Bulletin of Pure and Applied Sciences
Zoology (Animal Science), Vol.43B, No.2s,
July-December 2024: P. 688-704

Print version ISSN 0970-0765
Online version ISSN 2320-3188

Original Research Article

Content Available online at: www.bpasjournals.com

SOFTWARE-DEFINED NETWORK SECURITY AND ROUTING MECHANISMS: AN OVERVIEW

[1*]Mrs.J.Delshi Julie, [2]Dr.R. Beulah

[1*]Assistant Professor, Department of Computer Science, Bishop Ambrose College,
Coimbatore -641 045.

[2]Assistant Professor, Department of Computer Science, PPG College of Arts and Science,
Coimbatore -641 035

ABSTRACT: The Data Plane (DP) and network Control Planes (CP) are separated by Software Defined Networking (SDN). Applications are separated from the underlying network infrastructure, which is conceptually centralised in terms of network intelligence and state. By providing global insight into the network state and facilitating easy conflict resolution from the logically centralised control plane, SDN improves network security. As a result, the SDN design gives networks the ability to actively monitor traffic and identify dangers, which makes it easier to do network forensics. To examine risks to the SDN's application, CP, and DP in this study. Various security techniques for network-wide security in SDN are described after the security platforms that safeguard each of the planes. The focus on Deep Reinforcement Learning (DRL) for optimizing routing in SDN involves the application techniques to improve network performance and adaptability. In summary, this article outlines the routing difficulties that SDN faces today and in the future, as well as potential paths for safe SDN.

KEYWORDS: Routing, Software Defined Network (SDN), Security, Denial-of-service (DoS)

INTRODUCTION

SDN is a crucial network topology for enhancing network administration and driving innovation in communication networks. It separates the data forwarding plane and network control. Logically centralised, the CP makes it easy for the forwarding aircraft to respond to choices made by the CP [1]. By implementing software-based reasoning in the CP of SDN, additional control functions may be added. This logic then installs decision logic in the forwarding plane using standard interfaces. The control plane's Network Operating System (NOS) links all of the network's services and apps to various implementations that are built on top of the control plane [2].

With its centralised behaviour control, global network state visibility, and run-time traffic forwarding rule modification, SDN improves network security. SDN's centralised networking architecture reduces the possibility of policy clash and allows for the enforcement of network-wide security regulations [3]. Through the controller and the data path, a NS (Network Security) application (like security monitoring programme) can request flow samples [4]. Following a security analysis, the data path parts can be redirected by the security application to limit traffic inside a certain network jurisdiction, reroute to security middle boxes, or block traffic altogether [5]. Furthermore, rather than altering the hardware or firmware of the controller platform, modifying security rules in SDN necessitates updating the security applications or adding security modules.

Conversely, traditional networks are made up of several manually programmable devices that are customised to each vendor and dispersed over multiple networks. Based on function-specific logic in each device, these devices are hardwired with particular algorithms used to route, control, and monitor data flow [6]. Because of this, it is challenging to seamlessly integrate them into a single domain with all of the proprietary programmes, protocols, and interfaces. As a result, outdated network designs struggle to implement and maintain consistent network-wide policies and lack global awareness of the network state. The complexity and integration flaws make it more difficult to maintain reliable and consistent network security [7]. For instance, it is essentially impossible and expensive to update or modify security settings in these systems in response to attacks or changes in traffic behaviour.

In network topologies that are outdated, network security is viewed as an optional feature that depends on manually adjustable perimeter-based solutions [8]. Network operators must use vendor-specific low-level instructions to configure each device in order to apply a high-level NS policy. However, configuration errors and intra- and inter-domain policy conflicts can lead to significant security breaches and threats as the NS tools are manually configured on extensive sets of devices [10], these tools including firewalls, Intrusion Detection System (IDS), Intrusion Prevention Systems (IPS), and IPsec. Manual low-level setups on individual devices cause corporate firewalls to enforce rule sets that go beyond established security principles and cause security breaches according to a quantitative research on firewall configuration errors [11].

