sustainability

MDPI

Article

A Security Policy Protocol for Detection and Prevention of Internet Control Message Protocol Attacks in Software Defined Networks

Edeh Michael Onyema [1,2], M. Anand Kumar [3], Sundaravadivazhagn Balasubaramanian [4], Salil Bharany [5,*], Ateeq Ur Rehman [6], Elsayed Tag Eldin [7] and Muhammad Shafiq [8,*]

1 Department of Vocational and Technical Education, Faculty of Education, Alex Ekwueme Federal University, Ndufu-Alike, Abakaliki P.M.B. 1010, Nigeria
2 Adjunct Faculty, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai 602105, India
3 Department of Computer Applications, Graphic Era (Deemed to be University), Dehradun 248002, India
4 Department of Information Technology, University of Technology and Applied Sciences, P.O. Box 191, Al Mussanah 314, Oman
5 Department of Computer Engineering & Technology, Guru Nanak Dev University, Punjab 143005, India
6 Department of Electrical Engineering, Government College University, Lahore 54000, Pakistan
7 Faculty of Engineering and Technology, Future University in Egypt New Cairo, New Cairo 11835, Egypt
8 Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, Korea
* Correspondence: salil.bharany@gmail.com (S.B.); shafiq@ynu.ac.kr (M.S.)

Abstract: Owing to the latest advancements in networking devices and functionalities, there is a need to build future intelligent networks that provide intellectualization, activation, and customization. Software-defined networks (SDN) are one of the latest and most trusted technologies that provide a method of network management that provides network virtualization. Although traditional networks still have a strong presence in the industry, software-defined networks have begun to replace them at faster rates. When network technologies emerge at a steady rate, SDN will be implemented at higher rates in the upcoming years in all fields. Although SDN technology removes the complexity of tying control and data plane together over traditional networks, certain aspects such as security, controllability, and economy of network resources are vulnerable. Among these aspects, security is one of the main concerns that are to be viewed seriously as far as the applications of SDN are concerned. This paper presents the most recent security issues SDN environment followed by preventive mechanisms. This study focuses on Internet control message protocol (ICMP) attacks in SDN networks. This study proposes a security policy protocol (SPP) to detect attacks that target devices such as switches and the SDN controller in the SDN networks. The mechanism is based on ICMP attacks, which are the main source of flooding attacks in the SDN networks. The proposed model focuses on two aspects: security policy process verification and client authentication verification. Experimental results shows that the proposed model can effectively defend against flooding attacks in SDN network environments.

Keywords: bandwidth; attacks; controller; flooding; ICMP; security; software-defined networks; virtualization

Citation: Onyema, E.M.; Kumar, M.A.; Balasubaramanian, S.; Bharany, S.; Rehman, A.U.; Eldin, E.T.; Shafiq, M. A Security Policy Protocol for Detection and Prevention of Internet Control Message Protocol Attacks in Software Defined Networks. *Sustainability* **2022**, *14*, 11950. https://doi.org/10.3390/su141911950

Academic Editors: Limei Peng, Yi Sun and Ali Kashif Bashir

Received: 26 August 2022
Accepted: 16 September 2022
Published: 22 September 2022

1. Introduction

The latest advancements in software-defined networks (SDN) have provided new mechanisms for simplified network operations [1]. This approach provides a simple abstraction mechanism for network operators by removing the complexity of the network topology. The separation of hardware and software components provides a flexible way to design and program a network [2]. The complexity of switches in the network is further reduced using the concept of dynamic and adaptive network management. The control

system and monitoring are part of the controller, and the components of the data plane are organized according to the instructions of the controller.

The SDN controller collects all information on the network topology from components such as host location, link status, and switch information. The collected information is used by the controller for further processing [3]. The developers can modify the SDN controller task by including their inventions, strategies, and applications at the top of the controller. The open-flow protocol acts as an interface between the switch and controller for data transmission [4]. The main responsibility of the controller is to define the traffic flow between components in the network. The following Figure 1 shows the architecture of SDN.

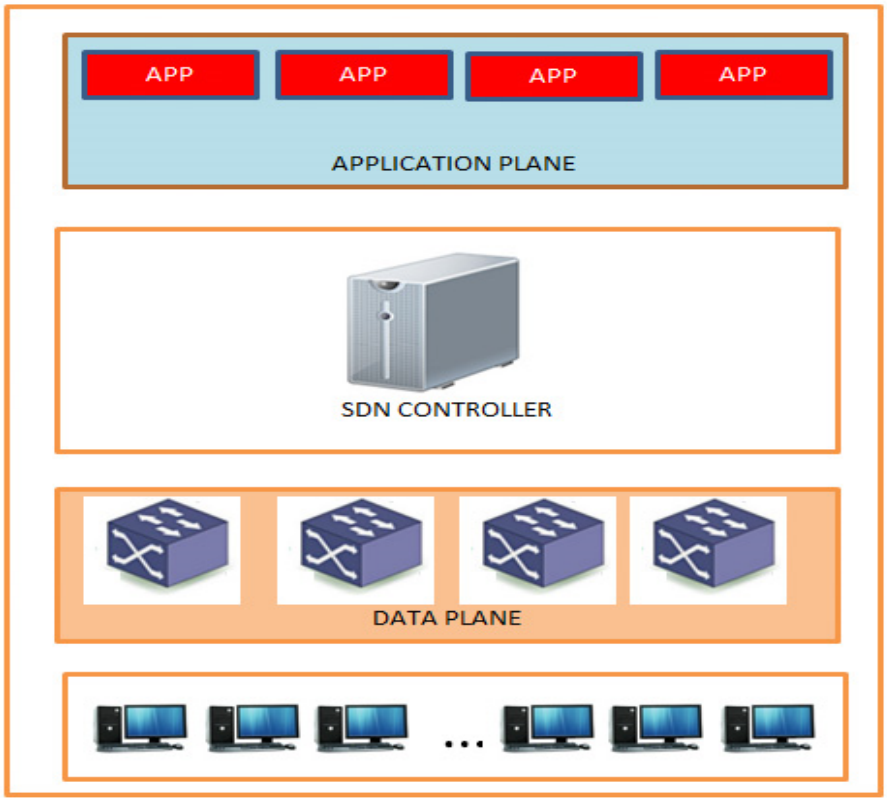

Figure 1. SDN architecture.

The application layer of a software-defined network comprises several network tools, devices, and a variety of commercial applications that interact with the control layer through an SDN controller [5]. The infrastructure layer is the foundation layer of the SDN architecture. The main responsibility of this layer is to forward network traffic and collect network statistics, usage, and network topology. This layer is responsible for handling packets constructed in the direction of the SDN controller [6]. It contains virtual and physical network equipment such as routers, switches, and other network devices that are used to forward network traffic. The control layer acts as the interface between the application and infrastructure layers. The northbound interface enables the communication between the applications and the SDN controller. The southbound interface is responsible for communication between the SDN controller and the infrastructure layer. The application layer instructions are processed through the southbound interface and network components through a southbound interface [7].

Traditional networks consist of static protocols for network equipment, such as routers and switches, where it is not possible to implement network protocols. Therefore network administrators find it difficult to define custom routing protocols [8]. The SDN controller eliminates routing issues in the SDN network with the use of logical connections. This technology opens a gateway for clients to track environmental changes. Once clients in

the network have complete controller information, they can execute a malicious attack on the controller [9]. This is a huge challenge for administrators in preserving the network information in SDN networks. The first goal of this research was to distinguish malicious attacks in SDN networks. The second goal is to provide a security policy protocol to avoid unauthorized attacks on the network. The final goal was to evaluate the proposed model in a different scenario with several parameters to identify the suitability of the model for deployment in an SDN environment.

The remainder of this paper is organized as follows. Section 2 presents recent works done in the software-defined networks followed by SDN security in Section 3. Section 4 details the Internet control message protocol, followed by ICMP attacks in Section 5. Section 6 presents the proposed methodology, followed by experimental setup in Section 7. Section 8 presents experiments and results. Finally Section 9 concludes the paper.

The contribution of the paper includes but not limited to:

- The study highlights ICMP protocol-based attacks and their impact on SDN environments.
- It presents a new security policy protocol (SPP) and client authentication model to avoid unauthorized attacks on SDN networks.
- The proposed solution proved to be accurate in tackling potential attacks in SDN and the performance metrics were evaluated using parameters such as CPU utilization, channel bandwidth, packet delivery ratio, response time, and a number of flow requests.
- The proposed model has the capability to detect different attacks including ping flood and Smurf attacks that often originate from undefended legacy equipment.
- It adds to the knowledge relating to the security of Internet control message protocol and indeed cybersecurity in general.

2. Literature Review

Security has been identified as an unnerving task in communication networks because of the nature of underlying network complexities and parameter-based security solutions that are difficult to manage [10]. The authors of ref. [11] proposed an approach to divert traffic from an attacked device and remove unwanted instructions from the attacked switch. However, there is no consideration of attacks on controller resources, which is an important security aspect of SDN.

Wang et al. [12] stated that sniffing a network is possible without any significant impact on the SDN controller. This work is not suitable for detecting or preventing slow attacks on the controllers. The authors of ref. [13] presented an approach in which the controller checks the authentication for every incoming packet and installs certain instructions to prevent the intruder from using the underlying network resources. These methods require more instructions to be stored in the affected device, which is more vulnerable for the controller and affects the performance.

