EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

A Review on DDoS Attack in Controller Environment of Software Defined Network

Gunjani Vaghela[1,*], Nishant Sanghani[2], Bhavesh Borisaniya[3]

[1] Atmiya University, Rajkot, Gujarat, India
[2] Gujarat Technological University, Ahmedabad, Gujarat, India
[3] Shantilal Shah Engineering College, Bhavnagar, Gujarat, India

Abstract

Distributed Denial of Service (DDoS) attacks pose a significant threat to the security and availability of networks. With the increasing adoption of Software-Defined Networking (SDN) and its multi-controller architectures, there is a need to explore effective DDoS attack detection mechanisms tailored to these environments. An overview of the current research on detecting DDoS attacks in SDN environments, with a focus on different detection techniques, methodologies and problems is presented in this survey paper. The survey attempt to identify the limitations and strengths of current approaches and propose potential research directions for improving DDoS detection in this context.

Keywords: Software Defined Network,DDoS, Detection, Machine Learning

Received on 10 03 2024; accepted on 10 06 2024 published on 24 07 2024

doi:10.4108/eetsis.5823

1. Introduction

Networking has become an essential element of our lives in a modern digital world, enabling us to share information and resources easily through the use of ICT. It has transformed the way we communicate, collaborate, and access valuable resources. Networking has become an integral part of various domains, ranging from personal communication to business operations and global connectivity.

The traditional network architecture, serving as a global communication pathway for connecting terminals worldwide, experiences a substantial amount of network traffic. However, the advent of emerging technologies such as mobile computing, cloud computing, and edge computing has introduced significant variations in patterns of network traffic. Particularly, the proliferation of mobile devices has enabled users to access the Internet anytime and anywhere, resulting in diverse traffic sources. Furthermore, the rise of cloud computing and edge computing has introduced a multitude of "east-west" traffic flows, which differ from the conventional "north-south" traffic pattern. Additionally, the growing number of cloud services generates passive traffic across wide area networks [1].

The traditional architecture of network, showcased by conventional routers and switches, faces several challenges in adapting to the evolving network traffic demands. One challenge is the need for network managers to log into individual routers or switches and manually modify configurations using vendor-specific management interfaces. This process proves time-consuming and hampers the ability to respond quickly to rapid changes in network traffic. Another challenge lies in the traditional network's limited scalability, Meeting the scalability requirements of numerous network service providers poses a significant challenge.

To overcome these challenges, innovative approaches have emerged to enhance network flexibility, adaptability, and scalability. One such approach is SDN a paradigm that divides the data from the control plane. and centralizes network management. By abstracting network control, SDN enables rapid reconfiguration of network policies and rules to accommodate dynamic traffic patterns.

Currently, SDN has found applications in different domains, including data centers, wide area networks,

*Corresponding author. Email: vaghelagunjani22@gmail.com

and wireless networks. However, despite its potential, there remain certain security barriers that impede the widespread adoption of SDN in global networks. To ensure the robustness and security of SDN deployment on a broader scale, these obstacles must be overcome [2].

SDN has two main architectural approaches: single controller and multi-controller.

1. Single Controller: Entire network is managed and control by centralised controller in a single controller architecture. This controller acts as the brain for making decisions and enforcing policies for all network devices. It maintains a global view of the network topology and is responsible for processing and responding to control messages from the switches.

2. Multi-Controller: Each multiple controllers are responsible for managing a subset of network devices or a specific domain. These controllers collaborate and coordinate their actions to ensure the overall functioning of the network. They can communicate with each other through standard interfaces and protocols.

Among the myriad of network security challenges, DDoS attacks is the most formidable and destructive threats. The impact of DDoS attacks can be highly devastating, posing significant risks to the availability, performance, and reliability of targeted systems and services.

Along with a large number of services provide by SDN, they are vulnerable to different attacks such as Network manipulation , Traffic diversion , DDoS attack etc. Effect DDoS attack on SDN network. DDoS attack on data plane switch has switch have limited flow table size. DDoS passing the number of packets into switch and switch cannot find the match and it give the request to the controller and packet consume the bandwidth of controller. DDoS attacks target a wide range of different resources and sites, posing big challenges to their managers and users. Google's Threat Analysis Group (TAG) updated its blog on October 16, 2020, regarding how threats and threat actors are changing their tactics due to the 2020 U.S. election.

In 2020, our Security Reliability Engineering team measured a record-breaking UDP ampli- fication attack sourced out of several Chinese ISPs (ASNs 4134, 4837, 58453, and 9394) which remains the largest bandwidth attack [3].

SDN allows for network design, construction, and operation. Attacks via distributed denial- of-service (DDoS) represent a serious risk to data centers. New security concerns and assaults, particularly Distributed Denial of Service (DDoS) attacks, are frequently launched against SDN networks [3].

A DDoS attack refers to a malevolent endeavor aimed at confusing the normal operations of a network, service, or website. This is achieved by inundating the target with an overwhelming influx of incoming traffic. By inundating the target with a massive volume of requests, the assaulter aims to overload its infrastructure and disrupt its normal operations.

The attack scenario can be compared to a situation where a crowd of customers congregates outside a shop, seeking entry and causing disruptions that hinder the entry of genuine customers [4].

The single architectural design of SDN makes it susceptible to DDoS attacks from multiple angles. Controller offers a centralized perspective of the network topology.it becomes vulnerable to various threats. Attackers can exploit this characteristic to manipulate the functionality of the entire SDN network simply by compromising the controller.

To address these security concerns, one approach is to adopt a multi-controller setup. In this scenario, if one controller is compromised by an attacker, another controller can seamlessly take over and continue providing network resources to users. This redundancy and distributed control help increase the flexibility and security of the SDN infrastructure against DDoS attacks. However, ensuring the security of SDN networks and protecting them from various threats remains an ongoing and active area of research. This paper concentrates on the detection of DDoS attacks in both SDN single controller and multi-controller architectures using machine learning techniques.

The review paper primarily contributes by focusing on several key aspects. It starts by identifying vulnerabilities within SDN architecture that are susceptible to exploitation by DDoS attacks, while also categorizing various types of DDoS attacks targeting SDN controllers. Researchers then delve into analyzing the potential impact of these attacks on both SDN controllers and network performance. This analysis encompasses understanding how attacks disrupt network operations, degrade service quality, and impact the efficiency of traffic management and resource allocation. Furthermore, the paper evaluates different detection and mitigation strategies specifically tailored for SDN environments, employing machine learning techniques. Lastly, it offers recommendations aimed at enhancing the security resilience of SDN controllers against DDoS threats. These suggestions may involve protocol refinements, updates to security policies, or the integration of advanced machine learning for more adaptive threat responses.

The rest of the paper organized as follows: Section 2 provide introduction to SDN working and comparison with traditional network. Section 3 discusses SDN controller with its types. Section 4 explains DDoS attacks, its impacts and its effect on SDN planes. Section

5 describes related work for DDoD attack detection in SDN controller with its mitigation approaches. Section 6 details multi-controller in SDN with its advantages. Section 7 provides potential research direction in the area of DDoS detection in SDN environment followed by conclusion and references at end.

2. Background

In the realm of network infrastructure management, traditional networking approaches have long relied on hardware-centric architectures, where the control and data plane functionalities were tightly integrated within networking devices. However, with the rapid evolution of networking demands and the rise of cloud computing, the limitations of these traditional architectures became increasingly apparent. This led to the emergence of SDN, a paradigm that decouples the control plane from the data plane, enabling centralized control and programmability of network devices through software-based controllers. This fundamental shift has revolutionized the way networks are designed, deployed, and managed, offering unprecedented flexibility, scalability, and agility to meet the dynamic requirements of modern applications and services.

2.1. Architecture of SDN

The SDN is a network architecture approach that isolates the control from the data plane in traditional networking. In SDN, there the control plane is centralised and programmable; data planes are still distributed. The main goals of SDN architecture are to enhance network flexibility, agility, and scalability by separating the control and data planes, enabling centralized control, and providing programmability as shown in Figure 1.

