*Journal of Telecommunication, Electronic and Computer Engineering*
ISSN: 2180 – 1843 e-ISSN: 2289-8131 Vol. 14 No. 2
jtec.utem.edu.my

# SDN Multi-Domain Supervisory Controller with Enhanced Computational Security Count

Adamu Abubakar[1], Abdul Razaq Atayee[1], and Ibrahim A. Lawal[2]
[1]*Deparment of Computer Science, International Islamic University Malaysia, 53100, Kuala Lumpur, Malaysia.*
[2]*Department of Information Technology, Bayero University Kano, Nigeria.*
*adamu@iium.edu.my*

**Article Info**

***Article history:***
Received June 22nd, 2022
Revised June 28th, 2022
Accepted June 30th, 2022

***Index Terms:***
SDN supervisory controller
Multi-domain environment
Security, dependency
Latency

**Abstract**

As a new paradigm, software-defined networks (SDNs) are becoming increasingly popular in the network world. From the available research, it can be concluded that SDN multi-domain environments have been under-protected. In areas where security was not a primary concern, management and policies received the vast majority of attention. Previous studies have proposed a "distributed SDN Supervisory Controllers in Multi-Domain Environment" but still suffer from an operational count limitation. An improved framework for distributed SDN supervisory controllers operating in a multi-domain environment is the main focus of this paper. Additionally, we implemented an SDN supervisory controller on the network layer and a global SDN supervisory controller to improve operational counts for security. Furthermore, a security layer with a local and a global security controller was implemented. To test the proposed framework's compatibility, we constructed a network using Mininet, consisting of virtual hosts, switches, controllers, and links. In order to connect the various domains and the control centre, the network uses a wide area network. The switch takes an average of 120 milliseconds, with a packet loss rate of 1.89 percent on average, according to simulation and experiment results. A more efficient security architecture has been put forth, and it is superior to the one currently in place.

## I. INTRODUCTION

The traditional network was downright difficult to manage due to its intricate design, making the task particularly challenging [1]. The difficulty lies not only in the configuration of the network following the predefined policies and its reconfiguration in response to faults, load, and change. The vertical integration of the control and data planes also contributes to the challenges. Specifically, the difficulty lies in that predefined policies must be followed when configuring the network [2]. Both of these aspects present potentially difficult situations. SDN is a promising technology that solves this problem by breaking the vertically integrated control and data plane and separating the routers and switches from the network control logic [3]. In other words, SDN breaks the vertical integration of the control and data planes. In addition, it promotes the centralized control of networks and the programmability of networks. The management of a network is carried out by software rather than through the direct intervention of humans [4].

SDN is an emerging network technology that addresses the challenges currently faced by network infrastructure in the form of its complexity, inability to scale, and reliance on a limited number of vendors [5]. This is made possible by the technology's three primary principles: the separation of software and the physical layer, the centralized control of information, and network programmability. These principles allow the technology to accomplish this goal. A network needs resilience, scalability, and extension to interconnect data centres and various enterprises with wide networks successfully. This allows for successful interconnection [5]. However, software-defined networking (SDN), which creates new links that present new security challenges that were not present in traditional networks, has introduced new attack surfaces [6].

Numerous potential entry points for attackers have been identified as a direct result of the fact that the data plane and the control plane are kept separated from one another [7]. These include attacks on the device that is a part of the data plane; attacks on the links that connect the devices that are a part of the data plane; attacks on the control plane; attacks on the link that connects the control plane to the data plane; and attacks on the application that is built explicitly on that controller [8].

The attacks on the device that is a part of the data plane and the attacks on the links that connect the devices OpenFlow are central to a communication protocol. That area resides between the data plane and the control plane. It does not have any kind of security and is highly vulnerable to being breached by an adversary due to its lack of protection because it does not have any security [9]. In addition, the SDN's centralization gives rise to a variety of security risks, the vast majority of which are linked with the concept of centralization itself. Attackers will likely spend the majority of their attention on centralized data. There is a significant possibility that this will happen [10].

Even though centralized network administration is essential, it is clear from this example that it poses a significant threat to network security and has the potential to create a single point of failure (SPOF), which would result in the destruction of the network as a whole. Researchers devised the Distributed SDN supervisory controller as a solution to the problem of SPOF [8]. This controller is designed to avoid having a single point of failure, which means that an attack on any controller will only affect that specific controller [11]. However, several attacks will still be in various parts of the SDN Multi-domain environment. The application layer of an SDN supervisory controller is susceptible to various assaults, including Denial of Service attacks and Malformed packet attacks, according to several studies that have demonstrated this vulnerability [12]. As a result of these flaws in the SDN architecture, we can see a significant need to improve the SDN architecture's level of security. Therefore, to strengthen the robustness of distributed SDN supervisory controllers operating in multi-domain contexts, we provide a security framework for distributed SDN supervisory controllers in this research. [13].

