Detection approaches for software-defined networking controller-based distributed denial of service attacks: a critical analysis

Padma. I.S [1],

Department of CSE,

Arunachala College of Engineering for Women,

Pooja45luck@gmail.com

J. Caroline Misbha [2],

Assistant professor, Department of CSE,

Arunachala College of Engineering for Women,

Caroline.misbha@gmail.com

ABSTRACT

More complex security risks have emerged as a result of the last ten years' widespread adoption of telecommunications technologies. A novel networking framework known as Software-Defined Networking (SDN) divides the network control plane from the data plane, providing enhanced capabilities to detect and counter security risks. With its adaptable programmable nature, SDN empowers network operators to oversee and modify their networks efficiently, facilitating streamlined network administration. The threat posed by Distributed Denial of Service (DDoS) attacks is one of the main security concerns brought about by the new technology. This paper provides an extensive analysis of the most recent methods for identifying DDoS attacks against SDN controllers. It explains SDN technology in the first place before going into detail about how DDoS attacks affect SDN. Furthermore, every major DDoS detection technique is described in this paper, and they are categorized at a very high level based on the methods or techniques that are employed. Using a variety of metrics defined by the author, the current survey is qualitatively compared to previous surveys. Lastly, this work offers recommendations for further study into DDoS detection methods against SDN controllers.

KEYWORDS - **Software-Defined Networking (SDN), Detection Methods, Distributed Denial of Service (DDoS), Entropy**

INTRODUCTION

The complexity and rigidity of traditional network architecture management are often attributed to the challenges of adapting or controlling the network to meet evolving business needs [1]. The development of computer and communication technologies has been completely transformed by the Internet, and the addition of new technologies, like radio and mobile phones, has given users access to even more services and capabilities [2]. The traditional network's inability to meet new demands for scalability, security, flexibility, dependability, reliability, etc. is a result of the integration of multiple technologies.

As a modern network architecture, software-defined networking (SDN) [3], [4] is gaining popularity due to its simplified implementation and management compared to traditional approaches. SDN separates the network control plane from the data plane, providing a streamlined abstraction layer over the hardware/software infrastructure to meet network management needs effectively. Additionally, direct programming presents another alternative for configuring networking devices [5].

SDN incorporates a logically centralized controller capable of analyzing traffic and deploying new instructions to switches' tables. The controller, also known as the network's brain, is responsible for overseeing all network traffic flows, gathering statistics about incoming packets, and making decisions based on traffic flow analysis.

These characteristics give the SDN the capacity to recognize and respond to alterations or anomalies in the network. Nevertheless, separating the network control plane from the data plane introduces a new avenue for potential attacks that hackers may exploit, potentially resulting in the discovery of novel security vulnerabilities.

The severity of the Distributed Denial of Service (DDoS) attack's impact on the network over the past ten years has made it a genuine threat to SDN. It can cause the entire network to collapse in addition to denying legitimate users access to resources or services. An SDN's primary component is the centralized controller. Attackers find the controller to be a desirable target because any threat to it has the potential to bring down the network [6, 7]. Stated differently, the network's performance, availability, and reliability may be directly impacted by the controller's status as a single point of failure. Nevertheless, protecting the

SDN controller from DDoS attacks is a difficult and resource-intensive task that lessens the controller's ability to manage the network. This is especially more so in light of the various DDoS attacks on SDN that exist [8]. Consequently, any attempt to protect SDN infrastructure from DDoS attacks necessitates a thorough comprehension of SDN characteristics, key elements of network traffic that define DDoS attacks against SDN, and DDoS attack behaviors in SDN. A study on DDoS attacks within SDN revealed numerous distinctive traits and behaviours that could serve as indicators for detecting such attacks.

This survey paper aims to provide the following benefits to the research community: (i) Comprehensive examination of different types of DDoS attacks targeting SDN controllers and the methodologies employed for their detection; (ii) Qualitative comparison of this investigation with other surveys within relevant domains; and (iii) Proposals for future research avenues concerning DDoS detection techniques in SDN environments.

Here's the organization of the remaining sections in the paper. A research background on SDN and SDN controller is given in Section II. A thorough explanation of SDN security concerns, including the effects of DDoS attacks and the most typical kinds of DDoS attacks on the SDN controller, is given in Section III. The findings and a discussion of the qualitative comparison with previous reviews on DDoS attack detection methods against the SDN controller are presented in Section IV. The ways to identify denial-of-service (DDOS) attacks against the SDN controller are covered in Section V. Lastly, Sections VI and VII offer recommendations for further research and a conclusion, respectively.

BACKGROUND

A. Software-Defined Networking

SDN represents a modernized network infrastructure that offers programmability for efficient network administration and boasts greater elasticity and adaptability compared to conventional network architectures in regulating traffic flows. For years, researchers have been searching for ways to protect networks from attacks, but their efforts have been hampered by issues with performance, scalability, reliability, and security [10]–[12]. The research and security communities are excited about SDN technology's emergence because it offers new and creative ways to solve problems [13].

Network attack protection is made possible by creative security solutions made possible by the SDN environment's design, which separates the control plane from the data

plane. By means of a logical and centralized control function that instructs the data plane to forward network traffic, it enables the network to be managed dynamically [14]. However, because of the network's extreme dependence on it, the centralized control feature may end up becoming a liability as it becomes susceptible to single points of failure. As a result, it would seem that DDoS attacks are drawn to the centralized SDN controller because a successful attack could cause the network to deteriorate or even crash. Attackers also take advantage of the data plane switches' limitations, such as their memory size. When a DDoS attack is directed towards an SDN controller, its primary goal is to overwhelm and deplete its resources. This is usually accomplished by flooding the network with spoof IP packets, which causes congestion and eventually causes the network to collapse or deteriorate.

