When SDN Meets Low-rate Threats: A Survey of Attacks and Countermeasures in Programmable Networks

DAN TANG, College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
RUI DAI, College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
YUDONG YAN, College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
KEQIN LI, Department of Computer Science, State University of New York, New York, United States
WEI LIANG, School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China
ZHENG QIN, College of Computer Science and Electronic Engineering, Hunan University, Changsha, China

Low-rate threats are a class of attack vectors that are disruptive and stealthy, typically crafted for security vulnerabilities. They have been the significant concern for cyber security, impacting both conventional IP-based networks and emerging Software-Defined Networking (SDN). SDN is a revolutionary architecture that separates the control and data planes, offering advantages such as enhanced manageability, flexibility, and network programmability, as well as the ability to introduce new solutions to address security threats. However, its innovative design also poses new vulnerabilities and threats, especially susceptibility to low-rate threats. To this end, this article presents a comprehensive overview of low-rate threats in programmable networks. It explores low-rate threats and countermeasures within the SDN architecture, encompassing the data plane, control plane, control channel, and application plane, together with traditional low-rate threats and countermeasures in SDN. Furthermore, the article offers detailed insight into threats and countermeasures against low-rate attacks exploiting SDN vulnerabilities and low-rate attacks related to the programmable data plane. Additionally, it presents a comparative analysis and discussion of low-rate attacks versus high-volume attacks, along with suggestions for enhancing SDN security. This thorough review aims to assist researchers in developing more resilient and dependable countermeasures against low-rate threats in programmable networks.

CCS Concepts: • **Security and privacy → Network security**; • **General and reference → Surveys and overviews**

This work was supported in part by the National Natural Science Foundation of China (62472153).
Authors' Contact Information: Dan Tang, College of Computer Science and Electronic Engineering, Hunan University, Changsha, China; e-mail: dtang@hnu.edu.cn; Rui Dai (Corresponding author), College of Computer Science and Electronic Engineering, Hunan University, Changsha, China; e-mail: dairui@hnu.edu.cn; Yudong Yan, College of Computer Science and Electronic Engineering, Hunan University, Changsha, China; e-mail: yudongyan@hnu.edu.cn; Keqin Li, Department of Computer Science, State University of New York, New York, NY, USA; e-mail: lik@newpaltz.edu; Wei Liang, School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China; e-mail: wliang@hnust.edu.cn; Zheng Qin (Corresponding author), College of Computer Science and Electronic Engineering, Hunan University, Changsha, China; e-mail: zqin@hnu.edu.cn.

ACM 0360-0300/2024/12-ART103
https://doi.org/10.1145/3704434

Additional Key Words and Phrases: Low-rate threats, programmable data plane, programmable networks, software-defined networking, security vulnerabilities

ACM Reference Format:
Dan Tang, Rui Dai, Yudong Yan, Keqin Li, Wei Liang, and Zheng Qin. 2024. When SDN Meets Low-rate Threats: A Survey of Attacks and Countermeasures in Programmable Networks. *ACM Comput. Surv.* 57, 4, Article 103 (December 2024), 32 pages. https://doi.org/10.1145/3704434

1 Introduction

Modern computer networks typically comprise numerous devices from different network vendors, such as switches, routers, and middleboxes (firewalls, proxies, load balancers, etc.) [59]. These devices operate on closed and specialized software that offers restricted management tools and limited configuration abilities [35]. Network operators typically manage devices using different tools from various vendors, which increases the management's complexity and operation cost. In addition, network operators and researchers are faced with another challenge, referred to as *Internet ossification*. It stems from the extensive deployment and the limited ability of network devices to flexibly expand with new features and capabilities, making it difficult to scale the protocol and performance of the network [72].

Programmable networks and **Software-Defined Networking (SDN)** bring innovative ideas to tackle these problems. Programmable networking is a way to facilitate network evolution that has existed for many years before the advent of SDN, and the concept has regained significant momentum thanks to the SDN [72]. Unlike traditional networks integrating control and operation, SDN decouples control decisions from forwarding hardware, enhancing overall manageability [43]. The application plane offers an interface for developers to deploy various programs. The control plane usually comprises several controllers that carry out distributed cooperation and network management together. The data plane maintains the most attributes of traditional networks, such as routing and forwarding. Controllers can communicate with the data plane device by the control channel such as the well-known OpenFlow protocol [66]. In addition, SDN is also regarded as a powerful helper to do better performance in Cloud [38], BlockChain [58, 115], IoT [42], and VANET [73], among others.

Benefiting from the attributes of SDN, this architecture can solve some security problems that often threaten the traditional network [27]. The programmability of SDN brings many new ideas to the solution of security threats. However, the flexibility of the data plane is still low due to its inability to extend functions by itself. As technology advances, the next generation of SDN featuring the **Programmable Data Plane (PDP)** [16, 35, 59] has been developed. The PDP can customize protocol parsing and packet processing rules without control plane intervention. This advancement in flexibility opens new avenues for research in SDN. Researchers have begun to consider deploying solutions directly on the PDP to take pressure off the control plane [48, 134, 136].

Despite the many advantages of SDN, its novel architecture also brings new vulnerabilities and threats [63, 83]. For example, **Denial of Service (DoS)** attacks and **Distributed Denial of Service (DDoS)** attacks can overwhelm the SDN controllers and switch flow tables [63]. Therefore, it is crucial to develop new security mechanisms to address these security threats in SDN. Among the many threats, low-rate attacks [7, 103] pose a tremendous threat to SDN security. There are numerous surveys on low-rate attacks (e.g., [7, 65, 84, 112, 135, 138]). Table 1 compares these surveys in terms of volumetric low-rate, semantic low-rate, conventional network, SDN network, SDN vulnerability analysis, and defense mechanisms. These previous investigations (e.g., [65, 112, 138]) have mainly focused on the volumetric low-rate attacks (e.g., **Low-rate Denial of Service (LDoS)** attacks [101] and application layer DoS attacks [19]) in conventional networks. Although

Table 1. Comparison with Some Existing Surveys on Low-Rate Attacks

Authors	Volumetric Low Rate	Semantic Low Rate	Conventional Network	SDN Network	SDN Vulnerabilities Analysis	Defense Mechanisms
Zhu et al. [138]	✓[a]	✗	✓	✗	✗	✗
Mathew and Katkar [65]	✓[a]	✗	✓	✗	✗	✓
Wu et al. [135]	✓[a]	✗	✓	✓	✗	✓
Rios et al. [84]	✓[a]	✗	✓	✓	✗	✓
Tripathi and Hubballi [112]	✓[b]	✗	✓	✗	✗	✓
Alashhab et al. [7]	✓[a]	✗	✓	✓	✓	✓
This survey	✓	✓	✓	✓	✓	✓

[a] LDoS attacks only, no investigation of other volumetric low-rate attacks.
[b] Application-layer DoS attacks only, no investigation of other volumetric low-rate attacks.

Wu et al. [135] and Rios et al. [84] surveyed LDoS attacks in SDN, they mainly investigated the literature from the perspective of leveraging SDN features to defend against LDoS attacks. In addition, they focused only on LDoS attacks and did not investigate other low-rate threats. Alashhab et al. [7] investigated low-rate DDoS attacks in SDN, primarily reviewing the literature on the application of machine learning. However, they did not cover other low-rate threats, including volumetric low-rate attacks [21, 91] and semantic low-rate attacks [52, 60], resulting in a lack of comprehensive analysis of low-rate threats in SDN. Note that low-rate threats can be both volumetric and semantic. The volumetric low-rate attack refers to attacks depleting the network resource or overwhelming the network function (LDoS attacks [104], low-rate flow table overflow (LOFT) attacks [21, 22], and slow saturation attacks [91], etc.), whereas the semantic low-rate attack refers to attacks using carefully crafted packets to specific mechanisms, then damage the credibility of target resources thereby causing direct or indirect harm (API abuse [38, 113], malicious flow rule injection [47, 86], and side-channel attacks [60, 94], etc.).

The security issues arising from the special structure of SDN have received much attention. Table 2 compares previous overviews of SDN threats. Most reviews (e.g., [9, 44, 95, 114]) have focused on volume attacks (e.g., DDoS attacks) in SDN, whereas review on low-rate attacks is lacking. Compared with volume attacks, low-rate attacks in SDN have lower rates and are more stealthy but achieve similar or even stronger attack effects. Maleh et al. [63] reviewed the security threats and solutions for each plane of SDN, but they did not analyze the low-rate threats. In addition, some reviews of SDN focus more on discussing the types of attacks and defense mechanisms, lacking analysis from different planes of SDN. Chica et al. [27] analyzed the SDN attack surface, but they did not review SDN security solutions according to the SDN plane. Yoon et al. [127] investigated security issues and defense measures within the SDN architecture but omitted an analysis of the application plane. Abdou et al. [1] only comparatively analyzed the security of the control plane in SDN and traditional networks. Rauf et al. [83] only reviewed application threats in SDN without analyzing threats in other planes. Consequently, a systematic review of low-rate threats in SDN is crucial to address the security challenges and ensure the overall security of SDN.

Compared with previous reviews, this article provides a systematic analysis of the low-rate threats in programmable networks, in which we analyze the characteristics of each SDN plane and its vulnerability to low-rate attacks, comprehensively survey the low-rate threats to SDN, and discuss countermeasures against these low-rate threats. First, we provide a review of the low-rate attacks within the SDN architecture, and also investigate the traditional low-rate attacks that may

Table 2. Comparison with Some Existing Surveys on SDN Threats

Authors	Data Plane	Control Channel	Control Plane	Application Plane	Programmable Data Plane	Analysis of Vulnerabilities Related to Low-Rate Threats	Low-Rate Attacks
Maleh et al. [63]	✓	✓	✓	✓	✗	✗	✗
Chica et al. [27]	✗	✗	✗	✗	✓	✗	✗
Singh and Behal [95]	✓	✗	✓	✓	✗	✗	✗
Kaur et al. [44]	✓	✓	✓	✓	✗	✗	✗
Valdovinos et al. [114]	✓	✗	✓	✓	✗	✗	✗
Ali et al. [9]	✗	✗	✗	✗	✗	✗	✗
Yoon et al. [127]	✓	✓	✓	✗	✗	✗	✗
Abdou et al. [1]	✗	✗	✓	✗	✗	✗	✗
Rauf et al. [83]	✗	✗	✗	✓	✗	✗	✗
This survey	✓	✓	✓	✓	✓	✓	✓

be suffered in SDN. Next, we present a survey of countermeasures to low-rate threats in each SDN plane, as well as countermeasures to traditional low-rate attacks. Then, we offer detailed insight into threats and countermeasures against low-rate attacks exploiting SDN vulnerabilities and low-rate attacks related to the PDP. Finally, this article provides a comparative analysis and discussion on low-rate attacks versus high-volume attacks and gives some suggestions for SDN security. The main contributions are as follows:

— Compared with previous reviews, we provide a systematic review and analysis of low-rate threats and countermeasures in different planes from the perspective of SDN architecture. To our knowledge, this work is the first comprehensive overview of low-rate threats and countermeasures in programmable networks.
— We analyze the underlying causes of SDN vulnerabilities and abstract low-rate attacks as fundamental flaws in SDN components. We list and analyze several low-rate attacks exploiting SDN vulnerabilities and discuss defense mechanisms that seek to mitigate the impact of these attacks by demonstrating vulnerabilities in low-rate attack scenarios.
— We expand the investigation scope of low-rate threats to the next generation of SDN, which further opens up the PDP. On the one hand, we explore the promising opportunities that the PDP creates for defending against low-rate threats and present research on using the PDP to defend against low-rate attacks. On the other hand, we investigate the possible low-rate threats to PDP applications and the countermeasures against them.

The article is structured as follows. Section 2 describes the background for this work. Sections 3 and 4 systematically overview the low-rate attacks and countermeasures in each SDN plane. Section 5 provides an insightful discussion on SDN vulnerabilities related to low-rate threats, low-rate threats and countermeasures related to the PDP, low-rate attacks vs high-volume attacks, and suggestions for securing SDN. Section 6 concludes the article and outlines future directions.

2 Background

2.1 SDN Framework and Security Threats

2.1.1 SDN Framework. SDN brings flexibility, abstraction, programmability, and virtualization to overcome the shortcomings and inconveniences of traditional networking architecture [95]. As shown in Figure 1, SDN typically consists of three distinct planes:

— The *data plane* comprises devices responsible for packet processing and forwarding. Data plane devices leverage the southbound interface to communicate with controllers, such as the well-known OpenFlow protocol [59].

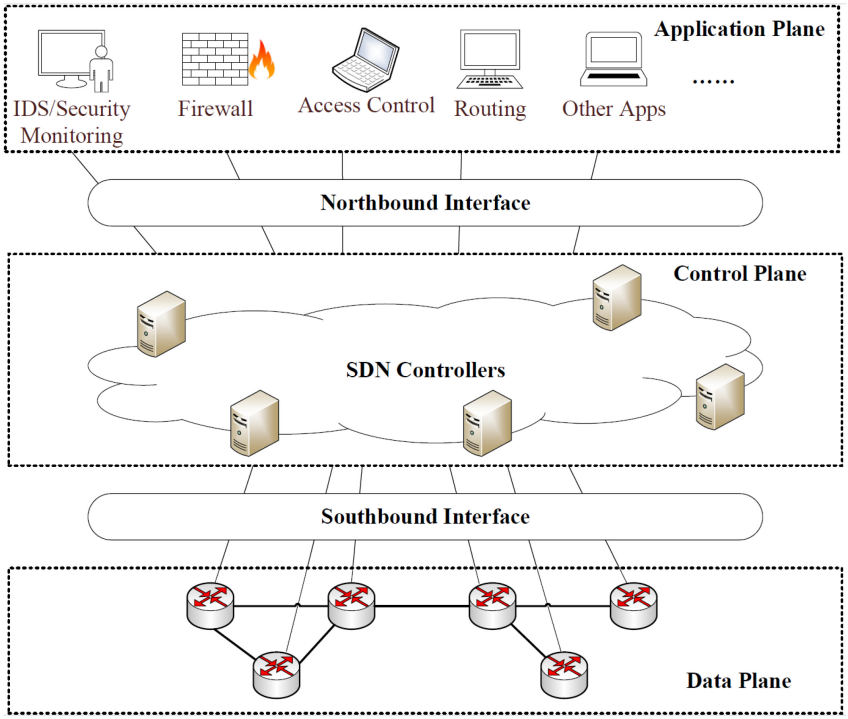

Fig. 1. The SDN architecture.

— The *control plane* comprises multiple controllers that abstract network logic to manage various responsibilities, offering the network flexibility to incorporate new functionalities through programming interfaces [1].
— The *application plane* includes applications that take advantage of SDN-enabled services such as security monitoring, routing, access control, and virtualization provided by the control plane, and the two planes interact via the northbound interface [83].

