Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.063139

ARTICLE

Detecting and Mitigating Distributed Denial of Service Attacks in Software-Defined Networking

Abdullah M. Alnajim[1,*], **Faisal Mohammed Alotaibi**[2,#] **and Sheroz Khan**[3,#]

[1]Department of Information Technology, College of Computer, Qassim University, Buraydah, 51452, Saudi Arabia
[2]College of Computer Engineering and Sciences, Department of information system, Prince Sattam bin Abdulaziz University, Al-Kharj, 16273, Saudi Arabia
[3]Department of Electrical Engineering, College of Engineering and Information Technology, Onaizah Colleges, Qassim, 56447, Saudi Arabia

*Corresponding Author: Abdullah M. Alnajim. Email: najim@qu.edu.sa

[#]These authors contributed equally to this work

Received: 06 January 2025; Accepted: 02 April 2025; Published: 19 May 2025

ABSTRACT: Distributed denial of service (DDoS) attacks are common network attacks that primarily target Internet of Things (IoT) devices. They are critical for emerging wireless services, especially for applications with limited latency. DDoS attacks pose significant risks to entrepreneurial businesses, preventing legitimate customers from accessing their websites. These attacks require intelligent analytics before processing service requests. Distributed denial of service (DDoS) attacks exploit vulnerabilities in IoT devices by launching multi-point distributed attacks. These attacks generate massive traffic that overwhelms the victim's network, disrupting normal operations. The consequences of distributed denial of service (DDoS) attacks are typically more severe in software-defined networks (SDNs) than in traditional networks. The centralised architecture of these networks can exacerbate existing vulnerabilities, as these weaknesses may not be effectively addressed in this model. The preliminary objective for detecting and mitigating distributed denial of service (DDoS) attacks in software-defined networks (SDN) is to monitor traffic patterns and identify anomalies that indicate distributed denial of service (DDoS) attacks. It implements measures to counter the effects of DDoS attacks, and ensure network reliability and availability by leveraging the flexibility and programmability of SDN to adaptively respond to threats. The authors present a mechanism that leverages the OpenFlow and sFlow protocols to counter the threats posed by DDoS attacks. The results indicate that the proposed model effectively mitigates the negative effects of DDoS attacks in an SDN environment.

KEYWORDS: Software-defined networking (SDN); distributed denial of service (DDoS) attack; sampling Flow (sFlow); OpenFlow; OpenDaylight controller

1 Introduction

Since its inception, the number of internet users has increased exponentially, leading to the emergence of architectures that can justifiably be termed the Internet of Things (IoT) or the Internet of Everything (IoE). This evolution facilitates the convergence of the physical and digital worlds, heralding an era where sensations are experienced by objects and humans across vast distances remotely, akin to direct interactions among users. This architectural framework incorporates internet-connected devices, smart objects, sensors, and associated web-based services that require processing prior to communication—collectively referred to as the Internet of Things (IoT) or the Internet of Everything (IoE) today. Currently, over three billion users

access the internet for various activities, including banking transactions, shopping from vendors around the globe, web-based monitoring and control, social media interactions, and many other applications [1].

As the web of everything continues to evolve, addressing the diverse requirements for intelligent and seamless connectivity will be essential. This involves leveraging advanced technologies, enhancing interoperability, and ensuring that all components of the network work together efficiently to meet the needs of end-users. These requirements cannot be effectively addressed through the intervention of network administrators, operators, or end-users alone; rather, they can only be fulfilled if the internet is engineered to be self-sustaining and self-responding. The growing number of IoT-based services in areas such as smart homes, healthcare, smart cities, agriculture, and supply chains all contribute to enhancing living conditions. However, the sheer volume of devices presents numerous challenges within the current legacy networking paradigm, which is often outdated. For instance, network manageability remains a significant challenge. The integration of edge computing, blockchain, and wireless resources, combined with the advancements in SDN, presents a unique opportunity to overcome existing limitations. By fostering interoperability and leveraging the capabilities of the software community, developers can create more dynamic, efficient, and manageable network architectures that meet the evolving demands of modern applications [2].

Finally, information security and privacy remain prominent areas of concerns. This raises awareness for practitioners and researchers who are concerned about their own safety and the safety of others as a result. The increasing interconnection of devices and systems, coupled with the diversity of hacker motivations, creates a challenging security environment. Ensuring robust security mechanisms while maintaining ease of use is critical to protecting sensitive information and mitigating the risks associated with network security breaches. Addressing these challenges is critical to protecting both businesses and consumers in an evolving digital environment [3,4].

There have been no fundamental changes to traditional network architectures, which consist of groups of routers and switches, along with devices from various manufacturers. These devices are meant to be used together to build heterogeneous networks. Because these devices may be from different manufacturers and used by different technicians under different brands, they require the construction of heterogeneous networks. Different systems from different vendors might be incompatible, making heterogeneous networks expensive to configure and difficult to operate, monitor, and maintain. Configuring different systems increases the vulnerability of hardware and software resources. This requires improving network architectures designed to meet the challenges of these growing networks, making them easy to manage, dynamic enough to accommodate different devices from different vendors, and capable of meeting changing workload requirements by scaling up or down resources with the help of cloud infrastructure. Software-defined networks (SDN) are one of the most promising approaches to improving current hardware-based network systems. Therefore, this project aims to develop a model for detecting distributed denial of service (DDoS) attacks on dynamic and diverse Internet of Things (IoT) or Internet of Everything (IoE) systems, which can be applied in smart environments to infer that the geographic distribution of attack sources follows specific patterns [5,6].

While IoT environments enhance productivity and convenience in a digitally connected society, the increasing complexity of these systems exposes them to significant risks, including distributed denial of service (DDoS) attacks. Addressing these vulnerabilities through effective security measures and proactive management is critical to protecting the integrity and reliability of IoT systems. This is due to the volatility of attack methods and patterns used by attackers.

DDoS attacks present significant challenges to Internet service availability, necessitating effective strategies to identify and differentiate between legitimate and malicious traffic congestion. Software-Defined Networking (SDN) offers a robust solution by simplifying network management and enhancing the ability

to respond dynamically to varying traffic conditions, ultimately improving resilience against DDoS attacks. Furthermore, DDoS attacks can range from the misuse of application-level vulnerabilities to high-volume flooding on a network. Although, it is easy to probe the service availability and to ease traffic on the network, the most significant challenge lies in differentiating between legitimate congestion-based traffic and attacker-generated congestion. SDN addresses these shortcomings by separating network control and management from the data plane to minimise complexity through implementing flow rules in thousands per server rack [7], through effective network security and memory management at both the network control and data planes. One way of combating such DDoS menaces is through identifying for network users the legitimate IoT traffic from anomalous DDoS-generated traffic, bearing thus the SDN with the ability to detect in order to respond to abnormalities in the network in timely manner to analyze the impact of various traditional DDoS attacks imposing threats on SDN architectures [8].

The emergence of software-defined networking (SDN) represents a revolutionary approach to networking, particularly in addressing security challenges. Its integration with the Internet of Things, along with features such as resource pooling and on-demand self-service via cloud computing, is establishing SDN as a pivotal area of research and application in modern networking environments, making the deployment of threat/attack detection strategies in the IoT environment an essential part of the operational ecosystem [9].

Software-Defined Networking (SDN) presents significant advantages in collecting deployability information and improving security through dynamic control and advanced application capabilities. By leveraging these features, SDN can effectively enhance network resilience and provide robust solutions for detecting and mitigating various types of attacks.

In this framework, the authors have proposed an SDN-based security system that aims at protecting the SDN network from DDoS attacks. The proposed algorithm in this system uses statistical data collected by the sFlow and OpenFlow protocols to detect and mitigate the DDoS attacks. Also, the proposed algorithm can identify high network traffic to block it based on selected threshold values. It represents a significant advancement in the application of SDN technology for enhancing network security, particularly in the context of increasingly sophisticated DDoS threats. The main contributions of the paper are summarised as below:

- A novel framework for detecting and mitigating the DDoS is executed in an SDN environment using the sFlow and OpenFlow protocols.
- An algorithm that uses statistical data collected by the sFlow and OpenFlow protocols to detect and mitigate DDoS attacks.
- The overall contribution of this work is proposing a system that uses sFlow to detect DDoS attacks and OpenFlow to mitigate these attacks.

The rest of the paper is organized as follows: Section 2 provides background on software-defined networks (SDNs), distributed denial-of-service (DDoS) attacks, defense techniques, and related work. Section 3 introduces the problem modeling and statement, Section 4 explains the proposed method, Section 5 describes the testing and evaluation environment, and Section 6 presents contemporary work. Section 7 concludes with concluding results and remarks, followed by Section 8 for further work.

2 Background

The Internet of Tings (IoT) is the interconnection of smart devices that integrate together as a single network via various services or protocols. The IoT enables the collection of sensitive information from smart devices to perform critical operations. It also allows smart devices to communicate with each other at high speed, and the SDN provides orchestration for network management by decoupling the control plane and

the data plane. This section presents a review of the Software Defined Networking (SDN) and Distributed Denial of Service attack (DDoS) supported by related work and its discussions.