Concurrently, a centralized SDN controller has the capability to operate as a virtualized network, gathering network statistics from incoming packets and identifying devices communicating with the controller, thereby rendering the network highly manageable and flexible. By utilizing its programmability and flexibility, the SDN controller may also help to enhance network performance [15]. Specifically, any network packets lacking corresponding rules in the flow table will be forwarded to the controller due to the segregation of the control plane and the data plane [19]. Put another way, the controller handles two different kinds of objects in order to enable the monitoring of network traffic flows. The first object pertains to network control and contains the switch table's packet forwarding policies. The second component pertains to network surveillance and is symbolized by network status, facilitating the examination of traffic patterns within the network. The characteristics of SDN that make it easier to identify DDoS attacks are compiled in Table 1 [9], [16], and [17].

Feature	Feature Description
Decoupling of control plane and data plane	Facilitates network traffic engineering, maintaining a network policy and security via programmable platform to implement experiments and virtualization environment of Network Functions.
Centralized Logical Controller	Ability to control, monitor, and analyze traffic behaviors from potential security threat. Help create proportionate security rules.
Programmability	Assist control on network behavior via software to simplify operations, enhance dynamic configuration of networks.
Updating of entry flow rules	Allows updating of entry flow rules to match abnormal behavior (attacks) to detect attack traffic entering the network.

TABLE 1.SDN features for DDoS attack detection.

More than half of network traffic flows are anticipated to be handled by SDN in the not too distant future. Additionally, as predicted by a Cisco study [18], a large portion of network operators will adopt SDN, whether fully or partially, to control traffic flows because it will help data centers manage costs and traffic more effectively. Additionally, a number of solutions have been put forth in recent years to address problems with data center security and simplify SDN management, which will boost SDN adoption going forward [19].

As previously indicated, SDN is a cutting-edge networking technology that outperforms its predecessor because of its numerous features, including virtualized logical networks, open programmable interfaces, switch management protocols, third-party network services, logically centralized control, and centralized monitoring units [4, 10, 11, 20, 30]. A comparison of SDN and conventional network architecture is shown in Figure 1. It is challenging to handle every network operation due to the traditional network's complexity and rigidity. The reason behind the rigidity of the conventional network is that control, data, and application all live in the same layer and do not differentiate between each other when handling incoming packets. Yet, SDN accomplishes its objective of simplifying network complexity by partitioning the functional elements that enable centralized management and control of the entire network into three separate layers [12], [13], and [14]. Applications can now manage traffic flow with a centralized visibility and a network-wide view thanks to this separation. Moreover, it offers the ability to virtualize the complete network infrastructure, which will make setting up and maintaining the network even easier.

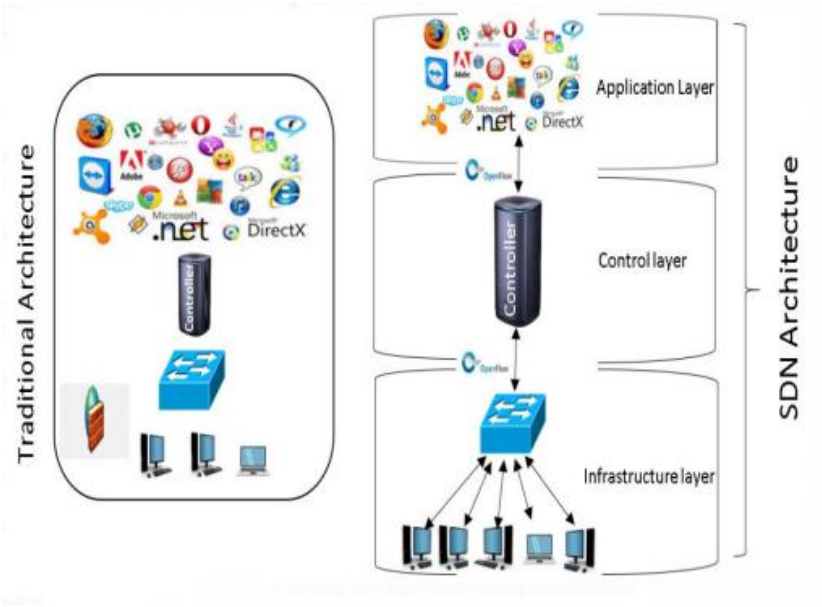

FIGURE 1. SDN Architecture vs. Conventional Network [15]

SDN separates the control feature from data planes, enabling network configurations that may further boost and improve performance and pave the way for security innovations in network operations and architecture [21]. Additionally, it offers immediate network status, allowing for effective flow handling and control procedures while maintaining the control plane's flexibility and intelligence [16].

The need to increase network security makes SDN properties of network operations crucial. Nonetheless, due to the challenges associated with managing and controlling extensive volumes of data, enhancing network performance becomes a daunting task. As illustrated in Figure 2, the advent of SDN offers a considerable opportunity to enhance network performance by empowering a centralized controller to oversee and manage network traffic flows comprehensively. Application programming interfaces (APIs), which are positioned in between the layers to link the networks together, are used by the SDN to manage the entire network [22]. In contrast, the single-package nature of traditional network design makes it challenging to further improve performance and efficiently manage data traffic [15].