SDN is not without its own difficulties and constraints, though, particularly with regard to security, scalability, and maintenance. Among these difficulties, security has been at the forefront. Since the whole network is managed by a centralised controller, a security breach in the controller might jeopardise the network as a whole. Security flaws in controller data path communication can also result in unauthorised access to and use of network resources [12]. SDN allows apps to communicate with the control plane in order to deploy new features, access network resources, and control network behaviour. However, protecting the network from malicious apps or apps that behave strangely is a significant security risk with SDN [13]. A networking technology's ability to maintain network security is essential, and communication networks need to offer end-to-end communication security.

In this work, it describes the security concerns in SDNs and provide suggested security platforms and solutions [14]. The NS in SDN is offered in the study. A synopsis of the NS risks in SDN and a description of some of the current models for improving security in SDN. As mentioned in, fewer SDN security solutions exist despite the fact that security has been acknowledged as a benefit of SDN. Nevertheless, lacks coverage of the most current developments in SDN security and has a narrow scope [15]. Our goal in this work is to offer a thorough and current review of SDN security by addressing security issues and solutions specific to the application, control, and data planes, three different SDN planes.

LITERATURE REVIEW

Shuvro et al [16] modifies the transformer design for time-series vehicle data in the forecast traffic flow. The dataset is used to create time-series sequences that capture temporal dependencies. We have designed our suggested transformer-based approach to capture both inter-sample and inter-feature correlations. When compared to Transformers and LSTM-based models, the 2D-Transformers model exhibits a notable reduction in inaccuracy. The model's forecast can be distributed among a network of automobiles. Thus, a comprehensive networking paradigm is put out in which Software Defined Networks (SDNs) serve as the backbone network and cars are connected to Road-side Units (RSUs). In contrast to SDN, which consistently delivers centrally programmable mechanisms, traditional design concepts, which combine DP, CP, and administration planes in a single network device, are unable to adapt to this level of data expansion, (BW)bandwidth, speed, security, and scalability. For traffic direction, the transformer model's trained parameters can be distributed across the network.

Wassie et al [17] proposed to reduce network congestion in advance, this project attempts to develop a traffic prediction model that can detect (EF) Elephant Flows. We are therefore driven to create models for predicting elephant flow and provide clear explanations of those models for use by network managers in the SDN network. The suggested model was created using H2O, Deep (AE)Autoencoder, and autoML prediction algorithms like XGBoost, GBM, and GDF. The efficiency of the prediction frameworks of EF yielded validation accuracy scores under construction errors, by utilizing XGBoost (XGB), GBM, and GDF algorithms. Additionally, Explainable Artificial Intelligence (AI) was used to provide a clear explanation of the models. Elephant flows can therefore be identified by paying close attention to packet size and byte size characteristics.

Nyaramneni et al [18] proposed to predict SDN traffic, we used and the 2 ML (Machine Learning) frameworks as Random Forest (RF) and XGB are compared this work. Our investigation shows that Random Forest trails behind the eXtreme Gradient Boosting technique, despite the fact that both algorithms have performed admirably in forecasting the network traffic. XGB outperformed RF, as evidenced by the fact that its root mean square error (RMSE) was lower. For traffic prediction in SDN networks, the eXtreme Gradient Boosting method works well since resource utilization matters when working with network controllers. Therefore, it is better to use eXtreme Gradient Boosting rather than RF.

Shukla et al [19] proposed to ensure the accurate anomaly identification is by the recursive network architecture utilized in this article to monitor traffic flow. The proposed technique improves the efficacy of cyberattacks in SDN. By minimizing network forwarding performance deterioration. In the case of Distributed DoS attacks, the suggested model provides superior (AD) Attack Detection efficiency. The proposed technique aims to educate users on matching traffic flows in a way that prevents overloading the SDN data plane while increasing granularity. Cyber-attack detection performance may be enhanced by applying a learned traffic flow matching control strategy, that provides the best traffic data to be collected throughout execution for identifying irregularities.

Yuqing et al. [20] suggested traffic prediction as a key method for performance monitoring and network planning in SDN. The proposed algorithm extracts traffic information from the DP using flow-based forwarding in SDN, resulting in higher levels of accuracy with less overhead.

In SDN, flow traffic is modeled using time-correlation theory, while network traffic is characterized using time-series analysis and regressive modelling.A novel traffic prediction method is introduced, specifically designed for SDN applications. Experimental validation demonstrates the viability and effectiveness of the suggested procedure in forecasting SDN traffic accurately.