The fact that it is abundantly evident that the centralization of data poses a substantial security risk since it has the potential to result in a SPOF, which would lead to the destruction of the network, is the basis for highlighting the difficulties that have been described above [8]. One of the reasons why SPOFs are sometimes referred to as "single points of failure" is because of this fact. Previous research has devised a solution called the Distributed SDN supervisory controller to address the issue of single points of failure in networks (SPOF) [8-13]. An attack on any controller will only affect that particular controller, even if multiple controllers are targeted in the attack because this controller was created to avoid having a single point of failure [14]. Despite this, some attacks will still be in the various components that make up the SDN Multi-domain environment [15].

A number of studies have demonstrated this vulnerability. Another justification is that an SDN supervisory controller's application layer is vulnerable to various attacks, according to several studies that have demonstrated this vulnerability. These attacks include attacks that result in a denial of service, as well as attacks that result in malformed packets [16]. When we consider each of these vulnerabilities in the SDN architecture, we can see an urgent need to improve the level of security offered by the SDN architecture [17]. Consequently, this research proposes a security framework for distributed SDN supervisory controllers that operate in contexts containing several domains. This architecture will increase the failure-resilience of distributed SDN supervisory controllers operating in multi-domain contexts [18].

## II. RELATED WORK

The identification of malicious attacks on SDN has been the subject of several studies that have been carried out. The SDN is also the target of a significant number of attacks and dangers coming from a variety of directions. The most crucial aspect of any prevention strategy is figuring out how to recognize typical assaults, such as distributed denial-of-service attacks and numerous instances of malicious node isolation, as well as entropy in the origin of the attack 19]. One of the significant efforts that have been made to ensure that SDN is secure is the research work carried out by Cabaj et al. [20]. According to the study's findings, using software such as CryptoWall and Locky makes it possible to identify threats disguised as HTTP message sequences and the content sizes of those messages. RouteGuardian is a dependable and security-focused SDN routing mechanism that effectively analyses abnormal traffic and isolates malicious nodes by merging the SDN switch node and a framework for network security virtualization. This method was developed by Wang et al. [21]. Diogoet al. [22] found many security flaws in SDN and introduced AuthFlow as a solution to protect against these types of attacks. AuthFlow is a system for authenticating users and controlling access to resources. It is predicated on host credentials.

Researchers have discussed several different security measures to provide consistent and transparent security policies to guarantee network safety, performance, and scalability and to keep the network functioning correctly in the event of a failure. A distributed controller is a solution proposed by Mattos et al. [23] to ensure security, performance, and scalability. Additionally, it ensures the correct operation of the network even when nodes are in a state of failure by providing consistency to the control plane and keeping track of their status. Aslam et al. [24] created a distributed denial-of-service attack detection and mitigation system for SDN-enabled IoT based on adaptive machine learning. It was discovered that integrating SDN into IoT network devices enables additional security measures and reduces a significant amount of the computational overhead generally associated with those devices.

It has been determined that in order to meet information security objectives in an environment where threat actors are continually becoming more sophisticated and new threats are constantly emerging, it is necessary to make use of the most recent technologies and tools. A thorough exploratory analysis of SDN-based cyber defence is presented in Yurekten and Demirci's [25] research report. The research conducted by Kakkavas et al. [26] investigates the effectiveness of monitoring in SDN-enabled 5G networks to develop network tomography. It was discovered that network tomography could respond to the current monitoring challenges by complementing and working together with SDN. This combination produces accurate estimations with low overhead while exploiting SDN capabilities such as the centralized view of the entire network, direct flow-level measurements, and controllable routing.

An SDN-assisted safety message dissemination framework for vehicular critical energy infrastructure was proposed by Prathiba et al. [27]. It was discovered that the simulation results demonstrate that the migrating consignment region is superior to other regions in terms of the amount of network overload. An improved 5G SDN/NFV edge was proposed by Paolucci et al. [28], which would allow software and SDN platforms to support the heterogeneity and complexity of various requirements. In SDN-based fog computing systems, Phan et al. [29] developed a dynamic offload from one fog to another fog. Through real-time monitoring of fog node status and the selection of optimal offloading nodes as well as end-to-end routing paths between overloaded and offloaded nodes, the system could identify nodes that had reached their capacity and were considered to be overloaded.

## III. METHODOLOGY

A secure framework for distributed SDN supervisory controllers in a multi-domain environment requires an SDN design process and the method utilized to design the security processes. A design science research methodological approach is selected for this current study [30]. The specific method involved building a framework for distributed SDN supervisory controllers in a multi-domain environment. This involves designing a supervisory control for a communication network to connect multi-domain networks, as shown in Figure 1.

### *A. Development of the Conceptual Model*

The framework lies within the scope of the job of a Distributed SDN supervisory controller in multi-domain environments. As presented in Figure 1, end-to-end flow management purposes were set to communicate with neighbouring domains to exchange aggregated network information. This allows network operations to be managed in real-time and verify their accuracy. With distributed SDN supervisory controller's global view, malicious assaults and unintentional errors may be detected and isolated so that abnormal actions can be quickly corrected.

