Received: 21 May 2026; **Revised:** 25 August 2026; **Accepted:** 01 September 2026; **Published:** 06 September 2026
Archs Sci. (2026) Volume 76(1), Pages 95-109, Paper ID 2026110.

Adaptive Security for Hybrid Distributed SDNs

Ajit Karki[1,*] and Abhijit Biswas[2]
[1]Department of Computer Science and Engineering, PhD Scholar, ICFAI University, Tripura, Agartala, India, Faculty of Science and Technology, The ICFAI University, Sikkim, India
[2]Department of Computer Science and Engineering, ICFAI University, Tripura, Agartala, India
Corresponding Author's E-mail: ajitkarki@iusikkim.edu.in

Abstract Software-Defined Networking (SDN) has revolutionized the current networking environment, separating the control plane and the data plane and allowing network control to be centralized, networks to be programmed, and networks to be much more visible and managed. Such functionality greatly enhances the versatility, scalability and use of network resources. However, when it comes to implementing and maintaining SDN infrastructures, the distributed parts of the network need to be carefully orchestrated and a particular skill set is required. Given the concerns about scalability, reliability and fault-tolerance (FT), SDN architectures have been continuously developing from centralized to distributed and hybrid architectures. However, security still stands out as one of the top issues limiting the adoption of SDN. SDN is also centralized in its control architecture and programmable which means that there is new attack surface open to adversaries, which can affect SDN network availability, integrity, and confidentiality. This paper presents a flexible and lightweight security layer in SDN world with its main goal to provide efficient management of security restrictions, real-time detection of intrusion, and proactive defense against zero-day attacks while ensuring the performance of SDN controller/forwarder devices. Apart from that, the key security issues in SDN are analysed in depth, then there is a detailed description of the proposed framework architecture, operational workflow and finally it shows the implementation and effectiveness of the proposed security layer using experimental evaluations.

Index Terms Hybrid distributed SDN, Open flow, Network security and SDN security, firewalls, DDoS (denial of service), Zero day's attacks.

I. Introduction:

The rapid pace of information technology has brought about many new challenges such as the need to protect your network, lower bandwidth costs, be able to support new technologies at deployment and maximize performance. However, the traditional network structure cannot meet these needs and requirements, with the lack of flexibility and complex network management mechanisms. The need for a new networking paradigm that can support the needs of the modern network thus continues to grow, as it needs to support dynamic and large-scale networks.

Software Defined Networking (SDN) is one of the revolutionary networking technologies with software-centric networking of separating control plane and data plane, which allows centralized and programmable network. Programmatic management of network devices through SDN can help lower both capital and operating costs, and enable network customization, optimization and innovation [1], [2]. Moreover, SDN is also advantageous in terms of policy driven management, automation, flexibility, agility, and capable of supporting the different technologies, services and applications deployed in modern communication networks. A fully Software-Defined Networking (SDN) infrastructure may involve replacement of the existing legacy network devices to SDN enabled devices, which may be very expensive and difficult to operate. Several studies [3], [4] have been proposed to solve this issue such as hybrid and distributed SDN architectures. In such designs, SDN-enabled devices reside alongside legacy networking, enabling a phased rollout to a fully programmable network at an affordable price. While SDN promises a number of benefits, such as centralized control, flexibility and programmability of the network, one of the major limitations that is hindering the large-scale adoption of SDN is security. Centralized control plane, open interfaces, create new attack vectors, which compromise network integrity and availability. To address such risks, many SDN security solutions have been suggested to address such threats [4], [5]. Many of these solutions have drawbacks like decreased network usability, more complex configuration and extra performance overhead, however. Moreover, they are frequently ineffective against unknown threats, advanced persistent attacks, and zero day exploits, indicating a demand for novel security mechanisms of adaptive and intelligent protection for SDN environments.

In this paper, we propose an all-encompassing security architecture for logical distributed hybrid Software-Defined Networking (SDN) environments. The proposed framework involves the presence of a security plane with the following modules: Firewall module, Network Intrusion Prevention System (NIPS) module and an anomaly detection module. In addition, we introduce a Security-as-a-Service (SECaaS)

ARCHIVES DES SCIENCES
A Multidisciplinary Journal

cloud platform to provide improved security protection of the management plane and secure administration of the network resources. Moreover, we suggest a completely novel secure communication protocol for SDN control plane to allow for secure communication between the different SDN controllers and secure exchange of data. The proposed security layer will provide the SDN infrastructures with a higher level of overall resilience, providing high availability, proactive threat mitigation, and defenses against sophisticated cyberattacks-such as new exploits or zero-day attacks that may be unknown. The architecture seeks to offer comprehensive security in a flexible and scalable manner by combining various security features within a single platform.

The rest of this paper can be divided as follows. Section 3 explores background concepts and SDN security-related studies. Section 3 puts forth the proposed security architecture and elaborates the designs and functionality of each component. The implementation of the framework and results from the experiments are described in Section 4. A security analysis is carried out in Section 5 and its results are discussed in detail. Finally, Section 6 concludes the paper and gives the possible directions for future work in securing SDN.

II. Security Issues in SDN: Challenge and Related Works

Centralized control paradigm and multi-layered architecture of Software-Defined Networking (SDN) create some significant security issues and vulnerabilities, which can affect the reliability and performance of networks. Understanding the following key issues of security in SDN environments is important:

- One of the most important security concerns of SDN is the centrality of the controller, which makes decisions in the network. Consequently, the controller is a possible single point of failure. Successful compromise of the controller could result in the attacker having control over a significant area if not the entire network infrastructure and it could have serious security and operational implications.
- SDN separates the control plane from the data plane and moves network intelligence out of the forwarding devices into focused controllers. This distinction adds to the programmability and flexibility of networks but also diminishes the security strength of the devices while exposing them to potential attacks and unauthorized operations.
- Hybrid Architecture Challenges: Hybrid SDN deployments involve working with a mix of devices that are SDN-enabled and traditional networking equipment. Whilst this may promote a smooth moreover, it additionally brings with it the various security problems traditional networks do have. The attack surface is thus extended, and requires more security management, because the various components are different.
- Controller Communication Risks: In a distributed SDN architecture, multiple controllers are required to communicate with each other to keep the network consistent and coordinated. These transmissions could be vulnerable to interception, monitoring, or manipulation if they are not properly protected, leading to potential exposures of sensitive data and to a compromise of the control plane integrity.

These challenges point to the need for powerful, scalable and flexible security mechanisms that can secure all layers of the SDN architecture without compromising the flexibility and performance associated with it. Below mentioned Figure 1 shows an overview of the different security issues in a Hybrid distributed SDN architecture.

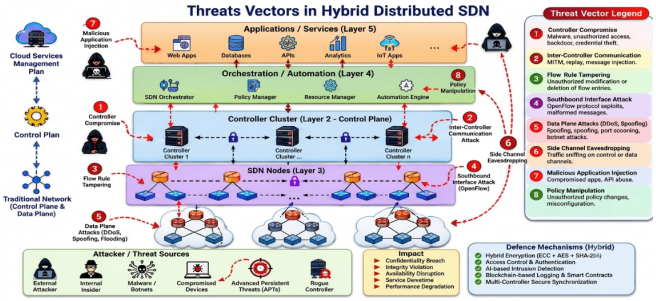

Figure 1: Threats Vectors in Hybrid Distributed SDN

A wide range of cyberattacks, which pose risk to SDN environments' security, availability and reliability, can exploit the aforementioned vulnerabilities. The most notable attacks into SDN infrastructure are:

- Scanning Attacks: These attacks allow an adversary to learn a lot about network topology, network-connected devices, open ports and services. Often these reconnaissance activities are actually the first phase of more advanced attack.
- Spoofing Attacks: Spoofing is the act of representing oneself as another entity, like a fake identity, credentials or network information. This can enable attackers to access the network without permission, disrupt the network, or circumvent security measures.
- Hacking into communication sessions, network devices, or control channels in the data plane or control plane can be considered a hacking attack or "hijacking". These attacks can cause interruptions in network operations and the loss of sensitive information.
- Distributed Denial-of-Service (DDoS) Attacks: DDoS attacks create excessive amount of traffic attempting to overload or inhibit the services that the network, controllers either or forwarding devices are providing to legitimate users. As SDN controllers are centralised, the impact of these attacks is severe and the attacks themselves are much easier to execute on compromised devices.