Changqing et al [21] introduces Virtual Network Function Resource Prediction based on Heterogeneous Information Network (VNF-RPHIN), a novel method for predicting Virtual Network Function (VNF) resource requirements by extracting traffic features. It highlights the complexity of resource management in Network Function Virtualization (NFV) due to dynamic traffic demands and the importance of effective resource prediction methods. The approach combines Heterogeneous Information Network (HIN) construction, HIN2Vec feature representation, and an attention mechanism to assign weights to features before inputting them into a Multilayer Perceptron (MLP) model. The superior performance of VNF-RPHIN over traditional machine learning and common deep learning (DL) models in predicting VNF resource requirements accurately and it is demonstrated by the outcomes.

Chen et al [22] proposed an automated reinforcement learning (RL)-based load-balancing architecture (ALBRL) for SDN is proposed. In this architecture, it modifies the Deep Deterministic Policy Gradient (DDPG) method to establish a near-optimal routing between network hosts and build an optimisation model for load balancing in cases with large traffic loads. The suggested ALBRL enhances the random extraction technique of the empirical-playback mechanism in DDPG by updating the experience pool using the SumTree structure through sampling. With a higher chance of updating the network, it extracts a more relevant experience that can effectively increase the convergence rate. The test outcomes demonstrate that the suggested ALBRL considerably increases network throughput and has a quicker training speed than current RL methods.

Liu et al [23] suggested a Deep RL-based routing (DRL-R) is the idea put out in this work. Initially, it provides a technique that recombines many network resources using various metrics. Specifically, it recombines BW and cache by calculating their respective contribution scores to delay reduction. They also provide a routing strategy that utilises resource-recombined state. A DRL agent installed on a SDN controller continuously communicates with the network to adaptively execute acceptable routing based on the network state by distributing network resources for traffic as efficiently as possible. To construct the DRL-R, it employs the deep Q-network (DQN) and the DDPG. Lastly, it uses comprehensive simulations to show the efficacy of DRLR. With a global perspective on continuous learning, DRL-R overtakes OSPF in terms of resilience, throughput, load balancing, and flow completion time. DRL-R can also overtake other DRL-R systems since it optimizes network resources.

Eyobu and Edwinah [24] offers a DL-R Method in a WMN (wireless mesh network) that guarantees a specified optimal Quality of Service (QoS). A WMN simulation environment is constructed and a network data feature set is compiled in order to fulfil the study's objectives. This data set is then utilised to train a Long Short-Term Memory (LSTM)-DL model, which determines the route with the best (QoS) quality of service. To validate the generated dataset, several learning models such as MLP, Logistic Regression (LR), and RF are trained. Our findings demonstrate that the LSTM-based model's chosen routes offer the highest throughput and packet delivery ratio (PDR). In addition, our findings demonstrate that the learning models (MLP, LR, and RF) outperform the conventional Ad-hoc On-demand Distance Vector (AODV) Routing Protocol (RP) in terms of PDR and throughput.

Chen et al [25] provide a revolutionary model-free architecture known as the Spatiotemporal DPG(STDPG) agent for dynamic routing in SDN. The CNN-LSTM- (Temporal Attention Mechanism) TAM combination, which combines a convolutional neural network (CNN) and a LSTM with a TAM, provides the foundation for both the actor and critic networks. CNNLSTM-TAM facilitates improved learning from experience transitions for the STDPG agent by effectively using temporal and spatial data. In addition, to quicken the model training convergence, they use the Prioritised Experience Replay (PER) technique. The experimental findings demonstrate that STDPG may achieve strong convergence and automatically adjust to the current network environment. STDPG delivers superior routing solutions than some cutting-edge DRL agents, in terms of average (E2E) end-to-end latency.

By utilizing RL-training, the controller of SD Wireless Sensor Networks (SDWSNs) is optimized to enhance routing patterns. This method integrates SDN and RL, with RL being used to generate routing tables for the SDN controller. Moreover, four distinct incentive functions are suggested to boost network performance. RL-based SDWSN outperforms RL-based routing algorithms by 23% to 30% by lifespan. Its intelligent learning of routing paths at the controller makes RL- SDWSN highly effective. Additionally, it exhibits a rate of quicker convergence of the network related to RL- WSN.