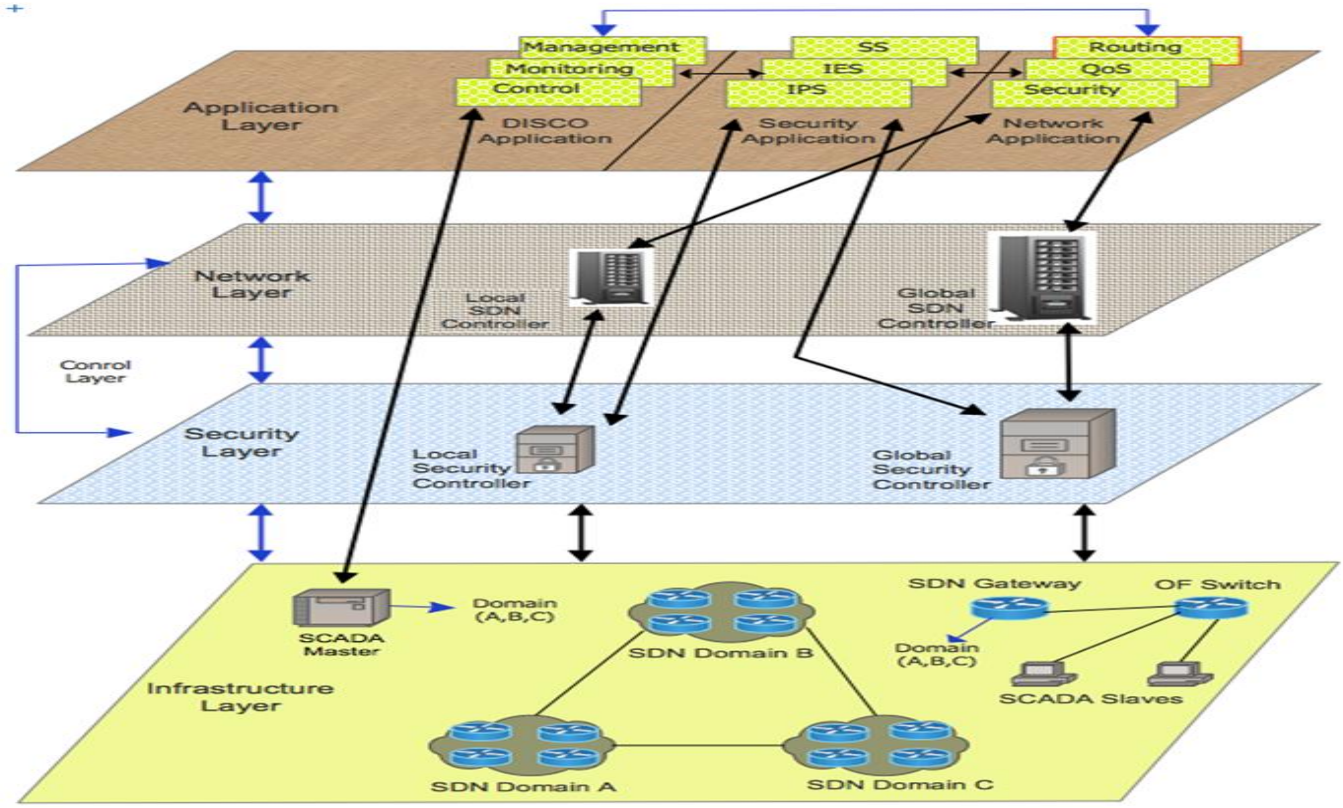

Figure 1. Secure distributed SDN supervisory controller in a multi-domain environment.

The infrastructure, control, and application layers are depicted in Figure 1. A multi-domain SDN control plan distributed SDN supervisory controllers and OF switches that comprise the infrastructure layer. Sub-layers of the control layer include security and networking. A global and local security controller is located in the security layer. An SDN supervisory controller for the control centre and a local SDN supervisory controller for the substation make up the network's sublayer. Other components such as SDN supervisory controllers, security controllers, supervisory control and data acquisition masters are all run by the application layer. The SDN supervisory controller runs applications for device management and provisioning of routing and QoS. The supervisory control and data acquisition master control, configure and manage supervisory control and data acquisition slaves with the help of application programmes. Three application programmes run on the security controller: i) a security system (SS) that generates and manages the cryptographic keys; ii) an intrusion elimination system (IES) that eliminates the attackers detected by IDS; and iii) an intrusion detection system (IDS) that monitors all devices and their activities in the control centre, generates an alarm, and notifies IES once attacked is detected.

We also use digital signatures for broadcast or multicasting messages and MACs for unicasting message devices to ensure message authentication in platform communication. Device A signs the broadcast or multicast messages with its private key and sends the signed messages and their ID to the recipient. When device B receives the signed message, it can use A's public key to verify the signature. Using the secret session key for the message, device A generates a MAC tag and sends the message and the MAC tag to device B. In addition, the secret key from the received message from A is used to generate a MAC tag for device B. It then performs a MAC tag comparison with the MAC tag received from A and the one generated. Device A is authenticated to B if both tags are the same.