The authors of [14] addressed several security treads in the control plane of an SDN. This study also proposed an enhanced security framework based on attribute-based encryption. Tree-structure-based encryption was used to achieve a fine-grained access control mechanism for SDN. Liang et al. [15] presented a security architecture for the SDN-based 5G networks. They focus mainly on mobile networks by implementing network and security domains with a low degree of coupling, which makes it easy to deploy the services or equipment without disturbing normal functionality.

The authors of [16] presented a pictorial model for attack detection using a graph theory approach. An attack path prediction model was developed to identify critical components and devices in an SDN network. They have mainly focused on reconnaissance, topology poisoning, and forensic attacks. Vijay et al. [17] proposed a hybrid architecture with a security management application on the SDN controller to detect the attacking device before a request is sent from the attacker host to the controller. They also addressed the dynamic management of security policies for data planes for flooding and injection

attacks. This study focused only on a single SDN domain where there will be a limited number of attack types.

The authors of ref. [18] presented an approach based on the extension of the controller to deal with only topology poisoning attacks using fingerprint methods of the device for authentication. The main issue with this approach is that all fingerprints should be maintained only by the controller, which will create more burdens for the controller in complex environments. The authors of [19] presented a policy-based security architecture for distributed SDN network platforms. They implemented an access policy rule to validate the MAC address and the original IP address of the end devices such as switches to drop the packet when the address is spoofed. They primarily focused on man-in-middle and spoofing attacks.

Hau et al. [20] addressed the integrity issues of the link layer discovery protocol, which is primarily used in network topology discovery. They proposed a detection algorithm for worm-hole attacks based on path latencies in SDN environments using three topologies: Nsfcnet, Shentel, and Neol. They also introduced a gravity model to generate network traffic by using real data. The authors of [21] addressed DDoS and IP spoofing attacks in SDN environments and proposed a variable security management solution. They developed an abstract grammar to implement security policies with compilers and employed an optimal algorithm to place the rules across the switches to avoid unwanted traffic.

Most recent studies focused on either diverting network traffic to remove unwanted instructions from attacked devices or using encryption methods to fine-grain the access control of DCN environments. From the literature, it was identified that using security policy mechanisms is an ideal solution for the detection of spoofing attacks in SDN networks and to prevent them from taking full control over network environments. This study focuses on ICMP attacks, namely man-in-the-middle attacks and flooding attacks, which are critical security attacks in SDN environments.

3. Software Defined Network Security

Software-defined networks provide capable solutions for handling the complications of traditional networks in the modern era. Although these models offer more advantages for concerned organizations, attackers can execute different forms of attacks in SDN environments [22]. The controller is a vital component used by the attacker to execute security attacks. Mischievous traffic can be generated to attack the controller and control plane communication. Once this is completed, the clients that are connected to the switches can execute the attacks. Flooding attacks are critical attack that fails in an entire network.

These attacks target flooding the control plane first and then the data plane and SDN controller bandwidth. Because the controller acts as the intelligence agent of the entire network that controls a large number of devices and applications, attacks block the entire traffic and fill up the total memory of the SDN switch [23]. Once the total memory is full, it is not possible to accept any new upcoming requests or configure the rules from the SDN controller, which leads packet dropping. The main reason behind this is that the degree of inward flows is very high because of malicious requests, which make the buffer memory full, leading to higher bandwidth consumption.

Attackers use different approaches to execute attacks, including network-based approaches such as ICMP, UDP, or TCP packets, to exploit the memory structure, algorithms, or authentication protocols [24]. Figure 2 presents the attack scenario in which the ICMP protocol is used to flood the controller bandwidth from host D. Switch S2 and the controller are the victims where, after a certain time interval, the entire traffic will be congested, and the new request will be discarded. These types of attacks not only affect specific hosts but also all the devices in SDN environments, as shown in Figure 3. Figure 4 clearly shows the utilization of available resources by ICMP attacks at specific time intervals, and Figure 5 shows the effects of ICMP on total traffic. This clearly shows that, in a short period, ICMP attacks use 90% of the available bandwidth in the entire network.

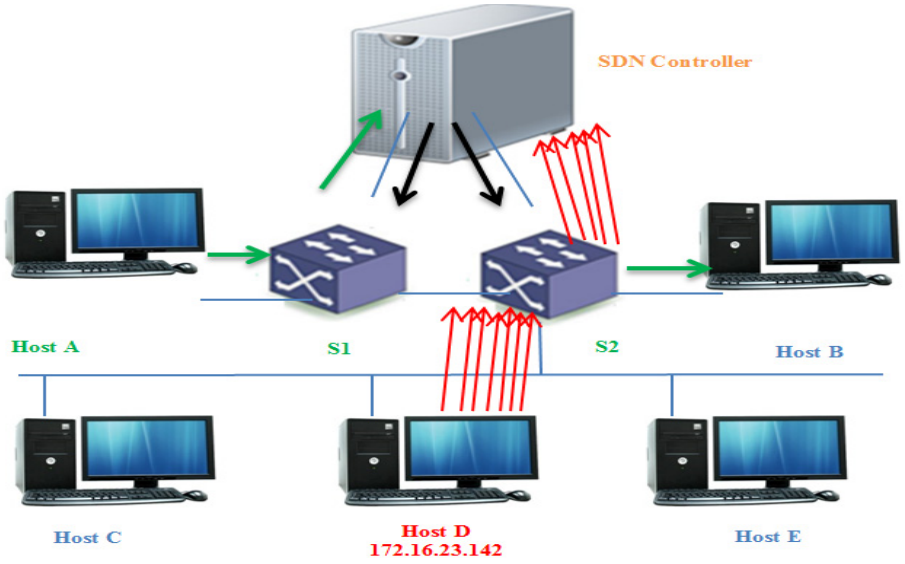

Figure 2. ICMP attack scenario.

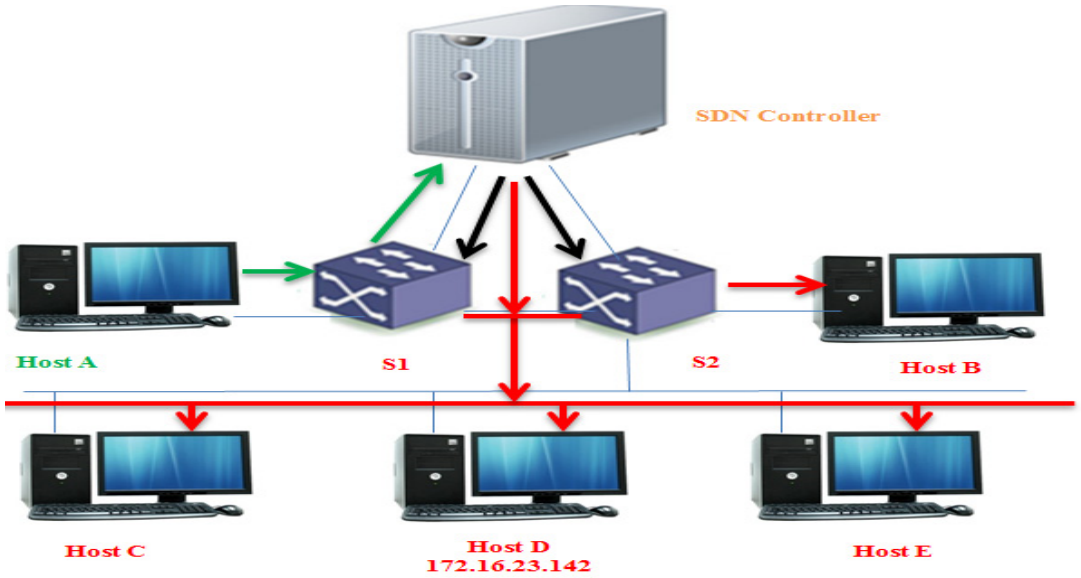

Figure 3. ICMP attack scenario (entire network).

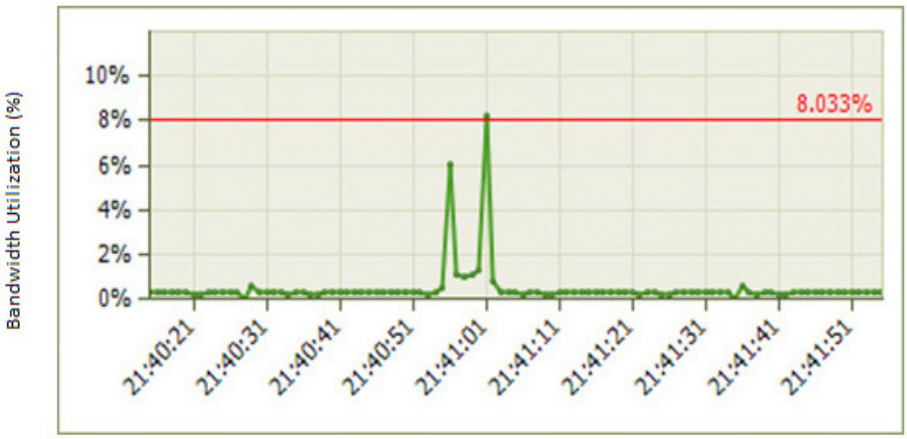

Figure 4. Bandwidth utilization in a host.