SDN is a revolutionary technology that transforms traditional fixed and complex networks into dynamic and programmable networks. It represents a major breakthrough in the field of networking, allowing for greater flexibility and efficiency in network management and configuration. SDN architecture devided into following components:

Application Layer. In the SDN architecture, the application plane is the initial plane were all business and security applications are managed. It is comprised of applications that use northbound APIs to communicate with the controller. These applications are tasked with delivering network services, including security, load balancing, and traffic management [5].

Northbound Interface (NBI) API. The NBI serves as the bridge between the controller and the various applications or services that operate on the controller's platform. It allows these applications to interact with the controller to configure network policies, retrieve network status information, and manage the behavior of the network.

The NBI offers a standardized collection of APIs that developers can utilize to construct applications capable of communicating to controller. These APIs abstract the intricacies of the underlying network infrastructure and offer a simplified programming model for creating network applications. The NBI enables the development of a vast range of applications of network, including network monitoring, traffic engineering, security, and virtual network services.

For example Frenetic [6], NetKAT [7], Procera [8], and FML [9] etc are commonly used Northbound Interface APIs.

Control Layer. It is the brains of the SDN architecture. The control layer is the second layer in the SDN architecture and is responsible to manage the forwarding devices in the data layer. The SDN controller functions as the central intelligence of the network, taking charge of the management. It accomplishes this through the Network Operating System (NOS), offering a comprehensive perspective of all network resources from a centralized standpoint [10].

The SDN architecture includes a centralized controller that establishes communication with the application layer via northbound APIs, as well as with the data plane through southbound APIs. The control layer is in charge for the following tasks:

- *Network topology discovery:* The controller gathers information about the topology by communicating with the switches in the data layer.
- *Network policy management:* The controller is responsible for implementing network forwarding rules and policies based on the network requirements
- *Network traffic engineering:* It can control the flow of traffic by manipulating the forwarding rules in the switches.
- *Network security:* The controller can detect and respond to security threats by implementing security policies.

NOX [11], POX [12], FloodLight [13], Ryu [14] and Beacon [15] are examples of controllers.

Southbound Interfaces (SBI) API. In the SDN architecture, the SBI designates the communication interfacing the control with the data plane. It plays a crucial role in making communication easy between the network devices and SDN controller, such as routers and routers. The commonly used SBI in SDN is the OpenFlow protocol (OFP).

OFP is used to manage the flow of network traffic by managing the forwarding tables in network switches.

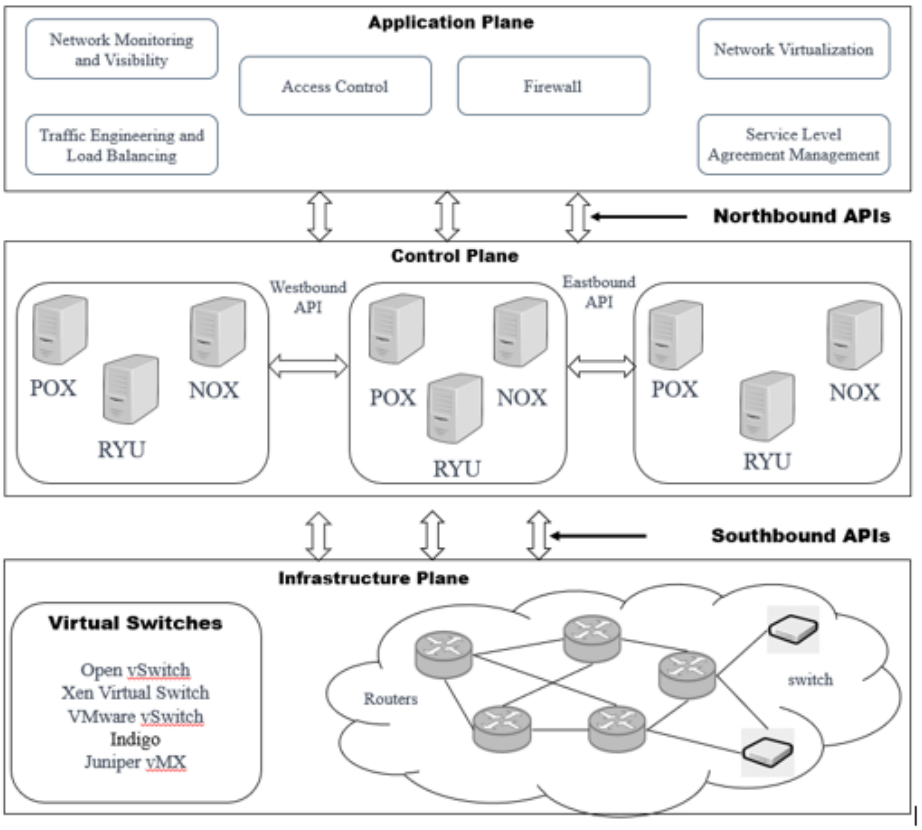


Figure 1. SDN Architecture

To manage and configure them the SDN controller establishes communication with them through the southbound interface. There are several protocols used in SDN, but the most commonly used protocol is OpenFlow. Other protocols used in SDN include NETCONF [16], PCEP [17], and Protocol-Oblivious Forwarding (POF) [18].

Data Layer. Its also called as the forwarding plane, encompasses the physical or virtual switches responsible for forwarding packets according to the flow rules installed by the controller. The data plane in SDN includes various types of network devices such as switches, routers, wireless access points, and virtual switches.

The data plane or forwarding plane is responsible for the actual forwarding of network traffic. Its primary function is to forward packets according to the policies and rules set by the SDN controller.

2.2. Working of SDN

OpenFlow is a key protocol in the realm of SDN, serving as the foundation for network programmability and centralized control. It enables the separation of the control from the data plane, allowing network administrators to dynamically manage traffic flow and implement policies through a central controller. With OpenFlow, switches and routers become simple forwarding devices, forwarding packets based on instructions received from the controller. This flexibility empowers networks to adapt rapidly to changing demands, optimize resource utilization, and enhance security by implementing fine-grained policies. In essence, OpenFlow revolutionizes network management by providing a standardized interface for configuring and managing network devices in SDN environments.

The workflow of an SDN OpenFlow switch can be summarized as follows: upon the arrival of a packet at the switch, it scrutinizes the header of packet and searches for a match within its flow table. If a matching flow entry is discovered, the switch executes the designated actions specified in the corresponding flow table entry.

In the event that packet not matched is found, then packet is transmitted to the controller for additional processing and decision-making. Once the controller has determined the appropriate actions, it installs a new entry in the switch's flow table to control similar packets in the future.

The OpenFlow switch then applies the new flow entry to subsequent packets. This process guarantees that network traffic is dynamically and efficiently managed, taking into account the policies and rules established by the SDN controller.

2.3. SDN vs. Traditional Networks

Compared with traditional networks, SDN represents a paradigm shift in network architecture. The division between the data and the control plane is one main the key differences. These two planes are tightly integrated into the networks equipment, e.g. routers and switches, where they jointly manage transport of data packets and management of network operations.

This controller provides a global view of the network, allowing for centralized control, programmability, and automation of network policies. In traditional networks, the scope for testing new policies is often. At its core, SDN encompasses several key aspects that differentiate it from traditional networking approaches is summarized in Table 1.

Control and Data Planes. In traditional networks, it is difficult to manage and scale the network because of an integrated both the planes. In SDN, the control plane is segregated from the data plane so that it can be centrally managed and programmed.

Programmability. SDN allows for more programmability and flexibility than traditional networks. Network administrators can use software to manage and configure the network, instead of relying on manual configuration of individual devices.

Security. SDN offers enhanced security features, such as the ability to prevent and detect DDoS attacks, through the use of software-defined security solutions. Traditional networks may be more vulnerable to attacks due to their reliance on distributed control planes and complex configurations.

3. SDN Controller

The paper focuses on exploring the two variations of SDN: SDN with a single controller and SDN with multiple controllers.

In the single controller architecture as shown in Figure 2a, there is a central controller in charge of governing the entire network. It maintains a overall view of the network topology, receives information from network devices, and makes decisions on how to handle network traffic. The single controller communicates with the network devices through southbound interfaces, configuring them and directing their forwarding behavior. This architecture offers a centralized control plane, simplifying network management and enabling consistent policy enforcement.

Single controller architectures are typically suitable for smaller networks or environments with less complex network configurations where the benefits of centralization outweigh the scalability and redundancy requirements.