A central control room houses the global security controller, whereas a substation houses the local security controller. A visual representation of the local security controller workflow and a visual representation of the global security controller workflow can be found in Figures 2 and 3. Security controllers send a packet to an IDS based on whether or not the verification process has been completed successfully. If the authentication tag is missing, the packet is discarded by both security controllers (signature or MAC). In addition, the security controllers verify the packet's authentication and integrity again.

The local connection collects the measurement data periodically from supervisory control and data acquisition slaves and verifies for suspicious data. It generates an alarm and notifies the connection if suspicious data is detected. Otherwise, the local connection allows sending the measurement data from the substation to the control centre. It further monitors the control commands sent by supervisory control and data acquisition master executed on supervisory control and data acquisition slaves. The global connection collects the measurement data from the substations. It verifies the measurement data for bad data detection and identification. Then, the global connection measures the consequences of control commands issued by either the SDN supervisory controller or supervisory control and data acquisition master. It generates an alarm message and notifies the connection if suspicious data is found, as shown in Figure 2.

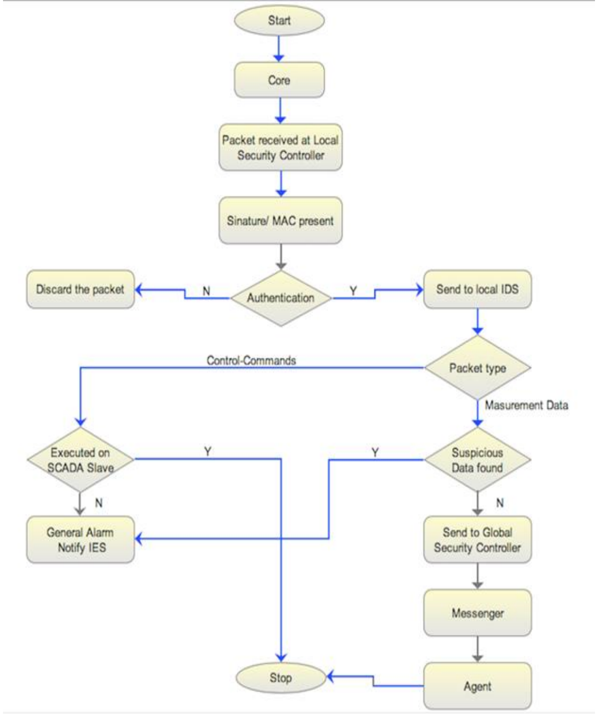

Figure 2. Workflow diagram at local security controller

Data is sent to messenger once it has been filtered and checked. In addition to receiving packet IN messages from the Core module, it can write its packet OUT messages, make calls and store information in the extended database, read the Floodlight configuration file upon startup, and subscribe to receive packet IN messages from that module. Figure 3 shows the configuration settings that must be met: message server type, listening port, and agent list. With messenger, neighbouring domain controllers can be discovered using an expanded version of the Link Layer Discovery Protocol. When it is required to use Open Flow, it must be associated with the use messages.

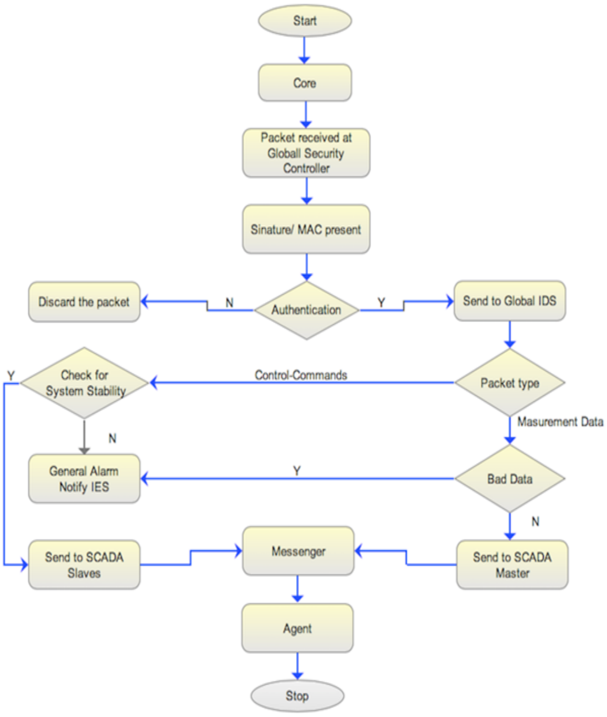

Figure 3. Workflow diagram at global security controller

Agents use messenger to exchange information with neighbouring domains. It has implemented four agents: Monitoring, Reachability, Connectivity and Reservation. Every second, a monitoring agent at the controller with identifier ID advertises the amount of bandwidth it has available for traffic transit. Upon receiving information from agents in neighbouring domains, the local agents store them in the extended Floodlight database. Local modules then use this information to make decisions on flows. These decisions are generally the outgoing peering link to choose for a given flow. The Reservation agents implement an RSVP-like reservation protocol for the end-to-end provision of resources. Agents thus exchange reservation requests and responses with flow descriptors.