2.1 Software Defined Networking (SDN)

The TSDN model is introduced to compensate for the shortcomings of the traditional network by offering programmability, compatibility, cost, manageability, and many other distinct features. The main idea of an SDN is to separate in network switches data plane and the centrally located device of control plane. It is typically centralized and manages network routing and policies. The data plane executes data routing based on the control plane's decisions. It consists of the physical network devices, such as switches and routers, that handle data packets, as shown in Fig. 1.

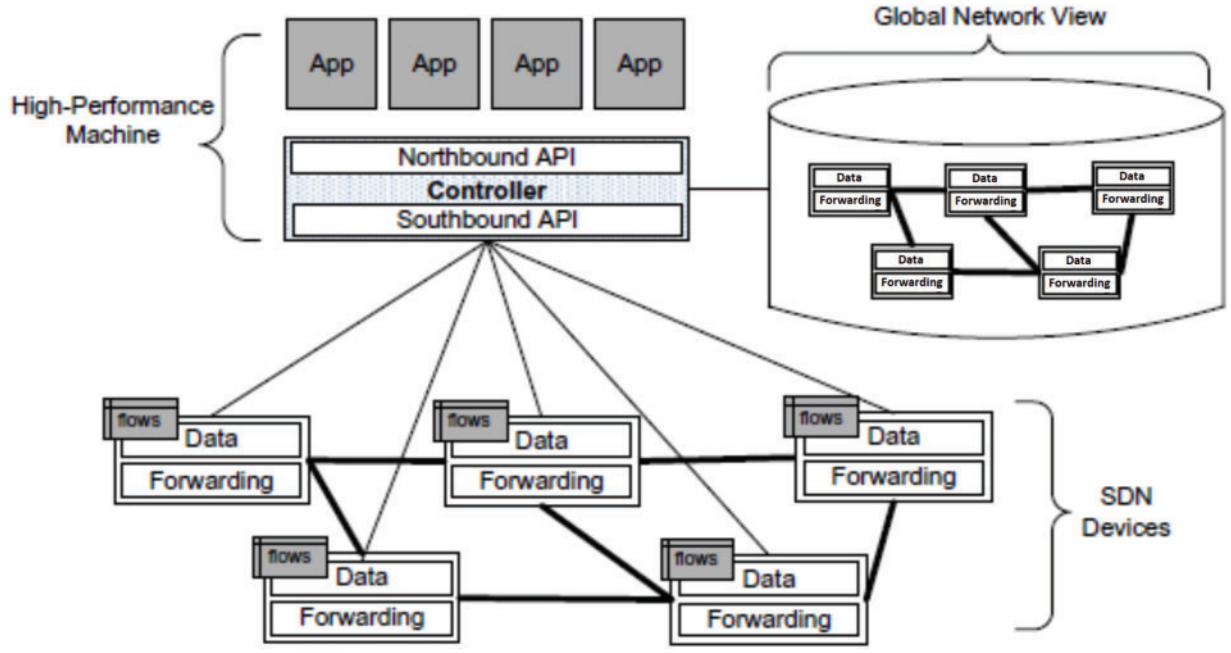

Figure 1: Software Defined Networking (SDN) drawn based on concepts from [7]

Furthermore, an SDN architecture generally consists of three functional components: 1) Applications: these are programs that send data to the SDN controller via an application programming interface (API); 2) Controllers: these are logical entities that receive instructions or requirements from the SDN application layer and transmit them up and down the hierarchy; and 3) Network devices: these devices control the routing and processing of data across network paths [10] in the SDN controller for network security. DDoS attacks transform the vulnerable IoT devices into becoming Botnets that cause severe disruptions to the IoT system by compromising the nodes.

OpenFlow is a communication protocol in the SDN environment. It is used by the SDN control plane and data plane for communication. Furthermore, the SDN controller determines the paths of packets across the SDN network. It configures network devices according to its flow rules. OpenFlow also enables communication between network devices, enabling control of devices from any vendor. Finally, OpenFlow can be used to remotely control the flow tables of layer three switches [11,12].

2.2 Distributed Denial of Service (DDoS) Attack

2.2.1 Definition

Distributed denial-of-service (DDoS) attacks are undoubtedly a major concerns for many businesses and organizations. These attacks are maliciously coordinated against online services, such as commercial websites, banking websites, government websites, and others. Furthermore, these attacks are typically carried out by a large number of independent programs, with the goal of disrupting system resources for a specific period of time until the targeted device is unable to provide services.

2.2.2 How DDoS Attacks Are Performed?

The first step in launching a distributed denial-of-service (DDoS) attack, which is not primarily aimed at stealing data, is to recruit a group of bots or malware. Furthermore, to turn a computer into a zombie, hackers develop and execute specialized malware. This malware can be installed on a large number of devices that have visited unsafe websites containing the malware. It can spread to targeted computers via email attachments, compromised websites, or corporate networks. As a result, legitimate users may be infected with malware that unwittingly turns them into bots. This type of malware allows attackers to gain access to infected or disabled computers. Once the computer is disabled, it connects to the attackers' commands, which begin controlling the server. It then receives their commands. Commands may include information about the targeted server, the time of the attack, the attack method, and its duration. Additionally, an army of bots is formed as a botnet, typically consisting of thousands of bots. This network weakens the security of IoT devices and exposes them to distributed denial of service (DDoS) attacks, as shown in Fig. 2.

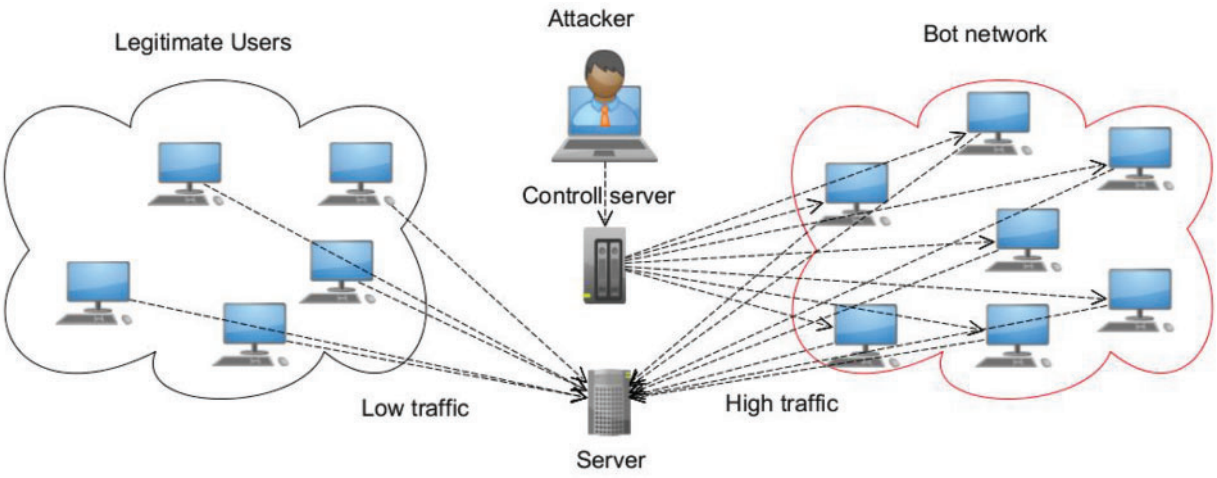

Figure 2: Distribute denial of service (DDoS)

The attackers send commands to the command and control server, which serves as the central management center for the botnet. The command and control server receives these commands and relays them to all connected bots within the botnet. The server coordinates the bots' operations and ensures they work in concert to execute the attack. Each bot, a compromised device, receives commands from the command and control server. The bots then launch a distributed denial of service (DDoS) attack on the victim's server to overwhelm the target. 3 derived from concepts presented in [13].

2.2.3 Target of DDoS Attack

In general, DDoS attackers, in order to cause denial of service to legitimate users, target the following network components: (i) routers, (ii) links, (iii) firewalls and defense systems, (iv) the victim's infrastructure, (v) the victim's operating system, (vi) existing connections, and (vii) the victim's applications [14,15].

Potential targets also include vulnerable IoT devices, which, when attacked, form botnets, threatening IoT system security through distributed denial-of-service (DDoS) attacks. Machine learning (ML) and deep learning (DL) techniques have demonstrated significant potential in detecting DDoS attacks, offering numerous advantages over traditional statistical or policy-based solutions.

2.2.4 DDoS Attack Types

As previously mentioned, modern computer networks rely heavily on a variety of network devices, each of which plays a critical role in ensuring seamless connectivity and high performance. Understanding the intricacies of these systems is crucial for effective network management and responding to changing conditions, given the variety of attack types, the most common of which are distributed denial of service (DDoS) attacks.