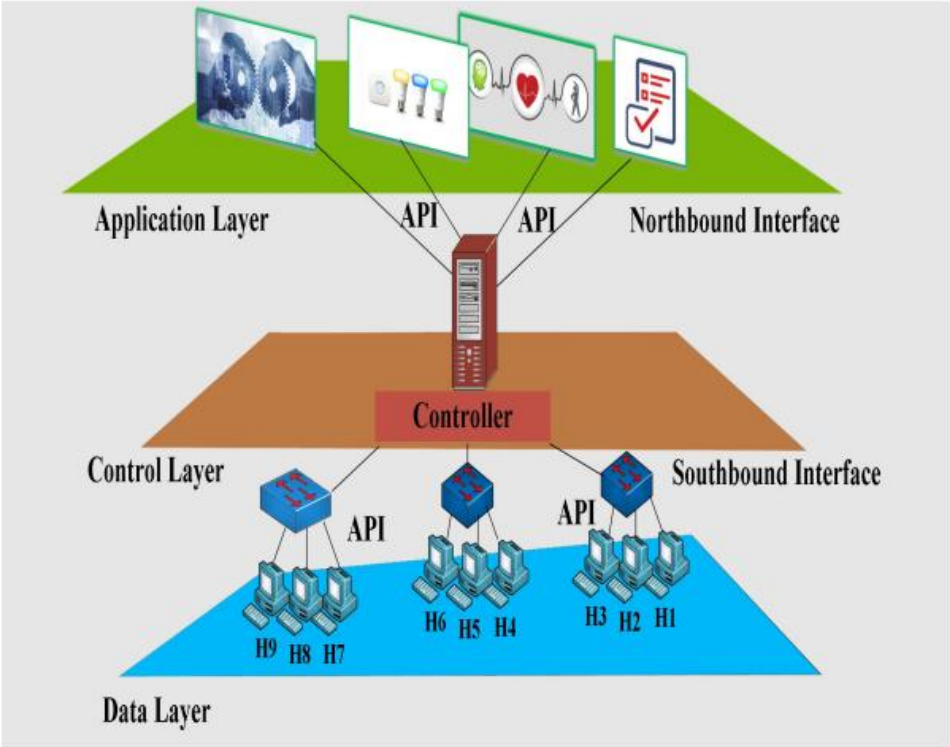

FIGURE 2. An Architecture of Software-Defined Networking with Layers. [23]

Given that the controller bears full responsibility for operating the control plane, the performance of SDN networks relies significantly on its functionality. Essentially, the switch table requests the controller for new or revised flow entries (rules/instructions). The controller and switch devices must communicate often in order to update rules and stay informed about the state of the network. In an SDN environment, a network switch is the first "station" that handles incoming packets. Each packet header feature is examined against the entry table (flow table rules). If no match is found, packets are securely transmitted to the controller for further processing. Consequently, this paper delves into the functions performed by the SDN controller in network security and its involvement in managing and processing incoming packets.

B. Controller for Software-Defined Networking (SDN)

The SDN controller performs a number of important functions in the network, including updating the infrastructure layer's flow table with new instructions, configuring the flow table, and monitoring networking devices through secure connections (switch's table) in order to detect incoming traffic. Before the switch could handle new incoming packets, the flow table had to be updated, and this could only happen because the controller could make new rules and update the switch [24]. Furthermore, by acting as a manager between the

infrastructure, the controller could oversee the entire traffic flow network traffic flow statistics gathered by the controller as a baseline input (information) to an attack detection method is used to determine whether the traffic flow is normal or abnormal. This is done at the application layer and network traffic layer through open API southbound, northbound, and east/westbound interfaces [25]. The controller collects statistical data about network traffic in a number of methods to achieve that [26]. As a result, the controller is crucial to any attempt to strengthen and advance SDN security against malevolent attacks.

In conclusion, by centralizing network control, the controller makes network operations simpler. Each incoming packet undergoes scrutiny based on the switch's flow table, which receives directives and policies from the controller. If a match is discovered, the packet is routed to the appropriate location; if not, it is dropped or routed to the controller for additional processing [27]. To put it another way, the controller decides which packets to forward when they enter the network, functioning as its brain.

III. SDN SECURITY ISSUE

SDN controllers stand out as one of the more dependable security solutions safeguarding networks against attacks. However, as the quantity of network traffic and user base grows as well as the likelihood of possible security problems rising [28]. Nevertheless, as of yet, no reliable method exists for detecting low-rate DDoS attacks with high accuracy and low false positive rate. Moreover, since the controller is the central and most important part of SDN, any issue there could potentially bring down the network as a whole [1], [17], [29], [30]. Consequently, many experts conducting research on security issues and challenges related to SDN architecture have proposed numerous recommendations or viable solutions to address some of these issues. One of the problems is that the SDN controller performs poorly when it is inundated with a large volume of incoming packets or flows, which makes it difficult for the controller to process incoming packets. There is undoubtedly a need for more research to address the question of how to enhance the SDN controller's performance even further [31]. Given that attackers often modify their attack strategies, accurately detecting attacks can be challenging, especially when the attack traffic is crafted to mimic legitimate traffic, making it challenging to distinguish [32]. As mentioned earlier, the objective of the attack is to overwhelm the network device capacity by inundating the SDN switch with a vast quantity of mismatched packets. The switch handles these mismatched packets as new packets and forwards them to the controller [33], [34]. However, SDN also has to contend

with other security threats, including denial of service attacks, malicious software, altered data, and misconfiguration problems [4], [35]–[37].