In a multi-controller architecture as shown in Figure 2b, multiple controllers are distributed across the network, responsible for a particular domain or subset of network devices. These controllers work collaboratively to manage the network, with each controller focusing on its assigned domain. Communication between controllers allows them to exchange information, synchronize their states, and make coordinated decisions. This distributed approach enhances scalability, fault tolerance, and performance, as the workload is distributed among multiple controllers.

In a multi-controller architecture, controllers can collaborate coordinate their actions to make network-wide decisions. They can distribute network policies, synchronize flow rules, and dynamically adjust network configurations based on real-time conditions. This distributed control approach enables better load balancing, faster response times, and improved overall network performance.

The choice between a single controller and multi-controller architecture depends on various factors, such as network size, complexity, and requirements. Single controller architectures are suitable for smaller networks or deployments with simpler network policies, offering centralized control and ease of management. In contrast, multi-controller architectures are beneficial for larger networks, geographically distributed deployments, or environments requiring higher scalability, fault tolerance, and distributed decision-making capabilities.

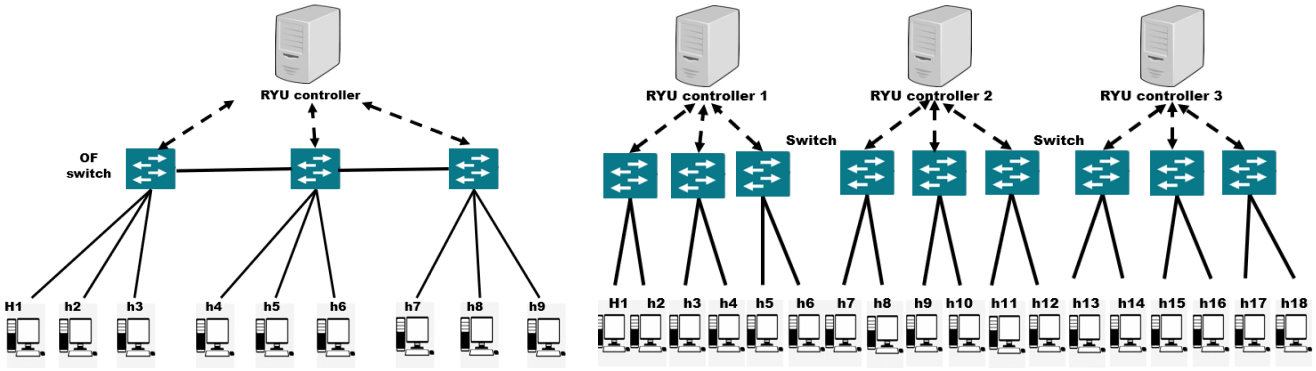
Overall, both single controller and multi-controller architectures in SDN provide flexibility, programmability, and centralized control, but they differ in terms of the scale and distribution of control within the network is summarized in Table 2.

4. DDoS

DDoS attacks are specifically designed to disrupt normal operations by inundating network devices with an overwhelming number of connection requests within a specific timeframe. The sheer volume of these malicious requests creates a burden on the target systems, causing them to experience slowdowns, crashes, or even complete shutdowns.The aim of attack is to render the targeted systems inaccessible or unresponsive to legitimate users, effectively denying them access to the services they require. By saturating the network devices and consuming their available

Table 1. Key differences between SDN and Traditional Networks

Key Differences	SDN	Traditional Networks
Control plane	Separated from data plane	Integrated with data plane
Network management	Centralized control	Distributed control
Network programming	Programmable via APIs	Not programmable
Hardware dependency	Less dependent	More dependent
Network scalability	Easily scalable	Less scalable
Traffic management	Dynamic and flexible	Static and rigid
Security	Centralized security management	Distributed security management
Network monitoring	Centralized monitoring	Distributed monitoring
Network intelligence	Intelligent and automated	Less intelligent and manual
Cost	Higher initial cost	Lower initial cost



(a) Single Controller Architecture

(b) Multi-controller Architecture

Figure 2. Single and Multi-controller Architecture

Table 2. Differences between Single Controller and Multi-Controller

Differences	Single Controller Architecture	Multi-Controller Architecture
Control Plane	Centralized control plane	Distributed control plane
Controller Nodes	Single controller instance	Multiple controller instances
Network Size	Suitable for smaller networks	Suitable for larger networks
Scalability	Limited scalability	Improved scalability
Fault Tolerance	Single point of failure	Enhanced fault tolerance
Performance	May experience performance bottlenecks	Distributed workload for improved performance
Network Complexity	Suitable for simpler network configurations	Handles complex network configurations
Decision Making	Centralized decision-making	Distributed decision-making
Management	Centralized network management	Distributed network management
Coordination	Less coordination required	Controllers collaborate for synchronization

resources, DDoS attacks disrupt the normal flow of operations and cause disruptions in service.

DDoS attacks come in various types, along with different characteristics and various methods of execution. Here are some common types of DDoS attacks:

1. Volumetric Attacks: Volumetric attacks have the objective of overwhelming the destination's network bandwidth by overloading it with an immense volume of traffic. The goal is to saturate the network capacity, making it challenging for legitimate traffic to pass through. Volumetric attacks can utilize various protocols such as ICMP floods, UDP floods, or even DNS amplification attacks.

2. Application Layer Attacks: Application layer attacks, which are also known as Layer 7 attacks, focus specifically on the application layer of a network stack. These attacks specifically target vulnerabilities in application layer protocols and exhaust server resources by sending special requests. Common application layer attacks include Slowloris attacks, HTTP floods,and Application Layer (Layer 7) DDoS attacks.

3. Protocol Attacks: In order to disrupt the target's infrastructure, protocol attacks exploit weakness in network protocols. For example, SYN floods overload the target's resources by sending a big number of SYN packets without the handshake process. ICMP floods the target with ICMP echo request packets, causing network congestion and unresponsiveness. These attacks exploit vulnerabilities in protocol implementations and can consume significant network resources.

4.1. Impacts of DDoS Attacks

DDoS attacks can have severe consequences for the targeted entities. These include:

1. Service Disruption: A DDoS attack aims to disrupt the availability of services, making it impossible for authorized users. This can result in financial losses, customer dissatisfaction, and reputational damage.

2. Network Congestion: The high volume of malicious traffic generated during a DDoS attack can congest network resources, impacting the overall network performance and causing collateral damage to other systems and services.

3. Loss of Revenue: Online businesses heavily rely on continuous availability. DDoS attacks can lead to loss of sales, ad revenue, or other monetary transactions, directly impacting the financial health of organizations.

4. Damage to Reputation: Successful DDoS attacks can tarnish the reputation of targeted organizations, eroding trust and credibility among customers, partners, and stakeholders.

5. Costly Mitigation Efforts: Organizations must invest in robust mitigation measures and DDoS protection solutions to counteract attacks, which can be financially burdensome.

4.2. Effect of DDoS on SDN Planes

DDoS attacks can have significant impacts on the different planes of an SDN architecture. SDN separates the control plane, data plane, and management plane, each of which can be affected differently by DDoS attacks. Here's an overview of the effects of DDoS on each plane:

1. Control Plane: It is for managing and controlling the network devices and their behavior. DDoS attacks targeting the control plane can disrupt the communication done between the network devices and the SDN controller, leading to a loss of centralized control and management. This can result in network instability, misconfiguration, or even a complete network outage. Attackers may overload the control plane by flooding the SDN controller with a large volume of traffic or by exploiting vulnerabilities in the control plane protocols.

2. Data Plane: It handles the forwarding traffic of network based on the instructions received from the SDN controller. DDoS attacks on the data plane aim to overwhelm the network switches or routers, impacting their processing capabilities and forwarding performance. By flooding the data plane with excessive traffic, attackers can cause congestion, packet drops, or resource exhaustion, leading to degraded network performance and potential service disruptions.

3. Application Plane: It is responsible for network configuration, monitoring, and maintenance. DDoS attacks targeting the management plane can disrupt network management functions, making Network administrators face challenges in effectively monitoring and controlling the network. These attacks can prevent administrators from accessing management interfaces, logging into network devices, or modifying network configurations, impeding effective mitigation and response to the attack.