Distributed denial-of-service (DDoS) attacks vary in their methods and objectives, making it essential for organizations to implement effective security measures to mitigate these threats. They can be classified into three main types: 1) Value-based attacks, including User Datagram Protocol (UDP) floods, Ping floods, and spoofed packet floods, 2) Protocol-based attacks, including SYN floods, fragmented packet attacks, Ping of Death, and Smurf DDoS attacks, 3) Application layer attacks, consisting low-level and slow attacks, GET/POST floods, and targeted attacks.

The programmability of SDN routers is decoupled from the control decisions that enable innovation and evolution. In SDN, network intelligence is logically integrated into the software-based control plane, while OpenFlow network switches become the data plane that can be programmable via the OpenFlow protocol with an open interface [16].

Denial-of-service (DoS) attacks are designed to disrupt services by overwhelming the target with excessive traffic, limiting access for legitimate users. Researchers in [17] categorize DDoS attacks, from a downstream impact perspective, into: (i) application attacks, where attackers target the application itself; (ii) resource and host attacks, which disrupt resources; (iii) network attacks, which target network bandwidth; and (iv) infrastructure attacks, which target the Domain Name Server (DNS).

2.2.5 Challenges of DDoS Mitigation

There are several challenges in building an SDN controller capable of detecting and mitigating DDoS attacks. These challenges include: 1) the size of the Botnet—for example, a large Botnet becomes extremely difficult to handle; 2) the accuracy of detecting abnormal traffic; 3) long-lasting attacks that last longer; and 4) large-scale testing to meet the needs of multi-Gbps networks.

2.2.6 Defense Approaches

There are three main defense mechanisms for detecting and mitigating DDoS attacks. The first approach, called the Proactive Defense Mechanisms framework, proposes a technique that does not directly mitigate DDoS attacks. However, this framework efficiently designs and builds infrastructure to withstand an imminent DDoS attack, known as cloud hosting. The second approach is called reactive defense mechanisms, and its main idea is to mitigate or, if possible, stop DDoS attacks when they occur. Finally, the third

approach is called post-attack analysis: This technique analyzes DDoS attacks and monitors their patterns to obtain information that enables tracking and preventing attackers from carrying them out. These defense mechanisms are determined based on various performance metrics that help ensure effective communication regardless of the sites involved [18–20].

Thanks to advances in the manufacture of various miniaturized sensor systems, the development of numerous web services, and cloud computing, it has become possible for almost any isolated system to communicate with similar devices, which are currently expected to number in the billions on the internet. These complex systems face numerous challenges, the most common of which are distributed denial of service (DDoS) attacks [21,22]. All of these attacks are designed to send large volumes of messages, making it difficult to find the exact signature of the nature of the attack.

3 Problem Modeling and Statement

Mitigating DDoS attacks is one of the main causes of concern for network administrators. SDN as shown in Fig. 3, is a promising networking paradigm that can be used to tackle DDoS attacks. Furthermore, it is assumed that the network consists of a set of nodes, $\mathcal{N} \neq \varnothing$, where the number of network nodes is $N = |\mathcal{N}|$, $N \in \mathbb{Z}$, and $N \geq 4$. In addition, there is a subset of these nodes as hosts $\mathcal{H} \subset \mathcal{N}$ such that the number of these hosts is $H = |\mathcal{H}|$, $H \in \mathbb{Z}$ and $H \geq 2$, and some of these hosts (or users) are legitimate in the sense $\mathcal{U} \subset \mathcal{H}$, the number of these legitimate users $U = |\mathcal{U}|$ and some of them are Bot or Zombie users $\mathcal{B} \subset \mathcal{H}$, the number of these bots are $B = |\mathcal{B}|$, and $B \in \mathbb{Z}$. Moreover, some of these nodes are switches $\mathcal{S} \neq \varnothing$, $\mathcal{S} \subset \mathcal{N}$, $S = |\mathcal{S}|$, and $S \in \mathbb{Z}_{>0}$. Finally, there is at least one controller in the network $\mathcal{C} \neq \varnothing$, $\mathcal{C} \subset \mathcal{N}$, the number of controller in the network is $C = |\mathcal{C}|$, and $C \in \mathbb{Z}_{>0}$. Each host can send traffic with rate $\gamma \geq 0$ and $\gamma \in \mathbb{R}$. The bandwidth of the link between switches and between users and switches is D, and $D > 0$. The main problem here is to detect bots in $\mathcal{B}$ that are generating high traffic, and stop them from overwhelming the switches and bandwidth. Further, to allow only legitimate users in $\mathcal{U}$ to send packets through links and switches. The proposed model is explained in the section that follows how it is designed to tackle the problem at hand by mitigating DDoS attack.

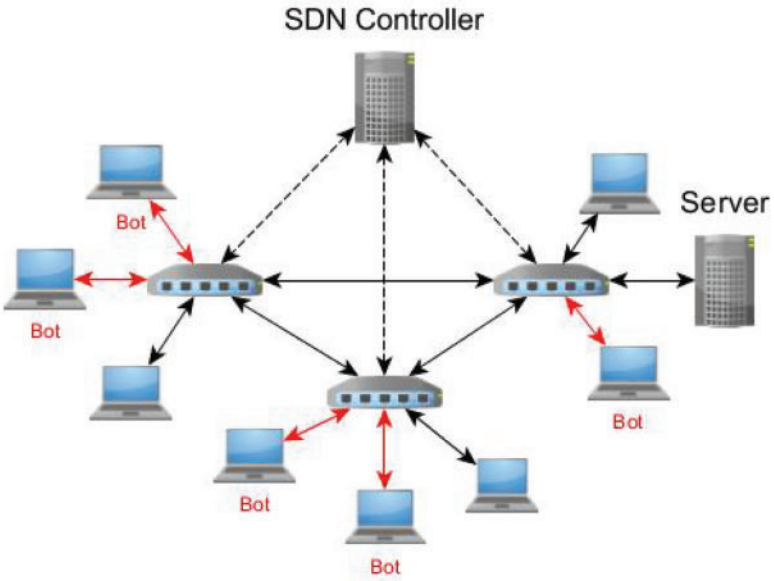

Figure 3: Distributed denial of service (DDoS) attack in Software Defined Networking (SDN)

4 The Proposed Method

The authors have proposed in this framework a defense approach to detect and mitigate DDoS attacks in an SDN environment.

DDoS Detection and Mitigation System Using sFlow

The proposed distributed denial-of-service (DDoS) detection and mitigation system relies on real-time traffic monitoring within an SDN environment as a defense against DDoS attacks. The security system uses statistics recorded while monitoring traffic passing through SDN OpenFlow switches to detect malicious traffic. Once the attack is detected, the proposed defense system tracks the hosts that caused it in order to block them for a specified period of time, allowing only legitimate users to use the network facilities by sending packets over OpenFlow switches and links in the SDN network. Traffic monitoring is critical to detecting and mitigating DDoS attacks using tools like Sampling Flow (sFlow).

Furthermore, sFlow is a sampling technique used to monitor network traffic. It can be used to detect various types of DDoS attacks, such as SYN floods, UDP user datagrams, and more. Furthermore, the OpenFlow and sFlow protocols provide SDN researchers with an integrated flow monitoring and control system where the OpenFlow controller can be used to determine which flows should be monitored by sFlow, making the controller the layer that sends packet output messages to software forwarding rules to switches. In this context, to make the most of sFlow, we will use the powerful analytics engine sFlow-RT to collect real-time statistics for switches. In general, sFlow-RT can be used to monitor traffic by recording relevant time-release statistics. It is also used for load balancing, DDoS attack detection, and mitigation. Additionally, there are three main components of sFlow: i) the sFlow aggregator in an external controller or in an SDN controller, ii) the sFlow agents integrated into Openvswitch, and iii) the sFlow protocol. Software-defined networking enables a logical and centralized separation of the network control plane from the data plane [23].

The main idea is to integrate sFlow agents into the OpenFlow switches within an SDN network. The main task of sFlow agents is to send samples of network traffic from a specific network device, such as a switch or router, to an sFlow aggregator in an external controller. The sFlow aggregator can then use tools like the inMon sFlow-RT to automatically detect and mitigate large and long-waiting packet flows in real time, as shown in Fig. 4. The sFlow agents embedded in the OpenFlow switches communicate with the sFlow cluster via the sFlow protocol, while OpenfFlow switches communicate with OpenDaylight via the OpenFlow protocol, as shown in Fig. 5. The sFlow agent encapsulates sample flow and interfaces counters into sFlow data-grams. These data-grams are sent by sFlow agents to sFlow collector in the external controller via sFlow protocol.

The sFlow agents sample data packets based on probabilities defined by network administrators, operators, or end users. They also send interface counters to the sFlow aggregator. Network users can also specify threshold values for data traffic and sampling rates. The primary role of the SDN aggregator is to analyze data samples and take necessary measures against DDoS attacks to mitigate or, where possible, stop them.