Security Issue	Description	Solutions
Scalability	Switch-controller latency burden the infrastructure and controller that cause delay in processing new incoming packets [40].	Avoid controller bottleneck by switch table. Put a device between the data plane and the control plane.
Reliability	A single centralized controller could fail under bombardment of packets. Thus, using a single controller is an unreliable method to detect D/DoS attack. [4].	Using multiple controllers to tackle new incoming packets.
Programmability	Easy for attackers to read un-complicated codes that allow modification to attack behavior [40].	Make codes more complex to prevent attackers from changing attack behavior to evade detection.
Dependability	Many gaps in security enable exploitation of SDN vulnerabilities such as limited memory size, and lacked a mechanism to detect abnormal traffic behavior [41].	Work to improve accuracy to detect abnormal behavior and achieve high security.

TABLE 2. Security Concerns with SDN.

Meanwhile, a few proposed approaches [42]–[44] monitor the traffic flow between the controller and switches to access the controller's rules in the switches' tables, enabling the boundary switches of SDN to intercept DDoS attacks. However, these methods rely on static switch rules and thus fail to detect DDoS attacks that regularly modify their attack behaviors. Switch flow tables and static switch rules prove ineffective against persistent attackers who alter attack traffic behavior to resemble legitimate traffic. A robust SDN controller security middleware has been recently proposed as a viable solution for handling erroneous traffic flow [45], [46]. However, relying on switches alone to protect the network from DDoS attacks is impractical, as the majority of SDN switches lack the intelligence required to promptly identify traffic flow fluctuations and detect DDoS attacks in time [34].

A. ATTEMPTS TO DISTRIBUTE A DENIAL OF SERVICE (DDOS)

Due to the extreme resource asymmetry between the victim and the network, DDoS attacks pose a serious threat to network security and stability because they typically originate from multiple sources and are dispersed geographically [47–50].

Attackers typically initiate their attack by scanning the network for a vulnerable host or exploiting security vulnerabilities present in a host. If discovered, the flaw is then used to take over and install harmful software on them.

Attackers use cutting-edge strategies to constantly modify their DDoS attack tactics in an effort to avoid being discovered. To evade detection, attackers often obscure the identity of the compromised host by spoofing the source IP address in their attack packets. DDoS attacks therefore represent a severe risk to the SDN network's quality, particularly if they have an impact on the SDN controller directly or indirectly [51]. DDoS attacks aim to block or restrict the access of authorized users to network resources and services [52].

The majority of DDoS attacks use a variety of attack scenarios or DDoS attack techniques, such as ICMP flooding, TCP flooding, and UDP flooding, in order to evade detection and maximize the likelihood of reaching the intended victim [53, 54]. Table 3 lists the most prevalent DDoS attack types.

Type of DDoS attacks	Principles of Operation	Target
SYN	Transmits many SYN packets to the victim using TCP connection to prevent return of ACK to the victim causing resources to be exhausted [55].	Victim's machine
HTTP Flood	Transmits massive number of requests to web server to overwhelm it and unresponsive to legitimate user request [56].	Server
UDP	Transmits a huge number of packets to random ports of victim causing the machine to look for applications on these ports. Thus, the machine has to send Unreachable Destination packet in response to each incoming packet. When incoming packets increase, the delays also increase [57].	Inaccessible machine
DNS	Transmits spoofed IPs and asks for the response which is more than what the victim when it is directed to it. Thus, they change the source IP address. It causes massive traffic [58].	Victim
ICMP	Transmits a huge number of ICMP pings to the victim to exhaust the victim's resources [59].	Server and Victim

TABLE 3. Typical DDoS Attack Types

The SDN architecture introduces programmability features to the network, potentially enhancing the capabilities of existing intelligent systems such as intrusion detection and prevention systems (IDS and IPS), thereby offering promising advancements in networking security [60].

Despite the fact that SDN properties have been shown to play a significant role in enhancing network security, the SDN controller's security needs are still unmet, leaving the network vulnerable to attacks. [61]

SDN properties have been demonstrated to significantly improve network security; however, the security requirements of the SDN controller remain unfulfilled, making the network susceptible to attacks. [61]

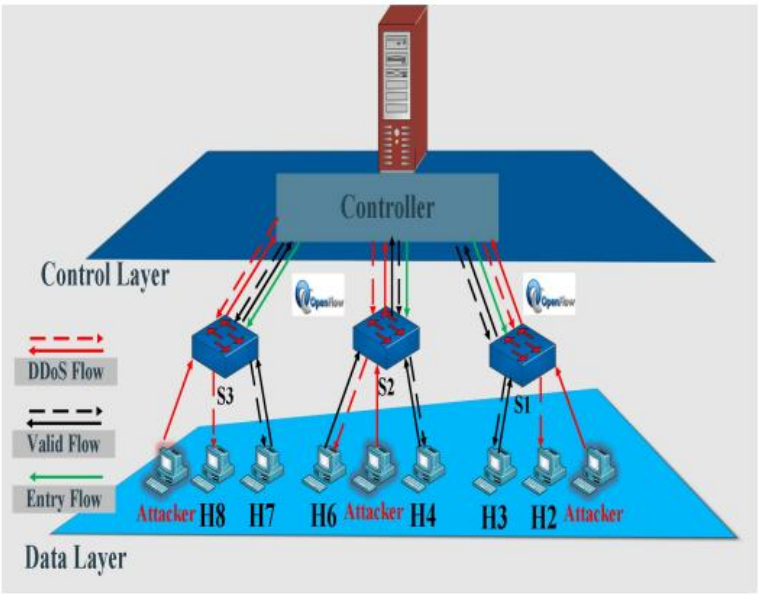

FIGURE 3. SDN Controller DDoS Attacks [64]

DDoS attacks can be categorized according to the layer or protocol they target. TCP, ICMP, or UDP packets are typically used in control layer attacks to deplete the victim's bandwidth. Application layer attacks typically aim to deprive authorised users of services by depleting the server's resources that supply the specific service [63].