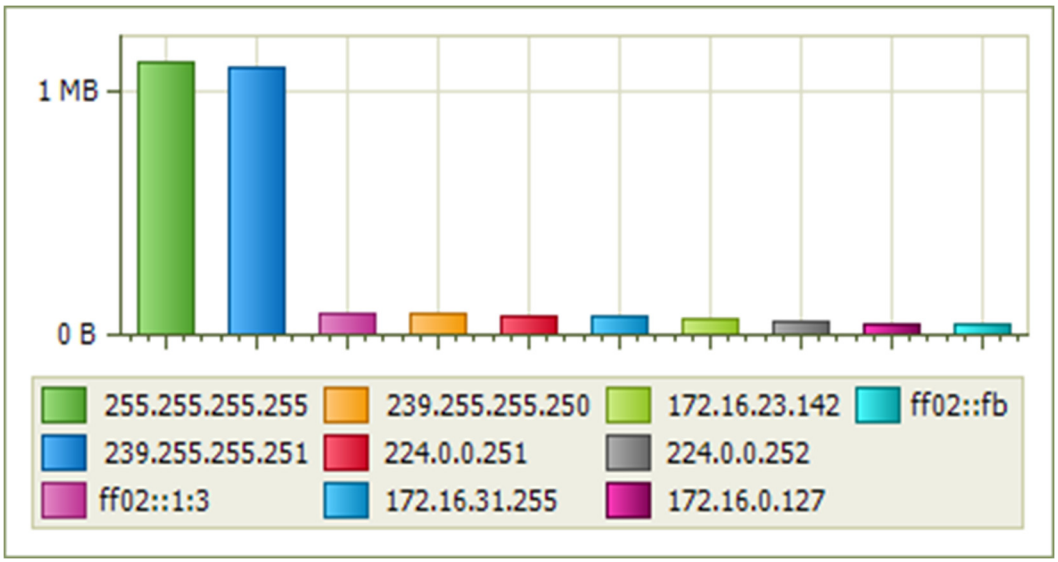

Figure 5. Traffic during ICMP attacks.

The attacker continuously sends ICMP request messages to the destination machine over the network. This makes the host busy replying the ICMP request messages. This leads to unwanted flooding in the network and degrades the network performance. The host machine with an IP of 172.16.23.142 is attacked by the victim, as shown in Figure 3. The host is used to send unwanted traffic in the form of a request to switch (S2) in the SDN network. Within a couple of minutes, the host sends n requests to the switch and creates unwanted traffic over the entire network. The following Figure 6 shows the situation of the attack. In this scenario, the other protocols are affected by the attack. HTTP protocol was used unnecessarily in this attack scenario to send and receive the resources. Figure 7 shows the protocol distribution for the attack scenario.

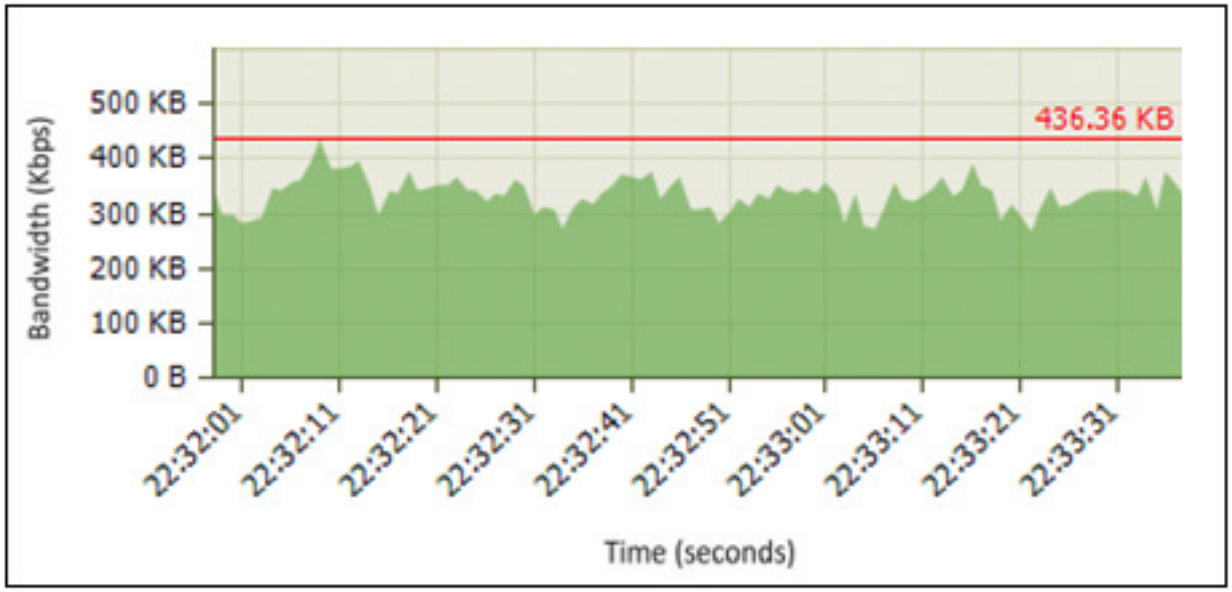

Figure 6. ICMP packet request.

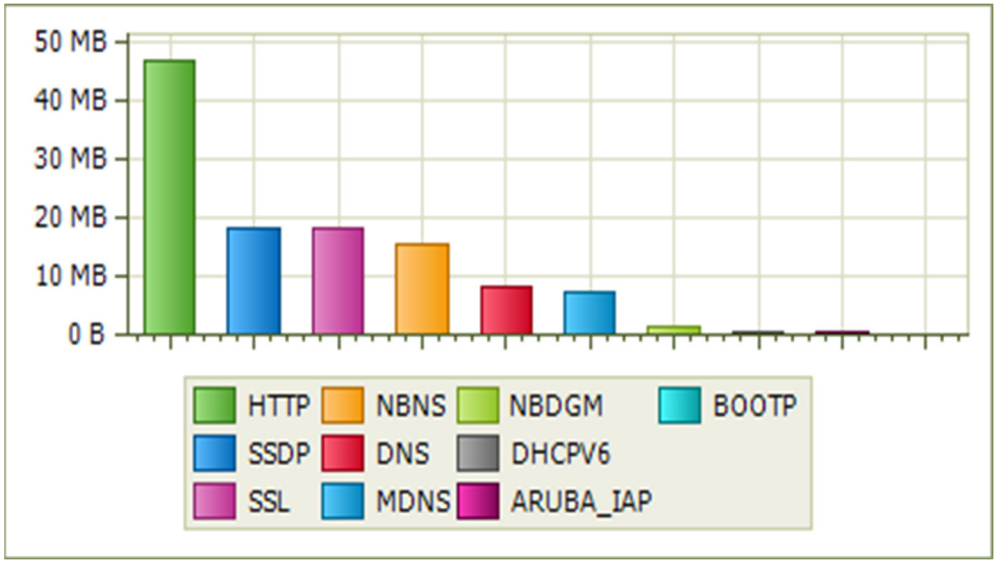

Figure 7. Protocol distribution during attack scenario.

4. Internet Control Message Protocol

All IP-enabled end systems and intermediate devices such as routers frequently use the ICMP protocol for troubleshooting the network [25]. The ICMP protocol is used to report issues in the network or intermediate devices such as routers, hubs, and switches. Some of the important features of ICMP protocols are reporting when end systems (ES) do not respond to the request, congestion in the network, IP header issues, and other network-related issues. The protocol is frequently used by network administrators to track the working functionality of end systems (ES). It is also used to check whether the router correctly direct packets to their intended destination. Its communications are not transferred directly to the data link layer; although it belongs to the network layer, messages will be encoded into IP datagrams before being sent to the lowest layer. The protocol field has a value of one, indicating that the IP data are an ICMP message category. Consequently, ICMP is primarily concerned with a mechanism for any IP-enabled device to deliver error messages to another IP machine in the network. ICMP has several message formats that allow the transfer of different types of data. In response to the message delivered by Host0 to Host1 and transmitted by router0, router R1 generates ICMP packets. When the MTU value of the link between routers 0 and router1 is less than the size of the IP packet, and when the packet has a do not fragment (DS) bit in the IP packet header, the ICMP message delivered to the Host0.

A. ICMP MESSAGES

One of the most significant protocols of the TCP/IP protocol suite is the Internet control message protocol (ICMP). It is mostly used by the underlying platform to transmit error messages to devices in the network. ICMP [26] is a critical element of the IP that must function. It differs from TCP and UDP in that it is rarely utilized for data transmission between the end systems. User network programs or devices rarely use this protocol, except for ping and traceroute commands. Unannounced network flaws, such as the inaccessibility of a host or a network portion owing to a malfunction, are among these issues.

ICMP sends a TCP packet or UDP packet to the specified port number in the network without any destination information. The router in the network buffers the packet when there are more packets to be transmitted within a specific time interval to assist in the troubleshooting process. The echo function in ICMP simply sends a message back and forth between the two hosts [27]. A ping command is a popular network administration tool for determining the availability of the device in the network. The ping sends out a series of packets to calculate the loss percentages and average round-trip times. Timeouts should be announced. When the TTL field of an IP packet is zero, the router or any intermediate

device discards the packet from the network and sends an ICMP message to the source to denote the packet delivery issue to the destination. Trace-route is a command that uses tiny TTL packets to map network pathways while monitoring ICMP timeout discoveries.