In response to these security issues, several research projects have put forth security frameworks and security defense measures to improve SDN security. Chapter 4 outlines representative studies aimed at protecting SDN architectures at a summative level in Table 1. In this table, each of the three

ARCHIVES DES SCIENCES
A Multidisciplinary Journal

Table 1: Related Works

References	Contributions	Data Plane C	Data Plane I	Data Plane A	Control Plane C	Control Plane I	Control Plane A	Channel C	Channel I	Channel A
Chiti et al. [6]	Integrated SDN–NFV architecture for secure 5G and IIoT environments.	✓	✓	✓	✓	✓	✓	✓	✓	✓
Shahraki et al. [7]	AI-driven SDN security framework for anomaly detection and adaptive threat mitigation.	✓	✓	✓	✓	✓	✓	✓	✓	—
Al-Mhiqani et al. [8]	Multi-layer intrusion detection system for SDN.	✓	✓	✓	✓	✓	—	—	—	—
Wang et al. [9]	Deep-learning-based DDoS detection framework for SDN.	✓	—	✓	—	—	✓	—	—	—

security objectives of Confidentiality, Integrity and Availability (C, I, A) are represented. Though there are significant developments made by current solutions, the majority of the proposed SDN frameworks are concentrated on the conventional SDN architectures and few of them support hybrid and distributed SDN environment. Moreover, many of these techniques make significant changes to network configuration, add network latency and add extra computational overhead that can reduce network performance. A further outstanding constraint is that the vast majority of solutions are geared to protect against recognised attack patterns with limited protection against new attacks, advanced persistent attacks, and zero day vulnerabilities.

This work introduces a new security architecture to address all these drawbacks and fully secure SDN in all architect layers. In particular, we present a secure security plane for distributing hybrid SDN environments. The proposed framework aims to fulfil the following purposes:

- Ensure a flexible and modular security solution suitable to secure SDN infrastructure against diverse cyber threats.
- Promote resistance to new attacks, advanced threats, and zero–day exploits through adaptive security mechanisms.
- Provide a strong security service without significant impact on performance and network latency.
- Gain high availability, reliability and fault tolerance for distributed SDN deployments.

Enable security management measures that meet both scalability and performance-to-pair smartly with heterogeneous SDN. In the design, these mechanisms are combined to construct a solid security system of next generation SDN networks, while maintaining the flexibility and programmability features of SDN networks.

There were a number of remarkable works focused on increasing the security of Software-Defined Networking (SDN) environments. Qui et al. (2017) [10] presented the GFT concept to provide security appliances with comprehensive details on the flow paths traversed within the SDN infrastructure and, at the same time, keep a complete picture of network activities worldwide. This helps to enhance visibility, which in turn helps implement a security policy throughout the network thereby aiding in the confidentiality, integrity and availability of data plane and communications. In the year 2016, Jantila et al. [11] have proposed the novel SDN architecture in order to mitigate Distributed Denial-of-Service (DDoS) attack. They use a client behaviour-based access control with their framework to further increase data plane security and increase the availability of the network in the event of an attack. Okwuibe et al. (2015) [12] introduced Host Identity Protocol (HIP), a secure communication protocol to protect communication between SDN Controller and network devices. New technologies like HIP use a PK-key based authentication system for the communication between two entities and provide communication security and trust, which improves the communication security in the control plane as compared to the traditional approaches where communication security services are provided based on the IP-address of the sender and receiver hosts. Lara et al. (2014) [13] composed and created an open-source security framework called OpenSEC that consists of embedding of advanced security services like deep packet inspection, intrusion detection and malware detection in SDN environments. OpenSEC leverages SDN programmability to make the network more discoverable and responsive to a wide variety of cyber threats. In the same vein, Shin et al. (2013) [14] created AVANT-GUARD, a security architecture that spans across the control plane and the data plane, which decreases the number of unnecessary interactions between the planes and monitors the behavior of the flows in case of any anomaly occurs. It is designed to block access and infiltration of communication channels and implement efficient counter defence against Denial of Service (DoS) attacks. While each of these studies showed some advancement in securing SDN infrastructures, most solutions were designed for specific security problems and for older SDN deployments, and do not offer any support for SDN architectures beyond the centralized and distributed versions, vouching for the importance of broader and adaptive security solutions.

III. A Centralized Secure Design for Hybrid Distributed SDN Architecture

A. Proposed Centralized Design Scheme

One of the most important concerns is the security of Software-Defined Networking (SDN) architectures, which still worries most network security practitioners and researchers [15]–[19]. Many security paradigms and security solutions have been introduced to compensate the vulnerabilities in SDN environment, but there are still some challenges, which need to be addressed.

For one, numerous security architypes wager greatly on

old security apparatuses, for example, access control, Firewalls and Intrusion Detection Systems. Although these are the fundamental elements of protection, they still do not offer a sufficient amount of protection against advanced cyber threats, advanced persistent attacks, and new types of attacks that are able to trespass past traditional security controls. Secondly, some of the suggested security architectures are intricate and need significant configuration and administration. This can add an extra layer of complexity, potentially waste network resources, and affect network performance and scalability.

In addition, most of the current approaches are geared toward traditional SDN deployments, and do not sufficiently account for security challenges posed by hybrid and distributed SDN setups. These environments include more complicated issues such as heterogeneous device management, the mixing of legacy and SDN-enabled system components, and how to handle inter-controller communication security. An additional crucial constraint is the fact that the majority of existing approaches discover and thwart established patterns of attack, supplying minimal defense against unanticipated dangers, zero-day vulnerabilities and adaptive attack methods. Moreover, the security of the management plane has relatively received little attentions in literature. This layer is also often provided on a cloud infrastructure remote from the enterprise, but managing security means more than securing the network cloud; it provides for the safe deployment, orchestration and delivery of such services to the end-users. As a result, there is a need for a complete SDN security architecture that covers both the data plane and the control plane as well as the management plane, and that can deliver strong security, high performance, scalability and availability. Recognizing these issues, this work introduces a centralized SDN security design scheme where advanced security mechanisms, providing comprehensive security for SDN deployments such as hybrid and distributed deployments, can be incorporated without incurring significant performance penalty and increasing resilience to both known and unknown cyber threats.

A preliminary application of the solution described in this work is a new security approach for overcoming the main security problems of distributed and hybrid Software-Defined Networking (SDN) scenarios. The overarching goal of the proposed framework is to provide the same protection at all the SDN layers and to detect, prevent and mitigate known cyberattacks, together with the previously unknown ones such as Zero day attacks.

The proposed architecture, for these purposes, proposes the addition of new functionalities that increase the SDN infrastructure's security and trustworthiness. The Security Plane is the first plane and provides security to the data plane, the control plane and the communication channels inside the network. This security plane consists of three components - each is complementary and independent while simultaneously contributing to proactive and reactive security services. The first part is the so-called "Anomaly Detection Module" that includes a honey controller, a Distributed Denial-of-Service (DDoS) detection system, and a behavioral analysis engine. These components monitor all network activities and controller interactions continuously, detect suspicious activities, security incidents, and unusual traffic patterns that can make us aware of any on-going attacks.

The second is the Network Intrusion Prevention System (NIPS) Module for advanced traffic inspection and threat protection as per SDN requirements. This module examines the network packets and flow information in real time and can detect and mitigate the malicious activities before they can influence the network operation.

Its third component is the Stateful Firewall Module, which provides an intelligent packet filtering mechanism according to connection security rules. The firewall will examine and regulate the network traffic flows, providing efficient communication between SDN entities while increasing network protection.

Moreover, in the control plane, there is a secure communication mechanism for controllers. The mechanism is designed to provide confidentiality, integrity, and authentication and secure coordination between the different controllers through the SDN infrastructure when exchanging sensitive information to each other. The details of each component of the proposed security architecture, its design, operation and its role of improving the overall security and reliability of distributed hybrid SDN environments are provided in the following sections.

B. Security Plane: Anomaly Detection Module

The main goal of the security architecture being proposed is to provide more security, reliability, and resilience in the context of Software-Defined Networking (SDN) while maintaining network performance and scalability. While many security solutions have been built for SDN deployments, the direct placement of traditional security devices can add to the complexity of the architecture, add further communication overhead, and impact network performance. Moreover, since SDN controllers are centralized and network components are programmable, the network poses new security risks that cannot be properly addressed if the network security is implemented based on traditional approaches alone.