4.3. DDoS Attack on SDN

SDN aims to enhance network control by facilitating quick responses to changing business needs. Conversely, the primary aim of DDoS attacks is to disrupt ongoing operations by overburdening network devices with a high volume of connection requests for a specific duration.

DDoS attacks can take different forms and target various entities for various reasons. For instance, some attacks may be aimed at disrupting the services of specific organizations or websites, while others may seek to steal sensitive data or extort money from victims. Moreover, as the internet continues to evolve, new types of DDoS attacks are emerging, making it a constantly evolving threat in the field of cybersecurity.

Volumetric Attack. Excessive traffic used to prevent genuine users from accessing by overwhelming a network are known as volumetric attacks. The production of many new flows which flood the control plane's bandwidth, OpenFlow switch and SDN controller will allow volumetric attacks to be performed.

Attackers often uses spoofed IP addresses which are not there in rules in an OpenFlow switch flow table. As a consequence, a table-miss scenario arises, leading the switch to generate an extensive influx of packet-in messages directed towards the SDN controller from the affected OpenFlow switch. This activity consumes memory, communication bandwidth, and CPU resources in both the data and control planes of SDN.

There are numerous types of volumetric attacks in SDN, including UDP floods, ICMP floods, and TCP SYN floods.

UDP flood attack occurs by sending huge number of UDP packets to random victim system port. Upon receiving these UDP packets, the victim system tries to identify the application that is expecting data on the destination port [19].

UDP floods occur when a big number of UDP packets are forwarded to the target device or network, overwhelming the device's resources.

It is a type of DDoS attack that floods the victim with large volumes of UDP packets. The objective of these attacks is to overload the victim's network with a large volume of traffic, which will result in decreased performance or total unavailability. A specific type of UDP flood attack is the DNS amplification attack. In this attack, the attacker falsifies the source IP address to match that of the victim and forward a small request to a DNS server. As a result, the DNS server sends a significantly larger response to the victim, potentially causing a substantial impact on the victim's network [20].

In ICMP flood attack the attacker generates an overwhelming flood of ICMP ECHO packets sent at the victim system [19]. During a Smurf attack, which is a type of ICMP flooding attack, the victim system responds to each ICMP request, resulting in the consumption of its CPU and network resources [19].

ICMP floods send a large number of ICMP packets to the target device or network, which can cause it to slow down or crash.

Application Layer Attack. Two commonly encountered types of application layer attacks are HTTP floods and SMTP floods. In order to facilitate the loading of websites and the transmission of forms content over the Internet, HTTP is a foundation for browser based Internet requests.

Some HTTP-based slow attacks [21] are as follows:

The HTTP header is broken down to several packets when attacked with a "Slow HTTP Header" attack, also known as 'Slowloris. The attacker then sends these fragmented headers to the server at a very slow rate, which results in an interrupted target service.

In a Slow HTTP POST attack, commonly referred to as RUDY (R-U-Dead-Yet), the body of the POST message is fragmented into multiple packets and transmitted to the server at a deliberately low rate, leading to a potential denial-of-service condition.

A slow read attack occurs when an attacker makes the server routine HTTP requests but purposefully waits to receive the server's response. This delayed response from the server can cause disruption and potential denial-of-service impacts.

5. DDoS Attack Detection in SDN Controller

DDoS attacks in SDN controllers involves leveraging the controller's centralized visibility and control over network traffic. The controller continuously monitors incoming traffic flows, analyzing various parameters such as packet rates, flow durations, and packet sizes. Deviations from normal traffic behavior are indicative of potential DDoS attacks. Additionally, anomaly detection algorithms, such as statistical analysis or machine learning models, can be implemented within the controller to detect unusual patterns or spikes in traffic volume. Furthermore, SDN controllers can collaborate with switches and routers to implement flow-based filtering policies, dynamically rerouting or dropping malicious traffic during an attack. Real-time communication between the controller and network devices enables rapid response and mitigation actions, enhancing the network's resilience against DDoS attacks. By centralizing detection and response mechanisms, SDN controllers play a crucial role in safeguarding network resources and ensuring uninterrupted service delivery in the face of DDoS threats.

Shideh et al. [22] presents a novel approach for detecting DDoS attacks in SDN networks using Support Vector Machines (SVM) and the Ryu SDN controller. The study focuses on addressing the challenge of identifying and mitigating DDoS attacks in the SDN environment, where traditional detection methods may not be directly applicable. The authors propose an SVM-based detection model that leverages features pluck out from network traffic data to train and classify normal and attack traffic. The results shows that current method is effective in accurately detecting DDoS attacks while maintaining less false positive rates. The study provides insight into the use of machine learning techniques, in particular SVM, to enhance security and resilience for SDN networks against DDoS attacks with a view to contributing to existing literature.

Abbas et al. [23] present a collaborative method for detecting and containing DDoS flooding attacks in SDN. The study addresses the challenge of effectively detecting and mitigating DDoS attacks, which can cause significant d isruptions t o n etwork services. The authors propose a collaborative framework that strengthens the programmability and centralized control of SDN to facilitate coordinated detection and response actions across multiple network domains. The approach involves monitoring network traffic using flow s tatistics a nd a pplying m achine l earning techniques for traffic cl assification. The pro posed framework collaboratively activates mitigation strategies, such as rate limiting, and traffic di version, to contain the attack. The experimental evaluation demonstrates the effectiveness of the collaborative approach in detecting and mitigating DDoS flooding a ttacks, thereby enhancing the overall security and resilience of SDN networks.

Hock et al. [24] aim at detecting DDoS attacks via ML techniques in the SDN domain. In order to ensure the availability and security of network services, the study addresses the increasing threat of DDoS attacks and the need for effective d etection m echanisms. T he authors explore the application of ML algorithms, such as SVM and Random Forest, for identifying DDoS attacks in SDN environments. They evaluate the performance of these algorithms using different f eature s ets extracted from network traffic da ta. Th e ex perimental results demonstrate the efficacy of th e pr oposed machine learning-based approach in accurately detecting DDoS attacks with high detection rates and low false-positive rates.

Kshira et al. [25] offer a machine learning approach for the prediction of DDoS traffic in SDN. The study addresses the growing need for proactive DDoS mitigation strategies by leveraging machine learning methods to accurately identify and predict DDoS attacks. The authors propose a framework that combines statistical features extracted from network traffic data with different machine learning algorithms, including Decision Trees and Support Vector Machines (SVM). Using real world network traffic data, they assess the performance of the proposed approach and compare it to traditional detection methods. The results of the trials show that machine learning techniques are effective when it comes to accurate prediction of DDoS traffic, with a large accuracy rate. The study contributes to the existing literature by providing insights into the application of machine learning techniques for DDoS traffic prediction in SDN, enabling network administrators to take proactive measures to mitigate potential DDoS attacks and enhance the security of their networks.

T. Chin et al. [23] propose an innovative approach for detecting and containing DDoS flooding attacks in SDN. The study addresses the need for efficient and effective mechanisms to mitigate and detect DDoS attacks, which can pose serious threats to network availability and performance. The framework integrates various techniques, including statistical analysis, machine learning, and traffic engineering algorithms, to detect abnormal traffic patterns and promptly respond to mitigate the attack. The effectiveness of the collaborative approach to determine and contain flooding DDoS attacks, while minimising impact on legitimate traffic, has been demonstrated by a trial evaluation carried out with network traffic data from actual world networks.The research adds to the current pool of knowledge by introducing a fresh method that utilizes SDN principles and cooperative methods to boost the identification and control abilities for DDoS attacks. This, in turn, enhances the general security and durability of network systems.

K. S. Sahoo et al. [26] propose a module designed to implement detection mechanisms for transport layer and application layer DDoS attacks within the context of SDN architecture. The module is flexible and adaptable to SDN environments, leveraging the capabilities of an ONOS controller. The study utilizes two datasets, namely CICDoS2017 and CICDDoS2019, to check the performance of the proposed work. The results indicate high detection rates of up to 95 highlighting the effectiveness of the module in identifying and flagging potential DDoS attacks at the application and transport layers. However, the paper falls short in terms of scalability as it does not provide extensive scalability analysis or discuss the potential limitations of the proposed module in handling larger and more complex network scenarios. Additionally, the paper lacks a mitigation module, focusing primarily on detection rather than offering comprehensive solutions for DDoS attack mitigation. Nonetheless, the study contributes to the field of SDN-based DDoS attack detection by presenting an approach that targets specific layers and achieves promising detection performance.