Ferrazani Mattos and Duarte [27] proposed AuthFlow to access control and authentication system that uses host credentials. We also outline the primary security risks associated with SDN. The idea offers three primary contributions. Initially, the deployment of a host AM positioned directly above the MAC layer in an OpenFlow network, minimal overhead and precise access control was also assured. Then, The authentication system relies on credentials to match host credentials with the specific flows associated with every host.Finally, by incorporating the host identity as a new flow field, the latest control application framework enables SDN controllers to establish forwarding rules.

The POX controller was utilised to test a prototype of the suggested method. The findings demonstrate that AuthFlow blocks access to sites that have either revoked authorization or invalid credentials. Eventually, it demonstrates how our approach grants various degrees of network resource access to individual hosts based on their credentials.

The SD security architecture introduced by Miranda et al [28] integrates collaborative IPS with detection of anomalies. The primary focus of an IPS-based authentication process is to supply a lightweight DP IPS. To furnish a cost-effective ID solution near the DP, a cooperative anomaly detection system is employed. Additionally, a Smart Monitoring System (SMS) is implemented in the CP to correlate the true positive alarms triggered by the Sensor Nodes (SN) located at the network edge.The effectiveness of the suggested model is assessed in various security scenarios and contrasted with alternative approaches, demonstrating the model's high security and decreased false alert rate.

An energy-efficient, E2E security solution for SD Vehicle Networks (SDVN) was presented by Raja et al [29]. Green IIoT services are made possible in large part by SDN's flexible network administration, energy-efficient E2E security system, and network performance. Thus, lightweight E2E security is provided by the recommended SDVN. There are two stages at which the E2E security goal is addressed: Under the RSU-based Group Authentication (RGA) scheme, every vehicle within the RSU range is issued a group id-key pair to facilitate secure communication. ii) The Private-Collaborative IDS (p-CIDS) employs collaborative learning to identify potential intrusions within the VANET architecture, ensuring privacy through the combination of homomorphic encryption and differential privacy. Results from simulating the SDVN in NS2 and MATLAB demonstrate improved energy efficiency over current frameworks with less communication and storage overhead. Furthermore, the SDVN invader is detected by the p-CIDS with 96.81% accuracy.

Singh et al [30] A secure software-defined industrial network is intended to be provided via a (BC) blockchain framework based on deep learning. A voting-based consensus process is used in this model's BC Technology (BCT) to validate all switches after they have been registered, checked, and confirmed using zero-knowledge evidence. A deep Boltzmann machine-based flow analyzer is implemented at the CP to detect odd switch requests. With a mininet emulator, the evaluation is carried out, and outcomes indicate the effectiveness of the suggested system.

An SD-IoT-based architecture was suggested by Bhayo et al. [31] that provides IoT network security services. C-DAD (Counter-based DDoS AD) was created for efficiently detect DDoS attacks. This software is based on counter data from several network assessments. C-DAD has undergone extensive testing with a range of network features, making it a versatile and reliable programmable solution.Through SDN, the algorithm shows high performance with improved outcomes. Furthermore, the suggested framework effectively and quickly identifies the attack while using minimal CPU and memory resources.

Due to its low (EC) energy consumption and enhance the SDWSN's efficiency, the fuzzy topology (DP) Discovery Protocol (FTDP), a fuzzy logic-based technique that will lengthen the network's lifespan and it was suggested by Kipongo et al [32] suggest. The link layer DP (LLDP) is used in this proposal to construct the SDN controller architecture for intelligent/smart management. IT-SDN platform, an SDN-based WSN framework, was used in the development and execution of the full system simulation setup. The efficiency o the technique is validated and contrasted, demonstrating that energy efficiency may be attained with an efficient SDWSN discovery policy.

1. Enhancing the Quality of Service by optimizing internet traffic to efficiently utilize network resources.
2. Maximizing network resources for improved Quality of Service by managing internet traffic effectively.
3. Improving QoS through efficient utilization of network resources by directing internet traffic appropriately.

TABLE 1: COMPARISON TABLE FOR ROUTING AND SECURE SDN WITH EXISTING METHODS AND PROPOSED METHODS

Author	Methods	Merits	Demerits
Shuvro et al [2023]	Modified Transformer Architecture	An integrated approach to traffic control in a growing SDN-VANET environment.	It is necessary for enhancing the operation in real-life situations with dynamic limitations.