In order to address those limits, the authors in this work propose a specific Security Plane, which will serve as an intelligent and centralized security layer in the SDN architecture. The proposed Security Plane will offer all-pervasive protection at the data-plane, control-plane and communication channels without affecting performance. The cloud-centric approach to security management allows for efficient threat detection, quick response, and coordinated security enforcement across the entire network, thanks to its advanced monitoring and detection capabilities.

The Security Plane combines three co-operative security modules:

- Anomaly Detection Module
- Network Intrusion Prevention System (NIPS) Module
- Stateful Firewall Module

ARCHIVES DES SCIENCES
A Multidisciplinary Journal

Flexibility and extensibility are achieved due to the modularity of the proposed architecture. Depending on the security needs of a particular deployment scenario, additional security components, such as whether a malware analysis engine, threat intelligence platform, machine learning based detection systems, security orchestration modules, or any other security applications are desired can all be easily incorporated.

C. Anomaly Detection Module

Most of the time, one of the first signs of security incidents and cyber-attacks is anomalous network behavior. These anomalies could be caused by a Distributed Denial-of-Service (DDoS) attack, malware, unauthorized access attempts and/or insider threats, command & control communications or zero-day exploits of which the owner is unaware. However, signature-based security module usually is useless when it comes to new threats; this is why it is necessary to add an advanced Anomaly Detection Module, which can detect deviations from the normal network behavior.

The overall goal of this module is to monitor the network activities on a continuous basis, build a network baseline and identify if there were any suspicious activities, which would indicate malicious activities. The mechanism was proposed to use behavioral analysis techniques, which allow detecting known and unknown threats, while traditional security solutions only focus on detecting known threats based on predefined attack signature.

D. Honey Controller and Behavioral Analysis Engine

This is one of the primary goals of SDN cybercrime campaigners: to rip off the controller that is the central network decision-making point. To gain unauthorized access to essential resources and compromise a component on a network, attackers typically use advanced persistent threats (APTs), privilege escalation, and sophisticated malware and stealthy attack vectors.

Intrusion Prevention Systems (IPSs) are known for providing protection against many known threats by using signature matching and traffic inspection, but they can often find it difficult to detect new variants of familiar malware, as well as Zero Day attacks, for which there are no signatures. This limitation is rectified in the proposed framework, where the following components are added: A behavioral analysis engine and a Honey Controller. The Honey Controller is an "intended decoy" SDN controller that draws filters and inspects traffic and events that are identified as suspicious or malicious, without affecting operational traffic. At the same time, the Behavioral Analysis Engine is constantly analysing the network flows, interactions with controllers, and behaviours of devices to identify acceptable traffic and suspicious behaviours. The anomaly detection and threat mitigation process goes like this:

1. Collect traffic statistics, flow information, controller requests and communication logs from SDN devices and controllers continuously.

2. Baseline Profile Generation: Using historical network data, a baseline model, or profile, is generated representing normal network behavior under a variety of operating conditions.

3. Continuous Monitoring: On an ongoing basis, real-time traffic and operation metrics are continuously compared with the baseline traffic profile.

4. Anomaly Detection: Anomaly alerts are generated when it detects any activities that are significantly different from the expected behavior, which potentially signifies a security occurrence.

5. Risk Assessment: Bunching together several alerts to determine the likelihood and threat of a current attack with the aim of diminishing false positives.

6. Traffic Redirection: When suspicious activities are detected, the appropriate traffic flows are automatically diverted to the Honey Controller's machine for further investigation and containment.

7. Threat Analysis: Security administrators and automated analysis systems verify the traffic redirected and identify the kind, origin, attack level and effect of the threat found.

8. Mitigation and Response: Possible countermeasures are taken- this includes traffic filtering, device isolation, policy enforcement and so on -to neutralise the threat identified.

9. Reporting and Knowledge Integration: Incident reports are detailed and passed on to the management layer and the knowledge therein is used to enrich security policies, detection models and bolstering security measures.

With the addition of the Honey Controller and Behavioral Analysis Engine, capabilities to detect advanced cyber threats, insider attacks, malware infections, and zero-day vulnerabilities have been greatly improved. The proposed solution constantly looks for behavioral anomalies, as these are the only ones that have not already been defined, as "signature", therefore offering proactive protection against attacks that have not yet been observed. However, anomaly detection is not enough for security. To address this module, the proposed architecture introduces more security features: Network Intrusion Prevention System (NIPS) and Stateful Firewall Module and combine security layers that work to secure distributed hybrid SDN environments.

IV. DoS and DDoS attacks

One of the biggest dangers to Software-Defined Networking (SDN) environments is Distributed Denial-of-Service (DDoS) attacks. However, because the control plane is logical, SDN controllers are appealing targets for attackers who want to wreak havoc on the network. Controllers, switches, and communications channels can be overwhelmed with so much traffic the attackers use that they cause a denial of service so they can take away the availability of critical services from legitimate users. In addition, DDoS could be executed from several compromised devices and could impact various layers of the SDN architecture, such as data plane, control plane and application plane.

Several research studies have introduced various DDoS Mitigation techniques for SDN environments. Many of these solutions, however, involve complicated traffic engineering

methods, distributed packet filtering systems or a number of rate-limiting solutions at different network layers. While these solutions work to a degree, they impose a significant amount of processing overhead, network latency and use up precious controller resources. As such, they can be a hindrance to large-scale SDN deployments in terms of scalability and performance.

Addressing these limitations, the proposed architecture introduces lightweight centralized DoS/DDoS Detection Module in the Anomaly Detection System of the Security Plane. The proposed framework does not require multiple traffic limiting mechanisms to be deployed across the network but only a single intelligent malicious traffic detection/mitigation module, which affects network performance as little as possible while detecting and mitigating malicious traffic. This design provides tremendous reduction of architectural complexity and is designed to detect attacks in real-time and respond quickly.

This is because most DoS or DDoS attacks are detected as anomalous surges in the number of traffic, number of connections, and flow creation rate or packet throughput within the Anomaly Detection Module. Continuous traffic monitoring and anomaly analysis can be very effective in detecting such deviations from normal operational behaviour. The mechanism can recognize known or previously unknown variations of flooding attacks without only using predefined signatures by treating each massive traffic surge as an out-of-behaviour event. Proposed DDoS Detection and Mitigation process can be divided into following stages:

1. Definition of Baseline Thresholds: Baseline thresholds are defined for normal operations for the packet throughput, flow generation rate, connection requests, and interactions with the controller based on history of network behavior.

2. Attack Probability Threshold Configuration: Defined probability threshold, which defines when abnormal traffic patterns are detected, it should be assumed to be a possible DoS and DDoS attack.

3. Continuous Traffic Monitoring: anomaly detection server periodically captures and analyses the traffic samples from various different layers of SDN, such as data plane, control plane, and communication plane.

4. Anomaly Identification: If the number of traffic observed is higher than the number of traffic already set (baseline), an anomaly will be alerted and recorded.

5. Attack Probability Estimation: Several alerts are collected together and correlated over a time period of observation to get an estimate of the percentage chance of attack in progress, whether DoS or DDoS.

6. If the probability of attack is higher than a predetermined value, the suspicious traffic flow or connection is automagically restricted so that further resources cannot be consumed; this is known as automated mitigation.

7. Incident Reporting: Creates an incident report with the attack description and all the information about traffic and attacks including affected resources, traffic statistics, and mitigation action and forwards it to the Management Plane for forensic analysis and policy update.

The proposed framework could also be extended to include machine learning and artificial intelligence algorithms to achieve adaptive detection of attacks. These would allow it to constantly learn from past traffic patterns, adaptively vary detection thresholds, and increase the detection accuracy in very dynamic SDN environments. Proposed DoS/DDoS Detection Module offers a number of benefits than the traditional Mitigation approaches. First, it reduces degradation by not having to deploy several filtering mechanisms throughout the network. Second, it is centralized, which makes security management and policies easier. Third, the anomaly-based detection approach can be used to identify known and previously unseen attack patterns, enhancing resilience against the new wave of evolving cyber threats. Lastly, the controllers' capabilities to rapidly detect and automatically mitigate events help provide network availability, controller stability and overall Quality of Service (QoS) in a distributed hybrid SDN environment.

When used together with the components of Honey Controller, Behavioral Analysis Engine, Network Intrusion Prevention System (NIPS), and Stateful Firewall groups, the proposed Security Plane brings an all-in-one multilayer defense scheme for SDN infrastructures capable of handling wide-ranging denial-of-service attacks and advanced cyberattacks.