### *B. Development of the Experimentatial Scenarios*

The experimentation scenario involves a network topology presented in Figure 4. Each network domain A, B, and C is governed by a local distributed SDN supervisory controller, which works in coordination with the distributed SDN supervisory controllers in its neighbouring networks. In Figure 4, each network controller is connected with an SDN gateway and a Supervisory Control and Data Acquisition master. The infrastructure layer, SCADA, supervisory controller, and SDN gateway are all marked by double-sided arrows. These arrows are bidirectional in any given setting, indicating that the design support two points on each end, i.e. internal and external. If the inter-domain link between A and C fails, monitoring agents will reconfigure themselves to transfer data from A to B, and then monitoring traffic will be passed through the weak link B to C.

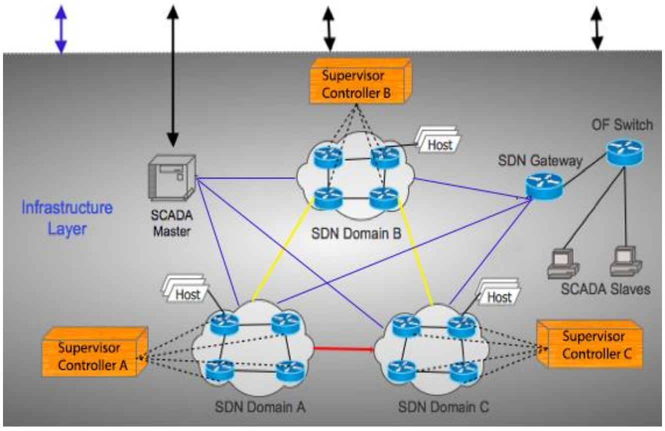

Figure. 4. Setup in Multi-domain SDN Topology

By disabling the connection between domains A and C, the security application in domain A will be able to send filtered data to domain A from the core, and the supervisory control and data acquisition master will be able to connect domain A to domain C. However, even though domains A and C are protected by security, they are unable to share data with any other domains. The treatment of attacks in distributed SDN supervisory controllers is illustrated in Figure 4, which is an important part of the Figure. The control layer manages the SDN supervisory controller, the supervisory control, and data acquisition master computers and servers. Several controls, including status master monitoring, under frequency load shedding, frequency and voltage control, are carried out by the supervisory control and data acquisition master.

The supervisory control and data acquisition master is responsible for processing the data that has been received, and it is the supervisory control and data acquisition slave that is responsible for receiving control-commands such as read, write or execute. The framework's integrity is vulnerable in two ways: i) There is the potential for a breach of the supervisory control and data acquisition master; ii) OF switches may be vulnerable to attack. The packets that carry measurement data or control commands from supervisory control and data acquisition slave or master to supervisory control and data acquisition master or slave are dropped, injected, and delayed by the OF switch. The main causes of inaccurate data are the poor calibration of supervisory control and data acquisition slaves, the failure of supervisory control and data acquisition slaves, and the injection of malicious measurement by supervisory control and data acquisition slaves. Mininet has been our tool of choice for the testing phase. Using the network emulator known as Mininet, we were able to construct a network consisting of virtual hosts, switches, controllers, and links. The Mininet is a tool that is utilized in the creation of rich topologies as well as the instantiation of Open vSwitch switches and virtual hosts. The Mininet is hosted on a VM that is dedicated exclusively to it. The controllers each have their own VM. With this configuration, we have granular control over the network.

## IV. RESULTS AND DISCUSSION

Within the context of this research simulation, Mininet is utilized to test the functionality of the system in order to run the experimental topology successfully. During this phase of testing, both the script and the simulation that runs the script for distributed SDN supervisory controllers were put through their paces in various settings. We can create a network consisting of virtual hosts, switches, controllers, and links by using Mininet. The network utilizes a wide area network to establish connections between the various domains and the control centre. The control centre essentially contains a global SDN supervisory controller and a virtual host responsible for running our programme. Mininet makes it simple to obtain a system's expected behaviour and conduct experiments with its topology. We tested the system on Mininet to ensure that an OpenFlow controller, modified switch, or host could move to a real system with minimal changes, which is necessary for real-world testing, performance evaluation, and deployment.

Figure 5 illustrates the performance metrics for network metrics and service interruptions. Now, the Virtual machine can be accessed from either of the two domains. We can see the drops in latency that occur at the precise moment when the flow's path is altered. In order to obtain average values, we carried out this experiment. The packet loss rate for the switch is 1.89 percent on average (see Figure 6), and it takes an average of 120 milliseconds (see Figure 6). In any case, the results of this experiment demonstrate the adaptability of distributed SDN supervisory controllers in an environment in which end-hosts are free to move between domains and in which the controllers can reroute the communication of the end-hosts without interruption.

