Q. Yan, Q. Gong, and F. R. Yu. 2017. Effective software-defined networking controller scheduling method to mitigate DDoS attacks. *Electron. Lett.* 53, 7 (2017), 469–471. DOI: http://dx.doi.org/10.1049/el.2016.2234

[31] Ligia Rodrigues Prete, A. A. Shinoda, C. M. Schweitzer, and R. L. S. de Oliveira. 2014. Simulation in an SDN network scenario using the POX controller. In *Proceedings of the 2014 IEEE Colombian Conference on Communications and Computing (COLCOM'14)*. 1–6. DOI: http://dx.doi.org/10.1109/ColComCon.2014.6860403

[32] Ming-Hung Chen, Jyun-Yan Ciou, I-Hsin Chung, and Cheng-Fu Chou. 2018. FlexProtect: A SDN-based DDoS attack protection architecture for multi-tenant data centers. In *Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia'18)*. ACM, New York, NY, 202–209. DOI: http://dx.doi.org/10.1145/3149457.3149476

[33] J. Zheng, Q. Li, G. Gu, J. Cao, D. K. Y. Yau, and J. Wu. 2018. Realtime DDoS defense using COTS SDN switches via adaptive correlation analysis. *IEEE Trans. Inf. Forens. Secur.* 13, 7 (Jul. 2018), 1838–1853. DOI: http://dx.doi.org/10.1109/TIFS.2018.2805600

[34] C. Buragohain and N. Medhi. 2016. FlowTrApp: An SDN based architecture for DDoS attack detection and mitigation in data centers. In *Proceedings of the 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN'16)*. 519–524. DOI: http://dx.doi.org/10.1109/SPIN.2016.7566750

[35] K. Hong, Y. Kim, H. Choi, and J. Park. 2018. SDN-assisted slow HTTP DDoS attack defense method. *IEEE Commun. Lett.* 22, 4 (April 2018), 688–691. DOI: http://dx.doi.org/10.1109/LCOMM.2017.2766636

[36] Mark Shtern, Roni Sandel, Marin Litoiu, Chris Bachalo, and Vasileios Theodorou. 2014. Towards mitigation of low and slow application ddos attacks. In *Proceedings of the 2014 IEEE International Conference on Cloud Engineering (IC2E'14)*. IEEE, 604–609.

[37] Thomas Lukaseder, Lisa Maile, Benjamin Erb, and Frank Kargl. 2018. SDN-assisted network-based mitigation of slow DDoS attacks. In *Proceedings of the International Conference on Security and Privacy in Communication Systems*. Springer, 102–121.

[38] Nikhil Tripathi, Neminath Hubballi, and Yogendra Singh. 2016. How secure are web servers? An empirical study of slow HTTP DoS attacks and detection. In *Proceedings of the 2016 11th International Conference on Availability, Reliability and Security (ARES'16)*. IEEE, 454–463.

[39] Tetsuya Hirakawa, Kanayo Ogura, Bhed Bahadur Bista, and Toyoo Takata. 2016. A defense method against distributed slow HTTP DoS attack. In *Proceedings of the 2016 19th International Conference on Network-Based Information Systems (NBiS'16)*. IEEE, 152–158.

[40] Clifford Kemp, Chad Calvert, and Taghi Khoshgoftaar. 2018. Utilizing netflow data to detect slow read attacks. In *Proceedings of the 2018 IEEE International Conference on Information Reuse and Integration (IRI'18)*. IEEE, 108–116.

[41] Truong Thu Huong and Nguyen Huu Thanh. 2017. Software defined networking-based one-packet DDoS mitigation architecture. In *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication (IMCOM'17)*. ACM, New York, NY, Article 110, 7 pages. DOI: http://dx.doi.org/10.1145/3022227.3022336

[42] Rishikesh Sahay, Gregory Blanc, Zonghua Zhang, Khalifa Toumi, and Hervé Debar. 2017. Adaptive policy-driven attack mitigation in SDN. In *Proceedings of the 1st International Workshop on Security and Dependability of Multi-Domain Infrastructures (XDOMO'17)*. ACM, New York, NY, Article 4, 6 pages. DOI: http://dx.doi.org/10.1145/3071064.3071068

[43] Rishikesh Sahay, Gregory Blanc, Zonghua Zhang, and Hervé Debar. 2017. ArOMA: An SDN based autonomic DDoS mitigation framework. *Comput. Secur.* 70 (2017), 482–499. DOI: http://dx.doi.org/10.1016/j.cose.2017.07.008

[44] Sufian Hameed and Hassan Ahmed Khan. 2018. SDN based collaborative scheme for mitigation of DDoS attacks. *Fut. Internet* 10, 3, Article 23 (2018). DOI: http://dx.doi.org/10.3390/fi10030023

[45] Lei Wang, Qing Li, Yong Jiang, Xuya Jia, and Jianping Wu. 2018. Woodpecker: Detecting and mitigating link-flooding attacks via SDN. *Comput. Netw.* 147 (2018), 1–13.