Anupama Mishra et al. [12] propose defense mechanisms for mitigating DDoS attacks in an SDN-cloud environment. The study focuses on utilizing entropy-based techniques to detect and prevent DDoS attacks. The POX controller is employed as the control plane for implementing the defense mechanisms. The paper presents various entropy-based algorithms and evaluates their effectiveness in mitigating and identifying DDoS attacks. The results demonstrate that the proposed mechanisms effectively detect and mitigate DDoS attacks, enhancing the resilience and security of the SDN-cloud infrastructure. The study contributes to the field by application of entropy-based techniques for DDoS defense in SDN environments, providing valuable information for

network administrators and researchers working on DDoS attack mitigation strategies.

Mohammed et al. [27] provide a compre-hensive overview of defense mechanisms employed to counter DDoS flooding attacks. The paper highlights the significance of DDoS attacks as a major threat to network security and analyzes various defense techniques and strategies. The survey categorizes the defense mechanisms into four main categories: prevention-based, detection-based, response-based, and hybrid approaches. The paper discusses the strengths and limitations of each approach and provides insights into their effectiveness a nd p racticality. A dditionally, the survey explores research directions and future challenges in the field of DDoS defense.

5.1. DDoS Detection using Machine Learning

Machine learning algorithms have emerged as effective tools for addressing complex problems, including the detection of DDoS attacks. ML based classifiers o ffer a higher level of performance than conventional signature based detection techniques. In order to identify abnormal network traffic be havior, th ese algorithms may be trained with greater accuracy. Several commonly used classifiers i n t he c ontext o f D DoS attack detection include SVM, K-nearest neighbor (KNN), Hidden Markov Model (HMM), Random Forest, Decision Tree (J48), Naive Bayes, Advanced Support Vector Machine, Logistic Regression, Binary Bat Algorithm, and Random Trees. These effective a lgorithms have been widely used by researchers to identify DDoS attacks, as shown by the findings t hat a re d escribed in Table 3 as follows.

Table 3 summarizes the outcomes of employing different m achine l earning t echniques a nd databases for the detection of DDoS attacks in SDN single controller environments. Accuracy rates for each method were evaluated based on the respective datasets used.

5.2. Limitations of DDoS Attack Detection

In a single controller-based SDN architecture, there are limitations when it comes to the detection of DDoS attacks. Here are some of the key limitations:

1. Limited Resources: A single controller has finite resources, including processing power and memory capacity. DDoS attacks can generate a big volume of traffic that overwhelms the resources of the controller, making it difficult to mitigate and detect the attack effectively. The limited resources can result in delays or inaccuracies in detecting and responding to DDoS attacks.

2. Single Point of Failure: In a controller-based architecture, the controller serves as the central control hub for the whole network. This central control in SDN architecture creates a vulnerability with a single point of failure. If it is compromised or suffers a DDoS attack that disables it, the entire network becomes susceptible, hindering the effectiveness of attack detection and mitigation. Relying on a single controller heightens the chances of disruptions and downtime.

3. Scalability Challenges: DDoS attacks can generate a vast amount of traffic that is processed and analyzed for effective detection. In a single controller-based architecture, scaling the resources to handle large-scale DDoS attacks can be challenging. The limited scalability can result in performance bottlenecks, delays, or even false negatives in DDoS detection, allowing the attack to bypass detection mechanisms.

4. Limited Visibility: DDoS attacks can target specific network segments or individual devices within the network. In a single controller-based architecture, the visibility into individual network elements and their traffic patterns may be limited. This lack of granular visibility can make it harder to accurately detect and isolate DDoS attacks, as the controller may not have sufficient information about the traffic patterns and behavior of individual network elements.

5. Slow Detection and Mitigation: mitigating and Detecting DDoS attacks in real-time is crucial to minimize their impact. However, in a single controller-based architecture, the process of responding and detecting to DDoS attacks can be slower due to the centralized nature of control. The time needed for the controller to receive, analyze, and respond to the attack traffic can result in significant delays, allowing the attack to cause damage before countermeasures are deployed.

To overcome these limitations, alternative approaches can be considered, such as multi-controller architectures or hybrid models that combine centralized control with distributed elements. These approaches distribute the control and detection mechanisms across multiple controllers or network devices, enabling better scalability, improved resilience against attacks, and enhanced detection capabilities. By leveraging distributed resources and decentralized decision-making, these architectures can provide more effective DDoS attack detection and mitigation in SDN environments.

Table 3. DDoS attack detection using machine learning method in SDN environments

Reference	ML method used	Controller	Dataset used	Disadvantages
Hüseyin Polat et al.[28]	KNN,ANN,DT, SVM	Single	InSDN	Additionally, the proposed method needs the large amount of network traffic data, which may be challenging in some environments.
Shi Dong and Mudar Sarem[29]	KNN algorithm	Single	custom dataset	The performance may be sensitive to the choice of hyperparameters, which could require significant tuning to achieve optimal performance.
Lingfeng Yang and Hui Zhao[30]	SVM algorthim	Single	KDD99	The single Data set is used for evaluation and it is not clear how the method would perform on other datasets.
V.Deepa et al.[31]	SVM-SOM	Single	custom dataset	The work is not compared with other existing approaches for detection of DDoS attack in SDN.
KM Sudar et al.[32]	DT,SVM	Single	KDD99	Limited visibility: Machine learning algorithms rely on the input data provided to them. If the data is incomplete or inaccurate, the algorithm's ability to detect attacks is limited.
S Haider et al.[33]	RNN,LSTM, CNN	Single	CICIDS2017	Attackers may try to bypass detection systems by deliberately crafting attacks that evade detection by machine learning models. Adversarial attacks can be addressed through the use of robust training methods and feature engineering techniques.
NN Tuan et al.[34]	KNN	Single	CAIDA and Custom	The effectiveness of work may be limited by the quantity and quality of the training data used for machine learning.The proposed scheme may not be effective against new or sophisticated DDoS attacks that are not covered by the training data.
L Tan[35]	K-Means and KNN	Single	NSL-KDD	The proposed framework uses machine learning algorithms for DDoS detection, which may impose a significant computational overhead on the SDN controller. This overhead may affect the overall performance and responsiveness of the SDN network.
K Naik et al.[36]	SVM	Single	NSL-KDD and DDoS	It not be scalable to large-scale SDN networks with a large number of controllers and switches, as the training process may become computationally expensive.he proposed method uses a limited set of features to represent network traffic, which may not capture all the relevant information needed for accurate DDoS detection.
F Musumeci et al.[37]	RF, KNN, SVM and ANN	Single	Real time dataset	The proposed work is limited to SYN DDoS flood attacks. Also training and testing is done on small dataset.
Sangodoyin et al.[38]	DA, KNN, DT and NB	Single	Custom dataset	It was trained with the default settings of hyper-parameters or control parameters.
Polat et al.[39]	SVM, NB, ANN and KNN	Single	Custom dataset	The ML classifiers achieve relatively low performance along with detection accuracy and the proposed approach's false positive rate is not discussed.

6. Multi-Controller in SDN

Using a single controller to detect DDoS assaults in SDN was covered in the previous Section 5 description. However, there are several defense mechanisms available to prevent and detect DDoS attacks in SDN. These mechanisms include flow-based anomaly detection, traffic sampling and analysis, threshold-based detection, and machine learning-based approaches.

Single controller architectures in SDN networks have several limitations. One major drawback is their vulnerability to DDoS attacks, as these attacks can overwhelm the controller and render it unable to manage the network effectively. Additionally, single controller architectures may struggle to handle large-scale networks with high traffic volumes, resulting in poor network performance and reduced scalability.

Using multiple controllers in an SDN network can also enhance DDoS defense by distributing the traffic load and enabling the network to quickly respond to any attack. Moreover, applying a combination of different defense mechanisms can improve the accuracy and efficacy of DDoS mitigation and detection.