B. ICMP MESSAGE TYPES

Network errors are reported to the host using ICMP messages. Faults may be in the network, router, or any other intermediate device. The source can quickly determine the cause of the errors by observing these types of messages. Query and error reporting are two types of ICMP messages that can be used to troubleshoot network issues. When intermediate devices such as the host or router process an IP packet, these error reporting schemes can report the errors encountered. Destination inaccessibility, source-quench, time exceed, parameter problem, and redirection are some of the error reporting messages provided by the ICMP protocol to the host devices or routers [28].

A pair of query messages will assist intermediate devices, such as hosts, routers, or network managers in obtaining error-related information from a host or router in the network [29–34]. Devices in the network can locate any router and collect router information for further processing. Even routers can assist devices (hosts) with redirection messages using updated information about the router and routing table. Echo messages, timestamps, router advertisements, and solicitation are the message types provided by the query message of the ICMP protocol [35–38]. The following are some key points to remember regarding the ICMP error messages:

(1) ICMP Messages will not be generated for the messages that contain error messages of ICMP type.
(2) There is no provision for using the ICMP error messages for fragmented datagram.
(3) ICMP messages will not be generated for messages that contain a multicast address.
(4) ICMP error reporting messages are not generated for a datagram that contains special address ranges such as 127.0.0.0 or 0.0.0.0.

C. ICMP MESSAGE FORMAT

An ICMP message's structure can be conceived as having a common component and a unique part [29]. The common part of all ICMP messages consists of three fields of the same size and meaning (but the values in the fields vary depending on the ICMP message type). Each message form has its own set of fields in unique portion. Figure 8 shows the ICMP packet format.

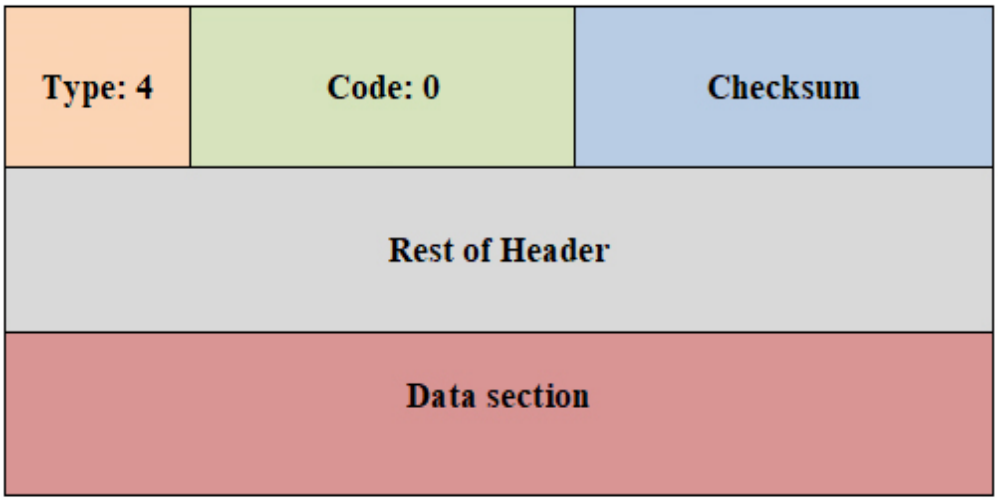

Figure 8. ICMP packet format.

Network troubleshooting involves identifying and resolving networking issues to maintain the best performance of a network. The primary role of a network administrator is to maintain network connectivity for all devices. To assist administrators, ICMP plays

a vital role in tracking the status of connections and improving their performance. ICMP can be used to accomplish this goal. First, ICMP traffic should be captured on the network to troubleshoot it. A network analyzer can be used to record all TCP/IP traffic while filtering the ICMP traffic. After configuring the network analyzer to filter the ICMP traffic, we examined the ICMP traffic that passes through the network. Although some redirect messages are common (especially during morning start-up hours), if one device is frequently routed before talking to other network devices, then it is necessary to designate a different default gateway.

5. Security Attacks in SDN

A. FLOODING ATTACKS

The services are affected by flooding attacks. Generally, denial of service (DoS) and distributed denial of service (DDoS) are common flooding attacks on SDN networks [30]. Server resources are exposed to the entire device in the network with this type of attack. It also slows down services that degrade the performance of servers and controllers, including memory, forwarding rates, and flow control in the SDN environment.

B. MAN-IN-MIDDLE ATTACKS

Most SDN networks permit third-party applications to be installed in a network. Thus, there is a possibility of active and passive attacks that directly route applications between the client and gateway. These applications provide false network information to the other devices in the network and initiate various attacks.

C. REPLICATION ATTACKS

This is considered to be one of the most dangerous attacks in an SDN network. The node was replicated in the network to manipulate a particular segment. As it is replicated, the node gains complete control over the network, including devices such as servers, switches, and controllers. It is very difficult to detect replicated nodes because of the complexity of the network.

D. NETWORK MANIPULATION ATTACK

Manipulation attacks provide false information in a network and execute the attack. Such attacks occur in the control plane and gain access to all the devices in network. Detecting manipulation attacks in SDN environments is very difficult owing to their complex nature.

E. TRAFFIC DIVERSION ATTACKS

These attacks are executed in the components of SDN networks, which redirect the traffic flow from a trusted path to a malicious path. Once this is done, the attacker can gain complete access to all components in the SDN network. After a certain period, the total services will be blocked.

Flooding attacks can be rectified in the SDN network using security middle boxes, such as IDS, anti-malware, and firewalls. These approaches should be integrated into virtualized environments to prevent security attacks. Several malware shields are available in the market to cope with the security challenges of SDN networks; however, the performance of such shields is very poor for massive attacks [30]. The use of machine learning classifiers also requires more computing resources, resulting in overhead for all network devices.

6. Proposed Methodology

The control plane waits until a stable topology is present in the network. Then, the control panel creates the forwarding table to send data from the source to the destination port via the forwarding plane. The client sends a request from the controller to the switch for certain services. Once the switch receives the request, it performs the following: (a) The SDN switch sends the request in message to the SDN controller, (b) drops packet in case of invalid authentication, and (c) provides a service based on the previous records. The main

aim of this research was to prevent malicious attacks from end hosts connected to SDN. It was also identified that if the nearest source of the attack was detected in the initial stage, then it would be possible to reduce the traffic to the controller and minimize the wastage of network bandwidth and complex computations. The proposed architecture consists of a security policy protocol (SPP) component that checks all incoming packets before they reach the controller. This component was placed close to the controller between the application and data planes. Thus, it is possible to implement security policies in the data plane. The proposed security component comprises a database that stores all authenticated host details. The database consists of complete details of authorized clients that are previously accessed resources in the SDN network. If the host requests a service for the first time, the SPP checks for a real IP address from the request and verifies it. Subsequently, it either adds an entry to the database or discards the packet. The data-path ID uniquely identifies devices in an SDN environment. SPP operates at two levels: (a) the security policy process and (b) client filtering process.

A. SECURITY POLICY PROCESS

The security policy process plays a vital role in providing security to all devices connected to an SDN network. In this study, 1 SDN controller, 16 intermediate switches, and 60 host machines were used in the initial stage. Once the security policies are formulated, it is possible to add multiple controllers and devices to the network. When a host requests a service from the SDN controller for the first time, the intermediate switch sends a packet-in message to the SPP, which in turn checks the packet header field information with the database, as shown in Algorithm 1. The host request is processed by the controller after the authentication process. Only authorized hosts are permitted to use the network resources. The SDN edge-switch drops the unauthorized packets from the network. Second, the SPP will check client filtering process to initiate a connection with the SDN controller. For instance, when the client initiates a connection to the SDN controller through switch S1, S1 queries SPP to identify the client. If the client passes the authentication by the SPP, it is allowed for the connection and other operations in the network; otherwise, the request is dropped by the end switch (S1). Table 1 shows the structure of the database using only the sample devices used in this experiment.

Algorithm 1 Security policy process

```
In Each Time Slot t
For Each Switch (S1, S2, . . . , Sn), I do
For each request r from Host (IP) arriving at Switch, do
IF the Host (IP) is in the SPSS Database SPSS Db, then
IF Host (IP) Equals (C_ID, S, DP_ID, PN, MAC)
Switch approves Request Grant (R_G) issues to Host (IP)
Else
IF Host (IP) = Real IP Address, then
Insert an entry in SPSS Db (C_ID, S, DP_ID, PN, MAC)
Switch approves Request Grant (R_G) issues to Host (IP)
Update SPSS
Else
Drop the Request r from the Host (IP)
End if
End if
End for
End for
```

Table 1. SPP Database.