[46] Hojjat Adeli and Shih-Lin Hung. 1994. *Machine Learning: Neural Networks, Genetic Algorithms, and Fuzzy Systems*. John Wiley & Sons, Inc., New York, NY.

[47] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the 21st International Conference on Machine Learning (ICML'04)*. ACM, New York, NY, 104. DOI: http://dx.doi.org/10.1145/1015330.1015341

[48] Teuvo Kohonen. 1998. The self-organizing map. *Neurocomputing* 21, 1 (1998), 1–6. DOI: http://dx.doi.org/10.1016/S0925-2312(98)00030-7

[49] J. Ashraf and S. Latif. 2014. Handling intrusion and DDoS attacks in software defined networks using machine learning techniques. In *Proceedings of the 2014 National Software Engineering Conference*. 55–60. DOI: http://dx.doi.org/10.1109/NSEC.2014.6998241

[50] Quamar, Weiqing Sun, and Ahmad Y. Javaid. 2016. A deep learning based DDoS detection system in software-defined networking (SDN). *CoRR* abs/1611.07400 (2016). http://arxiv.org/abs/1611.07400

[51] Chuanhuang Li, Yan Wu, Xiaoyong Yuan, Zhengjun Sun, Weiming Wang, Xiaolin Li, and Liang Gong. 2018. Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN. *Int. J. Commun. Syst.* 31, 5 (2018), e3497.

[52] M. E. Ahmed, H. Kim, and M. Park. 2017. Mitigating DNS query-based DDoS attacks with machine learning on software-defined networking. In *Proceedings of the 2017 IEEE Military Communications Conference (MILCOM'17)*. 11–16. DOI: http://dx.doi.org/10.1109/MILCOM.2017.8170802

[53] D. Hu, P. Hong, and Y. Chen. 2017. FADM: DDoS flooding attack detection and mitigation system in software-defined networking. In *Proceedings of the 2017 IEEE Global Communications Conference (GLOBECOM'17)*. 1–7. DOI: http://dx.doi.org/10.1109/GLOCOM.2017.8254023

[54] Yunhe Cui, Lianshan Yan, Saifei Li, Huanlai Xing, Wei Pan, Jian Zhu, and Xiaoyang Zheng. 2016. SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks. *J. Netw. Comput. Appl.* 68 (2016), 65–79. DOI: http://dx.doi.org/10.1016/j.jnca.2016.04.005

[55] Mohd Zafran Abdul Aziz and Koji Okamura. 2017. Leveraging SDN for detection and mitigation SMTP flood attack through deep learning analysis techniques. *Int. J. Comput. Sci. Netw. Secur.* 17, 10 (2017), 166.

[56] A. Santos da Silva, J. A. Wickboldt, L. Z. Granville, and A. Schaeffer-Filho. 2016. ATLANTIC: A framework for anomaly traffic detection, classification, and mitigation in SDN. In *Proceedings of the 2016 IEEE/IFIP Network Operations and Management Symposium (NOMS'16)*. 27–35. DOI: http://dx.doi.org/10.1109/NOMS.2016.7502793

[57] Jin Ye, Xiangyang Cheng, Jian Zhu, Luting Feng, and Ling Song. 2018. A DDoS attack detection method based on SVM in software defined network. *Security and Communication Networks* 2018 (2018).

[58] S. Lee, J. Kim, S. Shin, P. Porras, and V. Yegneswaran. 2017. Athena: A framework for scalable anomaly detection in software-defined networks. In *Proceedings of the 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'17)*. 249–260. DOI: http://dx.doi.org/10.1109/DSN.2017.42

[59] M. V. O. De Assis, A. H. Hamamoto, T. Abrão, and M. L. Proença. 2017. A game theoretical based system using holt-winters and genetic algorithm with fuzzy logic for DoS/DDoS mitigation on SDN networks. *IEEE Access* 5 (2017), 9485–9496. DOI: http://dx.doi.org/10.1109/ACCESS.2017.2702341

[60] C. C. Chen, Y. R. Chen, W. C. Lu, S. C. Tsai, and M. C. Yang. 2017. Detecting amplification attacks with software defined networking. In *Proceedings of the 2017 IEEE Conference on Dependable and Secure Computing*. 195–201. DOI: http://dx.doi.org/10.1109/DESEC.2017.8073807

[61] Pankaj Berde, Matteo Gerola, Jonathan Hart, Yuta Higuchi, Masayoshi Kobayashi, Toshio Koide, Bob Lantz, Brian O'Connor, Pavlin Radoslavov, William Snow, et al. 2014. ONOS: Towards an open, distributed SDN OS. In *Proceedings of the 3rd Workshop on Hot Topics in Software Defined Networking*. ACM, 1–6.