Denial-of-service (DDoS) attacks targeting the SDN controller are instigated by flooding it with substantial volumes of network traffic from diverse sources, often by spoofing their IP addresses. Actually, the use of various attack scenarios against the target gives DDoS attacks their ability to completely destroy their victim [32]. A sophisticated DDoS attack mimics the behavior of regular traffic and changes the traffic rate (e.g., low or high) to evade detection, which increases the attack's effectiveness. It also uses less bandwidth and fewer packets.

IV. COMPARING QUALITATIVELY WITH REVIEWS THAT ALREADY EXIST ON HOW TO DETECT DDOS ATTACKS AGAINST SDN CONTROLLERS

Numerous reviews cover the state-of-the-art methods for detecting DDoS attacks against SDN controllers. Consequently, a qualitative comparison is made in this section to highlight how special this review is in relation to others. The metrics listed in Table 4 serve as the foundation for the qualitative comparison. The metrics include: (i) the total number of techniques; (ii) the classification of mechanisms; (iii) the entropy-based detection technique; (iv) the low rate traffic flow detection technique; (v) the time-based detection technique; (vi) the intrusion detection system; and (vii) machine learning techniques. The author defined these metrics after thoroughly reviewing a large number of currently used detection methods. Such a comparison is necessary to comprehend the crucial problems associated with DDoS attacks against SDN in order to identify a better detection method. It might also act as a manual for researchers in the future who work in related fields. This review is compared to three previous reviews that have been published [34], [39], and [45].

Criteria	This work	[34]	[45]	[39]
Number of techniques	15	8	6	9
Mechanisms classified	Yes	Yes	No	No
DETECTION TECHNIQUE				
Entropy-Based Detection Technique	6	2	3	2
Low Rate Traffic Flow Detection Technique	3	1	-	-
Time-based Detection Technique	2	3	-	-
Intrusion detection system	2	-	1	-
Machine Learning Technique	2	-	1	2

TABLE 4. Qualitative comparison with reviews already in existence.

As previously noted, the SDN architecture provides a suitable environment and tools to enhance the management and control of traffic flows through the controller. This capability could potentially address some of the challenges associated with DDoS attack detection encountered in traditional network architectures.

But because the SDN controller is so important to the network, any issue or failure there could weaken or even bring down the entire SDN network. In order to locate, assess, and address incidents before they have a detrimental impact on the network, an effective and high-performing DDoS detection technique is imperative.

To safeguard the SDN network, a number of methods have been put forth to identify DDoS attacks against the SDN controller. Every method is examined in this paper based on its performance, accuracy, detection time, and traffic flow rate. Every technique possesses distinct attributes determined by the standards stated in Table 4. This section outlines the previous research on the detection of DDoS attacks against SDN and provides a summary of the conclusions (results) and limitations for each strategy in Table 5. This work represents the first attempt to categorize some of the DDoS attack detection methods currently in use according to their technique and features, threshold nature, and deployment location within the SDN environment, all of which are listed in Table 6.

Ref.	Findings	Drawbacks
[45]	Detect DDoS attack in its early stages. Low false positive and false negative.	Overload the controller. Unable to deal with large window size (over 50 packets).
[44]	Capable of early detection.	Only handles one type of DDoS attack. Only handle single victim.
[65]	Effectively protect the controller from DDoS attack.	Delay when handling enormous number of incoming packets
[42]	Able to distinguish attack traffic from legitimate traffic. Low false positive.	Only handles low traffic rate and threshold is fixed.
[66]	Traces attack source. Reduces workload on controller in early stage.	Works after controller received traffic flow which leads to flooding of controller by new incoming packets.
[67]	Reduce overload between controller and switches.	Unable to separate legitimate traffic from attack traffic.
[6]	Able to distinguish DDoS attack from flash crowd.	Overload the controller. Delay in DDoS attack detection.
[43]	Reduces controller overload. Quick reaction in detecting DDoS attacks.	Unable to deal with complex traffic flow in the switch.
[46]	Reduced resource consumption. High detection ratio on DDoS attack.	Requires high computing resources and processing power of SDN controller.
[61]	Prompt, versatile and accurate detection of DDoS attack. Limits false positive and false negative rate.	High resource consumption on the controller. Does not care in the temporal characteristics in order to accelerate detection process.
[68]	Reduces congestion of incoming packet at controller.	Consume time to process new network packets.
[69]	- Low false positive rate. - Increase detection accuracy.	Difficult to detect unknown or new type of DDoS attacks. Need time to detect the attack. Overload the controller

TABLE 5. A Synopsis of Current DDoS Attack Detection Methods in SDN.

It is difficult to identify any kind of attack on an SDN controller quickly and accurately. Nonetheless, to stop the network from deteriorating or possibly collapsing completely, it is imperative to identify the attack as soon as possible, even before it reaches

the controller. Various methods have been suggested to identify denial-of-service attacks. Every technique has its own unique features, benefits, and drawbacks. It also employs different approaches and parameters.