This module is used for Stateful firewall and NIPS (Network intrusion prevention system) operations. For Stateful firewall and NIPS (Network intrusion prevention system) operation, this module can be used. Securing Software-Defined Networking (SDN) infrastructures is a fundamental requirement and traffic filtering and intrusion prevention are fundamental requirements. Because SDN is programmable and centralised, a successful attack against the centralisation and/or the data plane will quickly propagate throughout the network, breaking several devices and services. Therefore, it is important to include features that can support monitoring, filtering, and control of network traffic in an effective security framework that still maintains performance and scalability benefits of SDN.

Firewalls are the first line of defense against unauthorized access to a network and malicious communications in conventional network settings. Firewalls, in the same way, are a necessary part of SDN architectures, as they examine traffic in and out of the network, apply security policies and defend against unauthorized traffic interactions. However, when it comes to firewall architecture, it becomes one significant challenge in SDN deployments. The state-less firewalls are most commonly used in current SDN security infrastructure - as they do not require significant calculation workloads or undue impact on network performance. Stateless Firewalls do not require store and retrieval of information regarding previously established connections. While this is certainly useful to help process things faster, it confers minimal immunity to advanced threats using sophisticated attacks that rely on session states, connection hijacking, manipulation of the protocol, or ad-

vanced persistent threats. Each attack transaction can then be masqueraded as a legitimate one, with the other traffic filtered out of view as long as the individual packets themselves are benign.

On the other hand, a Stateful Firewall keeps track of a lot of information about an active connection and communication sessions. Stateful firewalls can analyse network interactions and effectively detect malicious behavior using knowledge of the state of interactions. For large-scale SDNs with high traffic volumes and dynamic flow creation, traditional stateful firewall implementations, however, typically involve a significant overhead in processing, latency and lower network throughput.

In order to address such problems, in this work a new SDN-aware Stateful Firewall Architecture specially tailored for distributed/hybrid SDN architectures is proposed. Proposed firewall secures the programmable network with stateful firewall while meeting performance demands. Firewall work is pushed out of the controller and runs instead on the Security Plane, which frees up computational resources on SDN controllers and maintains network performance as all traffic does not need to pass through the SDN controllers to be inspected. The above-defended firewall design is able to handle traffic from the following sources at any one time:

- Flows dynamically added by decisions made by controller: Reactive.
- Policies that are proactively determined and used for forwarding.
- Legacy - network traffic flowing from traditional devices in hybrid networks.

Distributed hybrid architectures also include traditional networking devices, thus making interoperability essential. The proposed framework aims to tackle this issue by having protocol translation mechanisms in the data plane. These translators are used to transform the traditional network traffic to OpenFlow compliant representations, and on this way, the firewall can process and analyse the traffic based on a common and uniform representation, which is independent of the origin.

A. Key Components to the Proposed Stateful Firewall are:

- Message Listener Module: Listens for flow requests, packet-in messages and communication events from both SDN controllers and network devices. The Message Listener analyses packets (to the extent needed for preliminary analysis), and matches incoming packets with the information stored in the connection information within the state management database.
- State Table Module: The State Table stores full information about the ongoing sessions and the state of connections, including authentication details, statistics on traffic flow and history of flows. In context-aware firewalling, the firewall has the ability to recognize between real and possibly nonsuspicious communications, increasing the accuracy of detection, and minimizing false positives.
- Policy and Rule Management Engine: Generates, Updates and Enforces security Policies. It automatically enforces filtering policies according to network conditions, threats discovered and organization security requirements, and administrator policies. The engine can also be used for the automated adaptation of policies to new attack patterns. Traffic Inspection and Decision Engine: This module consists of examining the behavior of a packet against the current state of sessions and security policies, and hence deciding whether to allow, modify, rate-limit, temporarily quarantine, or block the traffic.
- The proposed architecture features a graphical user interface and advanced firewall functionality, along with the addition of a network intrusion prevention system (NIPS) for proactive network intrusion and cyberattack protection. Although Intrusion Prevention Systems are popular in traditional networks, they are not generally included in SDN deployments, as the resources required to keep IPSs and network performance may be significant.
- This design decision provides many potential security holes, especially concerning malware propagation, exploitation attempts, reconnaissance, and Advanced Persistent Threats. This problem is solved by implementing a light SDN oriented NIPS in the Security Plane and not directly in the controller infrastructure. This separation enables functions, which are related to security to run separately without putting the burden on the control plane.

The proposed NIPS will serve various security tasks such as:

- Real time Packet Inspection & Flow Analysis.
- Signature-based attack detection.
- Behavioral anomaly identification.
- Detection of the malware and its communication to and from the botnet.
- Prevention of exploit and vulnerability scanning.
- Identification and prevention of unauthorized entry attempts.
- Automatic mitigation of traffic flows that are suspected to be suspicious.
- Correlation of threat intelligence information from various network layers.

When NIPS identifies a malicious event, it can autonomously respond by blocking the flow, terminating the connection, isolating traffic, dropping the packets, or even dynamically updating the policy. In addition, intrusion events are represented to the Management Plane for incident analysis, forensic investigation and security policy improvements. The adaptive statefull firewall module for hybrid SDN is presented in Figure 2.

Syngers Stateful Firewall and NIPS modules enable a multi-layer defense approach that covers everything from SDN network components to legacy network components. The NIPS provides proactive traffic detection and threat prevention, and

ARCHIVES DES SCIENCES
A Multidisciplinary Journal

the firewall provides granular traffic control and secure session management. Combine it all into a single suite that provides greater confidentiality, integrity, and availability of network resources while preserving the scalability, flexibility, and performance demanded by distributed hybrid SDN networks.

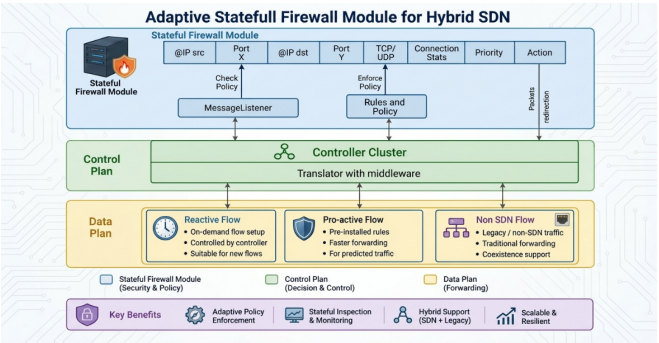

Figure 2: Adaptive Statefull Firewall Module for Hybrid SDN

The proposed framework offers a comprehensive security infrastructure that can withstand conventional cyber-attacks as well as sophisticated attacks that aim to target today's programmable networks, thereby enhancing the overall security and security awareness of SDN deployments in the future.

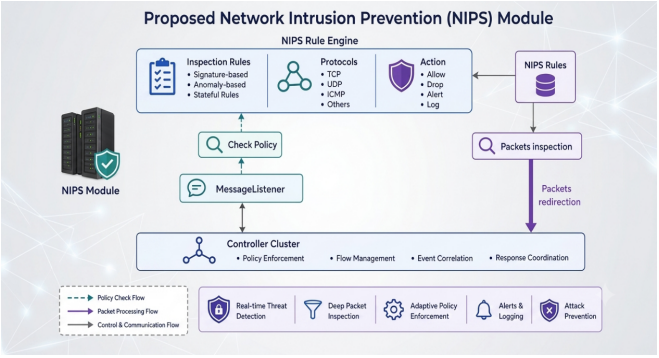

Figure 3: Proposed Network Intrusion Prevention Module

In this regard, a new Network Intrusion Prevention System (NIPS) for distributed and hybrid Software-Defined Networking (SDN) environment is proposed as depicted in Figure 3. The presented NIPS is in contrast to traditional intrusion prevention, which is deployed separately from the network control plane, and utilizes the recently provided global network visibility by SDN. This integration helps to proactively block at the source, automatically process and mitigate malicious traffic in real time, and puts less processing strain on SDN controllers. The proposed NIPS architecture includes three main components:

- Message Listener Module: Modular component that listens for a flow request, packet-in results, traffic statistics and communication events from the SDN controllers, forwarding devices and legacy network components. The Message Listener continually observes various types of network activities, which provides the main point of communication between the network and the NIPS, to forward to the NIPS for security inspection and analysis.
- NIPS Rule Table: The rule table stores detailed information about all active connection records, inspection policies, attack signatures, alert definitions and mitigation actions. It is the database on which the decision-making is done in the intrusion prevention system and it enables the quick classification of the network events based on security policies.
- NIPS Policy and Rule Management Engine: This allows for policy and rule management, including the addition and updating of malware signatures, attack rules to be detected and response strategies to be coordinated. New threat intelligence data is continually fed into the engine, with security policies evolving dynamically to cope with new forms of attack vectors and changing threat landscapes.