[62] Qiao Yan and Wenyao Huang. 2017. A DDoS detection and mitigation system framework based on spark and SDN. In *Smart Computing and Communication*, Meikang Qiu (Ed.). Springer International Publishing, Cham, 350–358.

[63] D. He, S. Chan, X. Ni, and M. Guizani. 2017. Software-defined-networking-enabled traffic anomaly detection and mitigation. *IEEE IoT J.* 4, 6 (Dec. 2017), 1890–1898. DOI: http://dx.doi.org/10.1109/JIOT.2017.2694702

[64] Adel Alshamrani, Ankur Chowdhary, Sandeep Pisharody, Duo Lu, and Dijiang Huang. 2017. A defense system for defeating DDoS attacks in SDN based networks. In *Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access (MobiWac'17)*. ACM, New York, NY, 83–92. DOI: http://dx.doi.org/10.1145/3132062.3132074

[65] Jing Liu, Yingxu Lai, and Shixuan Zhang. 2017. FL-GUARD: A detection and defense system for DDoS attack in SDN. In *Proceedings of the 2017 International Conference on Cryptography, Security and Privacy (ICCSP'17)*. ACM, New York, NY, 107–111. DOI: http://dx.doi.org/10.1145/3058060.3058074

[66] T. Xu, D. Gao, P. Dong, C. H. Foh, and H. Zhang. 2017. Mitigating the table-overflow attack in software-defined networking. *IEEE Trans. Netw. Serv. Manage.* 14, 4 (Dec. 2017), 1086–1097. DOI: http://dx.doi.org/10.1109/TNSM.2017.2758796

[67] Z. K. Khattak, M. Awais, and A. Iqbal. 2014. Performance evaluation of OpenDaylight SDN controller. In *Proceedings of the 2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS'14)*. 671–676. DOI: http://dx.doi.org/10.1109/PADSW.2014.7097868

[68] R. Durner, C. Lorenz, M. Wiedemann, and W. Kellerer. 2017. Detecting and mitigating denial of service attacks against the data plane in software defined networks. In *Proceedings of the 2017 IEEE Conference on Network Softwarization (NetSoft'17)*. 1–6. DOI: http://dx.doi.org/10.1109/NETSOFT.2017.8004229

[69] S. M. Mousavi and M. St-Hilaire. 2015. Early detection of DDoS attacks against SDN controllers. In *Proceedings of the 2015 International Conference on Computing, Networking and Communications (ICNC'15)*. 77–81. DOI: http://dx.doi.org/10.1109/ICCNC.2015.7069319

[70] Kshira Sagar Sahoo, Deepak Puthal, Mayank Tiwary, Joel J. P. C. Rodrigues, Bibhudatta Sahoo, and Ratnakar Dash. 2018. An early detection of low-rate DDoS attack to SDN based data center networks using information distance metrics. *Fut. Gener. Comput. Syst.* 89 (2018), 685–697.

[71] Tao Wang, Hongchang Chen, Guozhen Cheng, and Yulin Lu. 2018. SDNManager: A safeguard architecture for SDN DoS attacks based on bandwidth prediction. *Security and Communication Networks* 2018 (2018).

[72] R. Macedo, R. de Castro, A. Santos, Y. Ghamri-Doudane, and M. Nogueira. 2016. Self-Organized SDN controller cluster conformations against DDoS attacks effects. In *Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM'16)*. 1–6. DOI: http://dx.doi.org/10.1109/GLOCOM.2016.7842259

[73] Peng Zhang, Huanzhao Wang, Chengchen Hu, and Chuang Lin. 2016. On denial of service attacks in software defined networks. *IEEE Netw.* 30, 6 (2016), 28–33.

[74] R. Mohammadi, R. Javidan, and M. Conti. 2017. SLICOTS: An SDN-Based lightweight countermeasure for TCP SYN flooding attacks. *IEEE Trans. Netw. Serv. Manage.* 14, 2 (Jun. 2017), 487–497. DOI: http://dx.doi.org/10.1109/TNSM.2017.2701549

[75] Seungwon Shin, Vinod Yegneswaran, Phillip A. Porras, and Guofei Gu. 2013. AVANT-GUARD: Scalable and vigilant switch flow management in software-defined networks. In *Proceedings of the ACM Conference on Computer and Communications Security*.

[76] Biao Han, Xiangrui Yang, Zhigang Sun, Jinfeng Huang, and Jinshu Su. 2018. OverWatch: A cross-plane DDoS attack defense framework with collaborative intelligence in SDN. *Security and Communication Networks* 2018 (2018).