2.1.2 Security Threats. While the novel SDN designs offer significant benefits, the architecture is also prone to numerous vulnerabilities and may be subject to various security threats [63]:

— *Threats to the data plane*: The interconnected switches are vital components of the data plane, which take charge of packet forwarding and processing. The packet forwarding will be disrupted if the switches are compromised. As shown in Figure 2, there are two main low-rate threats to the data plane, which are LOFT attacks [21, 22] and side-channel attacks [54, 60, 75]. SDN flow tables typically use **Ternary Content Addressable Memory (TCAM)** storage and are crucial for the SDN data plane. Because TCAM capacity is very limited, the SDN flow table is susceptible to resource-consuming attacks (e.g., LOFT attacks). By sending malicious packets, an attacker can constantly occupy the flow table space or even overflow the flow table, leading to a decline in the forwarding performance. In addition, information leakage is also a threat to the SDN data plane. Side-channel attacks can deduce information about the SDN network that is not directly available through specific behaviors exhibited by data plane devices under certain network conditions.
— *Threats to the control plane*: The compromise of controllers can lead to network-wide disruption or damage, as the control plane manages and controls the behavior of data plane devices.

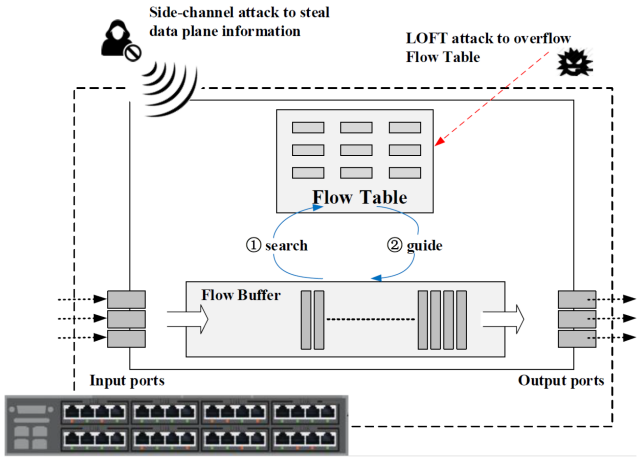

Fig. 2. Threats to the data plane.

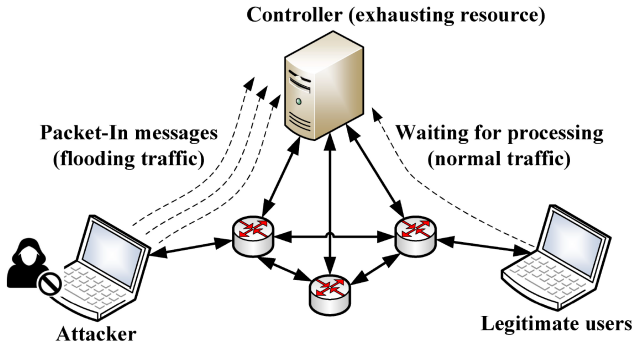

Fig. 3. Threats to the control plane.

As shown in Figure 3, DoS/DDoS attacks can make control plane services and functions unavailable by overwhelming the processing ability or memory resources of the controllers. Attackers can use their hosts or control other distributed zombie hosts to generate a massive flood of traffic to SDN-enabled networks in a short period.

— *Threats to the application plane*: SDN applications can obtain privileges through the controller, such as getting a global view and manipulating the switch infrastructure. Malicious or buggy applications can constitute a serious compromise to network security [83, 95]. SDN applications are usually developed by third parties, and most controllers lack inspection mechanisms for them, thus malicious applications can be deployed to victim controllers to launch well-designed attacks exploiting vulnerabilities in the **Network Operating Systems (NOS)**. As shown in Figure 4, the malicious application can attack other legitimate applications in the application plane or attack the underlying plane via NOS.

2.2 PDP Platform and Benefits against Security Threats

2.2.1 PDP Platform. The initial scenario for SDN insists on dividing the traditional network into two aspects, respectively for aggregated behavior control that provides administrators with flexible programming capacity, and ordinary data forwarding for in-network devices. However, with the emergence of the PDP platform, it is supported that customized programming can be

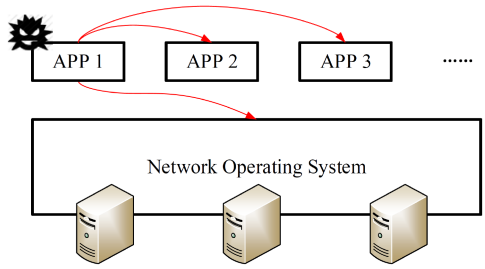

Fig. 4. Threats to the application plane.

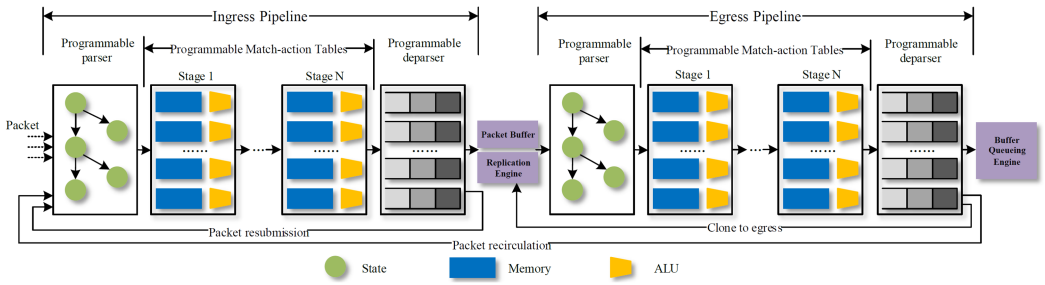

Fig. 5. Illustration of PISA.

fully reflected on in-network devices designed under the **Domain-Specific Architecture (DSA)** for the PDP.

The category of DSA for the PDP concretely refers to processors utilized by devices. On the one hand, the **Protocol Independent Switch Architecture (PISA)** is dedicated to devices integrated with the Tofino ASIC [40], which is designed based on the RMT (Reconfigurable Match-action Table) [17]. As shown in Figure 5, PISA consists of two reconfigurable independent pipelines located on ingress and egress and one built-in engine for the packet replication. Moreover, each pipeline is allocated by a pair of Parser and Deparser. And the gap between them exists abundant MUs (Match-action Units) that can be programmed and distributed in a range of stages. On the other hand, another architecture named *V1Model* is utilized for software-simulated and CPU-based devices represented by Behavioral Model version 2 (bmv2) [25]. This architecture is a truncated version of PISA, in which only one pipeline assists both ingress and egress.

What is more, the DSL (Domain-Specific Language) matched to DSA for the PDP mainly leverages P4 [16]. In the meantime, the corresponding P4Runtime APIs contribute to the implementation of southbound interfaces that bridge controllers and P4-driven devices.

2.2.2 PDP Benefits against Security Threats. The PDP develops the local complex processing ability for in-network devices so that aggregated control logic deployed in SDN controllers can be partially or fully offloaded to in-network [125, 134, 136]. To summarize, the PDP has the following benefits.

First, the PDP platform engaging per-packet processing is beneficial for fine-grained data collection. Since SDN controllers merely leverage polling in-network devices once for sampling flow statistics after each time interval finishes, these flow statistics are hardly complete and continuous for each packet that has traveled through polling objects. Conversely, per-packet processing ensures that the programming capabilities are applied to each incoming packet,

aiming for completeness and continuity in statistics. As a result, some fine-grained features can be extracted by the PDP platform.

Second, it is permitted that some defense-related solutions against security threats can be directly executed with a line rate that reaches the level of terabytes per second throughput by in-network devices. Compared with traditional CPU-based routers, Tofino ASIC switches leverage the PISA, chasing more powerful computational capacity, and achieving higher throughput. In other words, the gap between CPU and PISA under in-network processing is similar to the gap between CPU and GPU under image processing. With this high-powered performance provided by the PDP platform, the solutions can be normally deployed with basic packet forwarding and executed without a noticeable impact on normal line rate. Therefore, the PDP platform presents a more integrated and high-performance runtime environment for executing defense-related solutions at high speed.

In conclusion, the PDP platform goes beyond what the ordinary data plane can manipulate, and advocates placing SDN controllers' complex functions on in-network devices because of the benefits mentioned earlier like fine-grained data collection and executing defense-related solutions with a high line rate.

3 Low-Rate Attacks in Programmable Networks

3.1 Low-Rate Attacks in the SDN Data Plane

The data plane consists of basic packet switching devices including switches and routers. These hardware appliances are only in charge of parsing the arriving packets and matching them with the forwarding rules in the OpenFlow flow table to achieve the forwarding scheduling function for network traffic. However, the limited resources of the data plane make it susceptible to attacks that can degrade service quality or even cause interruptions. Low-rate attacks compromise the forwarding efficiency of the entire network by targeting the hardware devices and network protocols, or by probing network information through low-rate flows.

3.1.1 Low-Rate Flow Table Overflow Attack. Cao et al. [22] proposed the LOFT attack, targeting the TCAM's limited storage capacity and flow table's timeout mechanism. It installs malicious entries into the flow table through transmitting contrived packets at low rates to the destination switch, and occupies flow table space for a long time [74]. LOFT attacks are periodic in nature, sending a small set of packets in each attack cycle to achieve the soft timeout refresh for the installed malicious entries. In addition, the number of packets sent by LOFT attackers gradually increases with each cycle to install new malicious entries. This results in a relatively low average speed for the LOFT attack flows. Eventually, the ever-growing number of malicious entries can lead to flow table overflow, which results in normal flow entries being evicted from the flow table. Furthermore, the attacks can also increase the processing time of subsequent normal packets arriving at the switch, resulting in reduced TCP congestion windows and even a drop in overall throughput.

3.1.2 Side-Channel Attacks. Information leakage is also one of the threats to the SDN data plane. Side-channel attacks infer SDN network information that is not directly available through specific behaviors generated by data plane devices for certain network conditions [27], and may use these network information to launch further attacks against SDN [137]. KYE attacks [26] can obtain various information about the SDN, including the network's detection and defense measures against various attacks, network policies, and network virtualization, through the flow table side channel against individual switches. Leng et al. [54] assessed flow table capacity and usage by sending packets to trigger the addition or deletion of flow entries and observing the change in network

performance. Patwardhan et al. [75] inferred the timeout mechanism of the switch flow table through transmitting a few probe packets and observing the response time.

3.2 Low-Rate Attacks in the SDN Control Plane

The adversary can disrupt the target network due to the critical role of the controller. The heavy workload on the control plane caused by a large volume of requests can make it challenging to efficiently process all the requests. Shin and Gu [91] proposed the slow saturation attack and tested the attack time and bandwidth required for the saturation attack to consume controller resources. In slow saturation attacks, attackers continuously send saturation packets to the switch at a low rate, triggering a significant number of Table-Miss packets that consume control plane resources. The experiment shows that when the sending rate is between 50 and 600 packets per second, the bandwidth occupied by the attack is less than 0.25 Mbps, and the attack time does not exceed 35 seconds.

Ambrosin et al. [10] proposed the buffer saturation attack in which attackers only need to establish several TCP connections to a given host through the target OpenFlow switch. Note that each connection needs to store the state on the switch for transition. As the connections increase, the switch's memory may become saturated, preventing it from effectively providing services for additional connections. Some researchers [3, 4, 100] have studied a slow saturation attack that exploits the loophole in specific mechanisms (TCP control mechanism, keep-alive for HTTP, etc.) to compromise SDN resources. Unlike high-rate attacks, a successful attack can be performed with only a small percentage of traffic, typically around 10% to 20% [100]. Attackers can even generate attack traffic from multiple sources, making the controller busily employ vast Packet-In messages and finally causing the controller to crash.

3.3 Low-Rate Attacks in the SDN Control Channel

The control channel is responsible for the task of information transmission. It is the core of which the SDN brain could monitor and work. In addition to using only the transmission control protocol, its secure channel connection can use the enhanced protection protocol to encrypt messages. TLS/SSL is a well-known protection-based protocol for secure channels in the transport layer [133]. However, despite the protection of protocols, communication channels are still in danger of being attacked. Research shows that because of the importance of the control channel in SDN, attackers also take it as their attack target [119]. This survey classifies attacks against SDN control channels into three types: attacks based on flooding implementation, attacks based on the protocol itself, and attacks based on packet modification.

3.3.1 Attacks Based on Flooding Implementation. The main feature of attacks based on flooding implementation is that the attacker injects massive data packets into the target network, causing the network to be filled with meaningless data packets, and affecting normal information interaction [82, 90].

Link Flooding attack [82] is an attack that causes the original link to lose communication capability by injecting traffic into the target link. Attackers can achieve their goals by injecting massive data packets at one time, or by slowly injecting a quantity of harmful legal flows to preempt the link. Since the control channel is placed in an important position in communicating with the controllers, this attack is undoubtedly extremely damaging.

In addition, the attacker can jointly launch a bypass attack to disrupt the TCP handshake, in which the control traffic amplification attachment is reached, filling the control plane with buffered packets of these malicious copies [20]. This not only exhausts the bandwidth of the control channel but also ties up the controller's limited computing resources. According to Deng et al. [29], an attacker can even perform flooding attacks in the same way as packet injection.

3.3.2 Attacks Based on the Protocol Itself. An adversary typically launches such attacks by targeting the weaknesses of the protocols supported by the control channel. By analyzing and exploiting the vulnerability of protocols, the adversary can use the protocol itself to destroy the SDN control channel and achieve their objectives.

When a switch needs to process new packets and install new flow entries, it typically requests a configuration from the controller. However, with the increased request frequency, the limited TCAM space, controller resources, and channel bandwidth will be pushed to their limits. Based on the basic setup of the OpenFlow protocol, DoS attacks, saturation attacks, and flow table overflow attacks aimed at depleting limited controller resources and TCAM can all bring potential risks to the control channel [18]. Furthermore, low-rate attacks, due to their better concealment, can often make these attacks better covered up and cannot be detected in time, resulting in losses [122, 133]. Taking the slow saturation attack as an example, the attacker initially uses a LOFT attack to overflow the target switch's flow table. Then, the attacker accelerates the packet-sending speed, generating a significant number of control messages and overloading the control channel. The behavior of a slow saturation attack is more stealthy than directly flooding the control channel with a large number of packets [106].

In many SDN builds, the control message is protected by TLS/SSL as the default premise [133]. However, the TLS/SSL protocol itself may have security vulnerabilities, and attackers can secretly install client certificates to use TLS/SSL connections. For the vulnerabilities of TLS/SSL itself, common attacks include the downgrade attack and the **Man-in-the-Middle (MitM)** attack. The downgrade attack is a type of attack on communication protocols or computer systems in which the adversary intentionally causes the system to discard the new and more secure working mode, and instead utilize the old and less secure working mode designed for downward compatibility. Since the OpenFlow protocol came into existence, it has been changed in several versions, and the new version has continuously repaired the vulnerabilities in the old version. Since the old version of the protocol is still not prohibited in SDN, the downgrade attack can still be used by malicious attackers [96].