The suggested NIPS will serve not only one security purpose but multiple such purposes including deep packet inspection, flow analysis of the traffic, detecting known and unknown malware contents, preventing successfully from being exploited, and detecting any unauthorized access, identifying communication with botnets and correlating attacks. When suspicious activity is detected, the system can automatically initiate traffic filtering, traffic termination, traffic isolation, advanced access control implementation, and policy changes. In addition, incidents of security breaches are recorded and passed on to the management layer for forensic analysis for ongoing security assessment.

A big strength of the proposed NIPS is that it is centrally implemented in the Security Plane. As the module has a global picture of the network traffic and the interactions with controllers, it is able to detect the coordinated attacks that may be impossible to discern in a single network segment. Besides, processing overhead of the SDN controller is reduced greatly and the functions of intrusion prevention system are isolated from it, thus security action would not have bad influence on the network performance. This results in a well-balanced framework for security enforcement, scalability, and operation efficiency.

B. Secure Logical Distributed SDN Architecture

The Open SDN paradigm is based on a logically centralized control plane that has a single controller to manage the whole network infrastructure. While this makes network management and issuance of policies easier, it has some security, scalability and reliability implications that make it insufficient for large-scale or mission-critical networks. Centralized SDN architectures' biggest problem is that they have a single point of failure. In spite of the fact that the controller is the central decision making machine, if an attack against the controller is successful, attackers may be able to gain full control over the activity of the network. A compromise like this can result in unauthorized flow changes, disruption service, data leakage and even large network outages. An additional significant

worry is the susceptibility of centralized controllers to Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks. An adversary could overrun the controller with too many traffic volumes or bogus requests; this could consume resources and make the network unavailable to legitimate users. Since the controller is a vital component, even a temporary failure may have serious operational implications.

Besides security, centralised architectures are another drawback in terms of scalability and performance. However, with the scale-out of a network and the volume of traffic, an increasing number of flow requests, policy updates and management operations may become a controller's problem when implementing a single controller for a small network. This restriction can lead to higher latencies, lower throughputs, and lower Quality of Service (QoS).To overcome these shortcoming, a number of researchers have presented the logistically distributed SDN architectures [8] and [20]. A control function is cascaded among multiple controllers, each of which is responsible for cooperation to control the network as a whole. Each controller generally handles a particular part of a geographical location, administrative ranges and/or network parts, and coordinates with other controllers to secure consistency throughout the network globally. Some benefits of having distributed controllers include improved fault tolerance, reduced control-plane latency, increased network availability, load balancing and scalability. In addition, sharing the control duties reduces the effect of failures and increases resilience of targeted attacks in the controllers.

Nevertheless, distributed SDN architectures give rise to a new set of security issues. We need in particular continuous communication amongst the controllers, where they exchange topology information, updates on the routing decisions, policy decisions, and synchronization messages, as well as security events. These messages can often include very confidential data on network structure, state and security policy. For attackers, if intercepted, manipulated, or spoofed these communications, they may provide important information about the network or provide the opportunity to make an advanced attack on the control plane. Some threats are eavesdropping, altering messages, replaying messages, impersonation of the controller, hijacking of sessions and access to distributed control resources.

In order to address these risks, we advocate a secure communication scheme, tailored to distributed hybrid SDN infrastructure, between controllers. The proposed protocol enables secure communications between controllers by using powerful authentication, secure communication channels, integrity verification mechanisms, secure session establishment procedures, and trust-based controller coordination. The proposed framework guarantees the confidentiality, integrity, authenticity and availability of control-plane communications by protecting information exchange within a cluster of controllers.

Combined with the Security Plane, Stateful Firewall, NIPS and Anomaly Detection modules, it allows for the creation of a comprehensive defense architecture that can safeguard centralized and distributed SDN deployments to advanced cyber-threats all while also delivering high levels of scalability, performance and reliability.

The proposed distributed hybrid SDN architecture consists of distributed controllers with a set of security and management extension modules (provided by each controller) as shown in Figure 4. These modules aim to address the need to coordinate controllers, ensure network consistency, increase fault tolerance, boost security and provide high availability in distributed control plane.

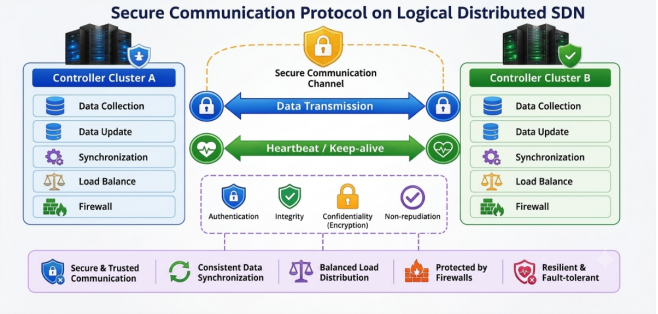

Figure 4: Secure Communication Protocol on Logical Distributed SDN

The presented controller design incorporates the following extension blocks:

- Data Collector Module
- Data Updater Module
- Synchronizer Module
- Load Balancer Module
- Host-Based Firewall Module

The Data Collector Module constantly collects data from underlying infrastructure, such as topology changes, flow stats, traffic patterns, controller status information, security events and device performance information. This information is fed back to controllers to give them a real-time and full overall view of the network environment so that they can make intelligent decisions, and enforce policies. Maintains consistency among distributed controllers: Data Updater Module. Periodically pushes routing information, flow tables, security policies, access control rules, network state information to clusters of controllers. Through this synchronization process, network coherence and collaboration guarantees that all controllers are working with the same network intelligence and valid data, minimizing the risk of stale network intelligence, inconsistent policies enforcing a decision, or conflicting configurations being applied.

A Synchronizer Module is also added in the proposed architecture to improve the synchronization between controllers. This component is responsible for inter-controller state synchronization and event dissemination, and for making sure controllers have the same configuration. The synchronizer ensures that topology changes, security alerts, network events discovered by one controller are quickly shared with the rest of the controller cluster. Hence, despite the controllers being

geographically and/or logically distributed, they all have a consistent global network view.

Enhancing scalability and fault tolerance, the Load Balancer Module can add more controllers to a pool and distribute the control plane workloads among these controllers. For large-scale SDN deployments, in some regions of the network, there could be many traffic requests whereas the other regions may be able to get away with less. The load balancer keeps a check on the utilization of the controllers, the processing loads, the memory usage and the communication round trip time to dynamically assign tasks to the controllers. This process limits the controller load, maximizes use of the resources, decreases response time and optimizes network performance under different traffic loads. For extra security, every controller includes a separate Host-Based Firewall Module. The security feature prevents the controller from unauthorized use, malicious communications and attacks (control-plane). The firewall implements access control policies, intrusion detection filtering, blocks unauthorized firewall management, and controls messages to the controllers to avoid exploiting vulnerabilities within the controllers. The framework secures individual controllers to greatly decrease attack area of the distributed control plane.

Secure communication is provided between controller clusters with encrypted communication channels (by TLS/SSL protocols) between controller clusters. To provide confidentiality, all the inter-controller messages are encrypted; meaning that all the information sent between controllers is encrypted upon transmission and protected from interception/mishandling by unauthenticated entities, such as routing information, topology information, security policies, authentication information and synchronisation information. The proposed framework in addition to encryption, features robust authentication and integrity verification. Controllers have to base themselves on each other's digital certificate and cryptographic credentials before they share information with each other. In this way, controller impersonation attacks are prevented and cluster communications are only processed using trusted controllers. Data modification, replay or unauthorized insertion of messages is protected for by additionally performing message integrity checks.