[77] K. Kalkan, G. Gür, and F. Alagöz. 2017. SDNScore: A statistical defense mechanism against DDoS attacks in SDN environment. In *Proceedings of the 2017 IEEE Symposium on Computers and Communications (ISCC'17)*. 669–675. DOI: http://dx.doi.org/10.1109/ISCC.2017.8024605

[78] J. Boite, P. A. Nardin, F. Rebecchi, M. Bouet, and V. Conan. 2017. Statesec: Stateful monitoring for DDoS protection in software defined networks. In *Proceedings of the 2017 IEEE Conference on Network Softwarization (NetSoft'17)*. 1–9. DOI: http://dx.doi.org/10.1109/NETSOFT.2017.8004113

[79] Giuseppe Bianchi, Marco Bonola, Antonio Capone, and Carmelo Cascone. 2014. OpenState: Programming platform-independent stateful openflow applications inside the switch. *ACM SIGCOMM Comput. Commun. Rev.* 44, 2 (2014), 44–51.

[80] Lobna Dridi and Mohamed Faten Zhani. 2017. A holistic approach to mitigating DoS attacks in SDN networks. *Int. J. Netw. Manage.* 28, 1 (2017), e1996.

[81] Zhuo Chen, Fu Jiang, Yijun Cheng, Xin Gu, Weirong Liu, and Jun Peng. 2018. XGBoost classifier for DDoS attack detection and analysis in SDN-Based cloud. In *Proceedings of the 2018 IEEE International Conference on Big Data and Smart Computing (BigComp'18)*. IEEE, 251–256.

[82] Pengpeng Wu, Lin Yao, Chi Lin, Guowei Wu, and Mohammad S. Obaidat. 2018. FMD: A DoS mitigation scheme based on flow migration in software-defined-networking. *Int. J. Commun. Syst.* 31, 9 (2018), e3543. DOI: http://dx.doi.org/10.1002/dac.3543 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/dac.3543

[83] H. Wang, L. Xu, and G. Gu. 2015. FloodGuard: A DoS attack prevention extension in software-defined networks. In *Proceedings of the 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. 239–250. DOI: http://dx.doi.org/10.1109/DSN.2015.27

[84] F. Rebecchi, J. Boite, P. A. Nardin, M. Bouet, and V. Conan. 2017. Traffic monitoring and DDoS detection using stateful SDN. In *Proceedings of the 2017 IEEE Conference on Network Softwarization (NetSoft'17)*. 1–2. DOI: http://dx.doi.org/10.1109/NETSOFT.2017.8004256

[85] Bing Wang, Yao Zheng, Wenjing Lou, and Y. Thomas Hou. 2015. DDoS attack protection in the era of cloud computing and software-defined networking. *Comput. Netw.* 81 (2015), 308–319. DOI: http://dx.doi.org/10.1016/j.comnet.2015.02.026

[86] Q. Yan, F. R. Yu, Q. Gong, and J. Li. 2016. Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. *IEEE Commun. Surv. Tutor.* 18, 1 (Firstquarter 2016), 602–622. DOI: http://dx.doi.org/10.1109/COMST.2015.2487361

[87] Narmeen Zakaria Bawany, Jawwad A. Shamsi, and Khaled Salah. 2017. DDoS attack detection and mitigation using SDN: Methods, practices, and solutions. *Arab. J. Sci. Eng.* 42, 2 (01 Feb. 2017), 425–441. DOI: http://dx.doi.org/10.1007/s13369-017-2414-5

[88] K. Kalkan, G. Gur, and F. Alagoz. 2017. Defense mechanisms against DDoS attacks in SDN environment. *IEEE Commun. Mag.* 55, 9 (2017), 175–179. DOI: http://dx.doi.org/10.1109/MCOM.2017.1600970

[89] Muhammad Imran, Muhammad Hanif Durad, Farrukh Aslam Khan, and Abdelouahid Derhab. 2019. Toward an optimal solution against denial of service attacks in software defined networks. *Fut. Gener. Comput. Syst.* 92 (2019), 444–453.

[90] Muhammad Azfar Yaqub, Syed Hassan Ahmed, Safdar Hussain Bouk, and Dongkyun Kim. 2016. Information-centric networks (ICN). In *Content-Centric Networks*. Springer, Berlin, 19–33.

[91] Bengt Ahlgren, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Borje Ohlman. 2012. A survey of information-centric networking. *IEEE Commun. Mag.* 50, 7 (2012).

[92] Qing-Yi Zhang, Xing-Wei Wang, Min Huang, Ke-Qin Li, and Sajal K. Das. 2018. Software defined networking meets information centric networking: A survey. *IEEE Access* 6 (2018), 39547–39563.

Received June 2018; revised December 2018; accepted December 2018