The MitM attack occurs when an attacker establishes separate connections with both communication endpoints and intercepts and alters the data between them. The attacker can eavesdrop on the communication and inject new content into it [55]. These attacks continue to send malicious packets to the network at a low rate in the early stage to obtain the required configuration information, so as to launch multiple attacks in the later stage [11]. MitM attacks against control channels can be broadly categorized into three scenarios. In the first scenario, MitM attacks are enabled through physical access or network access. While SDN infrastructure greatly facilitates IoT network management, it also introduces security risks due to the extensive number of connected devices. If some devices within the IoT LAN are vulnerable to firmware updating attacks, an external attacker can exploit this by launching a firmware modification to control these devices. Once the client certificates are installed, the attacker disconnects the controller from the gateway and conducts a KCI attack on the control channel, thereby executing an MitM attack [55]. In the second scenario, deficiencies in OpenFlow network topology management can enable MitM attacks. For example, the lack of integrity checking of **Link Layer Discovery Protocol (LLDP)** messages by OpenFlow controllers may lead to attacks on internal link discovery, which allows attackers to perform MitM attacks when false internal links are established [117]. In the third scenario, vulnerabilities in the authentication mechanism can enable MitM attacks. If the authentication mechanisms used by both parties to the communication are flawed or inadequate, an attacker can easily impersonate one of the parties. Downgrade attacks [96] cause systems to abandon the new and more secure working mode, greatly increasing the risk of MitM attacks.

3.3.3 Attacks Based on Packet Modification. The main feature of this type of attack is that the attacker sends the carefully modified and forged data packet to the target link at a low rate to detect sensitive information, then launches other attacks to achieve the purpose.

An attack called *Control Plane Reflection attack* [133] can be launched because the SDN switch has limited processing capacity for downlink messages, especially for two types of messages, Static Query and Flow-Mod, which cannot meet the high demand of control plane applications. This attack typically includes a probing phase and a triggering phase. In the probing phase, despite the protection provided by the TLS/SSL protocol, attackers can obtain the necessary configuration information by constructing timing binding packets and test packets. With the information gathered during the probing phase, an adversary can trigger direct events and indirect events in the data plane to make the control plane generate a substantial number of costly downlink messages to fill the limited control channel. This purposeful triggering of packets via a low-rate probing phase makes the attack more efficient and accurate than randomly generating messages to disrupt the control channel.

The controller acquires the global topology information via LLDP data packets, which necessitates information exchange during this process. This interaction process enables an attacker to alter fields, such as the port in the Packet-Out message. As a result, the controller receives misleading network information, preventing it from acquiring the correct LLDP packet needed to construct a real network topology [96]. An attacker can even directly modify the LLDP packet, further confusing the network construction for the controller.

3.4 Low-Rate Attacks in the SDN Application Plane

SDN applications can obtain privileges through the controller, such as getting a global view and manipulating the switch infrastructure. Malicious or buggy applications can constitute a serious compromise to network security [83, 95]. SDN applications are usually developed by third parties, and most controllers lack inspection mechanisms for them, thus malicious applications can be deployed to victim controllers to launch well-designed attacks exploiting vulnerabilities in NOS. These attacks are crafted to specific vulnerabilities and typically exhibit low rate and stealth. We have investigated low-rate attacks in the SDN application plane and roughly classified them into the following types: illegal function calling, API abuse, and malicious flow rule injection.

3.4.1 Illegal Function Calling. The malicious application can call the illegal function through the northbound interface to disrupt the execution of other legitimate applications and degrade network performance. For example, Lee et al. [53] revealed cases of attacks by malicious applications exploiting illegal function calling against three well-known controllers, including Floodlight, ONOS, and OpenDaylight. In the Floodlight case, the malicious application manipulates the Packet-In listener to erase the message's payload, causing the next legitimate application to throw an exception. In the ONOS attack case, the malicious application manipulates the target application's property through the configuration manager to degrade the network's performance. In the OpenDaylight attack case, the malicious application blocks the services the target application uses by accessing OpenDaylight's core service management, thus compromising the target application's ability to provide normal functionality to the network.

3.4.2 API Abuse. Due to the poor security of the northbound interface, malicious applications can launch crafted attacks with API abuse. For example, Tseng et al. [113] identified several scenarios in which APIs are abused to manipulate the flow rules in switches—for example, a malicious application can tamper with the flow rule installed by other legitimate applications with *UPDATE permission* and install higher priority flow rules with *ADD permission*, thus causing network traffic to be guided by the rules manipulated by that malicious application.

3.4.3 Malicious Flow Rule Injection. SDN's forwarding decisions are flow based, with switches forwarding packets in the network based on the rules in the flow table. Malicious applications can use rootkit techniques [85] to insert harmful rules to cause interference, such as blocking legitimate services and obfuscating other legitimate applications.

3.5 Traditional Low-Rate Attacks in SDN

Traditional low-rate attacks [105, 109, 130] exploit vulnerabilities in the network resource management defined by network protocols. These attacks legally occupy network resources, gradually exhausting them and thereby affecting network functionality. They are divided into two categories: TCP **Quality of Service (QoS)** attacks and Slow Request DoS attacks.

3.5.1 TCP QoS Attacks. These attacks periodically exhaust network bandwidth resources by sending attack traffic at a low average rate, leading to network congestion. As a result, the QoS of TCP connections continuously degrade due to the TCP congestion control mechanism. Kuzmanovic and Knightly [51] proposed specific TCP QoS attacks named *shrew attacks*. These attacks periodically send high transient intensity flow to trigger the transmission timeouts, which causes TCP senders to enter the slow start state frequently. Luo and Chang [61] proposed pulsing DoS attacks against TCP QoS. These attacks trigger transmission timeouts, and another congestion event called *duplicated ACKs*, which cause TCP senders to endlessly enter the fast recovery state. To sum up, these attacks finally reduce the TCP sender's congestion window size, decreasing the TCP QoS.

3.5.2 Slow Request DoS Attacks. These attacks craft packets with specific semantics or contents to gradually occupy the resources of targets until they are exhausted. These attacks target protocols leveraged by the *request-response* mode, including TCP (Three-way Handshake), HTTP, DNS, and SMTP. They exploit this request-response mode, send a series of crafted malicious requests, and increase the delay between these requests and corresponding responses. As a result, the resource in the server is occupied for a long time, and there is no spare resource for benign requests. Slow HTTP request DoS attacks are the most common in this category. There are some specific attacks with different launching approaches [19].

Slowloris attacks continuously send semi-connection requests, which only have a request header but no request body, so these requests are incomplete. Furthermore, the target HTTP server holds these semi-connections and waits for entire requests which are never received.

R-U-DEAD-YET (RUDY) attacks exploit the feature of the HTTP POST request, which can divide the request body into plenty of small blocks after establishing connections [70]. These blocks are transmitted slowly to increase time consumption for complete transmission. So these connections are held by the HTTP server and occupy resources.

Slow read attacks exploit the phenomenon that TCP receivers submit data to HTTP-based applications after receiving buffer is saturated or complete receiving data. So these attacks slowly send TCP segments to extend the wait time of the HTTP server, occupying its resources.

4 Countermeasures for Low-Rate Attacks in Programmable Networks

4.1 Countermeasures for Low-Rate Attacks in the SDN Data Plane

4.1.1 Countermeasures for Low-Rate Flow Table Overflow Attack. LOFT attacks use low-rate flows to deplete the flow table of switches in the data plane, resulting in DoS. To mitigate this attack, researchers have proposed three types of mitigation methods: active defense, passive defense, and rule management.

The *active defense* methods aim to identify or detect potential attack patterns in advance, then proactively delete malicious rules or block the sources of the attack. Such methods can be regarded

as the early detection for flow table overflow attacks, and the attack mitigation can be executed once the malicious rules are detected, which usually has a more sufficient mitigation effect than the passive defense approaches. However, these methods usually require real-time processing or management of plenty of rule-related flow table data and network status, even including the rule classification, which may result in higher mitigation time delay or extra system overhead.

The *passive defense* usually does not actively delete malicious rules or block such attacks from the source, but schedules or evicts the rules when a SDN switch is overflowed. Such methods can use a Table-Full message in OpenFlow protocol as a signal to initiate an attack mitigation action. Passive defense methods are simple to understand, implement, and deploy since few attack detection metrics or indicators are required to implement such approaches. However, massive malicious rules can enter the flow tables under passive defense, since the mitigation actions are not executed until the flow table overflows, and the attack mitigation effect may not be sufficient and timely.

In addition, another type of approach can focus on the flow table usage issues, and we call it the *rule management* method. These methods mainly aim at ensuring the flow table availability in large-scale topology or high-traffic scenarios. As pointed out in other works [13, 48], most network flows are *mice flow* (about 99% according to Kim et al. [48] and 80% according to Benson et al. [13]), whereas the *elephant flow* that makes up most of the network traffic can only take a small proportion (about 1% according to Kim et al. [48] and 20% according to Benson et al. [13]). Therefore, when a large number of forwarding rules flood into the SDN switch with limited TCAM, the flow with different features should also have different priorities. Such rule management methods can be assisted by the improved eviction algorithms or machine learning technologies to implement. The advantage of these methods is that they can effectively guarantee the schedulability and availability of flow table resources. However, these methods usually do not consider the attack situation. If the attack rate is high, the normal-network-state-aimed strategies may not be able to handle it.

To mitigate flow table overflow attacks, researchers have proposed some approaches based on the three types of methods listed previously. The summary and analysis for the related methods are listed in Table 3.

4.1.2 Countermeasures for Side-Channel Attacks. Attackers often infer SDN configuration information by sending specific packets and observing their response times. Therefore, mitigating side-channel attacks also involves optimizing the packet delay in the SDN or preventing attackers from obtaining the real response time.

The external security solution *Netkasi* [94] prevents information leakage by generating random response times for packets. To mitigate the proposed Flow Reconnaissance attack, Liu et al. [60] present three approaches. The first approach artificially adds a delay for the initial packets of each flow so that the attacker cannot infer whether the rule has been installed or not. The second approach requires the controller to actively install the rule in the switch's flow table. However, the first two approaches can affect the network's operational efficiency. The third way is to aggregate or split the rules in the flow table, making it difficult for the attacker to infer information. Conti et al. [26] proposed a solution called *traffic obfuscation* to stop KYE attacks, which prevents attackers from recognizing the flow rules they have installed by modifying information such as the identifiers and ports of the rules during transmission.

4.2 Countermeasures for Low-Rate Attacks in the SDN Control Plane

In this section, we review the countermeasures for low-rate attacks in the SDN control plane, including detection and mitigation. We have classified the detection and mitigation methods, as shown in Table 4.

Table 3. Summary of Countermeasures for Low-Rate Flow Table Overflow Attacks

Authors	Year	Method Type	Method Name	Description	Limitations
Yuan et al. [129]	2016	Passive Defense	Peer Support	Consider all switches in the SDN together as a single entity and schedule attack flows to other switches in the SDN with a free flow tablespace if necessary.	The approach allows attack traffic to enter the SDN, and once all peer switches are overflowed, it can lead to even more severe consequences.
Pascoal et al. [74]	2017	Passive Defense	SIFT	Randomly replace an existing rule to install a new rule when receiving Table-Full. The longer the attack lasts, the higher the probability of the new rule belonging to attacks and the lower the replacement probability.	An insufficient mitigation effect and high system overhead, and the control plane needs to store the information of all rules.
Xu et al. [124]	2017	Active Defense	Token Bucket	Consider a complex attack model, where the overflow attack target is the middle hop switch. Extract entropy features for attack detection, and use the bucket token to limit the attack rate.	The extracted features are not typical when traffic changes greatly. The token bucket does not prevent the attack flow from entering SDN.
Yang et al. [126]	2019	Rule Management	STEREOS	Simulate the traffic arrival, construct feature vectors for each flow, perform rule prediction and classification, and determine the rule that should be replaced in the current flow table by machine learning.	The complexity of updating features for each rule is high, and the proposed features are difficult to obtain from the actual flow table.
Phan et al. [78]	2020	Active Defense	DeepGuard	Use deep learning to predict the QoS and the switch flow table status; the forwarding granularity is dynamically adjusted to match only the MAC address when the flow table overflows.	The method performance and complexity are not ideal, and the mitigation is at the cost of forwarding granularity loss.
Yu et al. [128]	2020	Active Defense	/	Infer the attack existence and locate attack traffic based on adversary cache inference technology.	The proposed method is applicable to the specific network environment, and the general performance is not ideal.
Isyaku et al. [41]	2020	Rule Management	IHTA	Dynamically adjust idle timeout and hard timeout for traffic based on inter-group arrival time.	It is difficult to obtain accurate group counts, which affects the effectiveness of the flow eviction algorithm.
Phan et al. [77]	2020	Rule Management	Deepmatch	Propose a deep dueling neural network algorithm that provides the suitable traffic grain size in an adaptive manner and realizes the best flow rule matching strategy.	This method does not consider the priority of flow entry eviction.
Soylu et al. [99]	2021	Passive Defense	NFV-GUARD	Mitigate table overflow attacks through the use of fine-grained entries that are mechanically spread across the virtual fabric.	The mitigation method relies on known blacklists.
Xie et al. [121]	2021	Active Defense	SAIA	The mechanism on the basis of statistical diagnosis of small and imported flows.	The overflow prediction module of this method is only suitable for simple network situations.
Priyanka et al. [80]	2021	Active Defense	CEOF	Evict the extravagant entries using Hierarchical Agglomerative Clustering and use the Pareto Optimizer to realize entries optimization of each cluster.	This method may have an impact on legitimate traffic in the face of higher-rate attacks.
Kong et al. [49]	2022	Active Defense	TableGuard	Use the number of active flow rules as a metric for detection, and apply the Z-score method that can assist in filtering malicious flows.	Identification of individual attack ports relies on established thresholds.
Cao et al. [21]	2022	Active Defense	LOFTGuard	Dynamically migrate the flow rules by creating a cache buffer, and identify the attack flow to prevent its installation.	When the packet size is similar to benign packets, the identification of malicious flows will be reduced.
Tang et al. [108]	2023	Active Defense	SFTO-Guard	Detect attacks based on statistical analysis of flow tables, and implement flow entry eviction based on flow entry ordering and adaptive calculation of proportion.	This method is only applicable to slow flow table overflow attacks that attack the timeout mechanism of flow entries.

4.2.1 Countermeasures on the Attack Detection. We divide detection methods into *Machine learning-based methods* and statistics-based methods.