V. METHODS FOR IDENTIFYING DDOS ATTACKS AGAINST SDN

Due to the ten years of continuous DDoS attacks on networks, particularly SDN, researchers have developed numerous techniques for identifying DDoS attacks. The literature contains numerous studies on DDoS attack detection methods, including [6], [41], [43], [45], [46], [61], [65]–[67]. Based on the deployment location, DDoS attack detection techniques are categorized into two main groups for this survey, as shown in Table 6.

Ref.	Techniques	Features	Threshold	Deployment location
[45]	Entropy	Dst_IP	Fixed	Controller
[44]	Entropy	Dst_IP	Fixed	Out controller
[65]	TDDAD	Time feature	Number of thresholds	Out controller
[42]	Entropy	Dst_IP	Fixed	Out controller
[67]	Entropy	Dst_IP	Fixed	Controller
[6]	EDDM	Dst_IP	Fixed	Controller
[43]	Statesec	Dst_ IP, dst_ port, src_IP and src_ port	Fixed	Out controller
[46]	SOM	Src_ IP, dst_port	Fixed	Controller
[61]	SORT	Dst_IP address	-	Out controller
[68]	Entropy	Dst_ IP	Fixed	Out controller
[69]	Entropy	Flow duration, src_ IP, packet length, dst_port	Fixed	Controller

TABLE 6. DDoS Attack Detection Methods Using Current Approaches

1) SOURCE-BASED MODEL

To stop a DDoS attack before it starts, source-based tactics are placed close to the attack's origin. A DDoS attack typically overloads the switch table by flooding the network with spoof IP packets until the controller's resources are exhausted and the packet arrival rate is too high to handle. In such a scenario, the centralized SDN controller turns into a single point of failure.

Information distance (ID) was used by the authors in [42] to identify DDoS attacks with the fewest features possible within a predetermined detection window. Furthermore, they employed an entropy-based method that relies on the frequency of incoming packet

destination IP occurrences within a defined window size. Two thresholds were used in their method: information distance (ID) and entropy value. But because the threshold is fixed and the technique only addresses low-rate DDoS attacks, it may result in lower detection accuracy and a higher false positive rate.

Whenever a new flow is established, the controller dispatches fresh instructions to the switch without necessitating the addition of matching rules to the switch's table. However, since all flows lacking matching rules in the switch table must be directed to the controller, this process has the potential to overwhelm the controller. Nevertheless, efforts are underway to enhance the switches' capability to detect DDoS attacks autonomously, without relying on the controller, through a novel technique termed StateSec as proposed in [43]. This approach hinges on monitoring relevant traffic features and employing the entropy method. The authors asserted that their proposed technique boasts high detection accuracy, rapid response times, and the ability to prevent controller overload. However, the switch may need to conduct complex calculations to make decisions regarding DDoS attack detection instead of relying on the controller, potentially leading to delays in attack detection. Additionally, the switch is responsible for collecting statistics for the source IP.

Using the SDN's programmability and broad visibility, a novel detection method was put forth for the early detection and mitigation of TCP SYN flooding entropy approach to ascertain the flow's randomness [44]. The destination IP address and a few chosen TCP flag attributes are used to calculate the entropy. The average attack detection time, average false positive rates, and average detection accuracy rate were used to assess the suggested method. Nonetheless, the controller cannot reliably discern whether it is experiencing a DDoS attack using the entropy-based method, which relies on a single feature extracted from the packet header (such as the source IP or destination IP). As a result, using several features is recommended since it will significantly improve the accuracy of attack detection [61].

Time-Based Detection and Defense Scheme against DDoS (TDDAD) [65] is a method for identifying DDoS attacks by analyzing the temporal characteristics of the attack. Utilizing time features, this technique swiftly and efficiently detects and mitigates DDoS attacks. As incoming packets are automatically routed to the controller for processing, attackers may exploit OpenFlow switches to inundate the controller with a vast volume of packets rather than directly targeting the controller. However, since this method relies on the content feature to identify attacks, it is possible to get around the detection by changing the

malicious packet's content in some way. The five modules that make up TDDAD are port recovery, attack defense, attack detection, feature extraction, and statistics gathering.

The controller is in charge of establishing new rules in accordance with the flow and updating existing rules. However, attackers may exploit the delay in the controller's response time by inundating the SDN controller with numerous requests within this interval to initiate their attack, managing new network packets. To detect DDoS attacks using an entropy-based approach, a novel filtering method could decrease the entropy value by routing any new network packet directly to the security gateway instead of the controller. Subsequently, rules for these new requests are generated for the switch flow table [68]. This method utilizes the protocol, source IP address, and destination IP address as the three features to compute three distinct types of entropies. Nevertheless, processing new packet flows requires time for the detection method.

2) TECHNIQUES BASED ON DESTINATION

Most destination-based detection techniques employ detection and defense mechanisms against attack targets. The SDN controller is regarded as the attack's destination in this survey since it is the target.

[45] took advantage of the controller's advantageous position within the network to detect DDoS attacks in a lightweight and efficient manner. By selecting a fixed threshold, the suggested method used the entropy method to determine the likelihood of packets that are coming in randomly and identify an attack early on. The destination IP address is used to calculate the entropy. A single point of failure scenario could, nevertheless, come about in one of two ways. First, a bottleneck in the switch-controller communication channel caused by excessive traffic prevents legitimate traffic from getting to its destination. Second, the controller's processing capacity is exceeded by the volume of incoming packets that arrive at it. Therefore, in the event that either of the two occurs, the controller's resources will be depleted, preventing legitimate packets from reaching the controller. Consequently, the controller can assess the rate of incoming packets intended for a particular host or subnet using the entropy method.