Controller ID	Connection Status	State	Datapath–ID	Port	IP Address	MAC Address
C_ID_001	Connected	Active	000a009c02d81800	Port I	172.16.23.142	FC-4D-D4-49-87-4E
C_ID_001	Connected	Active	000a009c02d81800	Port I	172.16.3.201	fc-4d-d4-3c-ac-79
C_ID_001	Connected	Active	000a009c02d81800	Port I	172.16.5.109	78-e3-b5-b0-38-91
C_ID_001	Connected	Active	000a009c02d81800	Port I	172.16.7.41	fc-4d-d4-3b-34-ba
C_ID_001	Disconnected	No	0x000a44319261869e	Port II	172.16.8.144	68-14-01-25-d5-17
C_ID_001	Connected	Active	0x000a44319261869e	Port II	172.16.8.153	04-d4-c4-7d-fe-1e
C_ID_001	Connected	Active	0x000a44319261869e	Port II	172.16.9.38	48-e7-da-96-ac-e5

B. CLIENT FILTERING MECHANISM

Another important issue related to SDN is flooding attacks, as discussed in the previous section. ICMP attacks are considered dangerous malicious attacks that block machines or other network resources to end users. Most of the hosts/devices affected by ICMP attacks use high memory, CPU, and bandwidth, which will slow down the entire network and its components. To overcome these types of attacks, SPP uses a filtering algorithm to filter unwanted traffic in SDN. First, SPP monitors the total traffic in an SDN network. If unwanted traffic is found, for example, continuous ICMP flooding messages, filtering is executed to control the flow of ICMP requests in the network. Two different criteria were used to filter the ICMP packets: (i) If the number of ICMP packets exceeds 60, the filtering process applied for the check; (ii) if the size of the packet exceeds 78, the filtering process is applied as shown in the Algorithm 2. Two-way filtering is an ideal methodology for SDN. The following Figure 9 shows how the filtering process works in simulated SDN networks. In this experiment, both ICMP and IGMP packets were filtered by the SPP. There is no special condition check for the IGMP packets.

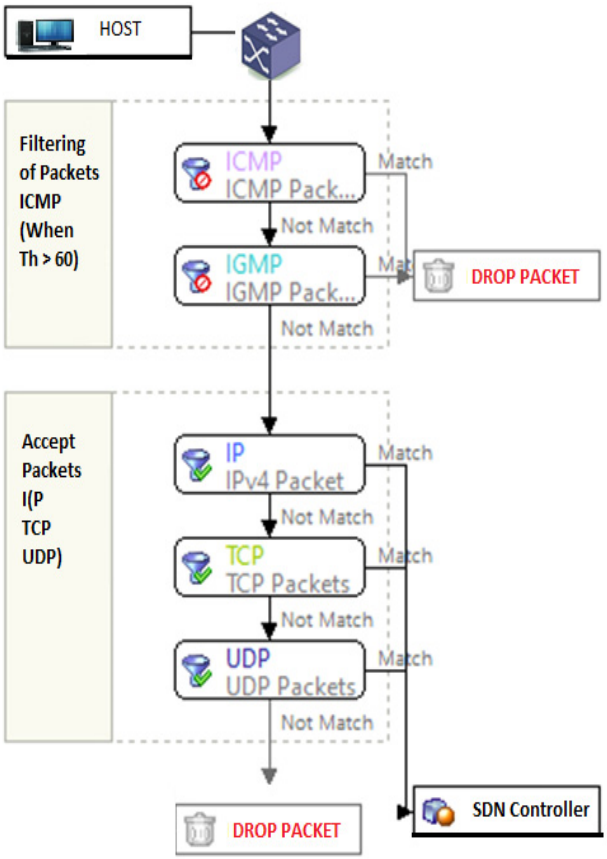

Figure 9. Filtering process of ICMP with a threshold value.

The filtering algorithm is executed only if the switch receives the maximum threshold value (n), where n = 60. If the value is less than 60, the ICMP packets are allowed in the network. An ICMP attack continuously generates a large number of flows with a small number of packets over a short period. Therefore, based on flow analysis, it is possible to determine the severity of ICMP attacks. The following formula was used to compute the percentages:

Of the ICMP packet is much smaller than that of the Ethernet frame size in the network. When the ICMP packets are affected, their size increases. It is possible to determine the degree of attack in the end switch based on the percentage of small bytes in ICMP packets.

$$FSB = \frac{\sum_{i}^{FlowSum} Flow_i\left(\overline{PacketsBytes}_i < T_{PB}\right)}{FlowSum} \quad (1)$$

Most of the ICMP flooding packets are invalid; thus, the corresponding flow rules issued by the SDN controller will not last for a long period before the timeout. The percentage of flow increased sharply over a short period. This can be determined using the following equation:

$$PFSD = \frac{\sum_{i}^{FlowSum} Flow_i(Duration_i < t)}{FlowSum} \quad (2)$$

Algorithm 2 Filtering process

```
In Each Time Slot t
For Each Switch (S1, S2, . . . , Sn), I do
For each ICMP request (IC_R) from host (IP) arriving at switch, do
IF the Host (IP) is in the SPSS Database SPSS Db, then
   IF Host(IP) Equals (C_ID, S, DP_ID, PN, MAC)
   Switch approves Request Grant (R_G) issues to Host(IP)
Else
IF (Host(IP) = Real IP Address) and IF (Threshold_val ≤ 60), then
IF (Packet_Size ≤ 78), then
   Switch Execute ICMP_Request Grant (R_G) issues to Host(IP)
   Update SPSS
   Else
   Drop the ICMP Request r from the Host(IP)
   End if
End if
End for
End for
```

7. Experimental Setup

The implementation was tested using Mininet simulator running on a virtual machine with Windows operating system. Hyper-V virtualization technology, which enables virtualized computer systems on the Windows platform, was used. Analytical modelling, measurement, and evaluation were identified as the three main approaches commonly used to evaluate communication network systems. The results of the evaluation were used to set the network performance indices given the traffic workload and network configuration. Sixty Core i7 CPU with 3.40 GHz, 1 IBM Intel server, 8 GB RAM, and Windows 64 bit operating systems were used for evaluation. The experimental topology is presented in the figure. To evaluate the proposed work, two existing models were used: RYU SDN Framework and the detection and mitigating DoS (DDS) [30–33,38–44]. Figure 10 shows the simulation topology.

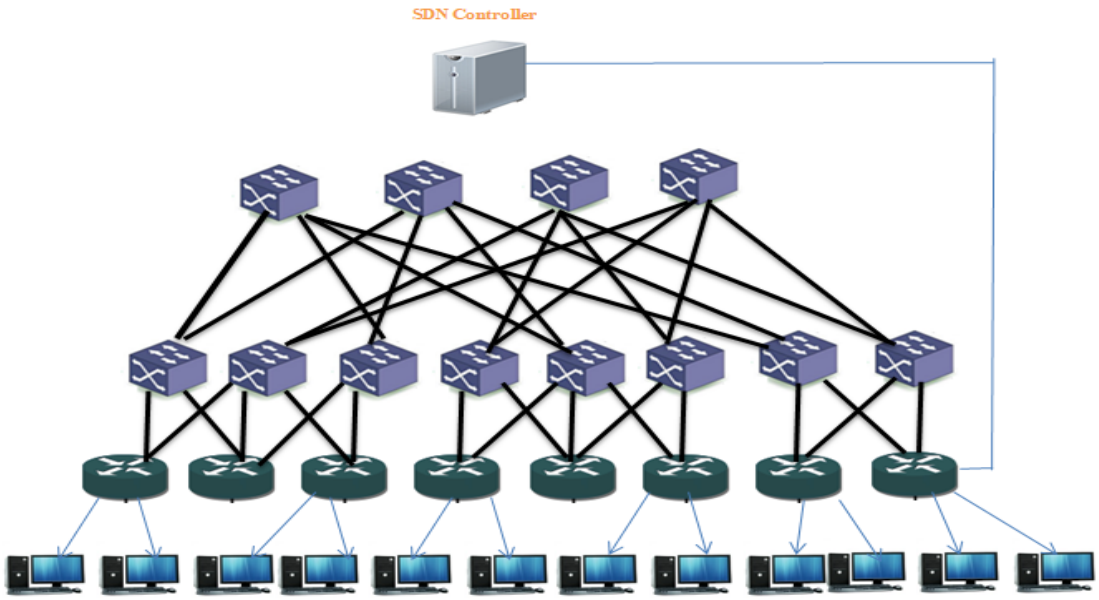

Figure 10. Simulation topology.

Before analyzing the impact of attacks in an SDN environment, threshold values were set for network components with different parameters. After the attacks were evaluated, these values were compared to obtaining accurate results. The threshold parameters were the CPU utilization, memory utilization, storage space, and throughput [44–47]. The following Table 2 presents the parameter values recorded before the attack scenario.

Table 2. Threshold Value before Analysis.

S. No	Parameter	Initial Threshold Values
1	CPU utilization	12%
2	Memory utilization	23%
3	Storage space	42%
4	Throughput	96%

8. Experiments and Results

The proposed model was evaluated using five parameters: CPU utilization, channel bandwidth, packet delivery ratio, response time, and number of flow requests. CPU utilization is important parameter for evaluating system performance in SDN networks. CPU utilization may vary depending on the deployment of additional security protocols particularly in SDN. This section presents some of the other parameters used to evaluate the proposed scheme.

A. CPU UTILIZATION

The total CPU processing power was used by ICMP attacks during the attack period. There will be a continuous installation of unwanted requests from the host machine in SDN networks during the attack period. Therefore, in this case, the normal traffic is affected and is only less provisional for trusted requests. The following Figure 11 shows a comparison of CPU utilization for the proposed model with RYU and DDS. The Table 3 presents the CPU utilization based on two sets of evaluation.