Machine learning-based methods typically detect slow saturation attacks by analyzing network traffic characteristics and employing machine learning techniques [45, 100]. Machine learning-based methods have shown good detection efficiency with an accuracy of more than 90%, which

Table 4. Summary of Countermeasures for Slow Saturation Attacks

Authors	Year	Detection Method Type	Mitigation Method Type	Controller	Advantages	Limitations
Shin et al. [93]	2013	/	Proxy based	POX	Lightweight and effective mitigation of SYN flooding attacks based on IP spoofing.	The scalability is weak, and it mainly improved the recovery ability of TCP SYN flooding attacks.
Mousavi and St-Hilaire [68]	2015	Statistics based	/	POX	An early work on DDoS attack detection in SDN controllers.	The minimum attack rate discussed is not very low.
Ambrosin et al. [10]	2016	/	Proxy based	POX	While mitigating SYN-based flooding saturation attacks, it also protects buffer vulnerabilities.	For other protocols, such as UDP and ICMP flooding attacks, its mitigation capability may not be as effective.
Gkountis et al. [33]	2017	Statistics based	Flow based	POX	Lightweight, simple, effective, and does not modify the controller.	Decision-making efficiency needs to be improved.
Sahoo et al. [88]	2018	Statistics based	Flow based	POX	Good detection effects.	The threshold is set to a fixed value and cannot be adjusted according to the network states.
Li et al. [57]	2019	Statistics based	/	Floodlight	Lightweight and scalable.	Lack of discussion on real time.
Khamaiseh et al. [45]	2019	Machine learning	/	Floodlight	It can detect unknown types of saturation attacks in SDN.	Detection performance will be affected by SDN environment settings.
Agrawal and Tapaswi [3]	2021	Statistics based	Flow based	OpenDayLight	IP tracing is introduced in the mitigation module, and the overall algorithm is real time.	The scalability is relatively weak.
Chen et al. [23]	2021	Statistics based	Proxy based	/	High resilience to strong attacks and low loss of service quality to legitimate traffic.	The effect of attack detection and mitigation for low-rate attack needs to be improved.
Aladaileh et al. [6]	2022	Statistics based	/	POX	It can adapt to changes in the attack traffic rate.	Lack of discussion on real time.
Ran et al. [81]	2022	Statistics based	Flow based	RYU	It has good detection and mitigation effects on both slow- and high-speed attacks.	Lack of discussion on real time.
Ahalawat et al. [4]	2022	Statistics based	Flow based	RYU	Realizes early detection and mitigation of LDDoS attacks.	Lack of discussion on overhead.
Sudar and Deepalakshmi [100]	2022	Machine learning	Flow based	POX	Good detection effects.	Lack of discussion on real time and overhead.
Ali et al. [8]	2023	Machine learning	/	/	High detection accuracy.	Lack of relevant assessment of online detection and mitigation.
Aladaileh et al. [5]	2023	Statistics based	/	POX	Both high-rate and low-rate attacks can be detected.	The detection effect of low-rate attacks is not good.

is promising, but these methods also have certain vulnerabilities. Khamaiseh et al. [46] studied the robustness of machine learning based methods to adversarial attacks in SDN. They proposed a countermeasure test that can generate countermeasure attacks by disturbing different traffic characteristics and avoiding the detection of several saturation attacks. According to their experimental results, adversarial attacks cause a significant deterioration in the detection performance of four saturation attacks (TCP-SYN, UDP, TCP-SARFU, and ICMP) by more than 90%.

Statistics-based methods usually use probability statistics, information measurement, threshold, and other mathematical statistics methods. Mousavi and St-Hilaire [68] first used entropy to detect DDoS attacks on SDN controllers. It realizes the early detection of DDoS attacks at different

rates according to the entropy of the target's attack IP address. After that, more and more information entropy-based methods [3, 5] are used in the detection of slow saturation attacks. Some researchers [4, 88] also adopt different information distance metrics of attack traffic on the controller to distinguish slow saturation attacks from normal traffic. Some detection schemes [6, 33] judged whether packets are malicious based on a set of efficient rules. Aladaileh et al. [6] proposed a rule-based mechanism that employs statistical analysis of incoming traffic, relying on the Rényi joint entropy to detect potential threats. Gkountis et al. [33] introduced two parameters, packet average and byte average, to count rule information and detect DDoS attacks. Besides, Chen et al. [23] proposed SDNshield, which starts the defense by setting the threshold of the Packet-In arrival rate and using CLP (Conditional Legality Probability) measured by Bayesian theory as the core measure to evaluate the legitimacy of the flows. Dong et al. [31] classified the flow events related to the interface, then used the Sequential Probability Ratio Test to make decisions to locate the damaged interface connected with malicious attackers or hijacked zombies.

4.2.2 Countermeasures on the Attack Mitigation. The mitigation strategies for saturation attacks mainly include flow-based methods and proxy-based methods.

Flow-based methods usually use the action field of the flow entries to handle malicious flows. Many flow-based mitigation methods [33, 76, 81, 88, 100] directly drop malicious flows by distinguishing between legitimate flows and malicious flows, thus blocking the attack source. Some researchers [3] introduced the IP tracing to directly block the traffic from malicious sources, reducing the impact of slow attacks.

Proxy-based methods typically use network proxy to detect and mitigate slow saturation attacks. Shin et al. [93] proposed Avant-Guard to address control plane saturation attacks. Avant-Guard uses connection migration, which involves using stateless TCP to implement SYN cookies and only reporting the complete TCP flow to the control plane. This approach effectively reduces the risk of link saturation. However, connection migration has a downside, as it can create vulnerabilities that can potentially shut down switches. The proxy must store state information such as timestamps, sequence numbers, source IP, and port for the duration of the entire connection, which can be exploited by attackers to disable switches. Recognizing this vulnerability, Ambrosin et al. [10] proposed LineSwitch, which includes probabilistic proxying and network traffic blacklisting to effectively mitigate saturation attacks based on SYN flooding and buffer saturation attacks. Chen et al. [23] used TCV (TCP Connection Verification) in the mitigation phase, which allows only TCP flows capable of completing three handshakes to pass through, as forged TCP flows cannot complete three handshakes.

4.3 Countermeasures for Low-Rate Attacks in the SDN Control Channel

The detection and mitigation of attacks against control channels in SDN are often not performed independently but are typically accompanied by monitoring the data and control planes. The main threat to the control channel comes from its own limited link bandwidth. Both attacks against the data and control planes can also achieve the purpose of destroying the control channel to a certain extent. Therefore, attacks on control channels are often achieved with overflow, saturation, and DoS at other levels. However, the impact of protocol-based vulnerabilities and attacks caused by forged packets on the control channel can also be attributed to the malicious saturation of the link. Because some protocols are not only designed for SDN but also shared with traditional networks and other architectures, this section only discusses some ideas and does not discuss the detection and improvement of protocols too much.

4.3.1 Detection and Mitigation Strategies for the Attacks Based on Flooding Implementation. The monitoring for control and data planes is the main deployment level for detecting and mitigating

low-rate attacks based on flooding implementation. For the detection part, machine learning and deep learning are commonly used methods for classification [82]. Commonly used features include bandwidth, packets, bytes, nodes, flow rules, ports, and processed features based on them [32]. Some scholars have also used network state information (e.g., TCP connection-related information) as well as packet payload and header field information [93]. Some scholars also set up a table containing DPID, MAC, and port number to match the legitimate flow with the attack flow for detection [29].

For the mitigation strategy, the general idea is to identify the flow with machine learning and deep learning techniques, then prevent the inflow of attack traffic and discard the attack flow [29, 82]. One approach is to integrate defense modules into the controller or data plane, filtering attack traffic and safeguarding legitimate packets [90]. Transferring the flow to the neighbor switch is also a method to alleviate flooding [93], but there is always a bottleneck in the control channel—that is, its limited bandwidth. In SDN, the design of control channel itself is also a challenge. When the interaction channel between the switch and the controller occupies the same link with a higher probability, the attacker can take the opportunity to launch a more effective attack. Therefore, it is necessary to consider the placement of the controller position, the matching between the switch and the controller, and the selection of the appropriate control channel that can be routed to further protect the control channel [62].

4.3.2 Mitigation Strategies for the Attacks Based on the Protocol Itself. There are several strategies for mitigating attacks based on the protocol itself. One approach is to strengthen the OpenFlow protocol, whereas another is to harden the controller.

For approaches to strengthen the OpenFlow protocol, in 2012, Dacosta et al. [28] introduced DVCert (Certificate Direct Validation) as a method for certificate validation without the need for a third party. DVCert enables domains to securely verify their certificates by utilizing pre-existing user authentication credentials instead of relying on an external authority. It not only strengthens server authentication with a robust cryptographic structure but also addresses certain limitations of third-party solutions, such as extensive deployment and operational costs, complex trust models, privacy risks, and interference with captive portals. Li et al. [55] proposed a lightweight countermeasure to detect MitM attacks using Bloom filters in 2017. The system can be used as a complement to various cryptographic technologies (e.g., TLS) to protect OpenFlow channels from MitM attacks. In addition, Agborubere and Sanchez-Valazquez [2] proposed to protect OpenFlow traffic in SDN by summarizing TLS security vulnerabilities and recommending ways to improve TLS security to secure OpenFlow traffic. They provide TLS extensions to mitigate attacks against TLS, such as MitM attacks.

For methods to harden the controller, Shin et al. [92] proposed FRESCO, which uses a modular and composable design architecture with 16 available submodules and callable APIs. In 2015, Porras et al. [79] proposed SE-Floodlight with an SEK (Secure Execution Kernel) layer, adding an SEK as an extension of the controller, and its functionality can also be directly applied to other OpenFlow controllers.

4.3.3 Mitigation Strategies for the Attacks Based on Packet Modification. Several mitigation strategies against attacks based on packet modification are to detect and filter forged and spoofed packets or messages. Zhang et al. [133] proposed the SWGuard method to detect anomalies in downlink messages, in which it prioritizes these messages according to the new monitoring granularity to mitigate them. Specifically, SWGuard introduced the behavior monitor module, which gathers downlink message events and utilizes a new abstraction called the *host-application pair* to identify anomalies in downlink messages. In 2015, Dhawan et al. [30] proposed a method for inspecting packets that learns network behavior based on control information and constructs flow

graphs for each observed stream to identify anomalies. In addition, Zhang et al. [132] proposed an integrated IP spoofing verification solution called *ISASA*, which can achieve IP prefix level verification granularity with minimal SDN device deployment.

4.4 Countermeasures for Low-Rate Attacks in the SDN Application Plane

As discussed in Section 3.4, low-rate attacks against SDN application plane vulnerabilities can constitute a potential threat to network security. In the following, we discuss various efforts to diminish the threat of low-rate attacks in the SDN application plane, thus providing help and guidance to researchers in mitigating security threats in the SDN network.

4.4.1 Countermeasures for Illegal Function Calling. Lee et al. [53] described the defense against malicious applications in two aspects: permission checking and static/dynamic analysis. In the *permission checking* mechanism, similar to Android applications, each SDN application sets up a permission file to specify the required permission set. The permission checking layer is designed to ensure that the application does not perform any permissions it is not authorized to use. In the *static/dynamic analysis* mechanism, the application is analyzed statically or dynamically before the network administrator downloads and executes the SDN application. The main problem faced by static/dynamic analysis mechanisms is the lack of information about the malicious application's behavior. In addition, Mansour and Chasaki [64] proposed to monitor the system calls of SDN applications periodically and check them with the baseline/benchmark to detect anomalous activities and malicious applications.

4.4.2 Countermeasures for API Abuse. Since static permission control cannot detect API abuse, Tseng et al. [113] proposed Controller DAC (the SDN Controller Dynamic Access Control System) to protect the SDN controller against malicious applications. The Controller DAC has a latency of less than 0.5% performance overhead and can effectively protect API requests. However, the assignment of application roles is done manually, which could be improved. Hu et al. [38] proposed the SEAPP mechanism, including a permission detection engine and a registration authorization engine to support rapid deployment and reconfiguration at runtime. However, insufficient entries in the sensitive API list may limit the accuracy of SEAPP.

4.4.3 Countermeasures for Malicious Flow Rule Injection. Khurshid et al. [47] proposed Veriflow to check the invariance validity of each rule as it is inserted, which is a layer deployed between the controller and the network device. Veriflow can detect faulty rules and optionally prevent them from causing abnormal network behavior. Röpke and Holz [86] proposed an approach that can detect and prevent hidden network manipulation. The approach detects hidden malicious applications by monitoring OpenFlow messages that change the network state, whereas malicious applications that do not directly affect the state of the network via OpenFlow messages are likely to go undetected.

4.5 Countermeasures for Traditional Low-Rate Attacks in the SDN

Owing to the SDN's software programming capacity, there are various methods, including analysis algorithms focused on the time and frequency domain, encryption algorithms, and machine learning related algorithms, which can be leveraged by online efficient detection and mitigation of traditional low-rate attacks. Moreover, APIs encapsulated by SDN southbound protocols address the difficulty of collecting real-time statistics of the flow to analyze. Consequently, SDN provides both simple and efficient solutions against traditional low-rate attacks.

Table 5 illustrates the work against traditional TCP QoS attacks based on SDN. Specifically, some research efforts (e.g., [4, 102]) leverage time-domain analysis algorithms covered by the

Table 5. Summary of Countermeasures for TCP QoS Attacks

Authors	Year	Feature Analysis Method	Decision-Making Method	Detection/Mitigation Mechanism	Testbed	Advantage	Defect
Xie et al. [120]	2019	Fast Fourier Transform, Mean Euclidean Distance	Threshold	SoftGuard	Floodlight, EdgeCore AS4610-54T	Lightweight	Fast Fourier Transform is not a robust enough method for analyzing the unstable throughput sequence.
Tang et al. [110]	2021	Statistics on packets or their fields, mean absolute temporal derivative, cumulative length of the waveform	HGBPLDT algorithm	HGB-FP	Ryu, Mininet	Low complexity, robust detection, and fast mitigation response	The method can only mitigate UDP-based attack flows.
Tang et al. [107]	2021	Statistics on packets or their fields, series of time-frequency analysis	Series of machine learning algorithms	Performance and features (P&F)	Ryu, Mininet	Versatile detection and mitigation framework	All decisions of the P&F are deployed on the control plane, making it fragile to Data-to-Control attacks.
Ahalawat et al. [4]	2022	Rényi entropy, information distance	Threshold	Rényi Entropy with Packet Drop (REPD)	Ryu, Mininet	Early detection and mitigation	The mitigation approach which is based on installing flow rules is easily targeted by flow table overflow attacks.
Tang et al. [111]	2022	SAX algorithm	Series of machine learning algorithms	PeakSAX	Ryu, Mininet	Lightweight, fast real-time detection and mitigation	The method is vulnerable to IP spoofing attack flows.
Tang et al. [102]	2023	Rank-Sum Ratio algorithm	FASSA-SVM algorithm	FSS-RSR	Ryu, Mininet	Low complexity, robust detection, and the ability to mitigate both UDP-based and TCP-based attack flows	The parameters selected for the SVM using the FASSA algorithm have low interpretability.

Rényi entropy and the Rank-Sum Ratio algorithm to analyze flow features. Tang et al. [111] leverage a specific encryption algorithm, the SAX (Symbolic Aggregate Approximation) algorithm to reach the same feature extraction target. In addition, referring to the results of the feature analysis, these efforts immediately make decisions against these attacks by some machine learning related algorithms or the threshold.