Wassie et al [2023]	Elephant flow prediction model	A traffic prediction model was designed for detecting the QoS.	Enhancements are required for the automated traffic prediction in validated SDN datasets.
Nyaramneni et al [2020]	Machine Learning Models	Enhancing the QoS by optimizing internet traffic to efficiently utilize network resources	Need to improve the resource utilization and automatic network traffic detection.
Shukla et al [2023]	Modified Recursive Learning	It tackles the traffic flow monitoring and achieve DDoS attack detection.	Enhance to improve the efficiency of cyber-attacks in SDN.
Yuqing et al [2020]	Flow-Based Forwarding Traffic Prediction Algorithm	It leveraging flow-based forwarding and time-correlation theory for accurate traffic forecasting.	For further real-world deployment testing to assess its scalability and performance in diverse network environments.
Changqing et al [2021]	Virtual Network Function Resource Prediction	It predicting VNF resource requirements based on traffic feature extraction, showing superior performance over traditional and common deep learning models.	Need to enhance the potential limitations in network traffic.
Eyobu and Edwinah [2023]	Deep Learning-Based Routing Method	The highest PDR and throughput was provided by the route of the LSTM-based model.	Multi-hop data must be used to identify the optimal hope size for optimal network efficiency.
Chen et al [2020]	Spatiotemporal Deterministic Policy Gradient (STDPG)	In terms of average E2E delay, STDPG outperforms other routing systems .	It will improve the centralized decision-making problem.
Younus et al [2020]	(SDWSNs)	When compared to RL-WSN, the faster convergence of network was offerred	The increasing the ability of the SDN controller that learns from the network background actual time

Ferrazani Mattos and Duarte [2016]	Auth Flow	Hosts with revoked authorization or those without valid credentials are denied access by AuthFlow.	By enabling the use of signed certificates as access credentials, it aims to expand AuthFlow to include additional authentication techniques like EAP-TLS.
Miranda et al [2020]	Software-Defined Security Architecture	It offers great security, minimal computing complexity, and a significant reduction of false alarms in SDWSNs.	It will investigate DL strategies for properly classifying and detecting unidentified anomalies in SDWSN systems.
Raja et al [2020]	Energy-Efficient E2E Security Solution	It reduces the network overhead and increases security by preventing several attacks.	Need to improve the differential privacy and homomorphic encryption schemes.
Singh et al [2020]	DL-Based BC Model	The findings indicate that it effectively managed to balance the cost of communication and the time required for computation.	Encryption, decryption, verification, validation, and block preparation time will be improved when the quantity of transactions increases.
Bhayo et al [2020]	C-DAD	Less utilization of CPU and memory resources, the model effectively detects the attacks in less amount of time.	Need to working with multiple IoT networks in heterogeneous backgrounds.
Kipongo et al [2020]	FTDP	Achieving energy efficiency is possible through an effective SDWSN discovery policy.	Need to improve the cyber physical systems like small grids.

INFERENCES

`The paper addresses significant issues in applying DRL to SDN for routing optimization, specifically focusing on exploration problems during the learning process, long convergence times, and traffic demand variability. Performance degradation caused by the exploration phase is mitigated by using a modeled network for offline training, protecting the actual network from suboptimal decisions. Convergence time is reduced by this offline learning approach, allowing the DRL agent to rapidly reach an optimal state without waiting for real-time feedback. Additionally, training the DRL model with synthetic traffic demands enhances its robustness and adaptability, enabling it to perform effectively under varying traffic conditions without frequent retraining. These strategies collectively improve the feasibility and efficiency of deploying DRL-based routing systems in real-world SDN environments.

CONCEPT OF SDN

The Open Networking Foundation (ONF) is a user-driven organisation whose mission is to produce open standards in order to promote and accelerate the adoption of SDN [33]. The OpenFlow standard protocol and the notion of SDN were developed in a 2012 ONF white paper. It outlined the three objectives of SDN: 1) show off the versatility of SDN structure and its capacity to foster creativity; 2) allow for extensive experimentation with campus production networks; and 3) allow for numerous concurrent experiments utilising virtualization and slicing on the same physical SDN structure.