The benefits of a multi-controller SDN architecture include:

- *Scalability:* With multiple controllers, the SDN network can scale better, handling more switches and flows without overloading a single controller.
- *Resilience:* If one controller fails, the other controllers can take over, ensuring that the network remains available and operational.
- *Load Balancing:* Multiple controllers can balance the load, distributing the work between them, and improving overall network performance.
- *Geographic Distribution:* Multi-controller SDN architectures can be deployed across different geographic locations, providing redundancy, and minimizing latency.
- *Better Resource Utilization:* With multiple controllers, different applications and services can be assigned to specific controllers, allowing for better resource utilization and more efficient management of the network.

A survey paper by Tao Hu et al. [40] is dedicated to the exploration of multi-controller based SDN, a networking paradigm that enables the separation of control and data planes, enhancing network flexibility and programmability. The paper delves into the architecture, advantages, and challenges associated with multi-controller based SDN. It also presents an overview of the existing literature on multi-controller based SDN, including various approaches and techniques used in research. The paper highlights the advantages of using multiple controllers in SDN, such as fault tolerance and scalability, while also discussing the challenges of ensuring consistency and synchronization between the controllers. Overall, this survey provides a comprehensive and informative overview of multi-controller based SDN, its benefits, and its challenges.

The design principles for multiple controllers can be analyzed from three key aspects: fault tolerance, consistency and availability.

By examining these aspects, one can identify the key considerations and best practices for designing and implementing a robust and reliable multi-controller SDN network.

6.1. Consistency

Consistency refers to the degree to which network controllers maintain a consistent view of network topology and forwarding rules. This is essential for ensuring that traffic is correctly and efficiently routed through the network. To achieve consistency, the controllers must communicate regularly and effectively with each other, and use consistent protocols and algorithms for managing the network.

Fetia Bannour et al. [41] suggest a self-adjusting consistency model designed for distributed SDN controllers, capable of effectively upholding consistency while adjusting to the network's specific attributes. It is based on the eventual consistency model and uses a set of mechanisms to achieve self-adaptivity. These mechanisms include conflict detection and resolution, adaptive quorum selection, and threshold-based consistency enforcement. The proposed model is evaluated through simulations and the results show that it can provide efficient and effective consistency management for distributed SDN controllers. The paper highlights the importance of consistency in distributed SDN controllers and presents a practical approach to achieve it.

Aslan et al. [42] utilized clustering methodologies, such as incremental K-means and sequential K-means, to construct a tailored consistency model adaptable to the precise needs of the network. The proposed tunable consistency model was evaluated using simulations, and the results showed that it could effectively manage consistency while adapting to changes in the network. This approach provides a practical solution for managing consistency in distributed SDN controllers while addressing the challenges posed by network dynamics.

Within the realm of SDN with multiple controllers, consistency typically encompasses three key aspects: version update consistency, state consistency, and rules update consistency.

6.2. Fault Tolerance

Fault tolerance refers to the capacity of a network to operate when one or more controllers fail or become disabled. To achieve fault tolerance, the network must be designed with redundancy and failover mechanisms, so that if one controller fails, another can take over its responsibilities without disrupting the network's operations.

Koponen et al. [43] introduces Onix, a distributed control platform designed for managing large-scale production networks. Onix utilizes a distributed publish-subscribe (pub-sub) messaging system, enabling flexible and efficient network programming. The platform is designed to handle the scale and complexity of modern networks, with features such as a distributed data plane and fine-grained control over network forwarding behavior. The authors describe the architecture and implementation of the platform, and evaluate its performance and scalability through a series of experiments.

Yuan Zhang et al. [44] The utilization of reactive fault management mechanisms in SDN can impose computational overhead on controllers and lead to extended recovery times. To enhance fault tolerance, a combination of preventive protection and recovery mechanisms can be employed. Preventive mechanisms involve the backup of routes or common fault messages, enabling quicker recovery times. On the other hand, recovery mechanisms implemented after a fault may introduce computational burdens and longer recovery durations.

6.3. Availability

Availability means the capability of the network to provide uninterrupted service to its users. This requires ensuring that the controllers are always available and responsive, and that they can handle the volume of traffic and requests that are generated by the network. To achieve high availability, the network must be designed with load balancing, scalability, and other mechanisms that can distribute the workload across multiple controllers and resources.

There are three methods available to improve controller availability:

1. **Rule Backup:** Dumitras et al. [45] proposed approach for achieving high availability in RuleBricks revolves around the concept of rules backup. In this system, each rule is represented by a specific color of "bricks," with the top bricks signifying the currently active rules. If a network node fails and certain colored bricks are lost, RuleBricks will replace them with new active rules. Through efficient management of brick selection and operation, RuleBricks prevents flow redistribution and excessive rule creation, ensuring smooth network operation.

2. **Controller Load Balancing:** S Mukherjee et al [46] proposed solution, known as Load-Constrained Control (LCC), seeks to dynamically manage traffic distribution among controllers by regularly monitoring the load window. As the load window fluctuates, LCC dynamically adjusts the controller pool size to align with the current demand. In instances where the load surpasses the maximum capacity of the controller pool, LCC incorporates additional controllers to ensure high availability and efficient network management.

3. **Proactive Rule Setup:** A. R. Curtis et al [47] proposed DevoFlow is a technique that categorizes network flows into two types: short and long. The switch applies different rules to handle these flows. Short flows are processed in the data plane directly, without requiring controller intervention. Only a small number of long flows are sent to the controller. By reducing the number of flows that require controller processing, DevoFlow minimizes the controller load and enhances controller availability.

A multi-controller platform has been developed to improve both the consistency and scalability of the network in summery table 4. By introducing multiple controllers, the platform aims to distribute the control plane functions across different entities, allowing for better fault tolerance and load balancing. The multi-controller platform offers enhanced scalability and reliability, providing a more robust and efficient network infrastructure.

6.4. Advantages of Multi-Controller

1. Scalability and High Availability: Multi-controller architecture enables better scalability and high availability compared to a single-controller architecture. By distributing the control plane across multiple controllers, the network can handle larger scale deployments and increased traffic volumes more effectively. The redundancy provided by multiple controllers ensures fault tolerance and minimizes the impact of a controller failure on the overall network.

2. Improved Performance: With multiple controllers, the processing load can be distributed, resulting in improved performance. Controllers can handle network events and updates in parallel, reducing processing bottlenecks and enabling faster response times. This distributed approach allows for better

Table 4. Summary table of Multi-Controller

Reference	Consistency	Scalability	Controller Type	Limitations
T Koponen et al.[48]	Consistent state replication across controllers	Scalable to large-scale networks	Onix distributed control platform	Lack of specific scalability metrics mentioned
Vignesh Sridharan et al.[49]	Consistent traffic engineering policies	Scalable to large-scale networks	Floodlight and Ryu controllers	Lack of specific evaluation metrics for consistency and scalability, limited to two controller platforms
Stephanos Matsumoto et al.[50]	Ensures consistent network policies and updates	Scalable to large SDN deployments	Custom controller (Fleet Controller)	Limited evaluation of scalability with only up to 20 switches and 160 hosts, and limited deployment of the system in a small testbed environment.
J Wang et al.[51]	Ensures consistency in multi-domain SDN deployments	Designed for scalable SDN architectures	Coordinate Controller	Limited evaluation and deployment in a specific multi-domain SDN scenario, scalability needs to be evaluated in larger and diverse network environments.
MN Yusuf et al.[52]	Consensus-based controller placement	Scalable to large-scale networks	Multiple controllers (not specific)	No specific scalability evaluation metrics mentioned
F Bannour et al.[53]	Introduces a self-adaptive consistency model	Focuses on improving scalability of distributed SDN	Distributed Controllers	Limited evaluation and validation of the proposed self-adaptive consistency model, scalability needs to be evaluated in larger distributed SDN deployments.
Abubakar Siddique Muqaddas et al. [54]	Focuses on supporting consistency in ONOS clusters	Addresses scalability considerations	ONOS cluster	The study may have specific limitations related to the experimental setup, deployment scenario, or scalability limitations specific to the ONOS platform.

resource utilization and more efficient handling of network operations.

3. Enhanced Fault Isolation: In a multi-controller architecture, failures or issues in one controller do not affect the entire network. The fault can be isolated to specific controller domains, allowing for easier troubleshooting and containment of issues. This isolation ensures that disruptions are localized and do not impact the overall network's functionality.