In addition, the REST APIs, also known an architectural style for an application program interface (API) help each application to retrieve metrics, configure flows, receive notifications, and set appropriate threshold values using *GET*, *PUT*, *POST* and *DELETE* data types, which refers to the reading, updating, creating and deleting of operations concerning resources. The sFlow agents can be embedded into OpenFlow switches by using the following code:

```
sudo ovs-vsctl – –id=@sflow create sflow agent=eth1
target= \“192.168.56.102:6343\” sampling=300 polling=15
– – set bridge s1 sflow=@sflow
```

The above instruction codes will configure or enable sFlow in OpenvSwitch. The sFlow agent of $eth1$ is embedded into the IP address of 192.168.56.102 for the controller. In addition, the sampling rate is set at 300. In other words, for every 300 packets captured by the sFlow agent, only one packet is sent to the sFlow collector. The polling period is set to 15 s. Alternatively, the client will send a data message to the sFlow collector every 15 s. Fig. 5 shows flow diagram of sFlow agent processing.

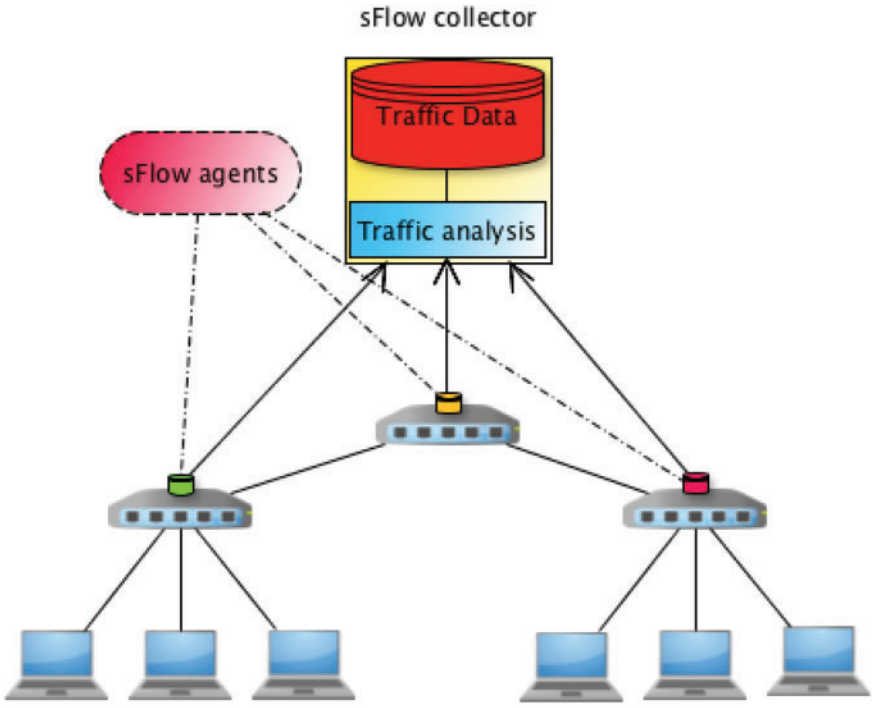

Figure 4: sFlow agents and sFlow collector

The sFlow agents are installed on OpenFlow switches, enabling them to sample network traffic in real time. These agents continuously monitor traffic and capture flow data, which is essential for analyzing network performance and security. The agents send the collected traffic data to the sFlow analytics engine in real time. This data includes information about packet flows, protocols, and bandwidth usage, allowing for comprehensive traffic analysis. The sFlow analytics engine processes the incoming data to identify anomalies or patterns that may indicate a distributed denial of service (DDoS) attack. By analyzing traffic trends, the engine can detect sudden spikes or unusual behaviors that are typical of DDoS attacks. When a potential DDoS attack is detected, the analytics engine sends immediate notifications to the affected application on the targeted server. This early alert mechanism enables a rapid response, enabling the application to implement countermeasures or strategies to mitigate the effects of the attack.

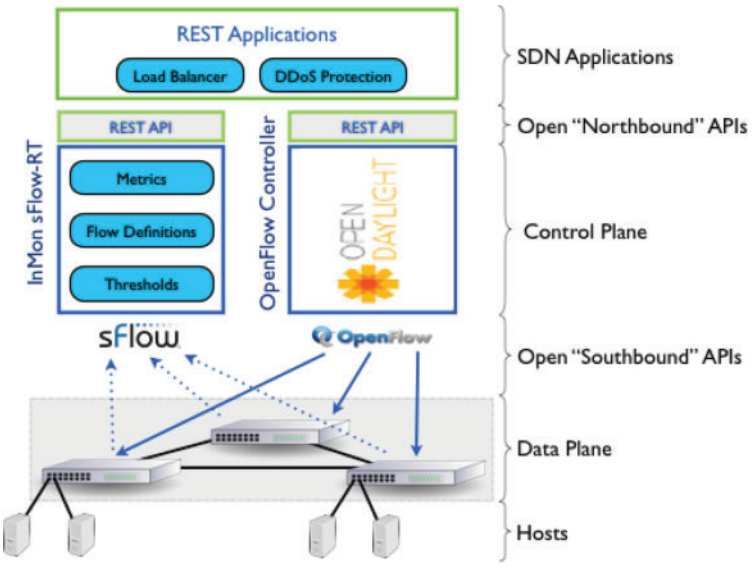

Figure 5: The sFlow agent embedded in switch drawn based on concepts from [24]

The listed Algorithm 1 demonstrates a pseudo-code designed for the detection of DDoS attacks and mitigation. The algorithm checks the traffic flow rate λ at any time t in all switches in the SDN network through sFlow that allows to check the usage of the bandwidth accordingly. Once the traffic reaches a pre-set threshold value θ that identifies to mean DDoS-based attack is detected. In this paper, the value of θ is kept equal to 10% of the bandwidth (D), $\theta_1 = (\frac{D}{100} \times 10)$. The utilization of bandwidth is u. It is noteworthy to note that θ could take any other value, depending on network administrator. The algorithm then blocks the users who behave like a Bot by keeping them on hold for T seconds, a value set for 60 s in this paper. The algorithm is made more complex by making the threshold value a range instead of single value, thus setting up a criterion for how long the time period, T_1, should be. However, if else the value of θ is kept equal or less than 15% of the bandwidth (D) $\theta_2 = (\frac{D}{100} \times 15)$, then the time period is kept as T_2.

Algorithm 1: Flow monitoring and DDoS attack detection

procedure ATTACK DETECTION
    Set all parameters.
    Start:
    for $\forall S \in \mathcal{S}$ **do** ▷ For all switches
        for $\forall P_i$ **do** ▷ For all packets
            Add source IP address of new flow to the flow table.
            Add OpenFlow rules to the flow table.
            Monitor packet flow (rate and duration) using sFlow.
            if $\theta_1 \geq \lambda \leq \theta_2$ **then** ▷ Check bandwidth usage
                The traffic is malicious traffic (DDoS attack).
                Block the attackers for a period of time, T_1.
            else
                if $\theta_2 \geq \lambda \leq \theta_3$ **then** ▷ Check bandwidth usage
                    The traffic is malicious traffic (DDoS attack).
                    Block the attackers for a period of time, T_2.
                else

(Continued)

Algorithm 1 (continued)

```
                    The traffic is legitimate; no action needed.
                end if
            end if
        end for
    end for
    goto Start.
end procedure
```

5 Testbed and Evaluation

In this work, Mininet was used as a network simulator and OpenDaylight as an SDN controller to develop a test environment to demonstrate how it can serve as a reference topology for security applications. The computer system used for simulation purposes was a Core i7-4790 processor at 3.60 GHz and 32 GB of RAM. An Oracle VM Box was also used to create a virtual machine to host the SDN network devices involved.

5.1 Network Emulator

An SDN network consists of five main components: 1) the application plane (AP), 2) the northbound interface (NBI), 3) the control plane (CP), 4) the southbound interface (SBI), and 5) the data plane (DP). The control plane, connected to SDN controllers, interacts with applications to transport data flows to their destinations, ensuring device sharing and quality-of-service (QoS)-compliant data routing between connected devices. Network simulators enhance fundamental understanding in the field of next-generation wireless network technologies worldwide.

There are several network simulators used in research, such as Network Simulator 3 (NS3), OPNET, OMNeT++, NetSim, REAL, QualNet, and J-Sim (R24). However, the researchers decided to use Mininet as a network simulator due to its simplicity and ease of implementation. Furthermore, Mininet can simulate an entire network of terminal receivers, controllers, and links for large SDNs operating on limited virtual machine resources in a single computer. Mininet switches also support OpenFlow protocols [25]. IoT devices are increasingly becoming targets for cyberattacks due to their inherent vulnerabilities including one emerging threat is the Mongolian DDoS attack that exploits the vulnerabilities of the distributed nature of the IoT.

5.2 SDN Controller

SDN controllers, also called network operating systems (NOS), are responsible for routing packets. These controllers have comprehensive monitoring of the SDN network through packet flow (from a viewpoint). They can monitor all network devices within their administrative scope. Several open source SDN controllers are available for research use, including but not limited to: a) Open Daylight, b) ONOS, c) Project Calico, d) Project Fast Data, e) Project Floodlight, f) Beacon, g) NOX/POX, h) vneio/sdnc, i) Ryu controller, j) Cherry, k) Faucet, and f) OpenContrail. However, in this work, the OpenDaylight SDN controller is adopted as an open source SDN architecture because it provides a comprehensive platform for the general design aspects of our framework.