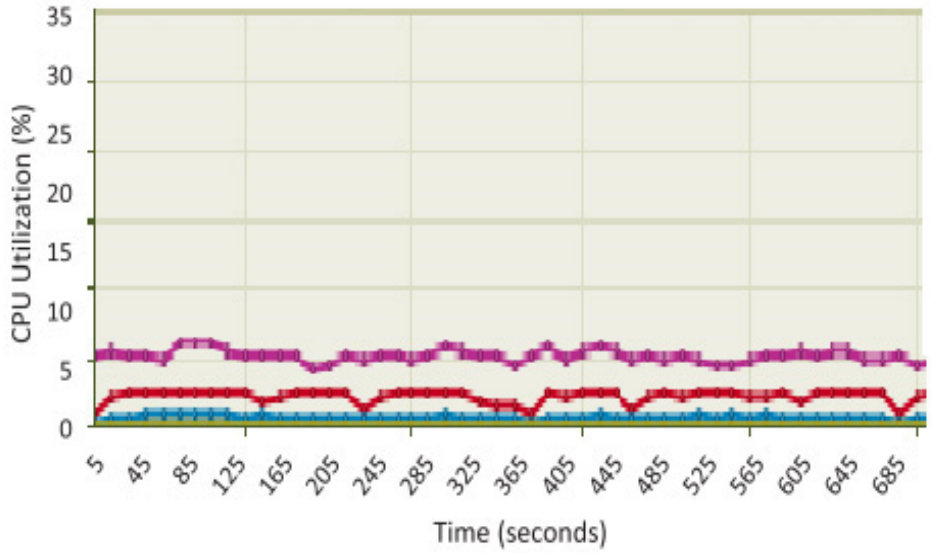

Figure 11. CPU Utilization.

Table 3. CPU Utilization.

Time in Seconds	Utilization (%)					
	Model (Test Run 1)			Model (Test Run 2)		
	RYU	DDS	SPP	RYU	DDS	SPP
100	7.2	3.2	0.5	7.1	3.2	0.5
200	5.0	3.2	0.5	5.0	3.4	0.5
300	6.1	2.3	0.6	6.0	2.6	0.6
400	7.2	2.5	0.5	7.2	2.5	0.5
500	5.7	2.5	0.7	5.5	2.4	0.6
600	7.2	3.8	0.7	7.2	3.5	0.7
700	7.3	2.5	0.6	7.1	2.4	0.6

B. PACKET DELIVERY RATIO

The packet delivery ratio is the ratio of the total packets sent by the source machine to the number of packets received by the destination machine. The packet loss ratio also plays a vital role in evaluating the packet delivery ratio. In our experiment, the TCP packets were sent from the source host to the destination host. The counter is used then to store the number of successful and unsuccessful packets. Table 4 presents the packet ratio units based on two sets of evaluation. The proposed scheme achieved a 98% delivery ratio compared with (87.25%) and DDS (73.25%) as shown in Figure 12. The formula used to calculate the delivery ratio is as follows:

$$FSB = \frac{Data\_Received}{Data - Received + Data\_loss} \quad (3)$$

Table 4. Packet Delivery.

Model	Packet Delivery (%)							
	Model (Test Run 1)				Model (Test Run 2)			
	Sent	Delivered	Dropped	Percentage	Sent	Delivered	Dropped	Percentage
RYU	54,325	53,719	4557	92	60,234	55,876	4358	93
DDS	45,378	44,765	6161	86.4	56,236	47,235	9001	84
SPP	51,123	51,109	873	98.2	65,000	64,129	871	98.6

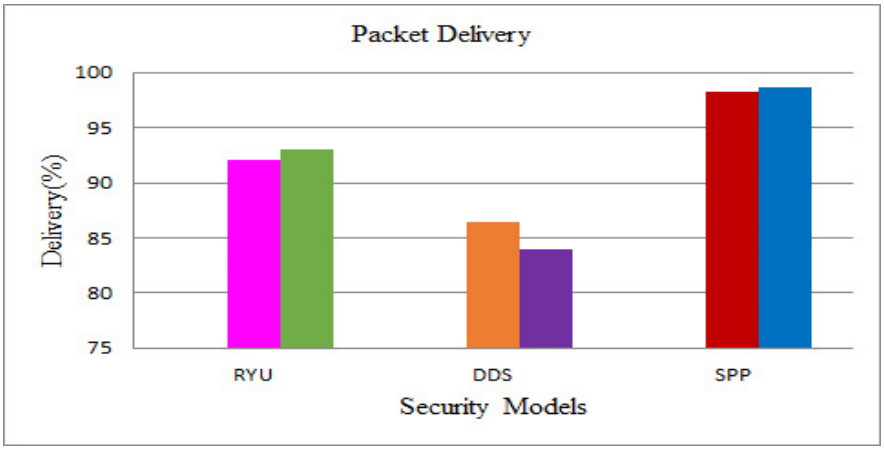

Figure 12. Packet delivery ratio.

C. CONTROL CHANNEL BANDWIDTH

When host requests pass through a control channel during ICMP attacks, the channel becomes unavailable owing to the lack of bandwidth. The proposed SPP protocol reduces the load by blocking malicious traffic when the threshold value reaches ≥ 60, as shown in the previous section (Figure 9). During the attack period, the bandwidth increased in both cases, whereas in the SPP model the bandwidth was constant. The reason for this is the blocking of malicious traffic in SDN. Table 5 presents the units for channel bandwidth. The proposed scheme achieved constant bandwidth as shown in Figure 13.

Table 5. Units for Channel Bandwidth.

Time in Seconds	Channel Bandwidth (Kbps)		
	Model (Test Run 1)		
	RYU	**DDS**	**SPP**
100	141.7	102.3	40.2
200	163.2	104.6	32.3
300	153.6	101.2	41.1
400	138.1	111.1	40.7
500	156.7	104.8	31.3
600	162.3	108.3	41.3
700	138.2	102.5	44.4

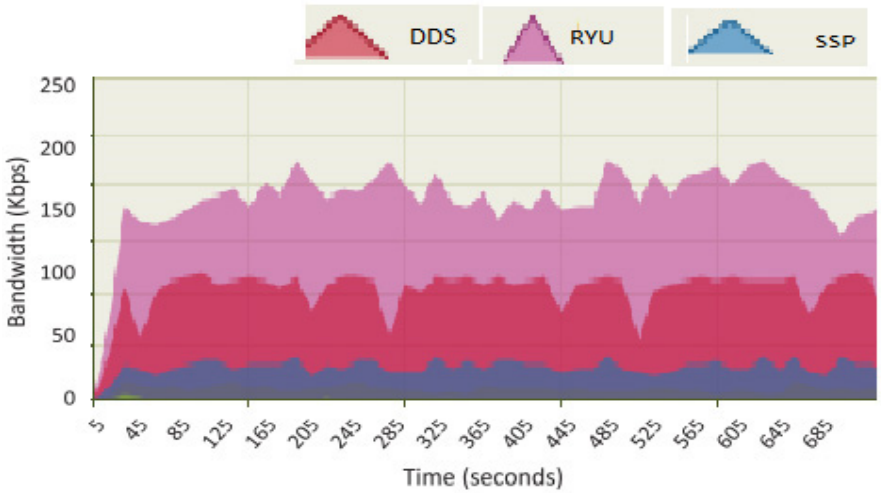

Figure 13. Control channel bandwidth.

D. FLOW CONTROL

Flow requests are vital component of SDN traffic. Because of DoS attacks, more flow rules will be installed by the end switch. Most attacks are executed only with flow request features to overload the SDN controller. The proposed scheme blocks all malicious attacks before they reach the controller. This is the reason why the SPP protocol is installed very close to the SDN controller. The figure indicates that SPP packet-in messages are much fewer when compared to the other two models during the attack. It also clearly shows that there are more unwanted packet-in requests in RYU during the attack. The average number of packets in the message is below 1000 messages per minute in the SPP model as shown in Figure 14, proving its better suitability to SDN environments than other approaches [48–51].

E. RESPONSE TIME

The response time increases during DOS attacks owing to fake requests on the controller. Therefore, there was a delay in the response time. Therefore, the proposed SPP overcomes this issue by blocking the unwanted traffic. The figure shows that the average response time of SPP is 5.23 milliseconds when compared to other models with 6.03 and 7.24 milliseconds, respectively, as shown in Figure 15.

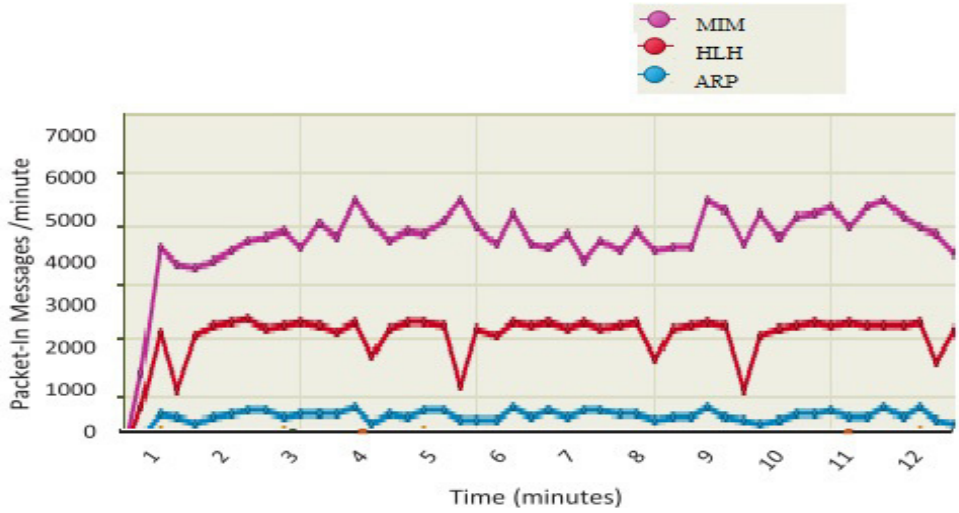

Figure 14. Flow control.