4. Load Balancing and Traffic Optimization: Multiple controllers can collaborate to distribute network traffic and balance the load across the network. By dynamically assigning tasks and traffic to different controllers, the architecture can optimize resource utilization, reduce congestion, and improve overall network performance.

5. Flexibility and Modularity: Multi-controller architecture offers greater flexibility in network design and deployment. It allows for modular expansion and the addition of new controllers as the network grows. This flexibility supports the scalability requirements of modern networks and facilitates the integration of new technologies and services.

7. Emerging Research Directions

The research on machine learning methods for detecting DDoS attacks, in single controller and multi-controller architectures, represents a prominent and evolving direction in the field of SDN. Despite considerable advancements in DDoS detection methods, there exists a noticeable gap in the current literature when it comes to applying machine learning algorithms within the context of SDN multi-controller environments.

Current work focuses on detection of DDoS attack in both single-controller and multi-controller SDN

environments. We analyze various machine learning techniques to identify effective methods for detecting DDoS attacks in SDN networks. In the context of a single controller, several existing methods are available for DDoS attack detection. However, when it comes to a multi-controller SDN setup, there is a gap in the existing literature, and there are limited methods available for effectively detecting DDoS attacks.

In a multi-controller environment, there are indeed numerous research papers available that focus on various aspects related to consistency, reliability, and fault tolerance. These topics are crucial in ensuring the robust and efficient operation of SDN architectures with multiple controllers.

Consistency in multi-controller SDN environments refers to the synchronization of the network state and policies across all controllers. Research papers explore techniques for achieving strong consistency to ensure that all controllers have a consistent view of the network topology and forwarding rules.

Reliability is critical to maintain the continuous operation of the SDN network, even in the presence of failures. Research papers in this area investigate fault-tolerant mechanisms to handle controller failures and ensure network resilience.

Fault tolerance techniques aim to prevent or recover from errors and failures in the network. This involves developing mechanisms to detect faults, isolate affected components, and ensure that the network remains operational.

Research topics in SDN multi-controller environments cover a broad spectrum of challenges and opportunities in distributed network management. Here are some potential research topics in this area:

Collaborative DDoS Attack Detection: Develop efficient and accurate collaborative DDoS attack detection mechanisms among multiple controllers in SDN to improve detection and mitigation capabilities.

Load Balancing and Resource Management: Investigate load balancing techniques and resource management strategies to distribute the control plane load evenly among multiple controllers, ensuring efficient resource utilization and high network performance.

Controller Placement and Scalability: Investigate optimal controller placement strategies and scalable designs to efficiently manage large-scale SDN deployments.

Research topics in SDN multi-controller using machine learning combine the advancements of machine learning with the complexities of distributed control in SDN environments. Here are some potential research topics in this area:

Distributed Traffic Classification: Investigate distributed machine learning algorithms for traffic classification in multi-controller SDN networks, enabling efficient and accurate identification of different traffic types.

Federated Learning in SDN: Explore federated learning approaches to train machine learning models across multiple controllers without centralizing data, ensuring data privacy and reducing communication overhead.

Auto-scaling and Elasticity: Explore machine learning techniques to enable auto-scaling and elasticity of controller resources to handle varying workloads in multi-controller SDN setups.

These research topics aim to leverage the capabilities of machine learning to enhance the intelligence and efficiency of SDN multi-controller environments, leading to more reliable, scalable, and secure network management.

This paper primarily focuses on tackling the issue of DDoS attacks within SDN environments. These attacks present a substantial threat to network availability and performance by inundating network resources with a large volume of traffic.

The paper focuses on exploring and evaluating various techniques for DDoS attacks detection in SDN. These techniques may include traditional machine learning algorithms, rule-based methods, and collaborative approaches among multiple SDN controllers.

By analyzing and comparing different detection techniques, the paper aims to provide the limitations and strengths of each approach and guide network administrators and security practitioners in choosing the most suitable method for their specific SDN deployment.

In summary, the primary focus of the paper is to study detection of DDoS attack in SDN environments and present a comprehensive analysis of the available techniques to improve network security and resilience against DDoS threats.

8. Conclusion

This survey paper offers an extensive and detailed examination of the present advancement in DDoS attack detection within SDN multi controller as well as single controller architectures. By evaluating existing approaches, tackling challenges, and proposing future research directions, the aim is to encourage further progress in this crucial field of network security. We can strengthen the resilience and security of SDN networks against distributed denial of service attacks and ensure the uninterrupted delivery of network services through continued efforts and innovative solutions.

References

[1] Cui, Y., Qian, Q., Guo, C., Shen, G., Tian, Y., Xing, H. and Yan, L. (2021) Towards ddos detection mechanisms

in software-defined networking. *Journal of Network and Computer Applications* **190**: 103-156.

[2] Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S. *et al.* (2013) B4: Experience with a globally-deployed software defined wan. *ACM SIGCOMM Computer Communication Review* **43**(4): 3–14.

[3] Dayal, N., Maity, P., Srivastava, S. and Khondoker, R. (2016) Research trends in security and ddos in sdn. *Security and Communication Networks* **9**(18): 6386–6411. doi:https://doi.org/10.1002/sec.1759, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1759.

[4] Wood, A.D. and Stankovic, J.A. (2002) Denial of service in sensor networks. *computer* **35**(10): 54–62.

[5] Singh, J. and Behal, S. (2020) Detection and mitigation of ddos attacks in sdn: A comprehensive review, research challenges and future directions. *Computer Science Review* **37**: 100279.

[6] Foster, N., Harrison, R., Freedman, M.J., Monsanto, C., Rexford, J., Story, A. and Walker, D. (2011) Frenetic: A network programming language. *ACM Sigplan Notices* **46**(9): 279–291.

[7] Anderson, C.J., Foster, N., Guha, A., Jeannin, J.B., Kozen, D., Schlesinger, C. and Walker, D. (2014) Netkat: Semantic foundations for networks. *Acm sigplan notices* **49**(1): 113–126.

[8] Voellmy, A., Kim, H. and Feamster, N. (2012) Procera: a language for high-level reactive network control. In *Proceedings of the first workshop on Hot topics in software defined networks*: 43–48.

[9] Khan, S., Gani, A., Wahab, A.W.A., Abdelaziz, A. and Bagiwa, M.A. (2016) Fml: A novel forensics management layer for software defined networks. In *2016 6th international conference-cloud system and big data engineering (confluence)* (IEEE): 619–623.

[10] Gude, N., Koponen, T., Pettit, J., Pfaff, B., Casado, M., McKeown, N. and Shenker, S. (2008) Nox: towards an operating system for networks. *ACM SIGCOMM computer communication review* **38**(3): 105–110.

[11] Priya, A.V. and Radhika, N. (2019) Performance comparison of sdn openflow controllers. *International Journal of Computer Aided Engineering and Technology* **11**(4-5): 467–479.

[12] Mishra, A., Gupta, N. and Gupta, B. (2021) Defense mechanisms against ddos attack based on entropy in sdn-cloud using pox controller. *Telecommunication systems* **77**: 47–62.

[13] Daha, M.Y., Zahid, M.S.M., Husain, K. and Ousta, F. (2021) Performance evaluation of software defined networks with single and multiple link failure scenario under floodlight controller. In *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)* (IEEE): 959–965.

[14] Chouhan, R.K., Atulkar, M. and Nagwani, N.K. (2022) A framework to detect ddos attack in ryu controller based software defined networks using feature extraction and classification. *Applied Intelligence* : 1–21.

[15] Manuel, T. and Goswami, B.H. (2019) Experimenting with scalability of beacon controller in software defined network. *International Journal of Recent Technology and Engineering* **7**(5S2): 550–555.

[16] Dallaglio, M., Sambo, N., Cugini, F. and Castoldi, P. (2017) Control and management of transponders with netconf and yang. *Journal of Optical Communications and Networking* **9**(3): B43–B52.

[17] Kukreja, N., Alvizu, R., Kos, A., Maier, G., Morro, R., Capello, A. and Cavazzoni, C. (2016) Demonstration of sdn-based orchestration for multi-domain segment routing networks. In *2016 18th International Conference on Transparent Optical Networks (ICTON)* (IEEE): 1–4.

[18] Song, H. (2013) Protocol-oblivious forwarding: Unleash the power of sdn through a future-proof forwarding plane. In *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*: 127–132.