Table 6 illustrates the work against Slow Request DoS attacks based on SDN. Except for some aspects of methods mentioned in Table 6, Yungaicela-Naula et al. [131] typically leverage deep reinforcement learning against Slow Request DoS attacks, and Benzaïd et al. [15] generate adversarial examples by the FGSM (Fast Gradient Sign Method) to train a robust machine learning model against these attacks.

In conclusion, thanks to programming flexibility and APIs contributed by SDN, traditional low-rate attacks can be efficiently defended online in SDN.

5 Discussion

Despite extensive research on countermeasures for low-rate threats in programmable networks, ensuring their security remains a pressing challenge. It requires researchers to focus on enhancing the security of programmable networks and developing more flexible and extensible security policy management frameworks. This section provides an opportunity to delve into the challenges and potential solutions for the programmable networks regarding SDN vulnerabilities related to low-rate threats, low-rate threats and countermeasures related to PDP, low-rate attacks vs high-volume attacks, and suggestions for securing SDN.

Table 6. Summary of Countermeasures for Slow Request DoS Attacks

Authors	Year	Feature Analysis Method	Decision-making Method	Detection or Mitigation Mechanism	Testbed	Advantage	Defect
Hong et al. [36]	2017	Duration of each request, count of connections related to each client	Threshold	Slow HTTP DDoS Defense Application (SHDA)	OpenFlow 1.5.1 supported controller, NS3	Consider the presence of slow clients.	The method uses only the threshold to determine whether the request or even the client is legitimate.
Benzaïd et al. [15]	2020	Statistics on packets or their fields	MLP algorithm with adversarial learning	A robust application-layer DDoS self-protection framework	ONOS, Open vSwitch	Ability against white-box attacks.	Adversarial examples generated by an FGSM attack are not sophisticated enough.
Yungaicela-Naula et al. [131]	2022	Statistics on packets or their fields, PCA	LSTM, DNN algorithm with deep reinforcement learning	An SDN-based security framework	ONOS, Mininet, Apache Web Server	Modular, flexible, and scalable, with consideration for the issue of legitimate clients being blocked due to false positive rates (FPR).	Per connection an agent, which is limited by available CPU capacity.
Aslam et al. [12]	2022	Statistics on packets or their fields	Series of machine learning algorithms	ONOS Flood Defender	ONOS, Mininet	Both slow and fast request DoS attacks can be detected and mitigated.	Designed for a single ONOS controller only.
Mohammadi et al. [67]	2023	Statistics on packets or their fields	Series of machine learning algorithms	HTTPScout	Ryu, Mininet	Lightweight, low complexity.	Long attack detection time, high TCAM overhead.
Gonçalves et al. [34]	2023	/	Whether to follow the HTTP redirection	A protection system against HTTP flood attacks	Ryu, Mininet	Simple, scalable.	Assume all ASes are reachable.

5.1 SDN Vulnerabilities Related to Low-Rate Threats

5.1.1 Low-Rate Attacks Exploiting SDN Vulnerabilities. Since the emergence of SDN, it has been accompanied by innovative and open source network technologies and applications, and plenty of open source SDN software has played an essential role during such a process. However, this open source environment can also lead to some unexpected risks. In SDN, software quality, robustness, and vulnerability are critical security issues. Among them, the defects in SDN software, including SDN controllers, are a significant concern, and the literature has uncovered some unique and widespread security vulnerabilities in different SDN layers. Consequently, such precise attacks against some specific vulnerabilities have also emerged as one of the serious threats to SDN. In this section, we list and analyze several low-rate attacks exploiting SDN vulnerabilities. Typically, these attacks can be launched by constructing a few specialized packets without consuming many resources. Compared with the traditional flooding attack, this attack causes similar damage to the network at a lower attack cost.

On the SDN Data Plane. OpenvSwitch is an integral element of SDN implementation and deployment, playing a key role in network functions such as matching, forwarding, and routing. In terms of switch matching, Cao et al. [20] discovered a matching hijacking vulnerability in OpenvSwitch. When an application tries to install flow rules, some SDN systems do not check the discrepancy between buffer IDs and match fields. As a result, an attacker can get around the forwarding matching domain and maliciously hijack the regular packets. Such an attack can cause extensive security threats in SDN data, control, and application planes. Regarding switch routing, a type of attack called *topology pollution* can disturb the data plane through abnormal ICMP and ARP packets, and interfere with the global visibility of the SDN network [37, 97, 116]. Furthermore, related research suggests that users on the data plane can obtain the core of SDN through the investigation method to obtain configuration information and topology, which means that SDN is facing the risk of

information leakage [75]. In SDN architecture, the switch itself may also turn into a bottleneck. Due to the switch's limited processing power and memory, an adversary can fabricate a vast number of bogus packets, causing the controller to issue the switch an overwhelming number of flow rules. Zhang et al. [133] studied the reflection attack, which makes use of data plane events to make the control plane send an excessive number of control messages to the switch in order to exhaust it. To strengthen the attack, they created a two-stage strategy to amplify the attack and make the reflection attack more stealthy and effective [133].

On the SDN Control Plane. The complex mechanisms in SDN controllers, coupled with the code execution status, can create opportunities and vulnerabilities for attackers. For example, Xiao et al. [118] have proposed the D^2C^2 attack, which is a harmful event injection threat based on the data dependency chain creation and chaining in the source code of two different SDN controllers. By carefully constructing a packet in the data plane, Packet-In can inject the malicious attack load into the SDN controller, causing a series of unexpected functions to become contaminated by malicious events. This contamination can result in the functions in the entire dependency chain being affected by malicious events. Another attack called *state manipulation* proposed by Xu et al. [123] can make use of SDN events or management messages to trigger vulnerabilities. These well-designed events can trigger race conditions among the network state management functions in the SDN control plane, injecting or pressuring malicious loads through attacks to make the key services, resources, or variables in the SDN controller compete with each other. These functions compete with the same resources, leading to waiting or deadlocks between network events, causing adverse effects. The Worm-Hole attack in SDN, proposed by Hua et al. [39], exploits weaknesses in the current controller topology management due to the LLDP, which lacks support for integrity verification. Attackers are capable of using LLDP packet manipulation to proclaim an invalid link to the controller, causing the controller to run the risk of accidentally sending flows to a dead connection, which could result in DoS, eavesdropping, and even hijacking attacks.

On the SDN Control Channel. The control channel is a possible target since it frequently carries sensitive network data and significant control decisions. The MitM attack in SDN, which makes use of the control channel's absence of message authentication, was proposed by Benton et al. [14]. By altering the control messages emitted by the controller, attackers can direct the switches to install incorrect entries in the flow tables. Eventually, attackers possess full control over all downstream switches and perform fine-grained eavesdropping attacks which are challenging to identify and mitigate. Due to the lack of encryption of control messages transmitted in the control channel, attackers can apply existing IP sniffing tools to achieve eavesdropping on the ongoing control messages to exfiltrate the topology and other sensitive data of both controllers and downstream switches [127].

5.1.2 Countermeasures for Low-Rate Attacks Exploiting SDN Vulnerabilities. As discussed in Section 5.1.1, numerous low-rate attacks that exploit vulnerabilities in SDN components have proven effective, severely compromising entire SDN networks. To combat these attacks, it is essential to identify the underlying causes of these vulnerabilities. In the following, we break down each attack to its fundamental flaws within the SDN components and conduct an in-depth analysis, where we discuss several possible defense measures that try to lessen the impact of the attacks mentioned earlier by presenting the vulnerabilities retrieved from the attack scenarios.

Countermeasures on the SDN Data Plane.

Architectural bottleneck: The fact that the control plane centrally maintains the network raises this vulnerability in the SDN design, which can be remotely exploited to reduce network availability. SWGuard [133], employing a multi-queue scheduling tactic to introduce variable delays for

different downlink messages, presents a promising approach to significantly mitigate the impact of Reflection attacks targeting this vulnerability. Another approach that protects data plane resources is FloodGuard [116]. It derives proactive flow rules to maintain network policy enforcement and submits cached flooded packets to the controller through rate limiting and round-robin scheduling.

Weak authentication: SDN controllers authenticate the devices such as switches and hosts using a distinct authentication process when establishing connections with them. According to research of Hong et al. [37] and Skowyra et al. [97], Floodlight as well as OpenDaylight employ weak network element authentication mechanisms. The attacker can leverage flaws in the present host tracking and connection discovery services to distort the overall network view and compromise its integrity. Hong et al. [37] proposed TopoGuard, which offers improved network element authentication, as a solution to this issue. Skowyra et al. [97] developed TopoGuard+, an extension to TopoGuard to mitigate and even prevent topology tampering caused by in-band LLDP port amnesia attacks, of which TopoGuard is insufficient to detect and mitigate. Concheck, developed by Cao et al. [20], creates mappings between buffered packets and buffered IDS and uses the mappings to check for inconsistencies between the two to match fields of Flow-Mod messages and stop the API request.

Architectural weakness: To handle packets that do not match any existing flow entries, OpenFlow-supported switches query the SDN controllers for Table-Miss entries in the flow table. However, this process can significantly delay flow processing and represents a potential architectural weakness in OpenFlow. Attackers can fingerprint the target network and even infer sensitive information, such as idle and hard timeout. Patwardhan et al. [75] designed a mitigation strategy to identify attack probe packets by examining the packet headers and data as well as the interval between subsequent packets from the same source.

Countermeasures on the SDN Control Plane.

Weak authentication: The widely used LLDP protocol has been shown to be vulnerable to message tampering, which can result in compromised network topology information being received by the SDN controller. This could potentially lead to the controller mistakenly routing flows through false links, causing network disruptions or security breaches. In response to this issue, Hua et al. [39] implemented a detection and mitigation strategy which uses the delay time to determine where Worm-Hole attacks are present and applies relay hosts to establish in-band channels between switches to prevent the attack.

Architectural weakness: Due to a logical flaw in the controller's implementation, which allows for many network events to occur on the shard network states, SDN's asynchronism frequently results in dangerous race conditions on the shard network states. To address this issue, Xu et al. [123] proposed ConGuard, which found 15 previously unknown harmful race conditions in three popular SDN controllers. They have reported these vulnerabilities to the developers and helped fix 12 of them. In addition, Xiao et al. [118] invented a novel tool, SVHunter, which detects 18 previously unknown data dependency creation and chaining vulnerabilities on Floodlight, ONOS, and OpenDaylight. They have reported these vulnerabilities to the vendors, and the vendors have patched 9 of them.

Countermeasures on the SDN Control Channel.

Weak authentication: Although SSL/TLS protocol can effectively protect control channel communication in OpenFlow, it is not widely adopted, resulting in a lack of practical encryption of control messages. Ndonda and Sadre [71] proposed the priority multipath routing strategy, which avoids delay peaks in the network caused by a basic multipath routing strategy. Their strategy makes use of OpenFlow's rule priority to make sure that each switch has a matched forwarding rule at all times. In addition, OpenFlow does not provide a control message integrity-checking function, opening the door for attackers to conduct MitM attacks.

5.2 Low-Rate Threats and Countermeasures Related to the PDP

5.2.1 Low-Rate Attacks Related to the PDP. The PDP is revolutionizing the SDN architecture, greatly enhancing the flexibility of network programming. With its advantages in fine-grained per-packet processing and high-performance line-rate forwarding, numerous network applications are favoring deployment in the data plane. On the one hand, research on low-rate threats has created promising opportunities, such as new developments in countermeasures for topology poisoning attacks and protocol abuse. On the other hand, applications deployed on the PDP are potentially vulnerable to low-rate threats.

Topology Poisoning Attack. Topology poisoning attacks [98] aim to disrupt a network topology by inducing changes in the routing path, leading to traffic being directed to malicious nodes. This can result in security issues such as network paralysis or data leakage. The technology used for this type of attack has advanced significantly in recent years. The most traditional and well-established form of topology poisoning attacks is ARP cache poisoning, or ARP spoofing [69], which takes the form of a MitM attack. Since the host updates its ARP cache based solely on the data in an ARP response message without first confirming the authenticity of the response, once attackers successfully deceive both parties involved in the connection, they can intercept and eavesdrop on their messages, and potentially destroy a larger network range by targeting the cache with a broadcast address.

Since both the controller and upper-layer applications rely on topology information to make decisions, tampering with this information can mislead applications and services in SDN, resulting in unpredictable consequences. Most SDN controllers utilize a host tracking service that maintains Host Profiles to understand the location and status of the underlying hosts. If the Packet-In information uploaded by the host conflicts with the Host Profiles, the controller updates its Host Profiles and considers its location has changed. Hong et al. [37] proposed the host location hijacking attack, which is initiated on the premise that the attacker possesses compromised devices through malware infection and has read/write privileges to packets. The Host Profile update process lacks an authentication mechanism, allowing an attacker to impersonate a target host. This can enable them to send an incorrect Packet-In message to the controller before the target host does so. As a result, the host tracking service may trigger an update operation incorrectly. Once this happens, all packets sent to the target host across the network will be redirected to the attacker, potentially leading to serious security breaches.

Similar to the routing discovery protocol, the Open Flow Discovery Protocol is used in SDN to provide link discovery service. Switches are required to broadcast LLDP packets to each port according to this protocol. The problem lies in the difficulty of ensuring the authenticity of LLDP packets and limiting the broadcast path only to OpenFlow switches, which gives attackers an opportunity to exploit. Thus, **Link Fabrication Attack (LFA)** [37] can be launched through two approaches: LLDP injection and LLDP relay. The former is to modify specific content by forging LLDP packets to declare a false connection, whereas the latter is to incorrectly forward the received LLDP packets to create an illusion of connectivity between two switches that do not exist.

Protocol Abuse Attack. There are also many network participants who behave improperly and violate the corresponding protocols, which can affect network security and performance in a subtle manner. In 1999, Savage et al. [89] found that if a receiver sends an ACK to the sender prematurely to acknowledge data that has not actually been received, it can mislead the sender into making an optimistic assessment of the network conditions, causing it to extend the congestion control window and boost the sending rate. This can potentially congest the bottleneck link and cause network performance to deteriorate, making it a dangerous attack. **Explicit Congestion Notification (ECN)** abuse [52] is also a common and easy-to-implement malicious behavior. ECN is used to inform hosts about congestion between them through certain flag bits in the packet header.

If a host indicates its support for ECN but then sets the relevant flags incorrectly, it can cause inaccurate congestion messages or just ignore the congestion notification, which can disrupt the normal congestion control of the network.

Low-Rate Threats to PDP Applications. Although the PDP offers numerous benefits, such as per-packet processing and line-rate forwarding, PDP applications are also susceptible to vulnerabilities that can be exploited by low-rate threats. In the following, we conduct our survey concerning two categories of common PDP applications that may suffer from low-rate threats: PDP applications that use **compact probabilistic data structures (CPDS)** and **in-band network telemetry (INT)** applications.