Another important consideration is high availability in the distributed SDN environment. For continuous monitoring of the controller health and to detect the failures in real time the proposed architecture uses a heartbeat based monitoring mechanism. -periodic heartbeat messages are sent from each controller, periodically, to the neighbouring controllers in the cluster. These heartbeat packets are used as liveness indications and for the controller to check whether the other controllers are working or not. If the controller does not get a heartbeat signal in a set period, it is assumed that something may have gone awry with the controller, there is a communication failure or that a security incident has occurred. This check triggers an alert to be immediately generated and sent to the Management Plane for further investigation. At the same time, the Load Balancer Module takes care of passing round the responsibilities of an affected controller to other ones that are still running, so that the network continues without interruption and services are maintained. In addition, there is an attack detection using the heartbeat mechanism. For instance, humanity of communication disruptions could be the indication of a Distributed Denial-of-Service (DDoS) attack, controller compromise, partitioning of a network or bad people trying to isolate the control plane parts. The framework integrates heartbeat monitoring with anomaly detection and security analytics, which can speed up the time it takes to identify and mitigate control-plane threats.

As a whole, the proposed secure inter-controller communication framework can provide a resilient, trustworthy distributed control plane, which can support the large-scale hybrid SDN deployment. The secure communication channels, controller synchronization, dynamic load balancing, host based protection and continuous health monitoring all add a layer above the programmability and flexibility that SDN provides to improve scalability, fault tolerance, availability and security. The design dramatically increases SDN infrastructures' resistance to controller failure, cyberattacks, connectivity loss and new security threats, allowing for reliable operation within today's enterprise, cloud, and critical network deployments.

V. Implementation and Experimental Results

A. Experiments' Environment

Distributed hybrid SDN Testbed was developed and deployed in a controlled experimental environment, to validate the effect of proposed AM-Sec security architecture. The goal of the implementation was to test the results and analyse the performance, scalability and security of the proposed framework subject to both normal and malicious traffic scenarios.

The configuration of testbed system is presented below. The configuration of the testbed system is given below.

The experimental platform was setup on an HP Proliant DL380 G6 server having the following hardware specification:

- 4 x 2000 MHz Quad-Core Intel Xeon E5504 CPUs
- 4 cores CPU with 4 MB cache memory
- The power consumption of the processor is 80 W.
- 24 GB RAM
- Gigabit Ethernet connectivity

To effectively use hardware resources and simulate a realistic multi-domain SDN environment, the virtualization technology was used. The entire infrastructure was deployed on the open-source virtualisation platform XenServer [21] to enable the deployment of isolated virtual machines to represent SDN controllers, Openflow switches, security services and infrastructure management.

Three "OpenDaylight (ODL)" controllers were implemented with 4G of RAM each installed on Ubuntu Server 16.04 (64-bit) for the control planeside. The SDN network topology was divided in three different administrative domains for simulating a logically distributed SDN network. OpenDaylight locally controlled each with flow management,

ARCHIVES DES SCIENCES
A Multidisciplinary Journal

Table 2: Performance Evaluation of the Proposed Security Modules

Security Component	Traffic Type	Traffic Rate (pps)	Threats Detected	Events Analyzed	Mitigation Decision	Controller CPU Utilization (%)
Stateful Firewall + NIPS	Legitimate Traffic	100	0	0	Forward	8.91
	Legitimate Traffic	150	0	0	Forward	9.48
	Legitimate Traffic	200	0	0	Forward	10.02
	Malicious Traffic	100	46	238	Block	10.47
	Malicious Traffic	150	103	386	Block	11.89
	Malicious Traffic	200	156	462	Block	13.21
Anomaly Detection Engine	Legitimate Traffic	100	0	0	Normal Operation	7.96
	Legitimate Traffic	150	0	0	Normal Operation	9.11
	Legitimate Traffic	200	0	0	Normal Operation	10.37
	Suspicious Traffic	100	338	338	Alert Generated	15.82
	Suspicious Traffic	150	451	451	Alert Generated	18.21
	Suspicious Traffic	200	698	698	Alert Generated	22.94

policy, and security coordination between the various OpenDaylight controllers. The data plane was built using the network emulator tool, Mininet [22] (version 2.2.1). For every network domain, there were three switches running OpenFlow and multiple virtual hosts running in virtual machines on Ubuntu Server 16.04 (64-bit) with 2 G of RAM. This enabled to generate a realistic network traffic with inter-domain communications, while keeping at the same time the experimental flexibility and reproducibility.

A server running Ubuntu Server 16.04 (64-bit) with 4GB RAM was set up as a "GRR Rapid Response" server to support forensic investigation and help in the post incident analysis. All SDN control devices were deployed with GRR agents, and, continuous monitoring and remote acquisition and incident response capabilities were deployed. This piece allowed the thematic analysis of participant's controller actions during attack situations, as well as trace data to be collected for security analysis.

An independent part of the proposed architecture is the security layer. The Linux Netfilter software was used to act as a stateful firewall that was used to filter and control traffic. The firewall enabled to filter out communications at a network and at a controller host level to prevent unauthorized access and malicious communications. The new Network Intrusion Prevention System (NIPS) and Anomaly Detection modules were also introduced into the Security Plane to detect and mitigate anomalies and attacks in real time.

VI. Experimental Methodology

A diverse set of traffic scenarios were devised and the success of the framework in each of these scenarios for different network loads was assessed. Valid and fake traffic was cross injected in the network. Testing: Primary attention in the experiments was given to testing:

- Intrusion detection capabilities.
- Detection of Malware/Traffic Anomalies.
- Effectiveness of DDoS attack mitigation.
- Packet filtering performance.
- Controller resource utilization.
- Effect of security mechanisms on to the network performance.

To simulate the traffic load and attack conditions, the traffic loads of 100 packets per second (pps), 150 (pps) and 200 (pps) were constructed. Security Plane was used in all the experiments to look, see the network activity, understand network traffic flows, trigger alerts and take appropriate mitigations.

The Implementation Results achieved by the Security Plane components; which are Stateless Firewall, NIPS, and Anomaly Detection and so on is presented in Table 2.

During normal traffic conditions, it was found that both the Stateful Firewall module and NIPS had not given any false alarm while processing network traffic. In low traffic condition (100pps, 150pps and 200pps), the controller's resource utilisation was relatively low (between 9.02% and 10.15%) indicating less performance overhead. It proves that the proposed security architecture is effective to secure the network traffic without affecting the efficiency of the controller.

With malicious traffic, the Firewall module and NIPS module were able to properly identify and deny the infected traffic. This increase is illustrated by the red line which displays what gets detected at different traffic intensities, and the blue line which displays the number of malicious activities analysed: 100 to 200pps (packets per second) for the red line, and 224 to 438 events for the blue line. The scalability and efficiency of the proposed intrusion prevention mechanisms were demonstrated since the utilization of the controllers was still below 13%, while the workload was increased.

The Anomaly Detection Module had excellent performance in detecting suspicious network activity. No abnormalities were reported during normal traffic conditions, the utilisation levels reached between 8.05% and 10.75%. With the inclusion of infected traffic, the anomaly detection system was able to raise alerts for abnormal network activity. The greater the value of the number of packets per second the more number of anomalies can be detected, and at 200 packets per second, there are 675 alerts detected. While there was some additional processing overhead with anomaly analysis, the controller utilization was still less than 24%, which was acceptable given the increased visibility and detection with the module.

The results of the experiments indicate some of the following. The proposed Security Plane first, separates the legitimate traffic from the malicious one without decreasing the controller performance. Second, all the security modules are centralized, enabling efficient monitoring and handling of traffic and attack mitigation with low computation requirement. Third, while the signature based security systems are incapable of detecting suspicious actions, the anomaly-based

ARCHIVES DES SCIENCES
A Multidisciplinary Journal

Table 3: Experimental Results of the Proposed Ddos Detection and Mitigation Module Under ICMP, UDP, and TCP Flooding Attacks

Domain	Attack Vector	Attack Intensity (pps)	Detection Time (ms)	Mitigation Time (ms)	Detection Accuracy (%)
SDN Area-1	ICMP Flooding	1450	121	415	98.4
SDN Area-1	UDP Flooding	890	69	298	99.1
SDN Area-1	TCP Flooding	2080	162	671	97.8
SDN Area-2	ICMP Flooding	1575	232	501	98
SDN Area-2	UDP Flooding	1745	241	522	97.6
SDN Area-2	TCP Flooding	1120	171	514	98.3

detection can detect them, resulting in a higher resistance towards unknown actions and Zero day attacks. The results indicate generally that the proposed architecture can achieve a balance between the performance and security. The Stateful Firewall, Network Intrusion Prevention System and Anomaly Detection Module provide added security to distributed hybrid SDN deployments without compromising on scalability, availability or controller efficiency. The results corroborate the appropriateness of the proposed framework for securing of next generation SDN infrastructures running in dynamic and large-scale networking environments.