[19] Deepa, V., Sudar, K.M. and Deepalakshmi, P. (2019) Design of ensemble learning methods for ddos detection in sdn environment. In *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)* (IEEE): 1–6.

[20] Zhang, B., Zhang, T. and Yu, Z. (2017) Ddos detection and prevention based on artificial intelligence techniques. In *2017 3rd IEEE International Conference on Computer and Communications (ICCC)* (IEEE): 1276–1280.

[21] Gupta, B.B., Joshi, R.C. and Misra, M. (2009) Defending against distributed denial of service attacks: issues and challenges. *Information Security Journal: A Global Perspective* **18**(5): 224–247.

[22] Mehr, S.Y. and Ramamurthy, B. (2019) An svm based ddos attack detection method for ryu sdn controller. In *Proceedings of the 15th international conference on emerging networking experiments and technologies*: 72–73.

[23] Chin, T., Mountrouidou, X., Li, X. and Xiong, K. (2015) An sdn-supported collaborative approach for ddos flooding detection and containment. In *MILCOM 2015-2015 IEEE Military Communications Conference* (IEEE): 659–664.

[24] Nadeem, M.W., Goh, H.G., Ponnusamy, V. and Aun, Y. (2022) Ddos detection in sdn using machine learning techniques. *Comput. Mater. Contin.* **71**(1): 771–789.

[25] Sahoo, K.S., Iqbal, A., Maiti, P. and Sahoo, B. (2018) A machine learning approach for predicting ddos traffic in software defined networks. In *2018 International Conference on Information Technology (ICIT)* (IEEE): 199–203.

[26] Yungaicela-Naula, N.M., Vargas-Rosales, C. and Perez-Diaz, J.A. (2021) Sdn-based architecture for transport and application layer ddos attack detection by using machine and deep learning. *IEEE Access* **9**: 108495–108512.

[27] Zargar, S.T., Joshi, J. and Tipper, D. (2013) A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. *IEEE communications surveys & tutorials* **15**(4): 2046–2069.

[28] Tonkal, Ö., Polat, H., Başaran, E., Cömert, Z. and Kocaoğlu, R. (2021) Machine learning approach equipped with neighbourhood component analysis for ddos attack detection in software-defined networking. *Electronics* **10**(11): 1227.

[29] Dong, S. and Sarem, M. (2019) Ddos attack detection method based on improved knn with the degree of

ddos attack in software-defined networks. *IEEE Access* **8**: 5039–5048.

[30] Yang, L. and Zhao, H. (2018) Ddos attack identification and defense using sdn based on machine learning method. In *2018 15th international symposium on pervasive systems, algorithms and networks (I-SPAN)* (IEEE): 174–178.

[31] Deepa, V., Sudar, K.M. and Deepalakshmi, P. (2018) Detection of ddos attack on sdn control plane using hybrid machine learning techniques. In *2018 International Conference on Smart Systems and Inventive Technology (ICSSIT)* (IEEE): 299–303.

[32] Sudar, K.M., Beulah, M., Deepalakshmi, P., Nagaraj, P. and Chinnasamy, P. (2021) Detection of distributed denial of service attacks in sdn using machine learning techniques. In *2021 international conference on Computer Communication and Informatics (ICCCI)* (IEEE): 1–5.

[33] Haider, S., Akhunzada, A., Mustafa, I., Patel, T.B., Fernandez, A., Choo, K.K.R. and Iqbal, J. (2020) A deep cnn ensemble framework for efficient ddos attack detection in software defined networks. *Ieee Access* **8**: 53972–53983.

[34] Tuan, N.N., Hung, P.H., Nghia, N.D., Tho, N.V., Phan, T.V. and Thanh, N.H. (2020) A ddos attack mitigation scheme in isp networks using machine learning based on sdn. *Electronics* 9(3): 413.

[35] Tan, L., Pan, Y., Wu, J., Zhou, J., Jiang, H. and Deng, Y. (2020) A new framework for ddos attack detection and defense in sdn environment. *IEEE Access* 8: 161908–161919.

[36] Sahoo, K.S., Tripathy, B.K., Naik, K., Ramasubbareddy, S., Balusamy, B., Khari, M. and Burgos, D. (2020) An evolutionary svm model for ddos attack detection in software defined networks. *IEEE Access* 8: 132502–132513.

[37] Musumeci, F., Fidanci, A.C., Paolucci, F., Cugini, F. and Tornatore, M. (2022) Machine-learning-enabled ddos attacks detection in p4 programmable networks. *Journal of Network and Systems Management* 30: 1–27.

[38] Sangodoyin, A.O., Akinsolu, M.O., Pillai, P. and Grout, V. (2021) Detection and classification of ddos flooding attacks on software-defined networks: A case study for the application of machine learning. *IEEE Access* 9: 122495–122508. doi:10.1109/ACCESS.2021.3109490.

[39] Polat, H., Polat, O. and Cetin, A. (2020) Detecting ddos attacks in software-defined networks through feature selection methods and machine learning models. *Sustainability* **12**(3). URL https://www.mdpi.com/2071-1050/12/3/1035.

[40] Hu, T., Guo, Z., Yi, P., Baker, T. and Lan, J. (2018) Multi-controller based software-defined networking: A survey. *IEEE access* 6: 15980–15996.

[41] Bannour, F., Souihi, S. and Mellouk, A. A self-adaptive consistency model for distributed sdn controllers .

[42] Aslan, M. and Matrawy, A. (2018) A clustering-based consistency adaptation strategy for distributed sdn controllers. In *2018 4th IEEE Conference on Network Softwarization and Workshops (netsoft)* (IEEE): 441–448.

[43] Koponen, T., Casado, M., Gude, N. and Stribling, J. (2014), Distributed control platform for large-scale production networks. US Patent 8,830,823.

[44] Zhang, Y., Cui, L., Wang, W. and Zhang, Y. (2018) A survey on software defined networking with multiple controllers. *Journal of Network and Computer Applications* 103: 101–118.

[45] Dumitras, T., Neamtiu, I. and Tilevich, E. (2009) Second acm workshop on hot topics in software upgrades (hotswup 2009). In *OOPSLA Companion*: 705–706.

[46] Dixit, A., Hao, F., Mukherjee, S., Lakshman, T. and Kompella, R. (2013) Towards an elastic distributed sdn controller. *ACM SIGCOMM computer communication review* **43**(4): 7–12.

[47] Curtis, A.R., Mogul, J.C., Tourrilhes, J., Yalagandula, P., Sharma, P. and Banerjee, S. (2011) Devoflow: Scaling flow management for high-performance networks. In *Proceedings of the ACM SIGCOMM 2011 Conference*: 254–265.

[48] Koponen, T., Casado, M., Gude, N., Stribling, J., Poutievski, L., Zhu, M., Ramanathan, R. *et al.* (2010) Onix: A distributed control platform for large-scale production networks. In *9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 10)*.

[49] Sridharan, V., Gurusamy, M. and Truong-Huu, T. (2017) Multi-controller traffic engineering in software defined networks. In *2017 IEEE 42nd Conference on Local Computer Networks (LCN)* (IEEE): 137–145.

[50] Matsumoto, S., Hitz, S. and Perrig, A. (2014) Fleet: Defending sdns from malicious administrators. In *Proceedings of the third workshop on Hot topics in software defined networking*: 103–108.

[51] Wang, J., Shou, G., Hu, Y. and Guo, Z. (2016) A multi-domain sdn scalability architecture implementation based on the coordinate controller. In *2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)* (IEEE): 494–499.

[52] Yusuf, M.N., Bin Abu Bakar, K., Isyaku, B., Mukhlif, F. *et al.* (2023) Distributed controller placement in software-defined networks with consistency and interoperability problems. *Journal of Electrical and Computer Engineering* **2023**.

[53] Bannour, F., Souihi, S. and Mellouk, A. (2017) Software-defined networking: a self-adaptive consistency model for distributed sdn controllers. *RESCOM 2017* .

[54] Muqaddas, A.S., Giaccone, P., Bianco, A. and Maier, G. (2017) Inter-controller traffic to support consistency in onos clusters. *IEEE Transactions on Network and Service Management* **14**(4): 1018–1031.