SDN structure

The control plane in SDN is programmable and isolated from the data plane [34]. Figure 1 depicts the fundamental structure of SDN. There are 3 layers in SDN. The infrastructure, which houses every network device, is the bottom layer. Devices in SDN networks lack control functions, in contrast to those in conventional networks. They function as basic, unintelligent packet forwarders. A new, unified control layer is created by abstracting their control functions. Through the control data plane interface, also referred to as the south-bound interface, the infrastructure layer communicates with the control layer.

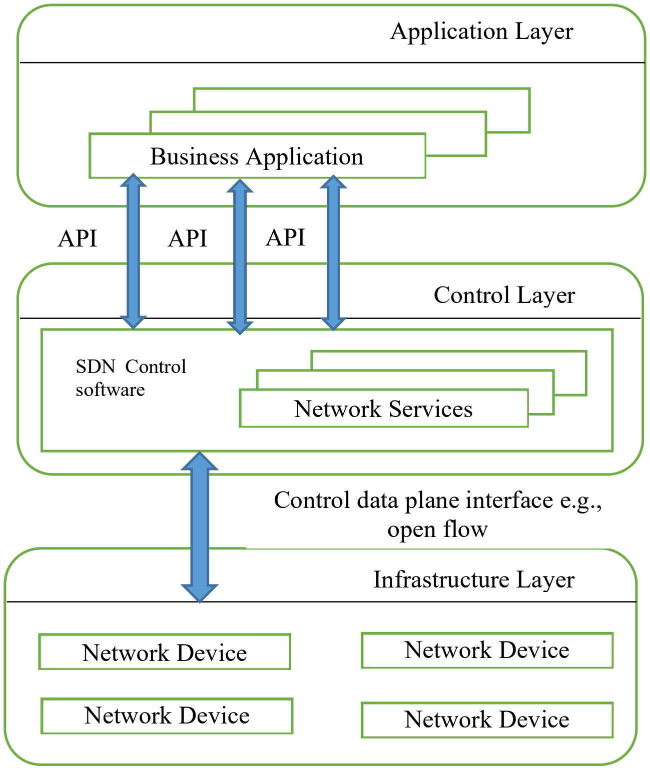

Figure 1. Basic Structure of SDN

It makes sense for all network intelligence to be centralised in the control layer, which is made up of numerous SDN controllers. SDN controllers oversee the underlying virtual and physical networks and use APIs, sometimes referred to as north-bound interfaces, to deliver services to higher layer applications. Device abstraction is a feature of the control layer that allows programmes to conceal device-specific information.

Network operators and application developers manage routing, access control, BW, traffic engineering, quality of service, processor and storage optimisation, EC, and other business needs through programmable interfaces in the application layer of the network. This removes the requirement for manual configurations, which are prone to errors.

OPENFLOW

A network switch or router's data plane can be accessed through the OpenFlow network communications protocol. With an internal flow-table and a standardised interface for adding and removing flow entries, it runs on Ethernet switches [35]. An OpenFlow Switch(OFS) simplified as shown in Figure 2.