5.3 Network Traffic Generator

Many software-based packet generation tools can be used to flood an SDN network with random packets or allow the user to create custom packet exchanges between hosts in an SDN network. These tools include, but are not limited to, AnetTest, Pktgen, IP Sorcery, Pierf, and Scorpy. In this work, the authors chose to use

Hping3 for its simplicity and ease of implementation. It specializes in generating, analyzing, and sending malicious IP packets for DDoS attacks.

5.4 Case Study

The proposed method for detecting and mitigating the DDoS attacks in an SDN environmental setup is implemented by utilizing sFlow and OpenFlow protocols for performance evaluation.

The star topology as shown in Fig. 6 has been used. The number of nodes in the SDN network is $N = 13$ nodes, there are 9 hosts in this network, $H = 9$, four of these hosts are legitimate users, that is, $U = 4$ and four of them are bots, $B = 4$ and one host is a server. Furthermore, there is also one controller $C = 1$ and four OpenFlow switches, $S = 4$. We will use an OpenDaylight controller (Nitrogen version) and OpenvSwitch to receive/forward commands using OpenFlow protocols. In addition, the legitimate user sends traffic at rate $\lambda = 10$ to 80 Kbps, whereas the bot sends traffic at rate λ between 1 and 10 Mbps. Table 1 lists more details about users and traffic rates for generating SDN traffic jam.

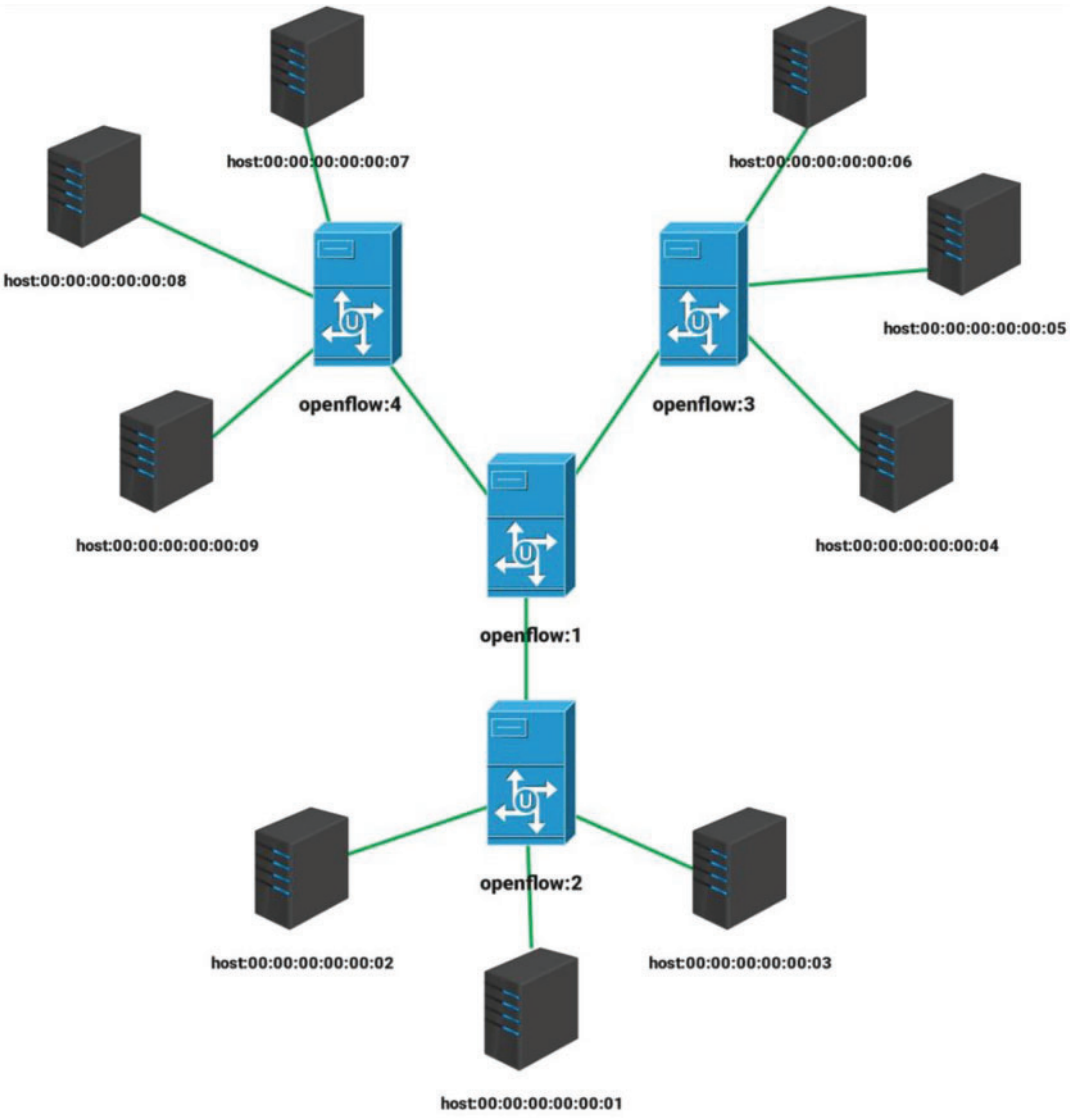

Figure 6: Star topology (OpenDaylight interface)

Table 1: SDN traffic details

Host number	Host type	Traffic rate
1	Legitimate user	10 Kbps
2	Legitimate user	30 Kbps
3	Legitimate user	20 Kbps
4	Legitimate user	80 Kbps
5	Bot	3 Mbps
6	Bot	4 Mbps
7	Bot	5 Mbps
8	Bot	10 Mbps

The sFlow agents have been configured for all OpenFlow switches of Fig. 6 described in Table 1. All the OpenFlow switches have been configured automatically using Python. The malicious traffic is generated by using the following *Hping*3 command:

```
Bot name hping3 –flood –tcp -k -s 80 Server name
```

The top four plots in Fig. 7 show regular traffic generated by legitimate users, while the high traffic generated by the *Hping*3 command to simulate DDoS attack behavior in an SDN network is generated by four bots as shown in the bottom four plots of the same figure.

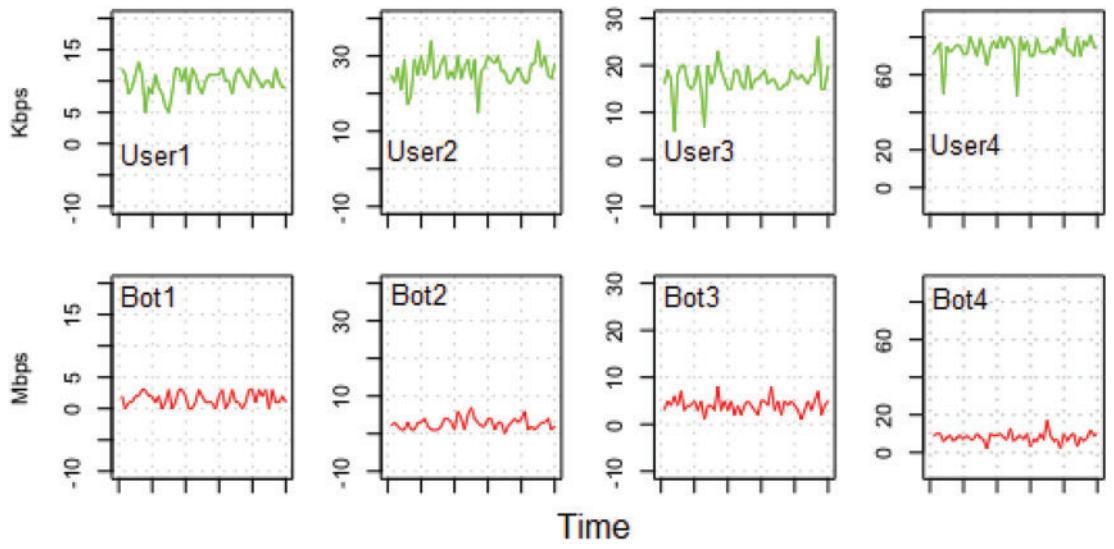

Figure 7: SDN network traffic (Kbps): the top four charts in this figure depict the sample of traffic generated by legitimate users whereas the bottom four charts demonstrate the DDoS attack

Fig. 8 illustrates the SDN network traffic of the device with IP (192.168.11.35) with and without the DDoS mitigation algorithm. As shown in Fig. 9, the traffic exceeds the pre-set threshold of 1.25 Mbps in the beginning, which happens because the detection and mitigation algorithms are not running as the controller had not been enabled. Then, after running the detection and mitigation algorithm, the traffic is kept below

the threshold value. The suspension of IP address has been set to last for around 60 s. Subsequently, the mitigation algorithm removes the suspension after 60 s to re-trigger provided the Bot has been still attacking.