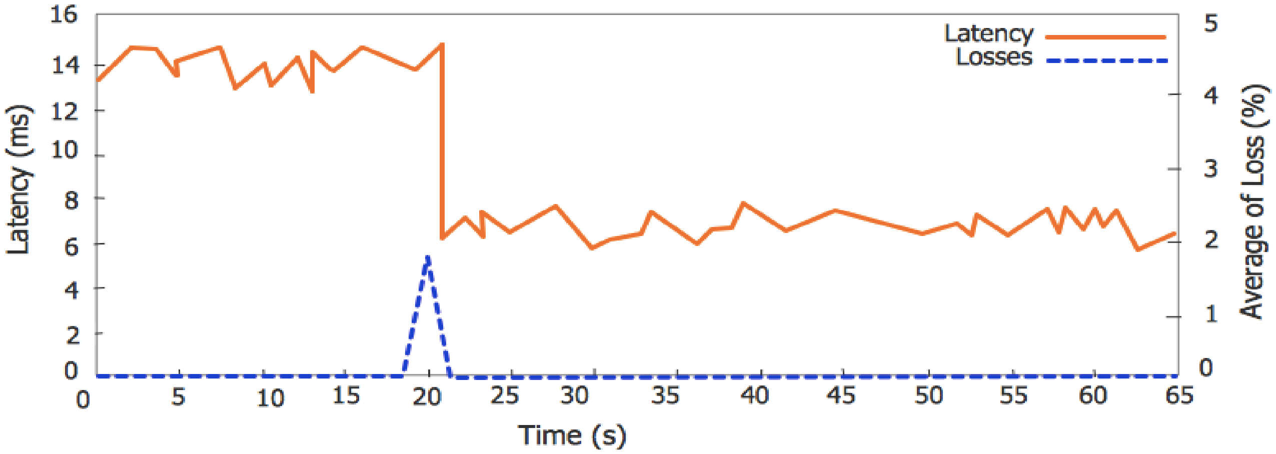

Figure. 5. The Result of the Impact on Flow Latency and Loss Rate

The results of the experimental study have further demonstrated that implementing a particular attack within the simulation yielded some results. Figure 6 is a representation of the proposed intrusion detection system (IDS) and intrusion elimination system detecting and neutralizing a denial-of-service attack (DoS attack) (IES). By sending an overwhelming number of Echo request packets using the Internet Control Message Protocol, an adversary can cause a denial of service (ICMP). We concluded that our intrusion detection system (IDS) threshold should be set at 100 packets per second.

This part of the article will explain how we decomposed the security framework for distributed SDN supervisory controllers. We offer a global security controller as a service to distributed SDN supervisory controllers operating in environments with multiple domains. This controller is responsible for ensuring the safety of communication devices. It established a connection with the neighbourhood SDN supervisory controller, which is responsible for regulating communication between devices operating within the same domain of an infrastructure application. There are master and slaves for supervisory control, data acquisition, and optical fibre switches inside the infrastructure application. Several domains are connected to an SDN gateway, an OF switch, and a supervisory control and data acquisition master controller. Consequently, we connected security controllers to each domain without a data center. We removed the datacenter from the data control system because if it received any malicious data, it would cause the shutdown of the entire system.

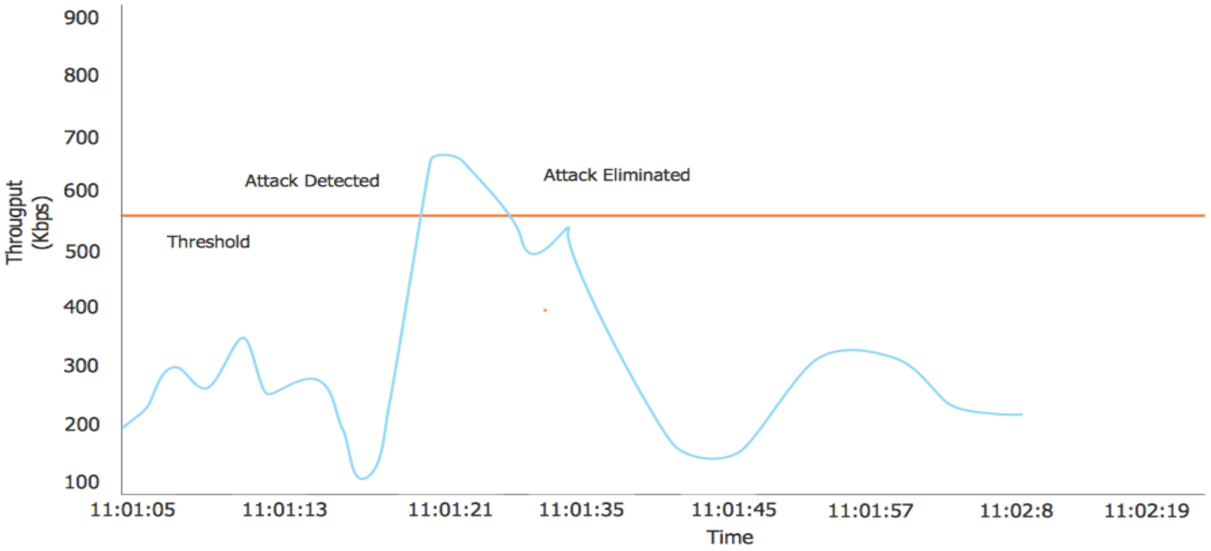

Figure. 6. DoS attack detection and elimination

The communication infrastructure can be made scalable, reliable, secure, and efficient through the linking of security controllers to distributed SDN supervisory controllers in an environment with multiple domains. Even if one of the domains does not work correctly, the functionality of the other domains will not be affected. In addition, data that has been filtered can move from the core to the messenger at the same time. If that is not the case, a functional attack, also known as a DOS attack, is simple to carry out in infrastructure communication.