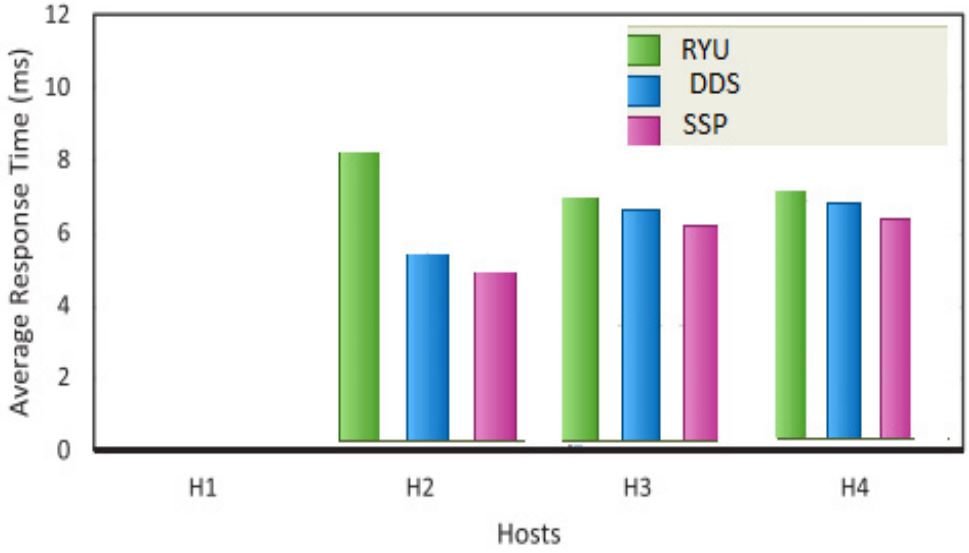

Figure 15. Response time.

9. Conclusions

When network technologies emerge at a steady rate, SDN will be implemented at higher rates in the upcoming years in all fields. Although SDN technology removes the complexity of tying control and data planes together over a traditional networks, it makes certain aspects, such as security, controllability, and the economy of network resources, vulnerable. Among these aspects, security is one of the main concerns to be viewed seriously as far as the applications of SDN are concerned. This study addressed recent ICMP protocol-based attacks and their impact on SDN environments. This work proposes a security policy protocol (SPP) and client authentication model to avoid unauthorized attacks on SDN networks. The proposed model was evaluated using parameters such as CPU utilization, channel bandwidth, packet delivery ratio, response time, and the number of flow requests. The final experimental results proved that the proposed model performed with higher performance and minimum overhead in terms of efficiency. This model can effectively defend against flooding attacks in SDN network environments. The proposed SPP protocol achieved 92% accuracy in ICMP detection Compared with traditional approaches.

Author Contributions: Conceptualization, E.M.O., M.A.K., S.B. (Sundaravadivazhagn Balasubaramanian), S.B. (Salil Bharany), A.U.R., E.T.E. and M.S.; methodology, E.M.O., M.A.K., S.B. (Sundaravadivazhagn Balasubaramanian), S.B. (Salil Bharany), A.U.R., E.T.E. and M.S.; software, S.B. (Salil Bharany); validation, E.M.O., M.A.K., S.B. (Sundaravadivazhagn Balasubaramanian), S.B. (Salil Bharany), A.U.R., E.T.E. and M.S.; formal analysis, S.B. (Salil Bharany); investigation. E.M.O., M.A.K., S.B. (Sundaravadivazhagn Balasubaramanian), S.B. (Salil Bharany), A.U.R., E.T.E. and M.S.; resources, E.M.O., M.A.K., S.B. (Sundaravadivazhagn Balasubaramanian), S.B. (Salil Bharany), A.U.R., E.T.E. and M.S.; data curation, S.B. (Salil Bharany).; writing—original draft preparation, E.M.O., M.A.K., S.B. (Sundaravadivazhagn Balasubaramanian), S.B. (Salil Bharany), A.U.R., E.T.E. and M.S.; writing—review and editing, E.M.O., M.A.K., S.B. (Sundaravadivazhagn Balasubaramanian), S.B. (Salil Bharany), A.U.R., E.T.E. and M.S.; visualization, S.B. (Salil Bharany); supervision, M.A.K.; project administration, S.B. (Salil Bharany); funding acquisition, E.T.E. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Future University Researchers Supporting Project Number FUESP-2020/48 at Future University in Egypt, New Cairo 11845, Egypt.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Thyagaturu, S.; Mercian, A.; McGarry, M.P.; Reisslein, M.; Kellerer, W. Software Defined Optical Networks (SDONs): A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2738–2786. [CrossRef]
2. Kafetzis, D.; Vassilaras, S.; Vardoulias, G.; Koutsopoulos, I. Software-Defined Networking Meets Software-Defined Radio in Mobile ad hoc Networks: State of the Art and Future Directions. *IEEE Access* **2022**, *10*, 9989–10014. [CrossRef]
3. Vasudevan, D.; Nayak, S. Software-Defined Networks. *IEEE Potentials* **2018**, *37*, 21–24. [CrossRef]
4. Liu, Y.; Zhao, B.; Zhao, P.; Fan, P.; Liu, H. A survey: Typical security issues of software-defined networking. *China Commun.* **2019**, *16*, 13–31. [CrossRef]
5. Abolhasan, M.; Lipman, J.; Ni, W.; Hagelstein, B. Software-defined wireless networking: Centralized, distributed, or hybrid? *IEEE Netw.* **2015**, *29*, 32–38. [CrossRef]
6. Chen, T.; Matinmikko, M.; Chen, X.; Zhou, X.; Ahokangas, P. Software defined mobile networks: Concept, survey, and research directions. *IEEE Commun. Mag.* **2015**, *53*, 126–133. [CrossRef]
7. Cao, X.; Yoshikane, N.; Popescu, I.; Tsuritani, T.; Morita, I. Software-defined optical networks and network abstraction with functional service design. *J. Opt. Commun. Netw.* **2017**, *9*, C65–C75. [CrossRef]
8. Hayawi, K.; Trabelsi, Z.; Zeidan, S.; Masud, M.M. Thwarting ICMP Low-Rate Attacks Against Firewalls While Minimizing Legitimate Traffic Loss. *IEEE Access* **2020**, *8*, 78029–78043. [CrossRef]