Due to limited memory of the PDP (e.g., only 120 Mbit SRAM and 6.2 Mbit TCAM on the Tofino1 switch [125]), many PDP applications employ CPDS to enable their functionality. However, CPDS are vulnerable to low-rate threats such as pollution attacks. Harish et al. [87] investigated the impact of adversarial network inputs on Bloom filters. They illustrated the viability of a pollution attack on FlowRadar [56], which distorts flow statistics using carefully crafted malicious flows, leading to a 99% degradation in FlowRadar's accuracy. In addition, Chen et al. [24] proposed the Stalker attack to undermine the accuracy of sketches operating on programmable switches. Stalker attack manipulates operations during the deployment of sketches, resulting in tampered sketches that inaccurately record flow data during runtime, thereby diminishing measurement accuracy.

INT embeds telemetry data into the payload of network traffic to monitor and analyze network conditions in real time, offering fine-grained insights and high monitoring accuracy. However, it also possesses vulnerabilities that attackers can exploit, resulting in low-rate threats. Kong et al. [50] presented four manipulation attacks that exploit INT weaknesses to easily cause serious damage to the network by manipulating INT packets at minimal cost, and the attacker can effortlessly bypass network monitoring and compromise network performance.

5.2.2 Countermeasures for Low-Rate Attacks Related to the PDP. In this section, we first summarize the contributions of the PDP in detecting topology poisoning and protocol abuse attacks. We then present the countermeasures for low-rate threats to PDP applications.

Countermeasures for Topology Poisoning Attack. In traditional SDN, detecting topology poisoning attacks requires forwarding suspicious traffic to the control plane, which introduces significant communication overhead and cannot exclude the risk of the controller being attacked. Moreover, centralized detection methods are ineffective against specific types of topology attacks, such as DACP (Data Plane ARP Cache Poisoning Attack) and relay-type LFA. With in-network customization of packet forwarding, PDP provides the solution to address this problem. SECAP Switch [98] is a defense system against topological poisoning attacks in the data plane, mainly targeting DACP and relay-type LFA, which are difficult to detect in traditional SDN. The method consists of two key parts, including source address verification and anomaly detection. A set of registers is maintained for each switch port to keep track of the mapping between MAC addresses and IP addresses. By using the P4 parser to progressively extract packet headers layer by layer, including MAC/IP, ARP, and LLDP, the forged packets used in the two attacks mentioned previously can be filtered out in the initial stage. However, relay-type LFA attackers who set the correct MAC address of the target network during the initial connection can bypass this check. The second part of SECAP, anomaly detection, is designed for this purpose. The feature used in this process is the LLDP interval, which has been experimentally found to perform well. It does not require involvement from the controller for measurement, but rather only needs to read and record the timestamp of the LLDP packet entering the switch and can fully differentiate between a real link and a fabricated link forged by attackers. SECAP has a small memory footprint and is able to block malicious traffic at the switch level, which prevents the control channel from being flooded with mirrored traffic.

Another advantage of SECAP is that it can adapt to different network environments by offering three modes of operation.

Countermeasures for Protocol Abuse Attack. Laraba et al. [52] proposed an approach to solve the protocol abuse problem using the PDP, specifically targeting misbehaving TCP end hosts. They concentrate on two primary types of protocol abuse: Optimistic ACK attack and ECN abuse. Their method utilized EFSM (Extended Finite State Machine) abstraction to achieve a security monitoring function and mapped a protocol's EFSM to the PDP using design primitives. In their method, the PISA and P4 programming language enabled data plane programmability without requiring changes to the end hosts or protocol specification. Additionally, this approach has reduced capital and operating costs by leveraging the programmability of the PDP.

Countermeasures for Low-Rate Threats to PDP Applications. Harish et al. [87] briefly discuss the mitigation of pollution attacks against Bloom filters in terms of four areas: best practices for system design, observing traffic response, modeling benign bloom filter growth, and ranking the flows. Chen et al. [24] discussed the feasibility of stateful verification to detect Stalker attacks. Due to the significant discrepancy between the flow data stored in sketches tampered by the Stalker attack and the actual flow data, quantifying the accuracy of the flow data measured by the sketch for stateful verification is an effective and low-overhead solution for detecting Stalker attacks. To address INT manipulation attacks, Kong et al. [50] designed SecureINT, which provides fast encryption and integrity verification supported by programmable switches to ensure the confidentiality and integrity of in-band network telemetry packets. Specifically, SecureINT employs Even-Mansour for encryption and SipHash for integrity verification.

5.3 Low-Rate Attacks vs High-Volume Attacks

The characteristics of low-rate attacks are that they have a slow attack speed and are not easily detected, whereas volumetric attacks overwhelm the target system with a large amount of traffic and requests. Therefore, different strategies and techniques are needed to address low-rate attacks and volumetric attacks. To deal with low-rate attacks, the main focus is on flow analysis and behavior analysis, and the deployment of a dedicated defense system. For volumetric attacks, the main strategy is traffic cleaning and diversion, which separates normal traffic from malicious traffic to ensure that normal network services are not affected. Traffic cleaning and diversion can be achieved through hardware devices and software systems, including firewalls, intrusion detection systems, and load balancers. Compared to low-rate attacks, high-volume attacks may be easier to detect and mitigate due to their distinct characteristics. However, they can still cause significant disruption and damage to the target network, especially if the network is not designed to handle large amounts of traffic. Therefore, it is important to consider both low-rate and high-volume attacks when designing an integrated security system for programmable networks, and to adopt a multi-layered approach that covers various aspects of the network architecture and operation.

5.4 Suggestions for Securing SDN

SDN has indeed brought greater flexibility and programmability to network innovation. However, it has also brought a series of security issues at the same time. Whether SDN is a true next-generation network or just a fleeting trend is a topic that has been explored by many scholars from various aspects such as centralization and distribution, technology, applications, and market. As stated in this article, each layer of SDN is more or less under security threats. If SDN security cannot be guaranteed, its further development will be hindered. First, we recommend maintaining each layer and prioritizing the control layer. Deploying specific security strategies for each layer can make the protection more targeted and goal-oriented, whereas focusing on the security of the controller is important to ensure that global control of the network is not compromised

or seized. Second, fully leverage the advantages of the PDP by deploying attack detection and defense strategies to programmable devices in the data plane as much as possible. The design of defense systems should aim to achieve universal abnormal traffic detection, reduce resource consumption, and lower communication bandwidth. Third, it is important to improve the security of the protocols themselves, as many attacks have seized the vulnerabilities of protocols and have the opportunity to exploit them. Even the widely used OpenFlow southbound protocol cannot guarantee the security of control channels. Moreover, the northbound and east-west interfaces of SDN are still undefined. By considering security issues during the initial design phase, the potential attack surface of SDN can be minimized.

6 Conclusion and Future Directions

6.1 Conclusion

SDN is a revolutionary architecture that brings flexibility, abstraction, programmability, and virtualization to overcome the shortcomings and inconveniences of conventional network architecture. However, this novel paradigm is also vulnerable to low-rate threats, making its security very challenging.

To comprehensively assess low-rate threats in SDN, we reviewed recent research works from the perspective of different SDN planes. Specifically, we comprehensively analyzed the low-rate threats on the data plane, control plane, application plane, and control channel. In addition, we analyzed low-rate attacks exploiting SDN vulnerabilities, traditional low-rate attacks in SDN, and PDP-related low-rate attacks. Besides this, we reviewed and summarized the countermeasures against different low-rate threats in SDN. Despite the considerable amount of research on countermeasures for low-rate threats in SDN, ensuring the security of SDN networks remains a pressing challenge. Finally, we provided a comparative analysis and discussion on low-rate attacks vs high-volume attacks and gave some suggestions to ensure SDN security. There are remaining challenges in defending against low-rate threats and maintaining SDN security, and it is our hope that research in this area can benefit from this survey.

6.2 Future Directions

With the continuous advancement of attack technology, low-rate attacks will continue to exist and evolve. Attackers will devote more effort to the research of low-rate attacks, which will make the attack strategy more complex and subtle, and move toward intelligence. For example, adaptive attacks can adjust their attack patterns based on the response of the target system to evade countermeasures. This also prompts defenders to seek more comprehensive and innovative security defense technologies, and the development and maturity of network devices and architectures may provide help from different perspectives. For instance, a distributed architecture and blockchain technology can be utilized to establish a distributed trust mechanism, which can enhance the security and reliability of the system. The advent of the PDP and network function virtualization further increases the speed and flexibility of security policy deployment and upgrades. In addition, developments in the field of artificial intelligence also provide many methods to boost attack detection and response capabilities.

References

[1] AbdelRahman Abdou, Paul C. van Oorschot, and Tao Wan. 2018. Comparative analysis of control plane security of SDN and conventional networks. *IEEE Communications Surveys & Tutorials* 20, 4 (2018), 3542–3559.

[2] Belema Agborubere and Erika Sanchez-Velazquez. 2017. OpenFlow communications and TLS security in software-defined networks. In *Proceedings of the 2017 IEEE International Conference on Internet of Things, IEEE Green Computing and Communications, IEEE Cyber, Physical, and Social Computing, and IEEE Smart Data.* 560–566.

[3] Neha Agrawal and Shashikala Tapaswi. 2021. An SDN-assisted defense mechanism for the shrew DDoS attack in a cloud computing environment. *Journal of Network and Systems Management* 29, 2 (2021), 12.
[4] Anchal Ahalawat, Korra Sathya Babu, Ashok Kumar Turuk, and Sanjeev Patel. 2022. A low-rate DDoS detection and mitigation for SDN using Renyi entropy with packet drop. *Journal of Information Security and Applications* 68 (2022), 103212.
[5] Mohammad Adnan Aladaileh, Mohammed Anbar, Ahmed J. Hintaw, Iznan H. Hasbullah, Abdullah Ahmed Bahashwan, Taief Alaa Al-Amiedy, and Dyala R. Ibrahim. 2023. Effectiveness of an entropy-based approach for detecting low-and high-rate DDoS attacks against the SDN controller: Experimental analysis. *Applied Sciences* 13, 2 (2023), 775.
[6] Mohammad Adnan Aladaileh, Mohammed Anbar, Ahmed J. Hintaw, Iznan H. Hasbullah, Abdullah Ahmed Bahashwan, and Shadi Al-Sarawi. 2022. Renyi joint entropy-based dynamic threshold approach to detect DDoS attacks against SDN controller with various traffic rates. *Applied Sciences* 12, 12 (2022), 6127.
[7] Abdussalam Ahmed Alashhab, Mohd Soperi Mohd Zahid, Mohamed A. Azim, Muhammad Yunis Daha, Babangida Isyaku, and Shimhaz Ali. 2022. A survey of low rate DDoS detection techniques based on machine learning in software-defined networks. *Symmetry* 14, 8 (2022), 1563.
[8] Muhammad Nadeem Ali, Muhammad Imran, Muhammad Salah ud din, and Byung-Seo Kim. 2023. Low rate DDoS detection using weighted federated learning in SDN control plane in IoT network. *Applied Sciences* 13, 3 (2023), 1431.
[9] Tariq Emad Ali, Yung-Wey Chong, and Selvakumar Manickam. 2023. Machine learning techniques to detect a DDoS attack in SDN: A systematic review. *Applied Sciences* 13, 5 (2023), 3183.
[10] Moreno Ambrosin, Mauro Conti, Fabio De Gaspari, and Radha Poovendran. 2017. LineSwitch: Tackling control plane saturation attacks in software-defined networking. *IEEE/ACM Transactions on Networking* 25, 2 (2017), 1206–1219.
[11] Markku Antikainen, Tuomas Aura, and Mikko Särelä. 2014. Spook in your network: Attacking an SDN with a compromised OpenFlow switch. In *Secure IT Systems*. Lecture Notes in Computer Science, Vol. 8788. Springer, 229–244.
[12] Naziya Aslam, Shashank Srivastava, and M. M. Gore. 2022. Onos flood defender: An intelligent approach to mitigate DDoS attack in SDM. *Transactions on Emerging Telecommunications Technologies* 33, 9 (2022), e4534.
[13] Theophilus Benson, Aditya Akella, and David A. Maltz. 2010. Network traffic characteristics of data centers in the wild. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*. 267–280.
[14] Kevin Benton, L. Jean Camp, and Chris Small. 2013. OpenFlow vulnerability assessment. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*. 151–152.
[15] Chafika Benzaïd, Mohammed Boukhalfa, and Tarik Taleb. 2020. Robust self-protection against application-layer (D)DoS attacks in SDN environment. In *Proceedings of the 2020 IEEE Wireless Communications and Networking Conference*. 1–6.
[16] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, et al. 2014. P4: Programming protocol-independent packet processors. *ACM SIGCOMM Computer Communication Review* 44, 3 (2014), 87–95.
[17] Pat Bosshart, Glen Gibb, Hun-Seok Kim, George Varghese, Nick McKeown, Martin Izzard, Fernando Mujica, and Mark Horowitz. 2013. Forwarding metamorphosis: Fast programmable match-action processing in hardware for SDN. *ACM SIGCOMM Computer Communication Review* 43, 4 (2013), 99–110.
[18] Wolfgang Braun and Michael Menth. 2014. Software-defined networking using OpenFlow: Protocols, applications and architectural design choices. *Future Internet* 6, 2 (2014), 302–336.
[19] Enrico Cambiaso, Gianluca Papaleo, and Maurizio Aiello. 2012. Taxonomy of slow DoS attacks to web applications. In *Proceedings of the International Conference on Recent Trends in Computer Networks and Distributed Systems Security*. 195–204.
[20] Jiahao Cao, Renjie Xie, Kun Sun, Qi Li, Guofei Gu, and Mingwei Xu. 2020. When match fields do not need to match: Buffered packets hijacking in SDN. In *Proceedings of the Network and Distributed System Security Symposium*. 1–15.
[21] Jiahao Cao, Mingwei Xu, Qi Li, Kun Sun, and Yuan Yang. 2023. The LOFT attack: Overflowing SDN flow tables at a low rate. *IEEE/ACM Transactions on Networking* 31, 3 (2023), 1416–1431.
[22] Jiahao Cao, Mingwei Xu, Qi Li, Kun Sun, Yuan Yang, and Jing Zheng. 2018. Disrupting SDN via the data plane: A low-rate flow table overflow attack. In *Proceedings of the International Conference on Security and Privacy in Communication Systems*. 356–376.
[23] Kuan-Yin Chen, Sen Liu, Yang Xu, Ishant Kumar Siddhrau, Siyu Zhou, Zehua Guo, and H. Jonathan Chao. 2022. SDNShield: NFV-based defense framework against DDoS attacks on SDN control plane. *IEEE/ACM Transactions on Networking* 30, 1 (2022), 1–17.
[24] Xiang Chen, Hongyan Liu, Qun Huang, Dong Zhang, Haifeng Zhou, Chunming Wu, Xuan Liu, and Muhammad Khurram Khan. 2023. Stalker attacks: Imperceptibly dropping sketch measurement accuracy on programmable switches. *IEEE Transactions on Information Forensics and Security* 18 (2023), 5832–5847.
[25] P4 Language Consortium. 2024. Behavioral Model Version 2 (bmv2). Retrieved July 23, 2024 from https://github.com/p4lang/behavioral-model