The introduction of SDN not only increases the programmability and flexibility of network management, but it also makes the system more appealing to hackers. Attackers impose undue stress and strain on the controller by continuously barrage it with attack packets; the controller must process all incoming traffic packets in order to identify and stop

any potential DDoS attack. Academic and industry researchers will continue to be interested in this problem until a dependable and practical way to protect the SDN network from DDoS attacks is discovered.

A novel approach to addressing this problem is the self-organizing mapping (SOM) network, which provides an early warning based on the likelihood of packets occurring during an event [46]. For UDP and TCP traffic flows, this approach is able to reduce resource consumption, false positive rates, and enhance detection ratio; however, it has a high false positive rate for ICMP traffic flows.

To achieve objectives such as identifying the route taken by attack traffic, ensuring prompt responses from the detection module, and addressing limitations associated with the fixed detection loop approach, an alternative method for detecting DDoS attacks was proposed [66]. This method incorporates four modules—attack detection trigger, attack detection, attack traceback, and attack mitigation—to facilitate rapid identification of DDoS attacks and reduce the workload on the controller.

Some researchers have proposed leveraging data collection and analysis from switch tables to detect DDoS attacks targeting SDN controllers. However, this approach adds to the burden of data collection due to the significant volume of data that needs to be transmitted between hosts, especially in large networks. Additionally, frequent communication between hosts may overload the switch-controller communication channel, making it challenging to capture every conversation between switches and the controller. Nevertheless, efforts are underway to manage flow statistics within the switch, reducing the need for extensive data collection from switches [67]. The authors have mitigated overhead from frequent flow collection and enhanced switch intelligence to proactively detect DDoS attacks at the switch using a lightweight, entropy-based technique for DDoS flooding attack detection implemented in OpenFlow. This technique reduces the flow collection load on the controller, resulting in reduced communication between the controller and switches.

A large number of earlier studies focused on SDN defense against DDoS attacks. A number of current studies experience false positives. [6] examined the similarities between the traffic patterns of DDoS attacks and flash crowds. They suggested a controller-based DDoS defense mechanism called entropy-based DDoS mitigation (EDDM). During flash, EDDM works to prevent valid packets from being dropped crowd events and thereby shield the network's legitimate users from denial of service attacks. This lowers the false positive

rate and improves the accuracy of DDoS attack detection. Simultaneously, this mechanism relies only on a single window in order to identify anomalous flows based on the entropy method, which demands quick response times. As a result, this technique might overburden the controller and postpone the DDoS attack detection.

Due to concerns about efficiency, numerous techniques for detecting DDoS attacks against the SDN controller hinge on a solitary packet attribute. However, the DDoS detection operation is severely limited when a single feature is used. A novel strategy based on the joint-entropy method and multiple packet header features was put forth in [69] in order to get around the restrictions. Because information theory is more scalable, less complex, and produces more accurate results, the authors used it in their methodology. Furthermore, by utilizing flow duration, source IP address, packet length, and destination port as features to lower false positive rate and increase detection accuracy, the method could detect both spoofing and non-spoofing DDoS attacks through online. The drawback is that it takes longer to identify an attack.

The Challenge of SDN DDoS Attack Detection

As was previously mentioned, by separating the control plane from the data plane, SDN provides network administrators with simplicity of management and programmability [1]. Despite the commendable efforts of researchers and the security community in detecting DDoS attacks in SDN environments, the incidence of such attacks continues to escalate. Some of these attacks have even taken on new forms and characteristics. The following lists some challenges associated with protecting SDN environments from DDoS attacks.

1. **Statistics**: In order to build their strategies, such as the process for extracting the essential characteristics of DDoS attacks, the majority of attack detection techniques require data collection from infrastructure layer switches packet header in order to identify unusual activity. The increasing frequency of DDoS attacks makes it more difficult and challenging to gather statistical data from traffic flows, particularly in cases where low-rate DDoS attacks are involved. Additionally, methods exist for distributing the data collection duties among several SDN network switches in order to balance the loads of data collection. But this makes it more difficult to get accurate information needed to identify DDoS attacks on SDN networks.

2. Algorithm selection: In an SDN environment, identifying anomalous traffic becomes more challenging due to the diversification of DDoS attack behavior. As a result, in order to

identify DDoS attack behaviors, numerous algorithms have turned to artificial neural networks, Bayesian classification, fuzzy logic, etc. But no single algorithm can handle every possible combination of DDoS attack tactics.

3. Quick reaction: In order to keep the network available, SDN controllers must react to DDoS attacks very quickly. However, a DDoS attack causes the controller to deal with enormous volumes of traffic that could deplete all of its resources, making it difficult for it to reply to requests from users who are using the system in a legitimate manner.

Several methods for identifying DDoS attacks against SDN controllers have been put forth in response to the aforementioned challenges. Nevertheless, there are a number of issues with the current DDoS attack detection techniques, such as the controller's excessive workload from processing a large volume of incoming packets in a brief amount of time, the incapacity to identify low-rate DDoS attacks, and the excessive use of network resources. Furthermore, invalid packets add to the controller's processing load, delaying the detection of attacks. Certain strategies were designed exclusively for low traffic volumes, which led to a high rate of false positives.