Not only has there been progress made in creating and building advanced network architecture, but there has also been progress made in breaching and compromising network communication due to the proliferation of technological advancement. Every single network device should prioritize security as the top priority. Once we are connected to the network, if there are no security measurements and policies in place, our communications and the data that is sent and received are extremely susceptible to any data breaches that may occur. In addition, we implemented a network layer consisting of a local supervisory controller for the SDN and a global supervisory controller for the SDN. A local and global security controller are both components of the security layer we have implemented. A local SDN supervisory controller is responsible for controlling the communication between devices located within the substation. A local security controller is responsible for providing security for switches, a gateway, and supervisory control and data acquisition slaves located within the substation. Whereas a global SDN supervisory controller is in charge of communication between control centres, a global security controller is in charge of providing security in communication devices, and a supervisory control and data acquisition master is in charge of controlling, monitoring, and managing distributed SDN supervisory controllers devices such as core, agents, and messenger. These responsibilities fall under the purview of the supervisory control and data acquisition master. We created a network of virtual hosts, switches, controllers, and links using mininet to test the compatibility of the proposed framework. Over a wide area network, the network establishes connections between the various domains and the control centre. Mininet uses standard Linux network software and switches support open flow for highly flexible custom routing and SDN. There is a high possibility that DOS and DDOS attacks will be directed toward the multi-domain environment because it is a newly adopted architecture in software-defined networks; however, due to the provided security framework, it will not allow any of these attacks to take place.

## V. CONCLUSION

Security in SDN is not just about detection, prevention, or correcting a problem; instead, it is about the computational cost of all these activities. As a result of the computational count that this work measured, it has made a significant contribution to computer science in associating resources and functions. According to the findings of this study, software-defined networking (SDN) is not only very important but also gaining more and more traction in the world of networks. Its applications and uses by enterprise organizations are highly crucial and essential to the field. As a consequence of this, the ongoing research has determined that there is other prior research that demonstrates concern regarding SDN security. In addition, the most recent research has demonstrated that efforts have been made to address the concerns regarding the safety of SDN. As a result, protective security management policies were implemented within an SDN environment consisting of multiple domains. Unfortunately, the security that was developed, known as "Distributed SDN Supervisory Controllers in Multi-Domain Environment," was not efficient enough. Because of this, the current research has proposed an improved version of "Distributed SDN Supervisory Controllers in Multi-Domain Environment." An enhanced framework for operating distributed SDN supervisory controllers in an environment with multiple domains has demonstrated the importance of having an SDN supervisory controller on the network layer. In addition, the study found that the performance count on enhanced distributed SDN supervisory controllers in a multi-domain environment takes an average of 120 milliseconds, with an average packet loss rate of 1.89 percent, according to the results of simulation and experimentation.

## REFERENCES

[1] M. Alsaeedi, M.M. Mohamad, & A.A. Al-Roubaiey,. Toward adaptive and scalable OpenFlow-SDN flow control: A survey. IEEE Access, 7, 2019, pp. 107346-107379.

[2] P.P. Ray., & N. Kumar. SDN/NFV architectures for edge-cloud oriented IoT: A systematic review. Computer Communications, 169, 2021, pp. 129-153.

[3] D. Kreutz, F.M. Ramos, P.E. Verissimo, C.E. Rothenberg, S. Azodolmolky, & S. Uhlig. Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1), 2014, pp. 14-76.

[4] K. Phemius, M. Bouet, & J. Leguay. Disco: Distributed multi-domain sdn controllers. In 2014 IEEE Network Operations and Management Symposium (NOMS), 2014, pp. 1-4.

[5] S. Khorsandroo, A.G. Sanchez, A.S. Tosun, J.M. Arco, & R. Doriguzzi-Corin, R. Hybrid SDN evolution: A comprehensive survey of the state-of-the-art. Computer Networks, 192, 2021, 107981.

[6] M.R. Haque, S.C. Tan, Z. Yusoff, K. Nisar, R. Kaspin, I. Haider, I., Nisar, S., Rodrigues, J.J., Shankar Chowdhry, B., Uqaili, M.A. and Prasad Majumder, S., Unprecedented smart algorithm for uninterrupted SDN services during DDoS attack. Computers, Materials & Continua, 70(1), 2022, pp.875-894.

[7] Deb, R., & Roy, S. (2022). A comprehensive survey of vulnerability and information security in SDN. Computer Networks, 108802.

[8] Zhang, X., Cui, L., Wei, K., Tso, F. P., Ji, Y., & Jia, W. (2021). A survey on stateful data plane in software defined networks. Computer Networks, 184, 107597.SPOF

[9] Mishra, A., Gupta, N., & Gupta, B. B. (2021). Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX controller. Telecommunication systems, 77(1), 47-62.