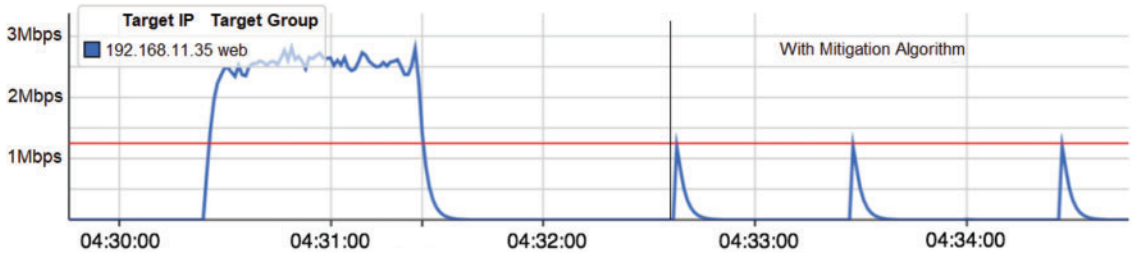

Figure 8: Real-time values of DDoS attack generated by the first Bot that triggers the mitigation algorithm when the threshold value is exceeded

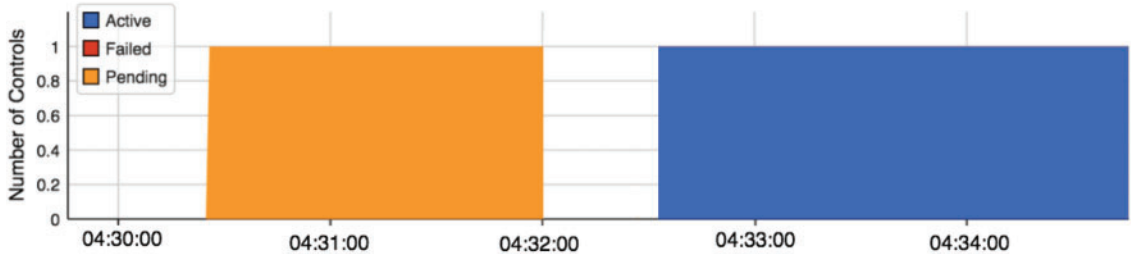

Figure 9: Shows time when Mitigation algorithm controller is Active (in blue), Pending (in yellow)

Fig. 8 shows the SDN network traffic for a device with IP address 192.168.11.35 with and without a DDoS mitigation algorithm. As shown in Fig. 9, the traffic initially exceeds the preset threshold value of 1.25 Mbps, because the detection and mitigation algorithms are not being triggered due to inactive controller.

It is clear that the DDoS attack has been carried out by only one Bot (Host 5) with IP address 192.168.11.35. The results in the Figure confirm that the proposed sFlow-based detection and mitigation algorithm is working well to mitigate a large number of DDoS attacks on SDN networks. This is illustrated in Fig. 9 when the sFlow controller is active (shown in blue) or suspended (shown in orange).

The results also confirm that the detection time is approximately three seconds in this setting. The researchers believe that the results may vary depending on the magnitude and type of the DDoS attack. Furthermore, a key advantage of using the sFlow algorithm is that it doesn't burden the controller with a huge amount of additional data, which could delay traffic by increasing packet latency in the network, because it only sends a sample. It's worth noting that this experiment was initially conducted to verify the effectiveness of sFlow with an SDN network in mitigating DDoS attacks crafted using HPing3 as the DDoS generator.

The results of this work, which detects any type of distributed denial of service (DDoS) attack, are compared to the results of a recent study that detected SYN flood attacks on edge routers connecting hosts to the Internet [26]. The proposed Adaptive Machine Learning-based and SDN-enabled DDoS Detection and Mitigation (AMLSDM) framework is a state-of-the-art solution designed to address the growing threat of DDoS attacks. The comparison was made at 1.25 Mbps, with detection occurring within 3 vs. 4 s, as shown. Fig. 7 in this work shows a comparison between legitimate users and botnet attackers, which reduces the detection and mitigation potential of DDoS attacks. Similarly, recently proposed techniques focus primarily on issues related to centralized visibility, link state detection, flow rule setting, and controller

load balancing in [27]. While the anomaly detection results are compared with [28] to restore the system to normal within 10 s, enabling more efficient use of network resources. The authors present a variety of controllers before presenting the Mininet tools used in the research.

Additionally, the work presented by the authors in [29] only detects high-volume traffic, whereas our method can detect and mitigate low-volume SYN DDoS attacks by combining aggregated sFlow traffic with traffic blocked using OpenFlow. The authors in [30] addressed DDoS attacks by resetting the controller, which redirected access control tasks to the data plane. This method, while effective in mitigating attacks, introduced latency issues due to the combination of three OpenFlow protocols. In the current paper, the authors propose a solution that maintains network performance even as the network size increases. They emphasize that increasing the network size does not adversely affect traffic overhead between the controller and the OpenFlow switches. The approach involves keeping the sampling rate low, which helps minimize traffic overhead. By efficiently managing the sampling rate, the system can handle larger networks without a significant increase in communication load between the controller and switches.The DDoS attacks are thwarted by resetting the controller to send access control to the data plane, which caused latency issues by combining three open flow protocols. However, in this paper, increasing the network size does not affect the traffic overhead adversely between the controller and the OpenFlow switches by keeping the sampling rate low to keep the traffic overhead low.

6 Contemporary Work

There are multiple solutions available to mitigate distributed denial-of-service (DDoS) attacks. DDoS attacks can be at the application level, the control level, or at a large scale. One common method used by large enterprises and service providers to protect against DDoS threats is BGP remotely activated blackhole (RTBH). This attack instructs routers to block data traffic at the edge before it enters the protected target network, reducing network overload. The number of internet users has increased significantly, now accounting for 57% of the world's population. Such a large number of internet users make them vulnerable to various security threats, including denial-of-service attacks via geographically distributed devices. Hence the name distributed denial-of-service (DDoS) is assigned to such attackers. Distributed denial-of-service (DDoS) attacks are a serious threat, exploiting network vulnerabilities and using malicious software, such as Trojans, to disrupt services. Understanding these mechanisms is critical to developing effective mitigation strategies to protect networks from these threats [31].

Various devices can also be used to mitigate distributed denial-of-service (DDoS) attacks by filtering out all types of attacks that drain resources, disrupting many services and thus degrading network performance. SDN technologies, such as Open Flow, provide multiple methods for controlling routing and switching, making existing networks more feature-rich, allowing network resources to be adjusted through so-called dynamic control defense to meet emerging demands [32].

With the development of computer networks, current network systems and data centers have become feature-rich, complex, and highly data-intensive, so that system designers often need to modify network software and coordinate network resources according to specific requirements. However, traditional network architectures do not meet these requirements from the perspective of enterprises, telecom companies, and end users. For example, decision-making power in legacy networks is distributed across different network components, making adding new devices or services to the network a daunting task. This is achieved by integrating AI into the data layer using software-defined networking (SDN) architectures [33,34].

SDN networks, combined with machine learning-based defense systems, significantly improve the classification and management of incoming requests. This capability enables more effective threat and anomaly detection, contributing to a more secure and efficient network environment in the face of the latest

DDoS attack scenarios in SDN and cloud computing [35,36]. These algorithms include PATMOS, a new hybrid flow-based processor with a protocol that uses a rumor-like approach to identify attacks [37–39], and an SDN scheduling algorithm that ensures SDN is unavailable during certain attacks. Using different algorithms, SDN can learn different DDoS techniques and counterattacks before they can harm the server.

SDNs are useful because they help repel automated botnet attacks and can detect DDoS attacks early, delaying server downtime. A new protocol called PATMOS is used to mitigate DDoS attacks in multi-controller SDNs by clustering controllers Additionally, SDN networks can operate automatically with minimal supervision to handle DDoS attacks such as UDP, HTTP, etc. Essentially, an effective method must be used to ensure DDoS attacks are detected by appropriate packet traffic management in an SDN network environment.

In [40,41], the authors propose an SDN-enabled adaptive machine learning-based distributed denial-of-service (AMLSDM) detection and mitigation framework for Internet of Things (IoT)-based networks. The authors used a combination of three techniques: the Snort intrusion detection system, the sFlow sampling standard, and the OpenFlow protocol. They used Snort to quickly detect DDoS attacks. Although the authors did not provide details of their system, the researchers behind this work believe that the proposed method is complex and may raise concerns about packet latency in such small networks [42].

The authors in [43] proposed a defense framework based on sFlow and OpenFlow. However, their proposed model did not include their algorithm. They were able to provide some preliminary results. They demonstrated that their proposed system can only detect massive DDoS attacks in secure environments and may not handle traffic in hostile environments.