Table 3 shows the achieved performance of the proposed DDoS detection framework concerning the size of the attack, the detection delay, and the delay of detected attack mitigation response. Experimental results showed that the proposed DDoS detector based on anomalies was able to detect and mitigate all the attack scenarios in both network domains. For the ICMP flooding attack, that occurred in Area 1, there had been an average of 1,423 connection requests per second, an average detection time of 127ms, the fastest detected of the various types of attacks evaluated, and an average mitigation time of 429ms. The TCP flooding attack type created the maximum number of connection requests per second (2,047), and was detected and mitigated in 168ms and 689ms respectively.

The same type of results was recorded in Area 2. The system detected with latency of 246ms and mitigated with latency of 517ms for ICMP flooding attacks consisting of 1,548 connection requests per second (CPS); and the system detected with latency of 247ms and mitigated attacks with latency of 534ms for UDP flooding attacks, effectively 1,720 CPS; and for TCP flooding attacks up to 1,069 CPS, the detection latency is 175ms and mitigation latency is 527ms.

The reported variations in the time of detection and mitigation are due to the nature of types of attacks. Traffic formats within UDP flood attacks are typically highly abnormal and do not have a concept of connection-state management, which means that they are easier to detect with anomaly based monitoring techniques. In comparison to an attack using the TCP protocol, in which more elaborate traffic pattern matching/correlation will need to be handled prior to taking mitigation actions, typically involves connections opened to other machines and extensive traffic patterns. Similarly, a large number of ICMP floods can result in a huge amount of traffic spikes, which require significant traffic analysis before it can exceed the pre-defined anomaly level.

The results demonstrated that the proposed detection mechanism has low response time, under high amount of attack traffic taking up to 2000 connection requests per second. The latency of detection was maintained below 250ms for every experiment and the latency of mitigation operations was maintained below 700ms, which is very crucial in SDN-based network scenarios to avoid drowning the controller resources and service disruption in the event of an attack.

Further, the deployment of DDoS Detection Module in the Security Plane results in a single, unified view of traffic vectors on a global scale across multiple domains. The advantage is that with this capability, the framework would be able to identify distributable attack pattern that may not be recognized by traditional localized security mechanism. In the proposed system, the attacks are successfully classified and the false-positive rate can be reduced by correlating traffic information of different segments of the network.

The other significant advantage of the proposed approach is that it is lightweight. The deployment of multiple packet-rate limiters/filter units at different locations within the SDN network is done in many existing SDN security solutions, but the proposed framework helps to avoid this. Along with keeping the protection against DoS attacks effective, this design has reduced the computation overhead with a minimal workload for the controllers and yet not affected the network's performance badly.

From the results in Table 3, we can conclude that the proposed DDoS Detection Module is a passive module that can be used in secure the distributed hybrid SDN environments. The framework demonstrates good attack detection capability and attack mitigation procedures are easily followed; operational overheads are low, which significantly contributes to the availability of resources and the stability of the network controller, and continuity of services. These results will reinforce our proposal on the practicality and scalability of the proposed solution that protects next generation SDN infrastructure against classical DDoS attacks and large-scale attacks. The results of the experiments and the security analysis sections are presented.

For incident investigation, threat hunting and post attack forensic analysis, a GRR Rapid Response (GRR) server was used; it was built on Ubuntu 16.04 (64-bit) virtual machine, set with 4 GB of RAM. To facilitate continuous monitoring, remote collection of evidence, inspection of memory, analysis of processes and reconstruction of events was extended via

installation of GRR agents on each OpenDaylight controller. Analysing these attacks from a forensic viewpoint helped determine controller activity that signals malicious activity as well as identify contaminated processes, unlawful activity, and potential attack paths.

In order to apply traffic filtering and access control policies, the proposed architecture is based on the Linux Netfilter framework, which is used here as a stateful firewall. Ubuntu 16.04 (64-bit) virtual machines with 2 GB RAM were created and the firewall deployed at network level and at host level (controller). This multilayer deployment ensures that any unauthorised access, malicious communication, lateral computing plane man-in-the-middle attacks, and control plane exploitation can be effectively protected with compatibility with SDN enabled and legacy network devices.

The goal of the experimental assessment was to verify the efficacy of the proposed AM-Sec framework in a realistic attack scenario. To obtain this objective, a number of cyberattacks were carried on the virtualized distributed SDN setting on objective. The experiments involved assessing the detection, prevention, mitigation, and recovery results for the proposed Security Plane, in both known and advanced attack scenarios.

Three types of attack on the data plane and control plane were in particular conducted:

- Flow tunnelling attacks
- Spoofing Attacks
- Malware Injection Attacks

These attacks were chosen for having been some of the toughest challenges in modern SDN infrastructures. Dynamic Flow Tunnelling Attacks are used to variable-flow traffic rules and policies to evade traditional security protection. Spoofing: Spoofed aim and illegal entry to the network by pretending to be trusted entities. Malware attacks have the goal of aiming to compromise controllers, switches, or hosts, and to gain persistent control of network resources.

The attacker's main goal in the experiments conducted was to break into an important server running in the SDN environment. This was done to mimic real-world security policies; the stateful firewall was set up to deny any external connection attempting to communicate with its protected server when it was at an unauthorized state. At the same time, the Anomaly Detection Module (ADM), Network Intrusion Prevention System (NIPS), and Stateful Firewall (SWF), which applied security policies, were always monitoring network activities.

Table 2 shows the implementation results to show the effectiveness of the proposed Security Plane. The results show that all attacks attempted in the experiments were detected, analysed, and mitigated by the integrated security modules. This included high level resistance to Dynamic Flow Tunnelling Attacks with the ability to alter forwarding rules and bypass the traditional firewall security profile. The anomaly detection subsystem quickly detected unusual behaviours in the flow and the NIPS and firewall components responded by implementing suitable mitigation strategies to stop the network resources being compromised.

Also, the centralized architecture of the Security Plane offered full visibility of the distributed SDN infrastructure for quick and easy correlation of security events from multiple domains. This worldwide visibility has considerable enhanced assault detection precision, while minimizing response time in contrast to traditional stand-alone security mechanisms.

Besides the malware attacks and intrusion attacks, Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks, which are among the most severe threats against SDN environments, were tested. Since SDN controllers are centralized, a successful DDoS attack can cause the controller to become out of memory resources, create difficulties with the flow control communications to controller or the SDN controller itself, or even render large parts of the network unavailable.

Using the network testing and attack-generation tool, Hping3, several attack campaigns were launched from the two experimental domains (Area 1 and Area 2) to assess the effectiveness of the proposed DDoS Detection Module. They used various traffic types like ICMP, UDP and TCP to simulate realistic flooding scenarios on the distributed controller infrastructure.

As detailed in Table 3 the proposed DDoS detection framework detected and mitigated the DDoS attacks before it had a major impact on controller availability and network performance. Anomaly detection was performed to constantly check traffic rates and the patterns of connection requests, thus enabling the system to easily detect abnormal traffic amount increases resulting from DDoS traffic. A series of auto fired Mitigation Policies were deployed preventing malicious traffic from affecting the network resources once predefined attack thresholds were hit.

Among the vital elements of the proposed framework is the embedding of the Honey Controller inside the Anomaly Detection Module. Selected attack packets were sent to the Honey Controller, instead of being discarded, for further investigation. This allows the valuable threat intelligence to be collected, such as attack signatures, source information, behaviour characteristics and attack methodologies. The data collected can then be analysed to enhance detection strategies, and further bolster security policies and incident response plans.

In conclusion, the experimental results from the proposed AM-Sec show the possibility of protecting a wide range of cyber-attacks in distributed hybrid SDN environments. The combination of anomaly detection, intrusion prevention, stateful traffic, and forensic analysis and DDoS mitigation capabilities allows the framework to fend off known attacks and even new ones under the radar with minimal performance impact. This experimental result validates that the design goals of this work, which are to improve network security, availability, and retain the programmability and scalability attributes of SDN are achieved.

VII. Conclusion and Future Work

The technology known as Software-Defined Networking (SDN) has presented a new paradigm in networking, providing unparalleled programmability, flexibility, centralized management and control and dynamic network control. However, the control/communication planes separation, the centralization of the controllers and the adoption of distributed/hybrid SDN architectures poses important security issues threatening the network confidentiality, integrity, availability and reliability. Thus, providing solid security is one of the most important needs for a successful application of SDN based infrastructures in enterprise, cloud, industrial, as well as next generation communication networks.