[26] Mauro Conti, Fabio De Gaspari, and Luigi V. Mancini. 2020. A novel stealthy attack to gather SDN configuration-information. *IEEE Transactions on Emerging Topics in Computing* 8, 2 (2020), 328–340.
[27] Juan Camilo Correa Chica, Jenny Cuatindioy Imbachi, and Juan Felipe Botero Vega. 2020. Security in SDN: A comprehensive survey. *Journal of Network and Computer Applications* 159 (2020), 102595.
[28] Italo Dacosta, Mustaque Ahamad, and Patrick Traynor. 2012. Trust no one else: Detecting MITM attacks against SSL/TLS without third-parties. In *Proceedings of the European Symposium on Research in Computer Security*. 199–216.
[29] Shuhua Deng, Xing Gao, Zebin Lu, and Xieping Gao. 2018. Packet injection attack and its defense in software-defined networks. *IEEE Transactions on Information Forensics and Security* 13, 3 (2018), 695–705.
[30] Mohan Dhawan, Rishabh Poddar, Kshiteej Mahajan, and Vijay Mann. 2015. Sphinx: Detecting security attacks in software-defined networks. In *Proceedings of the Network and Distributed System Security Symposium*. 1–15.
[31] Ping Dong, Xiaojiang Du, Hongke Zhang, and Tong Xu. 2016. A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows. In *Proceedings of the IEEE International Conference on Communications*. 1–6.
[32] Deyun Gao, Zehui Liu, Ying Liu, Chuan Heng Foh, Ting Zhi, and Han-Chieh Chao. 2018. Defending against Packet-In messages flooding attack under SDN context. *Soft Computing* 22 (2018), 6797–6809.
[33] Christos Gkountis, Miran Taha, Jaime Lloret, and Georgios Kambourakis. 2017. Lightweight algorithm for protecting SDN controller against DDoS attacks. In *Proceedings of the 10th IFIP Wireless and Mobile Networking Conference*. 1–6.
[34] Diego S. M. Gonçalves, Rodrigo S. Couto, and Marcelo G. Rubinstein. 2023. A protection system against HTTP flood attacks using software defined networking. *Journal of Network and Systems Management* 31, 1 (2023), 16.
[35] Frederik Hauser, Marco Häberle, Daniel Merling, Steffen Lindner, Vladimir Gurevich, Florian Zeiger, Reinhard Frank, and Michael Menth. 2023. A survey on data plane programming with P4: Fundamentals, advances, and applied research. *Journal of Network and Computer Applications* 212 (2023), 103561.
[36] Kiwon Hong, Youngjun Kim, Hyungoo Choi, and Jinwoo Park. 2018. SDN-assisted slow HTTP DDoS attack defense method. *IEEE Communications Letters* 22, 4 (2018), 688–691.
[37] Sungmin Hong, Lei Xu, Haopei Wang, and Guofei Gu. 2015. Poisoning network visibility in software-defined networks: New attacks and countermeasures. In *Proceedings of the Network and Distributed System Security Symposium*. 1–15.
[38] Tao Hu, Zhen Zhang, Peng Yi, Dong Liang, Ziyong Li, Quan Ren, Yuxiang Hu, and Julong Lan. 2021. SEAPP: A secure application management framework based on REST API access control in SDN-enabled cloud environment. *Journal of Parallel and Distributed Computing* 147 (2021), 108–123.
[39] Jingyu Hua, Zidong Zhou, and Sheng Zhong. 2021. Flow misleading: Worm-Hole attack in software-defined networking via building in-band covert channel. *IEEE Transactions on Information Forensics and Security* 16 (2021), 1029–1043.
[40] Intel. 2024. Intel Tofino Programmable Ethernet Switch ASIC. Retrieved July 23, 2024 from https://www.intel.com/content/www/us/en/products/network-io/programmable-ethernet-switch/tofino-series.html
[41] Babangida Isyaku, Maznah Bte Kamat, Kamalrulnizam bin Abu Bakar, Mohd Soperi Mohd Zahid, and Fuad A. Ghaleb. 2020. IHTA: Dynamic Idle-Hard timeout allocation algorithm based OpenFlow switch. In *Proceedings of the IEEE 10th Symposium on Computer Applications and Industrial Electronics*. 170–175.
[42] Saeed Javanmardi, Mohammad Shojafar, Reza Mohammadi, Mamoun Alazab, and Antonio M. Caruso. 2023. An SDN perspective IoT-Fog security: A survey. *Computer Networks* 229 (2023), 109732.
[43] Weiwei Jiang, Haoyu Han, Miao He, and Weixi Gu. 2024. ML-based pre-deployment SDN performance prediction with neural network boosting regression. *Expert Systems with Applications* 241 (2024), 122774.
[44] Sukhveer Kaur, Krishan Kumar, Naveen Aggarwal, and Gurdeep Singh. 2021. A comprehensive survey of DDoS defense solutions in SDN: Taxonomy, research challenges, and future directions. *Computers & Security* 110 (2021), 102423.
[45] Samer Khamaiseh, Edoardo Serra, Zhiyuan Li, and Dianxiang Xu. 2019. Detecting saturation attacks in SDN via machine learning. In *Proceedings of the 4th International Conference on Computing, Communications, and Security*. 1–8.
[46] Samer Y. Khamaiseh, Izzat Alsmadi, and Abdullah Al-Alaj. 2020. Deceiving machine learning-based saturation attack detection systems in SDN. In *Proceedings of the IEEE Conference on Network Function Virtualization and Software Defined Networks*. 44–50.
[47] Ahmed Khurshid, Wenxuan Zhou, Matthew Caesar, and P. Brighten Godfrey. 2012. VeriFlow: Verifying network-wide invariants in real time. In *Proceedings of the 1st Workshop on Hot Topics in Software Defined Networks*. 49–54.
[48] Sian Kim, Changhun Jung, Rhongho Jang, David Mohaisen, and Dae Hun Nyang. 2023. A robust counting sketch for data plane intrusion detection. In *Proceedings of the Network and Distributed System Security Symposium*. 1–17.
[49] Dezhang Kong, Chunming Wu, Yi Shen, Xiang Chen, Hongyan Liu, and Dong Zhang. 2022. TableGuard: A novel security mechanism against flow table overflow attacks in SDN. In *Proceedings of the IEEE Global Communications Conference*. 4167–4172.

[50] Dezhang Kong, Zhengyan Zhou, Yi Shen, Xiang Chen, Qiumei Cheng, Dong Zhang, and Chunming Wu. 2023. In-band network telemetry manipulation attacks and countermeasures in programmable networks. In *Proceedings of the IEEE/ACM 31st International Symposium on Quality of Service*. 1–10.

[51] Aleksandar Kuzmanovic and Edward W. Knightly. 2003. Low-rate TCP-targeted denial of service attacks: The shrew vs. the mice and elephants. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. 75–86.

[52] Abir Laraba, Jérôme François, Shihabur Rahman Chowdhury, Isabelle Chrisment, and Raouf Boutaba. 2021. Mitigating TCP protocol misuse with programmable data planes. *IEEE Transactions on Network and Service Management* 18, 1 (2021), 760–774.

[53] Seungsoo Lee, Changhoon Yoon, and Seungwon Shin. 2016. The smaller, the shrewder: A simple malicious application can kill an entire SDN environment. In *Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks and Network Function Virtualization*. 23–28.

[54] Junyuan Leng, Yadong Zhou, Junjie Zhang, and Chengchen Hu. 2015. An inference attack model for flow table capacity and usage: Exploiting the vulnerability of flow table overflow in software-defined network. *arXiv preprint arXiv:1504.03095* (2015).

[55] Cheng Li, Zhengrui Qin, Ed Novak, and Qun Li. 2017. Securing SDN infrastructure of IoT–fog networks from MitM attacks. *IEEE Internet of Things Journal* 4, 5 (2017), 1156–1164.

[56] Yuliang Li, Rui Miao, Changhoon Kim, and Minlan Yu. 2016. FlowRadar: A better NetFlow for data centers. In *Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation*. 311–324.

[57] Zhiyuan Li, Weijia Xing, Samer Khamaiseh, and Dianxiang Xu. 2020. Detecting saturation attacks based on self-similarity of OpenFlow traffic. *IEEE Transactions on Network and Service Management* 17, 1 (2020), 607–621.

[58] Wei Liang, Yang Yang, Ce Yang, Yonghua Hu, Songyou Xie, Kuan-Ching Li, and Jiannong Cao. 2023. PDPChain: A consortium blockchain-based privacy protection scheme for personal data. *IEEE Transactions on Reliability* 72, 2 (2023), 586–598.

[59] Athanasios Liatifis, Panagiotis Sarigiannidis, Vasileios Argyriou, and Thomas Lagkas. 2023. Advancing SDN from OpenFlow to P4: A survey. *ACM Computing Surveys* 55, 9 (2023), 37.

[60] Sheng Liu, Michael K. Reiter, and Vyas Sekar. 2017. Flow reconnaissance via timing attacks on SDN switches. In *Proceedings of the IEEE 37th International Conference on Distributed Computing Systems*. 196–206.

[61] Xiapu Luo and Rocky K. C. Chang. 2005. On a new class of pulsing denial-of-service attacks and the defense. In *Proceedings of the Network and Distributed System Security Symposium*. 1–19.

[62] Qian Lv, Jing Zhu, Fen Zhou, and Zuqing Zhu. 2020. Network planning with bilevel optimization to address attacks to physical infrastructure of SDN. In *Proceedings of the IEEE International Conference on Communications*. 1–6.

[63] Yassine Maleh, Youssef Qasmaoui, Khalid El Gholami, Yassine Sadqi, and Soufyane Mounir. 2023. A comprehensive survey on SDN security: Threats, mitigations, and future directions. *Journal of Reliable Intelligent Environments* 9, 2 (2023), 201–239.

[64] Christopher Mansour and Danai Chasaki. 2018. Design of an SDN security mechanism to detect malicious activities. In *Proceedings of the 10th International Conference on Ubiquitous and Future Networks*. 186–190.

[65] Rejo Mathew and Vijay Katkar. 2011. Survey of low rate DoS attack detection mechanisms. In *Proceedings of the International Conference and Workshop on Emerging Trends in Technology*. 955–958.

[66] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review* 38, 2 (2008), 69–74.

[67] Reza Mohammadi, Chhagan Lal, and Mauro Conti. 2023. HTTPScout: A machine learning based countermeasure for HTTP flood attacks in SDN. *International Journal of Information Security* 22, 2 (2023), 367–379.

[68] Seyed Mohammad Mousavi and Marc St-Hilaire. 2015. Early detection of DDoS attacks against SDN controllers. In *Proceedings of the International Conference on Computing, Networking, and Communications*. 77–81.

[69] N. Saritakumar, K. V. Anusuya, and Sreehari Krishnakumar. 2023. Detection of ARP spoofing attacks in software defined networks. In *Proceedings of the International Conference on Intelligent Systems for Communication, IoT, and Security*. 422–426.

[70] Maryam M. Najafabadi, Taghi M. Khoshgoftaar, Amri Napolitano, and Charles Wheelus. 2016. Rudy attack: Detection at the network level and its important features. In *Proceedings of the 29th International Florida Artificial Intelligence Research Society Conference*. 282–287.

[71] Gorby Kabasele Ndonda and Ramin Sadre. 2017. A low-delay SDN-based countermeasure to eavesdropping attacks in industrial control systems. In *Proceedings of the IEEE Conference on Network Function Virtualization and Software Defined Networks*. 1–7.

[72] Bruno Astuto A. Nunes, Marc Mendonca, Xuan-Nam Nguyen, Katia Obraczka, and Thierry Turletti. 2014. A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials* 16, 3 (2014), 1617–1634.

[73] Goodness Oluchi Anyanwu, Cosmas Ifeanyi Nwakanma, Jae-Min Lee, and Dong-Seong Kim. 2023. Optimization of RBF-SVM kernel using grid search algorithm for DDoS attack detection in SDN-based VANET. *IEEE Internet of Things Journal* 10, 10 (2023), 8477–8490.

[74] Túlio A. Pascoal, Yuri G. Dantas, Iguatemi E. Fonseca, and Vivek Nigam. 2017. Slow TCAM exhaustion DDoS attack. In *Proceedings of the IFIP International Conference on ICT Systems Security and Privacy Protection*. 17–31.

[75] Aditya Patwardhan, Deepthi Jayarama, Nitish Limaye, Shivaji Vidhale, Zarna Parekh, and Khaled Harfoush. 2019. SDN security: Information disclosure and flow table overflow attacks. In *Proceedings of the IEEE Global Communications Conference*. 1–6.

[76] Trung V. Phan, T. M. Rayhan Gias, Syed Tasnimul Islam, Truong Thu Huong, Nguyen Huu Thanh, and Thomas Bauschert. 2019. Q-MIND: Defeating stealthy DoS attacks in SDN with a machine-learning based defense framework. In *Proceedings of the IEEE Global Communications Conference*. 1–6.

[77] Trung V. Phan, Tri Gia Nguyen, and Thomas Bauschert. 2020. DeepMatch: Fine-grained traffic flow measurement in SDN with deep dueling neural networks. *IEEE Journal on Selected Areas in Communications* 39, 7 (2020), 2056–2075.

[78] Trung V. Phan, Tri Gia Nguyen, Nhu-Ngoc Dao, Truong Thu Huong, Nguyen Huu Thanh, and Thomas Bauschert. 2020. DeepGuard: Efficient anomaly detection in SDN with fine-grained traffic flow monitoring. *IEEE Transactions on Network and Service Management* 17, 3 (2020), 1349–1362.

[79] Phillip A. Porras, Steven Cheung, Martin W. Fong, Keith Skinner, and Vinod Yegneswaran. 2015. Securing the software defined network control layer. In *Proceedings of the Network and Distributed System Security Symposium*. 1–15.

[80] N. Priyanka, T. R. Reshmi, and Krishnan Murugan. 2021. CEOF: Enhanced clustering-based entries optimization scheme to prevent flow table overflow. *Wireless Networks* 28 (2021), 69–83.

[81] Longyan Ran, Yunhe Cui, Chun Guo, Qing Qian, Guowei Shen, and Huanlai Xing. 2022. Defending saturation attacks on SDN controller: A confusable instance analysis-based algorithm. *Computer Networks* 213 (2022), 109098.

[82] Raihan Ur Rasool, Khandakar Ahmed, Zahid Anwar, Hua Wang, Usman Ashraf, and Wajid Rafique. 2021. CyberPulse++: A machine learning-based security framework for detecting link flooding attacks in software defined networks. *International Journal of Intelligent Systems* 36, 8 (2021), 3852–3879.