The authors in [44] proposed a defense system against SYN DDoS attacks using sFlow and OpenFlow. However, the proposed model was unable to handle other types of DDoS attacks. After examining the above, it is clear that most of the proposed models are only able to detect and mitigate massive attacks by adaptively balancing attack detection coverage and accuracy. Some recommended modules also focused on a specific type of DDoS attack [45–47]. Table 2 is reproduced by comparing the work in this paper with the results of some contemporary research.

Table 2: Comparative analysis with contemporary techniques

Paper	Technique	Results	Comparison
[26]	SDN-based IoT security using Snort for intrusion detection, sFlow for attack detection, and OpenFlow for mitigation.	Fast SYN DDoS detection with Snort.	Complex and slow response.
[27]	The authors implement an SDN-based information security defense mechanism (ISDM) incorporating three OpenFlow management tools with sFlow standard for network intrusion detection system (NIDS), to perform anomaly detection, mitigation to reduce the loss caused by the DDoS attacks.	To guard against ICMP Flood attack, the SDN controller sends commands to OpenFlow switch for dropping datagrams, when the flow information entering. It shows that the flood traffic has reduced as shown in Fig. 8.	Complicated and time consuming in responding.

(Continued)

Table 2 (continued)

Paper	Technique	Results	Comparison
[28]	Proposes Safe-Guard Scheme (SGS) to implement anomaly traffic detection and controller dynamics detection by remapping controller to send access control to data plane.	Forged flows are differentiated from the legitimate ones by adopting a four-tuple vector to reduce the flow setup and controller response time.	Complicated and time consuming in responding.
[45]	An SD-IoT network model on the COOJA simulator where in some nodes in this model are configured to generate massive traffic to other nodes. The detection mechanism is deployed on the SDNWISE controller by applying IP Packet counter and Payload Detection techniques by analyzing packet logs.	It can detect the vulnerabilities in IoT devices or malicious traffic generated by IoT devices using the session IP counter and IP Payload analysis.	Complicated and time consuming in responding.
[46]	Proposes an SDN-based, four module DDoS attack detection and mitigation framework for IoT networks called FMDADM.	The experimental results show that the proposed framework performed better than most cutting-edge solutions currently available with the following benchmarks for accuracy, precision.	Complicated and time consuming in responding.
This work	The proposed method for detecting and mitigating the DDoS attacks in an SDN environmental setup is implemented by utilizing sFlow and OpenFlow protocols for performance evaluation.	The star topology as shown in Fig. 6 shows an SDN network is N = 14 nodes, there are 9 hosts in this network, H = 9, four of these hosts are legitimate users. In addition, the legitimate user sends traffic at rate λ = 10 to 80 Kbps, whereas the bot sends traffic at rate λ between 1 and 10 Mbps.	Easy implementation and faster response in detecting and mitigating traffic.

7 Conclusion

In conclusion, the findings of this framework have been examined and evaluated as a defense system for detecting and mitigating DDoS attacks in SDN networks. Preliminary simulation results confirm that the proposed defense system can efficiently detect and mitigate DDoS attacks within approximately three seconds. The proposed method has provided a comparative analysis of regular traffic from legitimate users while mimicking DDoS behavior using the Hping3 command to generate traffic congestion, as detailed in Table 1. The traffic analysis indicates that, with the DDoS mitigation algorithm in place, traffic remains

below the threshold value of 1.25 Mbps. Additionally, the system suspends the IP address for around 60 s if a bot continues to attack. These findings demonstrate that the proposed sFlow-based detection and mitigation algorithm is effective in handling a significant number of DDoS attacks on the SDN network. Moreover, results indicate that the proposed method does not adversely affect overhead traffic. Finally, by utilizing the suggested defense system, scalability concerns can be effectively managed, as network administrators can adjust the sampling rate to minimize overhead traffic between the controller and OpenFlow switch.

This article demonstrates that SDN can play a crucial role in reducing the impact of DDoS attacks. A preliminary framework for detecting DDoS attacks within an SDN architecture has been presented as a defense mechanism, utilizing overflow and sFlow technologies. Initial results indicate that the proposed model can effectively minimize the consequences of DDoS attacks in an SDN-based environment.

This work is planned to progress in three key directions using the OpenDaylight (ODL) controller for mitigating DDoS attacks. Firstly, a multi-level controller approach (comprising global and local controllers) will be explored to address DDoS attacks while minimizing overhead traffic between the data plane of switches and the control plane in a single-controller architecture. Secondly, the authors plan to investigate running OpenDaylight on a Raspberry Pi 3. The preliminary idea involves developing a DDoS mitigation script (mitigation algorithm) to implement on the Raspberry Pi 3 device architecture. Finally, the current system utilizes a randomly set sampling rate; the authors aim to identify methods for selecting the optimal sampling rate to ensure fast detection times and reduced packet latency.

8 Future Work

In addition to measuring how effective a system is at detecting and mitigating DDoS attacks, another security effectiveness metric is the false positive rate, which measures how often legitimate users are incorrectly identified as attackers. A high false positive rate can annoy legitimate users with recurrent disconnections, which can significantly weaken network availability. Although this study did not explicitly measured the FPR, reducing false positives remains a critical goal for improving the accuracy and reliability of SDN-based DDoS mitigation systems. Future studies will consider FPR analysis.

Acknowledgement: The researchers would like to thank the Deanship of Graduate Studies and Scientific Research at Qassim University for financial support (QU-APC-2025).

Funding Statement: This work is supported by the Deanship of Graduate Studies and Scientific Research at Qassim University for financial support (QU-APC-2025).

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Faisal Mohammed Alotaibi, Abdullah M. Alnajim; methodology, Faisal Mohammed Alotaibi, Abdullah M. Alnajim; software, Faisal Mohammed Alotaibi, Abdullah M. Alnajim; validation, Abdullah M. Alnajim, Sheroz Khan; formal analysis, Abdullah M. Alnajim, Faisal Mohammed Alotaibi, Sheroz Khan; investigation, Abdullah M. Alnajim, Faisal Mohammed Alotaibi; resources, Faisal Mohammed Alotaibi, Abdullah M. Alnajim; data curation, Faisal Mohammed Alotaibi, Sheroz Khan; writing—original draft preparation, Faisal Mohammed Alotaibi, Sheroz Khan; writing—review and editing, Abdullah M. Alnajim, Faisal Mohammed Alotaibi, Sheroz Khan; visualization, Faisal Mohammed Alotaibi, Abdullah M. Alnajim, Sheroz Khan; supervision, Faisal Mohammed Alotaibi, Abdullah M. Alnajim, Sheroz Khan; project administration, Faisal Mohammed Alotaibi, Abdullah M. Alnajim, Sheroz Khan; funding acquisition, Faisal Mohammed Alotaibi, Abdullah M. Alnajim. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The researchers declare that there are no data and materials for this work.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Khan LU, Han Z, Saad W, Hossain E, Guizani M, Hong CS. Digital twin of wireless systems: overview, taxonomy, challenges, and opportunities. IEEE Communicat Surv Tutor. 2022;24(4):2230–54. doi:10.1109/COMST.2022.3198273.
2. Goransson P, Black C, Culver T. Software defined networks: a comprehensive approach. San Francisco, CA, USA: Morgan Kaufmann; 2016.
3. Sarwar A, Alnajim AM, Marwat SN, Ahmed S, Alyahya S, Khan WU. Enhanced anomaly detection system for iot based on improved dynamic SBPSO. Sensors. 2022;22(13):4926. doi:10.3390/s22134926.
4. Bhattacharyya DK, Kalita JK. DDoS attacks: evolution, detection, prevention, reaction, and tolerance. Boca Raton, FL, USA: CRC Press; 2016.
5. Agrawal N, Tapaswi S. Defense mechanisms against DDoS attacks in a cloud computing environment: state-of-the-art and research challenges. IEEE Commun Surv Tutor. 2019;21(4):3769–95.
6. Wang A, Chang W, Chen S, Mohaisen A. Delving into internet DDoS attacks by botnets: characterization and analysis. IEEE/ACM Transact Network. 2018;26(6):2843–55. doi:10.1109/TNET.2018.2874896.
7. Hussain M, Shah N, Amin R, Alshamrani SS, Alotaibi A, Raza SM. Software-defined networking: categories, analysis, and future directions. Sensors. 2022;22(15):5551. doi:10.3390/s22155551.
8. Dayal N, Srivastava S. Analyzing behavior of DDoS attacks to identify DDoS detection features in SDN. In: 2017 9th International Conference on Communication Systems and Networks (COMSNETS). Bengaluru, India: IEEE; 2017. p. 274–81.
9. Qaddos A, Yaseen MU, Al-Shamayleh AS, Imran M, Akhunzada A, Alharthi SZ. A novel intrusion detection framework for optimizing IoT security. Sci Rep. 2024;14(1):21789. doi:10.1038/s41598-024-72049-z.
10. Aladaileh MA, Anbar M, Hasbullah IH, Chong YW, Sanjalawe YK. Detection techniques of distributed denial of service attacks on software-defined networking controller—a review. IEEE Access. 2020;8:143985–95. doi:10.1109/ACCESS.2020.3013998.
11. Rawat DB, Reddy SR. Software defined networking architecture, security and energy efficiency: a survey. IEEE Communicat Surv Tutor. 2016;19(1):325–46. doi:10.1109/COMST.2016.2618874.
12. Tr O. Principles and practices for securing software-defined networks. Palo Alto, CA, USA: Open Networking Foundation; 2015.
13. Batool S, Khan FZ, Shah SQA, Ahmed M, Alroobaea R, Baqasah AM, et al. [Retracted] lightweight statistical approach towards TCP SYN Flood DDoS attack detection and mitigation in SDN environment. Secur Commun Netw. 2022;2022(1):2593672.
14. Sangodoyin AO, Akinsolu MO, Pillai P, Grout V. Detection and classification of DDoS flooding attacks on software-defined networks: a case study for the application of machine learning. IEEE Access. 2021;9:122495–508. doi:10.1109/ACCESS.2021.3109490.
15. Aslam N, Srivastava S, Gore MM. A comprehensive analysis of machine learning-and deep learning-based solutions for DDoS attack detection in SDN. Arab J Sci Eng. 2024;49(3):3533–73. doi:10.1007/s13369-023-08075-2.
16. Eliyan LF, Di Pietro R. DoS and DDoS attacks in software defined networks: a survey of existing solutions and research challenges. Future Gener Comput Syst. 2021;122(3):149–71. doi:10.1016/j.future.2021.03.011.
17. Douligeris C, Mitrokotsa A. DDoS attacks and defense mechanisms: classification and state-of-the-art. Comput Netw. 2004;44(5):643–66. doi:10.1016/j.comnet.2003.10.003.
18. Carle G, Raumer D, Schwaighofer L. Future internet (FI) and innovative internet technologies and mobile communications (IITM). [cited 2025 Jan 1]. Available from: https://www.net.in.tum.de/publications/net/net-2015-09-1.html.
19. Mousavi SM, St-Hilaire M. Early detection of DDoS attacks against SDN controllers. In: 2015 International Conference on Computing, Networking and Communications (ICNC). Anaheim, CA, USA: IEEE; 2022. p. 77–81.
20. Keromytis AD, Misra V, Rubenstein D. Using overlays to improve network security. In: Scalability and traffic control in IP networks II. Vol. 4868. Bellingham, DC, USA: SPIE; 2002. p. 245–54.