VI. UPCOMING STUDIES

As discussed in Section VI, there are still a number of problems with the detection methods currently in use to protect SDN controllers from DDoS attacks. Future studies in this area ought to focus on the following areas:

1. Fixed threshold: Several researchers have proposed detection techniques to safeguard the SDN controller against DDoS attacks by employing a fixed threshold, determined based on a predetermined number of packets received within a specific time frame (e.g., 500 packets per t time). A high false positive rate is still problematic as a result. Therefore, it is necessary to develop and investigate a method for dynamically calculating the threshold. Based on the traffic statistics, researchers could potentially implement dynamic threshold features.

2. Detection of low-rate DDoS attacks: Relying solely on a single packet header feature makes the detection of DDoS attacks with low traffic rates nearly impossible [70]. From the attacker's perspective, numerous packets with falsified source IP addresses are generated and sent to a single network host. Consequently, the targeted host experiences an overwhelming influx of these packets at approximately the same time, leading to resource depletion. The packets seem to originate from multiple sources at a seemingly "normal" rate, making it

challenging for the controller to discern whether the specific traffic constitutes a DDoS attack. As a result, there is a high false positive rate and low detection accuracy. To enhance detection and identification of DDoS attacks with low traffic flow rates, the detection technique should rely on multiple packet header features instead of depending on a single feature.

3. Controller excess: At the controller, some SDN security techniques are implemented. Compounding the problem further, certain approaches require analyzing the complete traffic flow to detect DDoS attacks, which is an exceedingly resource-intensive process. This would impose unnecessary burdens and overhead on the controller. Therefore, the controller's overhead would be decreased by choosing suitable packet features and implementing the detection method somewhere other than the controller.

VII. CONCLUSION

By highlighting the significance of SDN features in managing, monitoring, and programming the network with an SDN controller, this paper presents an overview of the SDN concept.

Furthermore, the controller plays a crucial role in upholding network security against various risks, as elaborated in section II. Section III of this paper delves into the security challenges encountered by the SDN controller, elucidates the impact of DDoS attacks on SDN controllers, and provides further insight into some of the most common types of DDoS attacks. Section IV provides a thorough analysis of current DDoS detection methods and, based on predetermined criteria, compares it with three other current surveys. Moreover, this section provides an in-depth examination of DDoS attack detection techniques in SDN and presents the conclusions and constraints. Furthermore, this paper pioneers the categorization of certain DDoS attack detection methods currently employed, based on their methodology and attributes, their threshold characteristics, and the specific SDN environment where they have been deployed.

This study identifies the shortcomings of a number of DDoS attack detection strategies, which may be resolved by applying a more effective method that lowers the false positive rate and improves detection accuracy. Lastly, by combining different approaches, researchers can take advantage of their advantages or strengths to create a more complete detection strategy for DDoS attacks.

REFERENCES

[1] J. Chen, X. Zheng, and C. Rong, ‘‘Survey on software-defined networking,’’ in Cloud Computing and Big Data. CloudCom-Asia (Lecture Notes in Computer Science), vol. 9106, W. Qiang, X. Zheng, and C. H. Hsu, Eds. Cham, Switzerland: Springer, 2015, doi: 10.1007/978-3-319-28430-9_9.

[2] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. Wolff, ‘‘A brief history of the Internet,’’ SIGCOMM Comput. Commun. Rev., vol. 39, no. 5, pp. 22–31, Oct. 2009.

[3] A. Samson and N. P. Gopalan, ‘‘Software defined networking: Identification of pathways for security threats,’’ in Proc. Int. Conf. Informat. Anal. ICIA, 2016, p. 16.

[4] S. Scott-Hayward, S. Natarajan, and S. Sezer, ‘‘A survey of security in software defined networks,’’ IEEE Commun. Surveys Tuts., vol. 18, no. 1, pp. 623–654, 1st Quart., 2016.

[5] M. A. AL-Adaileh, M. Anbar, Y.-W. Chong, and A. Al-Ani, ‘‘Proposed statistical-based approach for detecting distribute denial of service against the controller of software defined network (SADDCS),’’ in Proc. MATEC Web Conferences, vol. 218, 2018, p. 2012.

[6] Y. Jiang, X. Zhang, Q. Zhou, and Z. Cheng, ‘‘An entropy-based DDoS defense mechanism in software defined networks,’’ in Communications and Networking. ChinaCom (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), vol. 209, Q. Chen, W. Meng, and L. Zhao, Eds. Cham, Switzerland: Springer, 2018, doi: 10.1007/978-3-319-66625-9_17.

[7] C. Bouras, A. Kollia, and A. Papazois, ‘‘SDN & NFV in 5G: Advancements and challenges,’’ in Proc. Innov. Clouds, Internet Netw. (ICIN) 20th Conf., 2017, pp. 107–111.

[8] Tom Bienkowski. (2018). 1.7tbps DDoS Attack Makes History | NETSCOUT. Accessed: Dec. 25, 2018. [Online]. Available: https://www. netscout.com/news/blog/security-17tbps-ddos-attack-makes-history.

[9] T. Jose and J. Kurian, ‘‘Survey on SDN security mechanisms,’’ Int. J. Comput. Appl., vol. 132, no. 14, pp. 32–35, Dec. 2015.

[10] A. Abdelaziz, A. T. Fong, A. Gani, U. Garba, S. Khan, A. Akhunzada, H. Talebian, and K.-K. R. Choo, ‘‘Distributed controller clustering in software defined networks,’’ PLoS ONE, vol. 12, no. 4, pp. 1–19, 2017