9. Yang, Z.; Yeung, K.L. SDN Candidate Selection in Hybrid IP/SDN Networks for Single Link Failure Protection. *IEEE/ACM Trans. Netw.* **2020**, *28*, 312–321. [CrossRef]
10. Ahmad, I.; Namal, S.; Ylianttila, M.; Gurtov, A. Security in Software Defined Networks: A Survey. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2317–2346. [CrossRef]
11. Alshra'a, A.S.; Seitz, J. Using INSPECTOR Device to Stop Packet Injection Attack in SDN. *IEEE Commun. Lett.* **2019**, *23*, 1174–1177. [CrossRef]
12. Wang, H.; Xu, L.; Gu, G. Of-guard: A DoS attack prevention extension in software-defined networks. In Proceedings of the 4th Annual Open Network, Santa Clara, CA, USA, 2–4 March 2014; pp. 1–2.
13. Deng, S.; Gao, X.; Lu, Z.; Gao, X. Packet injection attack and its defense in software-defined networks. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 695–705. [CrossRef]
14. Shi, Y.; Dai, F.; Ye, Z. An enhanced security framework of software defined network based on attribute-based encryption. In Proceedings of the 2017 4th International Conference on Systems and Informatics (ICSAI), Hangzhou, China, 11–13 November 2017; pp. 965–969.
15. Liang, X.; Qiu, X. A software defined security architecture for SDN-based 5G network. In Proceedings of the 2016 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC), Beijing, China, 23–25 September 2016; pp. 17–21.
16. Yoon, S.; Cho, J.-H.; Kim, D.S.; Moore, T.J.; Free-Nelson, F.; Lim, H. Attack Graph-Based Moving Target Defense in Software-Defined Networks. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 1653–1668. [CrossRef]
17. Varadharajan, V.; Tupakula, U. Counteracting Attacks From Malicious End Hosts in Software Defined Networks. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 160–174. [CrossRef]
18. Gray, N.; Zinner, T.; Tran-Gia, P. Enhancing SDN security by device fingerprinting. In Proceedings of the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal, 8–12 May 2017.
19. Varadharajan, V.; Karmakar, K.; Tupakula, U.; Hitchens, M. A Policy-Based Security Architecture for Software-Defined Networks. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 897–912. [CrossRef]
20. Hua, J.; Zhou, Z.; Zhong, S. Flow Misleading: Worm-Hole Attack in Software-Defined Networking via Building In-Band Covert Channel. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 1029–1043. [CrossRef]
21. Kumar, H.; HabibiGharakheili, H.; Russell, C.; Sivaraman, V. Enhancing Security Management at Software-Defined Exchange Points. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 1479–1492. [CrossRef]
22. Rahouti, M.; Xiong, K.; Xin, Y.; Jagatheesaperumal, S.K.; Ayyash, M.; Shaheed, M. SDN Security Review: Threat Taxonomy, Implications, and Open Challenges. *IEEE Access* **2022**, *10*, 45820–45854. [CrossRef]
23. Alhaj, A.N.; Dutta, N. Analysis of Security Attacks in SDN Network: A Comprehensive Survey. In *Contemporary Issues in Communication, Cloud and Big Data Analytics*; Lecture Notes in Networks and Systems; Sarma, H.K.D., Balas, V.E., Bhuyan, B., Dutta, N., Eds.; Springer: Berlin/Heidelberg, Germany, 2022; Volume 281.
24. Pradhan, A.; Mathew, R. Solutions to Vulnerabilities and Threats in Software Defined Networking (SDN). *Procedia Comput. Sci.* **2020**, *171*, 2581–2589. [CrossRef]
25. You, X.; Feng, Y.; Sakurai, K. Packet in message based DDoS attack detection in SDN network using OpenFlow. In Proceedings of the 2017 Fifth International Symposium on Computing and Networking (CANDAR), Aomori, Japan, 19–22 November 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 522–528.
26. Sayadi, S.; Abbes, T.; Bouhoula, A. Detection of Covert Channels Over ICMP Protocol. In Proceedings of the IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), Hammamet, Tunisia, 30 October–3 November 2017; pp. 1247–1252.
27. Arote, P.; Arya, K.V. Detection and Prevention against ARP Poisoning Attack Using Modified ICMP and Voting. In Proceedings of the 2015 International Conference on Computational Intelligence and Networks, Shenzhen, China, 7–9 December 2015; pp. 136–141.
28. Kim, H.; Kwon, D.; Ju, H. Analysis of ICMP policy for edge firewalls using active probing. In Proceedings of the 16th Asia-Pacific Network Operations and Management Symposium, Hsinchu, Taiwan, 17–19 September 2014; pp. 1–4.
29. Jiang, W.-H.; Li, W.-H.; Du, J. The application of ICMP protocol in network scanning. In Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies, Chengdu, China, 29 August 2003; pp. 904–906.
30. Durner, R.; Lorenz, C.; Wiedemann, M.; Kellerer, W. Detecting and mitigating denial of service attacks against the data plane in software defined networks. In Proceedings of the 2017 IEEE Conference on Network Softwarization (NetSoft), Bologna, Italy, 3–7 July 2017; pp. 1–6.
31. Onyema, E.M.; Dalal, S.; Romero, C.A.T.; Seth, B.; Young, P.; Wajid, M.A. Design of Intrusion Detection System based on Cyborg intelligence for security of Cloud Network Traffic of Smart Cities. *J. Cloud Comp.* **2022**, *11*, 26. [CrossRef]
32. Kaur, K.; Sharma, S.; Kahlon, K.S. A Middleware for Polyglot Persistence and Data Portability of Big Data PaaS Cloud Applications. *CMC-Comput. Mater. Contin* **2020**, *65*, 1625–1647. [CrossRef]
33. Kaur, K.; Sharma, D.S.; Kahlon, D.K.S. Interoperability and Portability Approaches in Inter-Connected Clouds. *ACM Comput. Surv.* **2018**, *50*, 1–40. [CrossRef]

34. Bharany, S.; Badotra, S.; Sharma, S.; Rani, S.; Alazab, M.; Jhaveri, R.H.; Reddy Gadekallu, T. Energy efficient fault tolerance techniques in green cloud computing: A systematic survey and taxonomy. *Sustain. Energy Technol. Assess.* **2002**, *53*, 102613. [CrossRef]
35. Bharany, S.; Sharma, S.; Badotra, S.; Khalaf, O.I.; Alotaibi, Y.; Alghamdi, S.; Alassery, F. Energy-Efficient Clustering Scheme for Flying Ad-Hoc Networks Using an Optimized LEACH Protocol. *Energies* **2021**, *14*, 6016. [CrossRef]
36. Kayes, A.S.M.; Kalaria, R.; Sarker, I.H.; Islam, M.S.; Watters, P.A.; Ng, A.; Hammoudeh, M.; Badsha, S.; Kumara, I. A Survey of Context-Aware Access Control Mechanisms for Cloud and Fog Networks: Taxonomy and Open Research Issues. *Sensors* **2020**, *20*, 2464. [CrossRef] [PubMed]
37. Iyappan, P.; Loganathan, J.; Kumar Verma, M.; Dumka, A.; Singh, R.; Gehlot, A.; Vaseem Akram, S.; Kaur, S.; Joshi, K. A generic and smart automation system for home using internet of things. *Bull. Electr. Eng. Inform.* **2022**, *11*, 2727–2736. [CrossRef]
38. Bharany, S.; Sharma, S.; Bhatia, S.; Rahmani, M.K.I.; Shuaib, M.; Lashari, S.A. Energy Efficient Clustering Protocol for FANETS Using Moth Flame Optimization. *Sustainability* **2022**, *14*, 6159. [CrossRef]
39. Talwar, B.; Arora, A.; Bharany, S. An Energy Efficient Agent Aware Proactive Fault Tolerance for Preventing Deterioration of Virtual Machines Within Cloud Environment. In Proceedings of the 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 3–4 September 2021.
40. Vourgidis, I.; Maglaras, L.; Alfakeeh, A.S.; Al-Bayatti, A.H.; Ferrag, M.A. Use of Smartphones for Ensuring Vulnerable Road User Safety through Path Prediction and Early Warning: An In-Depth Review of Capabilities, Limitations and Their Applications in Cooperative Intelligent Transport Systems. *Sensors* **2020**, *20*, 997. [CrossRef]
41. Bharany, S.; Sharma, S.; Khalaf, O.I.; Abdulsahib, G.M.; Al Humaimeedy, A.S.; Aldhyani, T.H.H.; Maashi, M.; Alkahtani, H. A Systematic Survey on Energy-Efficient Techniques in Sustainable Cloud Computing. *Sustainability* **2022**, *14*, 6256. [CrossRef]
42. Al-Dahhan, R.R.; Shi, Q.; Lee, G.M.; Kifayat, K. Survey on Revocation in Ciphertext-Policy Attribute-Based Encryption. *Sensors* **2019**, *19*, 1695. [CrossRef]
43. Bharany, S.; Kaur, K.; Badotra, S.; Rani, S.; Kavita; Wozniak, M.; Shafi, J.; Ijaz, M.F. Efficient Middleware for the Portability of PaaS Services Consuming Applications among Heterogeneous Clouds. *Sensors* **2022**, *22*, 5013. [CrossRef]
44. Shuaib, M.; Badotra, S.; Khalid, M.I.; Algarni, A.D.; Ullah, S.S.; Bourouis, S.; Iqbal, J.; Bharany, S.; Gundaboina, L. A Novel Optimization for GPU Mining Using Overclocking and Undervolting. *Sustainability* **2022**, *14*, 8708. [CrossRef]
45. Bharany, S.; Sharma, S. Intelligent Green Internet of Things: An Investigation. In *Machine Learning, Blockchain, and Cyber Security in Smart Environments*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2022; pp. 1–15.
46. Agarwal, S.; Oser, P.; Lueders, S. Detecting IoT Devices and How They Put Large Heterogeneous Networks at Security Risk. *Sensors* **2019**, *19*, 4107. [CrossRef]
47. Bharany, S.; Sharma, S.; Frnda, J.; Shuaib, M.; Khalid, M.I.; Hussain, S.; Iqbal, J.; Ullah, S.S. Wildfire Monitoring Based on Energy Efficient Clustering Approach for FANETS. *Drones* **2022**, *6*, 193. [CrossRef]
48. Sadiq, M.T.; Yu, X.; Yuan, Z.; Zeming, F.; Rehman, A.U.; Ullah, I.; Li, G.; Xiao, G. Motor imagery EEG signals decoding by multivariate empirical wavelet transform-based framework for robust brain–computer interfaces. *IEEE Access* **2019**, *7*, 171431–171451. [CrossRef]
49. Sadiq, M.T.; Yu, X.; Yuan, Z. Exploiting dimensionality reduction and neural network techniques for the development of expert brain–computer interfaces. *Expert Syst. Appl.* **2021**, *164*, 114031. [CrossRef]
50. Liu, Y.; Yin, J.; Cheng, J.; Zhang, B. Detecting DDoS attacks using conditional entropy. In Proceedings of the 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), Taiyuan, China, 22–24 October 2010; IEEE: Piscataway, NJ, USA, 2010; Volume 13, pp. V13-278–V13-282.
51. Ahuja, N.; Singal, G. DDoS attack detection & prevention in SDN using OpenFlow statistics. In Proceedings of the 2019 IEEE 9th International Conference on Advanced Computing (IACC), Tiruchirappalli, India, 13–14 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 147–152.