[10] Rahman, A., Chakraborty, C., Anwar, A., Karim, M., Islam, M., Kundu, D., Rahman, Z. & Band, S. S. (2021). SDN–IoT empowered intelligent framework for industry 4.0 applications during COVID-19 pandemic. Cluster Computing, 1-18.

[11] Ahmad, S., & Mir, A. H. (2021). Scalability, consistency, reliability and security in SDN controllers: a survey of diverse SDN controllers. Journal of Network and Systems Management, 29(1), 1-59.

[12] Ahammad, I., Khan, M. A. R., Salehin, Z. U., Uddin, M., & Soheli, S. J. (2021). Improvement of QOS in an IoT ecosystem by integrating fog computing and SDN. International Journal of Cloud Applications and Computing (IJCAC), 11(2), 48-66.

[13] Balasubramanian, V., Aloqaily, M., & Reisslein, M. (2021). An SDN architecture for time sensitive industrial IoT. Computer Networks, 186, 107739.

[14] Anerousis, N., Chemouil, P., Lazar, A. A., Mihai, N., & Weinstein, S. B. (2021). The origin and evolution of open programmable networks and sdn. IEEE Communications Surveys & Tutorials, 23(3), 1956-1971.

[15] Rahman, A., Islam, M. J., Montieri, A., Nasir, M. K., Reza, M. M., Band, S. S., ... & Mosavi, A. (2021). Smartblock-sdn: an optimized blockchain-sdn framework for resource management in IoT. IEEE Access, 9, 28361-28376.

[16] Novaes, M. P., Carvalho, L. F., Lloret, J., & Proença Jr, M. L. (2021). Adversarial Deep Learning approach detection and defense against DDoS attacks in SDN environments. Future Generation Computer Systems, 125, 156-167.

[17] Ren, X., Aujla, G. S., Jindal, A., Batth, R. S., & Zhang, P. (2021). Adaptive recovery mechanism for SDN controllers in Edge-Cloud supported FinTech applications. IEEE Internet of Things Journal.

[18] Sedaghat, S., & Jahangir, A. H. (2021). RT-TelSurg: Real time telesurgery using SDN, fog, and cloud as infrastructures. IEEE Access, 9, 52238-52251.

[19] Ni, T., Gu, X., Wang, H., & Li, Y. (2013). Real-time detection of application-layer DDoS attack using time series analysis. Journal of Control Science and Engineering, 2013.

[20] Cabaj, K., Gregorczyk, M., & Mazurczyk, W. (2018). Software-defined networking-based crypto ransomware detection using HTTP traffic characteristics. Computers & Electrical Engineering, 66, 353-368.

[21] Wang, M., Liu, J., Mao, J., Cheng, H., Chen, J., & Qi, C. (2017). RouteGuardian: Constructing secure routing paths in software-defined networking. Tsinghua Science and Technology, 22(4), 400-412.

[22] Mattos, Diogo Menezes Ferrazani, and Otto Carlos Muniz Bandeira Duarte. "AuthFlow: authentication and access control mechanism for software defined networking." Annals of Telecommunications 71.11-12 (2016): 607-615

[23] Mattos, D. M., Duarte, O. C. M., & Pujolle, G. (2016, May). A resilient distributed controller for software defined networking. In 2016 IEEE International Conference on Communications (ICC) (pp. 1-6).

[24] Aslam, M., Ye, D., Tariq, A., Asad, M., Hanif, M., Ndzi, D., Chelloug, S.A., Elaziz, M.A., Al-Qaness, M.A. and Jilani, S.F. (2022). Adaptive Machine Learning Based Distributed Denial-of-Services Attacks Detection and Mitigation System for SDN-Enabled IoT. Sensors, 22(7), 2697.

[25] Yurekten, O., & Demirci, M. (2021). SDN-based cyber defense: A survey. Future Generation Computer Systems, 115, 126-149.

[26] Kakkavas, G., Stamou, A., Karyotis, V., & Papavassiliou, S. (2021). Network tomography for efficient monitoring in SDN-enabled 5G networks and beyond: Challenges and opportunities. IEEE Communications Magazine, 59(3), 70-76.

[27] Prathiba, S. B., Raja, G., Bashir, A. K., Alzubi, A. A., & Gupta, B. (2021). SDN-assisted safety message dissemination framework for vehicular critical energy infrastructure. IEEE Transactions on Industrial Informatics.

[28] Paolucci, F., Cugini, F., Castoldi, P., & Osiński, T. (2021). Enhancing 5G SDN/NFV edge with P4 data plane programmability. IEEE Network, 35(3), 154-160.

[29] Phan, L. A., Nguyen, D. T., Lee, M., Park, D. H., & Kim, T. (2021). Dynamic fog-to-fog offloading in SDN-based fog computing systems. Future Generation Computer Systems, 117, 486-497.

[30] Carstensen, A. K., & Bernhard, J. (2019). Design science research–a powerful tool for improving methods in engineering education research. European Journal of Engineering Education, 44(1-2), 85-102.