The proposed Security Plane is based on three main security modules that are an Adaptive Stateful Firewall, Network Intrusion Prevention System (NIPS) and Anomaly Detection Module. Intelligent traffic filtering, and secure session management, for SDN-enabled traffic (reactive and proactive flows) and traditional (non-SDN) traffic makes the Adaptive Stateful Firewall a good fit in hybrid network environments. NIPS is a module that lets real-time monitoring, intrusion detection, malware detection and automated threat mitigation, including the minimal burden on the SDN controller. Additionally, the Anomaly Detection Module brings built-in behavioral analysis, Honey-controller functions, and DDoS detection and mitigation capabilities, all of which offer methods to detect and prevent against more advanced persistent threats, zero day and denial of service attacks that would otherwise never get detected via traditional signature based methods.

The proposed framework additionally presents an ensured inter-controller correspondence design covering encoded correspondence, controller synchronization, heartbeat screen, burden sharing and host based security measure strategies to fortify the security and solidness of the disseminated SDN deployments. All these features add up to greater fault tolerance, controller availability, scalability and operational reliability and safeguard sensitive control-plane communications from interception, tampering and unauthorized access.

For the proposed framework, the distributed SDN testbed was modelled virtually with multiple OpenDaylight controllers, OpenFlow switches, Mininet emulation environments, forensic analysis components and specific security modules were used for implementing and evaluating the proposed framework. The results of experiments showed that the proposed architecture was able to detect and mitigate different kinds of attacks like; Spoofing attacks, Malware infection, Dynamic flow tunnelling attacks, DoS attacks and the DDoS attacks. Results have shown that huge detection accuracy, a quick mitigation response, and a good threat containment with low utilization of controllers and minimal drop in controllers' performance were attained with the integrated security modules. Based on these results, one can confirm that the proposed approach is effective to secure the modern distributed hybrid SDN infrastructures. The AM-Sec framework as a whole, helps SDN security to advance with a scalable, modular, adaptive and performance aware security architecture that can protect against known and unknown cyber threats. With a myriad of complementary security mechanisms seamlessly woven into a single context, the network becomes more resilient and provides a solid foundation upon which to build safe next-generation programmable networks.

There are some directions that will be investigated in the future works, for enhancing the presented framework. On the one hand, machine learning and artificial intelligence (AI) technologies will be adopted in the anomaly detection engine, which will better predict attacks, analyse behaviours and adaptively mitigate threats. Second, there will be an integration of advanced threat intelligence and automated security orchestration mechanism enabling real-time response in case of sophisticated attacks. Thirdly, Blockchain based trust management and secure co-ordination among the controllers will be explored to result in a more secure communication among the controllers in large-scale distributed SDN environment. In the end, the framework will be tested to establish the feasibility and performance levels in actual SDN deployments and cloud-native infrastructures under realistic network scales and conditions to validate how scalable, robust and effective it can be in real operation. Hopefully these improvements will further increase the security, efficiency and robustness of future SDN ecosystems.

Ethics statement

This study did not involve human participants, animals, or the collection or processing of personal or sensitive data. The research was based exclusively on publicly available digital artifacts. Therefore, ethical approval and informed consent were not required.

Consent for publication

Not applicable.

Competing interests

The authors declare no competing interests.

Funding

There is no specific funding to support this research.

Data Availability

No original datasets were generated or analyzed in this study. The research is based exclusively on a systematic review and qualitative analysis of publicly available scientific literature and published sources. The sources reviewed are accessible through their respective publishers and academic databases.

References

[1] Masoudi, R., & Ghaffari, A. (2016). Software defined networks: A survey. *Journal of Network and Computer Applications, 67*, 1–25.
[2] Nadeau, T. D., & Gray, K. (2013). *SDN: Software defined networks*. O'Reilly Media.
[3] Sandhya, Sinha, Y., & Haribabu, K. (2017). A survey: Hybrid SDN. *Journal of Network and Computer Applications, 100*, 35–55.
[4] Blial, O., Ben Mamoun, M., & Benaini, R. (2016). An overview on SDN architectures with multiple controllers. *Journal of Computer Networks and Communications, 2016*, Article 9396525.

[5] Zkik, K., Tachihante, T., Orhanou, G., & El Hajji, S. (2016). A modular secure framework based on SDMN for mobile core cloud. In S. Boumerdassi, É. Renault, & S. Bouzefrane (Eds.), *Mobile, secure, and programmable networking* (Lecture Notes in Computer Science, Vol. 10026, pp. 137–152). Springer.

[6] Chiti, F., Morosi, S., & Bartoli, C. (2024). An integrated software-defined networking–network function virtualization architecture for 5G RAN–multi-access edge computing slice management in the Internet of Industrial Things. *Computers, 13*(9), Article 226.

[7] Shahraki, A., Abbasi, M., Taherkordi, A., & Kaosar, M. (2021, August). Internet traffic classification using an ensemble of deep convolutional neural networks. In Proceedings of the 4th FlexNets Workshop on Flexible Networks Artificial Intelligence Supported Network Flexibility and Agility (pp. 38-43).

[8] Bour, H., Abolhasan, M., Jafarizadeh, S., Lipman, J., & Makhdoom, I. (2022). A multi-layered intrusion detection system for software defined networking. *Computers & Electrical Engineering, 101*, Article 108042.

[9] Liu, Y., Zhi, T., Shen, M., Wang, L., Li, Y., & Wan, M. (2022). Software-defined DDoS detection with information entropy analysis and optimized deep learning. *Future Generation Computer Systems, 129*, 99–114.

[10] Qiu, X., Zhang, K., & Ren, Q. (2017). Global flow table: A convincing mechanism for security operations in SDN. *Computer Networks, 120*, 56–70.

[11] Jantila, S., & Chaipah, K. (2016). A security analysis of a hybrid mechanism to defend DDoS attacks in SDN. *Procedia Computer Science, 86*, 437–440.

[12] Okwuibe, J., Liyanage, M., & Ylianttila, M. (2015). Performance analysis of open-source Linux-based HIP implementations. In *2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS)* (pp. 60–65). IEEE.

[13] Lara, A., & Ramamurthy, B. (2014). OpenSec: A framework for implementing security policies using OpenFlow. In *2014 IEEE Global Communications Conference* (pp. 781–786). IEEE.

[14] Shin, S., Yegneswaran, V., Porras, P., & Gu, G. (2013). AVANT-GUARD: Scalable and vigilant switch flow management in software-defined networks. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security* (pp. 413–424). Association for Computing Machinery.

[15] Liyanage, M., Ahmed, I., Ylianttila, M., Llorente Santos, J., Kantola, R., Lopez Perez, O., Uriarte Itzazelaia, M., Montes de Oca, E., Valtierra, A., & Jimenez, C. (2015). Security for future software defined mobile networks. In *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies* (pp. 256–264). IEEE.

[16] Zhu, S., Bi, J., & Sun, C. (2014). SFA: Stateful forwarding abstraction in SDN data plane. In *Proceedings of the 2014 Open Networking Summit Research Track*. USENIX Association.

[17] Yao, G., Bi, J., & Guo, L. (2013). On the cascading failures of multi-controllers in software defined networks. In *2013 21st IEEE International Conference on Network Protocols (ICNP)* (pp. 1–2). IEEE.

[18] Braga, R., Mota, E., & Passito, A. (2010). Lightweight DDoS flooding attack detection using NOX/OpenFlow. In *2010 IEEE 35th Conference on Local Computer Networks* (pp. 408–415). IEEE.

[19] Pinheiro, A. J., Gondim, E. B., & Campelo, D. R. (2017). An efficient architecture for dynamic middlebox policy enforcement in SDN networks. *Computer Networks, 122*, 153–162.

[20] Alwabisi, S., Ouni, R., & Saleem, K. (2022). Using machine learning and software-defined networking to detect and mitigate DDoS attacks in fiber-optic networks. *Electronics, 11*(23), Article 4065.

[21] Dell Compellent. (2013). *Citrix XenServer 6.x best practices for Dell Compellent Storage Center* [Technical white paper]. Dell.

[22] Mininet Project. (n.d.). *Introduction to Mininet*. GitHub. Retrieved May 30, 2026, from https://github.com/mininet/mininet/wiki/Introduction-to-Mininet