21. Wang H, Zhang D, Shin KG. Detecting SYN flooding attacks. In: Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. New York, NY, USA: IEEE; 2002. p. 1530–9.
22. Limwiwatkul L, Rungsawang A. Distributed denial of service detection using TCP/IP header and traffic measurement analysis. In: IEEE International Symposium on Communications and Information Technology 2004 (ISCIT 2004). Sapporo, Japan: IEEE; 2004. p. 605–10.
23. Chemalamarri VD, Braun R, Abolhasan M. Constraint-based rerouting mechanism to address congestion in software defined networks. In: 2020 30th International Telecommunication Networks and Applications Conference (ITNAC). Melbourne, VIC, Australia: IEEE; 2020. p. 1–6.
24. Monika, Kaushik A, Shekhar M. Network simulators for next generation networks: an overview. Int J Mob Netw Commun Telemat. 2014;4(4):39–51. doi:10.5121/ijmnct.2014.4404.
25. Doshi K, Yilmaz Y, Uludag S. Timely detection and mitigation of stealthy DDoS attacks via IoT networks. IEEE Transact Depend Secure Comput. 2021;18(5):2164–76. doi:10.1109/TDSC.2021.3049942.
26. Aslam M, Ye D, Tariq A, Asad M, Hanif M, Ndzi D, et al. Adaptive machine learning based distributed denial-of-services attacks detection and mitigation system for SDN-enabled IoT. Sensors. 2022;22(7):2697. doi:10.3390/s22072697.
27. Alsaeedi M, Mohamad MM, Al-Roubaiey AA. Toward adaptive and scalable OpenFlow-SDN flow control: a survey. IEEE Access. 2019;7:107346–79. doi:10.1109/ACCESS.2019.2932422.
28. Gupta N, Maashi MS, Tanwar S, Badotra S, Aljebreen M, Bharany S. A comparative study of software defined networking controllers using mininet. Electronics. 2022;11(17):2715. doi:10.3390/electronics11172715.
29. Nugraha M, Paramita I, Musa A, Choi D, Cho B. Utilizing OpenFlow and sFlow to detect and mitigate SYN flooding attack. J Korea Multimedia Soc. 2014;17(8):988–94. doi:10.9717/kmms.2014.17.8.988.
30. Lin H, Wang P. Implementation of an SDN-based security defense mechanism against DDoS attacks. In: Proceedings of the 2016 Joint International Conference on Economics and Management Engineering (ICEME 2016) and International Conference on Economics and Business Management (EBM 2016); 2016; Philadelphia, PA, USA.
31. Singh J, Behal S. Detection and mitigation of DDoS attacks in SDN: a comprehensive review, research challenges and future directions. Comput Sci Rev. 2020;37(2):100279. doi:10.1016/j.cosrev.2020.100279.
32. Wang Y, Hu T, Tang G, Xie J, Lu JSGS. Safe-guard scheme for protecting control plane against DDoS attacks in software-defined networking. IEEE Access. 2019;7:34699–710. doi:10.1109/ACCESS.2019.2895092.
33. Wani A, Khaliq R. SDN-based intrusion detection system for IoT using deep learning classifier (IDSIoT- SDL). CAAI Transact Intell Technol. 2021;6(3):281–90. doi:10.1049/cit2.12003.
34. Varghese JE, Muniyal B. An efficient IDS framework for DDoS attacks in SDN environment. IEEE Access. 2021;9:69680–99. doi:10.1109/ACCESS.2021.3078065.
35. Haider S, Akhunzada A, Mustafa I, Patel TB, Fernandez A, Choo K-KR, et al. A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks. IEEE Access. 2020;8:53972–83. doi:10.1109/ACCESS.2020.2976908.
36. Dong S, Abbas K, Jain R. A survey on distributed denial of service (DDoS) attacks in SDN and cloud computing environments. IEEE Access. 2019;7:80813–28. doi:10.1109/ACCESS.2019.2922196.
37. Lim S, Ha J, Kim H, Kim Y, Yang S. A SDN-oriented DDoS blocking scheme for botnet-based attacks. In: 2014 Sixth International Conference on Ubiquitous and Future Networks (ICUFN). Shanghai, China: IEEE; 2014. p. 63–8.
38. Phan TV, Bao NK, Park M. novel hybrid flow-based handler with DDoS attacks in software-defined networking. In: 2016 International IEEE Conferences on Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld). Toulouse, France: IEEE; 2016. p. 350–7.
39. Tedeschi P, Sciancalepore S, Di Pietro R. Satellite-based communications security: a survey of threats, solutions, and research challenges. Comput Netw. 2022;216(18):109246. doi:10.1016/j.comnet.2022.109246.
40. Macedo R, de Castro R, Santos A, Ghamri-Doudane Y, Nogueira M. Self-organized SDN controller cluster conformations against DDoS attacks effects. In: 2016 IEEE Global Communications Conference (globecom). Washington, DC, USA: IEEE; 2016. p. 1–6.

41. Yan Q, Gong Q, Yu FR. Effective software-defined networking controller scheduling method to mitigate DDoS attacks. Electr Letters. 2017;53(7):469–71. doi:10.1049/el.2016.2234.
42. Zhang P, Wang H, Hu C, Lin C. On denial of service attacks in software defined networks. IEEE Network. 2016;30(6):28–33. doi:10.1109/MNET.2016.1600109NM.
43. Xu Y, Liu Y. DDoS attack detection under SDN context. In: IEEE INFOCOM 2016-the 35th Annual IEEE International Conference on Computer Communications; San Francisco, CA, USA: IEEE; 2016. p. 1–9.
44. Josbert NN, Wei M, Wang P, Rafiq A. A look into smart factory for Industrial IoT driven by SDN technology: a comprehensive survey of taxonomy, architectures, issues and future research orientations. J King Saud Univ-Comput Inf Sci. 2024;36(5):102069. doi:10.1016/j.jksuci.2024.102069.
45. Buzzio-García J, Vergara J, Ríos-Guiral S, Garzón C, Gutiérrez S, Botero JF, et al. Exploring traffic patterns through network programmability: introducing sdnflow, a comprehensive openflow-based statistics dataset for attack detection. IEEE Access. 2024;12:42163–80. doi:10.1109/ACCESS.2024.3378271.
46. Luo Z, Dai X. Reinforcement learning-based computation offloading in edge computing: principles, methods, challenges. Alex Eng J. 2024;108(6):89–107. doi:10.1016/j.aej.2024.07.049.
47. Rahdari A, Jalili A, Esnaashari M, Gheisari M, Vorobeva AA, Fang Z, et al. Security and privacy challenges in SDN-enabled IoT systems: causes, proposed solutions, and future directions. Comput Mater Contin. 2024;80(2):2511–33. doi:10.32604/cmc.2024.052994.