[83] Bilal Rauf, Haider Abbas, Muhammad Usman, Tanveer A. Zia, Waseem Iqbal, Yawar Abbas, and Hammad Afzal. 2021. Application threats to exploit northbound interface vulnerabilities in software defined networks. *ACM Computing Surveys* 54, 6 (2021), 1–36.

[84] Vinícius De Miranda Rios, Pedro R. M. Inacio, Damien Magoni, and Mário M. Freire. 2022. Detection and mitigation of low-rate denial-of-service attacks: A survey. *IEEE Access* 10 (2022), 76648–76668.

[85] Christian Röpke and Thorsten Holz. 2015. SDN rootkits: Subverting network operating systems of software-defined networks. In *Proceedings of the 18th International Symposium on Research in Attacks, Intrusions, and Defenses*. 339–356.

[86] Christian Röpke and Thosten Holz. 2018. Preventing malicious SDN applications from hiding adverse network manipulations. In *Proceedings of the 2018 Workshop on Security in Softwarized Networks: Prospects and Challenges*. 40–45.

[87] S. A. Harish, K. Shiv Kumar, Anibrata Majee, Amogh Bedarakota, Praveen Tammana, Pravein Govindan Kannan, and Rinku Shah. 2023. In-network probabilistic monitoring primitives under the influence of adversarial network inputs. In *Proceedings of the 7th Asia-Pacific Workshop on Networking*. 116–122.

[88] 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. *Future Generation Computer Systems* 89 (2018), 685–697.

[89] Stefan Savage, Neal Cardwell, David Wetherall, and Tom Anderson. 1999. TCP congestion control with a misbehaving receiver. *ACM SIGCOMM Computer Communication Review* 29, 5 (1999), 71–78.

[90] Gao Shang, Peng Zhe, Xiao Bin, Hu Aiqun, and Ren Kui. 2017. FloodDefender: Protecting data and control plane resources under SDN-aimed DoS attacks. In *Proceedings of the IEEE Conference on Computer Communications*. 1–9.

[91] Seungwon Shin and Guofei Gu. 2013. Attacking software-defined networks: A first feasibility study. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*. 165–166.

[92] Seungwon Shin, Phillip Porras, Vinod Yegneswara, Martin Fong, Guofei Gu, and Mabry Tyson. 2013. Fresco: Modular composable security services for software-defined networks. In *Proceedings of the Network and Distributed System Security Symposium*. 1–16.

[93] Seungwon Shin, Vinod Yegneswaran, Phillip Porras, and Guofei Gu. 2013. Avant-Guard: Scalable and vigilant switch flow management in software-defined networks. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*. 413–424.

[94] Faizan Shoaib, Yang-Wai Chow, and Elena Vlahu-Gjorgievska. 2021. Preventing timing side-channel attacks in software-defined networks. In *Proceedings of the IEEE Asia-Pacific Conference on Computer Science and Data Engineering*. 1–6.

[95] Jagdeep Singh and Sunny Behal. 2020. Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions. *Computer Science Review* 37 (2020), 100279.

[96] Michael Sjoholmsierchio, Britta Hale, Daniel Lukaszewski, and Geoffrey Xie. 2021. Strengthening SDN security: Protocol dialecting and downgrade attacks. In *Proceedings of the IEEE 7th International Conference on Network Softwarization*. 321–329.

[97] Richard Skowyra, Lei Xu, Guofei Gu, Veer Dedhia, Thomas Hobson, Hamed Okhravi, and James Landry. 2018. Effective topology tampering attacks and defenses in software-defined networks. In *Proceedings of the 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. 374–385.

[98] Dylan Smyth, Sandra Scott-Hayward, Victor Cionca, Sean McSweeney, and Donna O'Shea. 2023. SECAP switch–defeating topology poisoning attacks using P4 data planes. *Journal of Network and Systems Management* 31, 1 (2023), 28.

[99] Mustafa Soylu, Luis Guillen, Satoru Izumi, Toru Abe, and Takuo Suganuma. 2021. NFV-Guard: Mitigating flow table-overflow attacks in SDN using NFV. In *Proceedings of the IEEE 7th International Conference on Network Softwarization*. 263–267.

[100] K. Muthamil Sudar and P. Deepalakshmi. 2022. Flow-based detection and mitigation of low-rate DDOS attack in SDN environment using machine learning techniques. In *IoT and Analytics for Sensor Networks*. Lecture Notes in Computer Science, Vol. 244. Springer, 193–205.

[101] Dan Tang, Jingwen Chen, Xiyin Wang, Siqi Zhang, and Yudong Yan. 2022. A new detection method for LDoS attacks based on data mining. *Future Generation Computer Systems* 128 (2022), 73–87.

[102] Dan Tang, Chenjun Gao, Xinmeng Li, Wei Liang, Sheng Xiao, and Qiuwei Yang. 2023. A detection and mitigation scheme of LDoS attacks via SDN based on the FSS-RSR algorithm. *IEEE Transactions on Network Science and Engineering* 10, 4 (2023), 1952–1963.

[103] Dan Tang, Chenjun Gao, Wei Liang, Jiliang Zhang, and Keqin Li. 2023. FTMaster: A detection and mitigation system of low-rate flow table Overflow attacks via SDN. *IEEE Transactions on Network and Service Management* 20, 4 (2023), 5073–5084.

[104] Dan Tang, Siyuan Wang, Boru Liu, Wenqiang Jin, and Jiliang Zhang. 2023. GASF-IPP: Detection and mitigation of LDoS attack in SDN. *IEEE Transactions on Services Computing* 16, 5 (2023), 3373–3384.

[105] Dan Tang, Xiyin Wang, Xiong Li, Pandi Vijayakumar, and Neeraj Kumar. 2023. AKN-FGD: Adaptive Kohonen network based fine-grained detection of LDoS attacks. *IEEE Transactions on Dependable and Secure Computing* 20, 1 (2023), 273–287.

[106] Dan Tang, Yudong Yan, Chenjun Gao, Wei Liang, and Wenqiang Jin. 2023. LtRFT: Mitigate the low-rate data plane DDoS attack with learning-to-rank enabled flow tables. *IEEE Transactions on Information Forensics and Security* 18 (2023), 3143–3157.

[107] Dan Tang, Yudong Yan, Siqi Zhang, Jingwen Chen, and Zheng Qin. 2022. Performance and features: Mitigating the low-rate TCP-targeted DoS attack via SDN. *IEEE Journal on Selected Areas in Communications* 40, 1 (2022), 428–444.

[108] Dan Tang, Dongshuo Zhang, Zheng Qin, Qiuwei Yang, and Sheng Xiao. 2023. SFTO-Guard: Real-time detection and mitigation system for slow-rate flow table overflow attacks. *Journal of Network and Computer Applications* 213 (2023), 103597.

[109] Dan Tang, Siqi Zhang, Jingwen Chen, and Xiyin Wang. 2021. The detection of low-rate DoS attacks using the SADB-SCAN algorithm. *Information Sciences* 565 (2021), 229–247.

[110] Dan Tang, Siqi Zhang, Yudong Yan, Jingwen Chen, and Zheng Qin. 2022. Real-time detection and mitigation of LDoS attacks in the SDN using the HGB-FP algorithm. *IEEE Transactions on Services Computing* 15, 6 (2022), 3471–3484.

[111] Dan Tang, Zhiqing Zheng, Xiaocai Wang, Sheng Xiao, and Qiuwei Yang. 2023. PeakSAX: Real-time monitoring and mitigation system for LDoS attack in SDN. *IEEE Transactions on Network and Service Management* 20, 3 (2023), 3686–3698.

[112] Nikhil Tripathi and Neminath Hubballi. 2021. Application layer denial-of-service attacks and defense mechanisms: A survey. *ACM Computing Surveys* 54, 4 (2021), 33.

[113] Yuchia Tseng, Montida Pattaranantakul, Ruan He, Zonghua Zhang, and Farid Naït-Abdesselam. 2017. Controller DAC: Securing SDN controller with dynamic access control. In *Proceedings of the IEEE International Conference on Communications*. 1–6.

[114] Ismael Amezcua Valdovinos, Jesús Arturo Pérez-Díaz, Kim-Kwang Raymond Choo, and Juan Felipe Botero. 2021. Emerging DDoS attack detection and mitigation strategies in software-defined networks: Taxonomy, challenges and future directions. *Journal of Network and Computer Applications* 187 (2021), 103093.

[115] Lokendra Vishwakarma, Ankur Nahar, and Debasis Das. 2022. LBSV: Lightweight blockchain security protocol for secure storage and communication in SDN-enabled IoV. *IEEE Transactions on Vehicular Technology* 71, 6 (2022), 5983–5994.

[116] Haopei Wang, Lei Xu, and Guofei Gu. 2015. FloodGuard: A DoS attack prevention extension in software-defined networks. In *Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. 239–250.

[117] Xin Wang, Neng Gao, Lingchen Zhang, Zongbin Liu, and Lei Wang. 2016. Novel MITM attacks on security protocols in SDN: A feasibility study. In *Proceedings of the 18th International Conference on Information and Communications Security*. 455–465.

[118] Feng Xiao, Jinquan Zhang, Jianwei Huang, Guofei Gu, Dinghao Wu, and Peng Liu. 2020. Unexpected data dependency creation and chaining: A new attack to SDN. In *Proceedings of the IEEE Symposium on Security and Privacy*. 1512–1526.

[119] Renjie Xie, Jiahao Cao, Qi Li, Kun Sun, Guofei Gu, Mingwei Xu, and Yuan Yang. 2022. Disrupting the SDN control channel via shared links: Attacks and countermeasures. *IEEE/ACM Transactions on Networking* 30, 5 (2022), 2158–2172.

[120] Renjie Xie, Mingwei Xu, Jiahao Cao, and Qi Li. 2019. SoftGuard: Defend against the low-rate TCP attack in SDN. In *Proceedings of the IEEE International Conference on Communications*. 1–6.

[121] Shengxu Xie, Changyou Xing, Guomin Zhang, and Jinlong Zhao. 2021. A table overflow LDoS attack defending mechanism in software-defined networks. *Security and Communication Networks* 2021, 1 (2021), 6667922.

[122] Jianfeng Xu, Liming Wang, and Zhen Xu. 2020. An enhanced saturation attack and its mitigation mechanism in software-defined networking. *Computer Networks* 169 (2020), 107092.

[123] Lei Xu, Jeff Huang, Sungmin Hong, Jialong Zhang, and Guofei Gu. 2017. Attacking the brain: Races in the SDN control plane. In *Proceedings of the 26th USENIX Security Symposium*. 451–468.

[124] Tong Xu, Deyun Gao, Ping Dong, Chuan Heng Foh, and Hongke Zhang. 2017. Mitigating the table-overflow attack in software-defined networking. *IEEE Transactions on Network and Service Management* 14, 4 (2017), 1086–1097.

[125] Jinzhu Yan, Haotian Xu, Zhuotao Liu, Qi Li, Ke Xu, Mingwei Xu, and Jianping Wu. 2024. Brain-on-Switch: Towards advanced intelligent network data plane via NN-driven traffic analysis at line-speed. In *Proceedings of the 21st USENIX Symposium on Networked Systems Design and Implementation*. 419–440.

[126] Hemin Yang, George F. Riley, and Douglas M. Blough. 2019. STEREOS: Smart table entry eviction for OpenFlow switches. *IEEE Journal on Selected Areas in Communications* 38, 2 (2019), 377–388.

[127] Changhoon Yoon, Seungsoo Lee, Heedo Kang, Taejune Park, Seungwon Shin, Vinod Yegneswaran, Phillip Porras, and Guofei Gu. 2017. Flow wars: Systemizing the attack surface and defenses in software-defined networks. *IEEE/ACM Transactions on Networking* 25, 6 (2017), 3514–3530.

[128] Mingli Yu, Ting He, Patrick McDaniel, and Quinn K. Burke. 2020. Flow table security in SDN: Adversarial reconnaissance and intelligent attacks. In *Proceedings of the IEEE Conference on Computer Communications*. 1519–1528.

[129] Bin Yuan, Deqing Zou, Shui Yu, Hai Jin, Weizhong Qiang, and Jinan Shen. 2016. Defending against flow table overloading attack in software-defined networks. *IEEE Transactions on Services Computing* 12, 2 (2016), 231–246.

[130] Meng Yue, Minxiao Wang, and Zhijun Wu. 2021. Low-high burst: A double potency varying-RTT based full-buffer shrew attack model. *IEEE Transactions on Dependable and Secure Computing* 18, 5 (2021), 2285–2300.

[131] Noe M. Yungaicela-Naula, Cesar Vargas-Rosales, Jesús Arturo Pérez-Díaz, and Diego Fernando Carrera. 2022. A flexible SDN-based framework for slow-rate DDoS attack mitigation by using deep reinforcement learning. *Journal of Network and Computer Applications* 205 (2022), 103444.

[132] Chaoqin Zhang, Guangwu Hu, Guolong Chen, Arun Kumar Sangaiah, Ping'an Zhang, Xia Yan, and Weijin Jiang. 2017. Towards a SDN-based integrated architecture for mitigating IP spoofing attack. *IEEE Access* 6 (2017), 22764–22777.

[133] Menghao Zhang, Guanyu Li, Lei Xu, Jiasong Bai, Mingwei Xu, Guofei Gu, and Jianping Wu. 2021. Control plane reflection attacks and defenses in software-defined networks. *IEEE/ACM Transactions on Networking* 29, 2 (2021), 623–636.

[134] Changgang Zheng, Zhaoqi Xiong, Thanh T. Bui, Siim Kaupmees, Riyad Bensoussane, Antoine Bernabeu, Shay Vargaftik, Yaniv Ben-Itzhak, and Noa Zilberman. 2024. IIsy: Hybrid in-network classification using programmable switches. *IEEE/ACM Transactions on Networking* 32, 3 (2024), 2555–2570.

[135] Zhijun Wu, Wenjing Li, Liang Liu, and Meng Yue. 2020. Low-rate DoS attacks, detection, defense, and challenges: A survey. *IEEE Access* 8 (2020), 43920–43943.

[136] Guangmeng Zhou, Zhuotao Liu, Chuanpu Fu, Qi Li, and Ke Xu. 2023. An efficient design of intelligent network data plane. In *Proceedings of the 32nd USENIX Security Symposium*. 6203–6220.

[137] Yadong Zhou, Kaiyue Chen, Junjie Zhang, Junyuan Leng, and Yazhe Tang. 2018. Exploiting the vulnerability of flow table overflow in software-defined network: Attack model, evaluation, and defense. *Security and Communication Networks* 2018, 1 (2018), 4760632.

[138] Qiao Zhu, Zhang Yizhi, and Xie Chuiyi. 2011. Research and survey of low-rate denial of service attacks. In *Proceedings of the 13th International Conference on Advanced Communication Technology*. 1195–1198.

Received 4 May 2023; revised 26 July 2024; accepted 19 October 2024