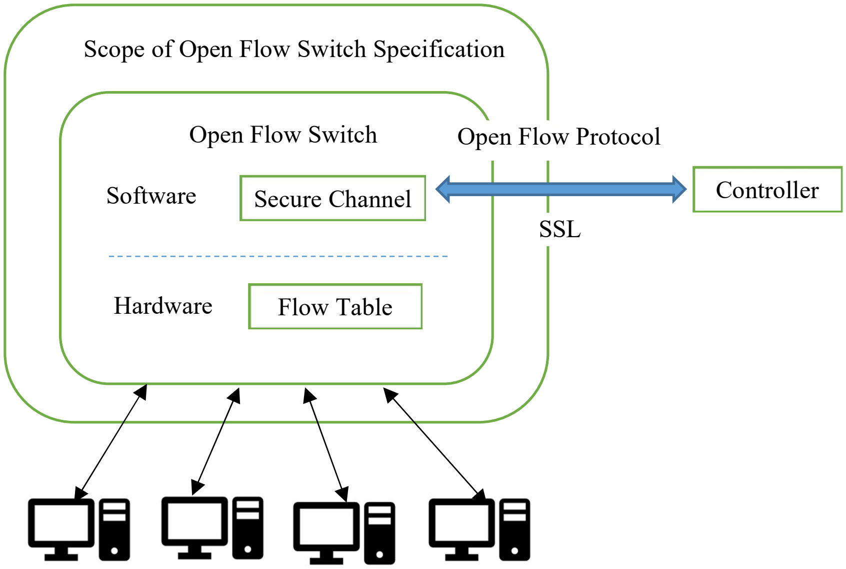

Figure 2. Open Flow Switch Design

Three features are supported by an OFS: the OpenFlow protocol, a secure channel, and flow table operations. The OpenFlow flow table is made up of user-defined or pre-established rules that OpenFlow uses to match and process network packets. The header field, action, and stats are the three components that make up each flow table entry as opposed to the typical IP quintuple routing entry.

After matching packets based on their header fields, the action (also known as instructions) in the flow entry determines how each packet should be handled. The network status is shown by statistics, which also include priority, counters, timeouts, cookies, and other variables. Flow control can be implemented at different granularities by network operators because any information in the header can be used for pattern matching. For instance, a wildcard can be used for all fields except than the destination IP field if the operator wants to alter packets with a certain IP address. To accommodate multiple OpenFlow features, the flow table's size could increase rapidly. The new OpenFlow standard uses OpenFlow pipeline technology, which takes its cue from the multi-level page table concept in memory management to conserve storage space. The process of matching packets against several tables in the pipeline is shown in Figure 3.

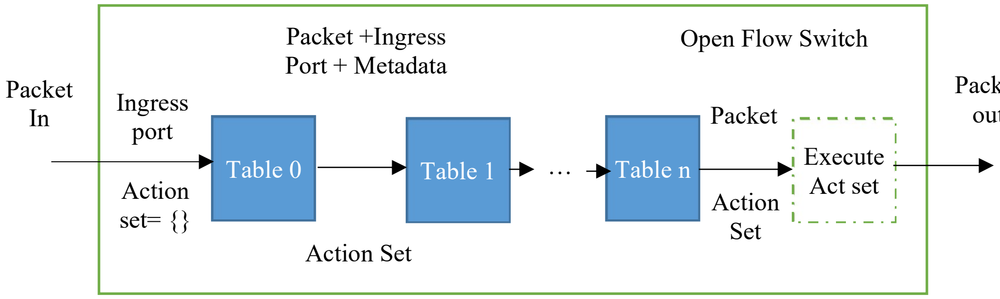

Figure 3. Packets are matched against multiple tables in the pipeline

Secure Channel:

Controllers and switches are connected via a secure channel. Every secure channel needs to be OpenFlow compliant [36]. Switches that forward packets and receive event notifications can be configured and managed by controllers.

OpenFlow Protocol:

Three different message kinds are supported by the OpenFlow protocol: symmetric, asynchronous, and controller-to-switch communications. There are several sub messages within each message [37]. The controller manages or queries switches by initiating controller-to-switch messages. The switch starts asynchronous messages to notify the controller of network events or changes in switch status. The switch or controller can start symmetric communications, such as echo requests, echo answers, and keep-alive messages, to help measure bandwidth, monitor latency, and confirm network availability between the controller and switch.

Version 1.4 of the OpenFlow standard has been updated by ONF. Version 1.0 of the protocol contained unchangeable header fields for flow table matching in packets, which limited flexibility. Version 1.0 is limited to forwarding LAN packets. Support for numerous flow tables, labelling, and tunnels (such as multi-path stream control transmission protocol and multi-protocol label switching, or MPLS) was added in version 1.1. IPv6 functionality was added in version 1.2, and the header field layout was changed to allow for flexible matching. It began to facilitate general message processing functions and instruction sets. Better flow action definitions and enhanced IPv6 extension header field support was introduced in version 1.3.2013 saw the introduction of version 1.4, which included bundle messages and a new flow table synchronisation technique.

When a packet reaches a switch, the switch checks to see if the header fields of the packet match any flow entries (referred to as rules throughout the remainder of the paper) in the flow table. In that case, the matching rule will determine how the packet is transmitted. If not, the switch sends the controller an asynchronous message. Based on pre-programmed policies, the controller forwards the message as an event to the relevant control application or applications. After processing the event, the applications reply with a message and any appropriate actions. Figure 4 shows how an OpenFlow switch handles packet forwarding in its entirety.

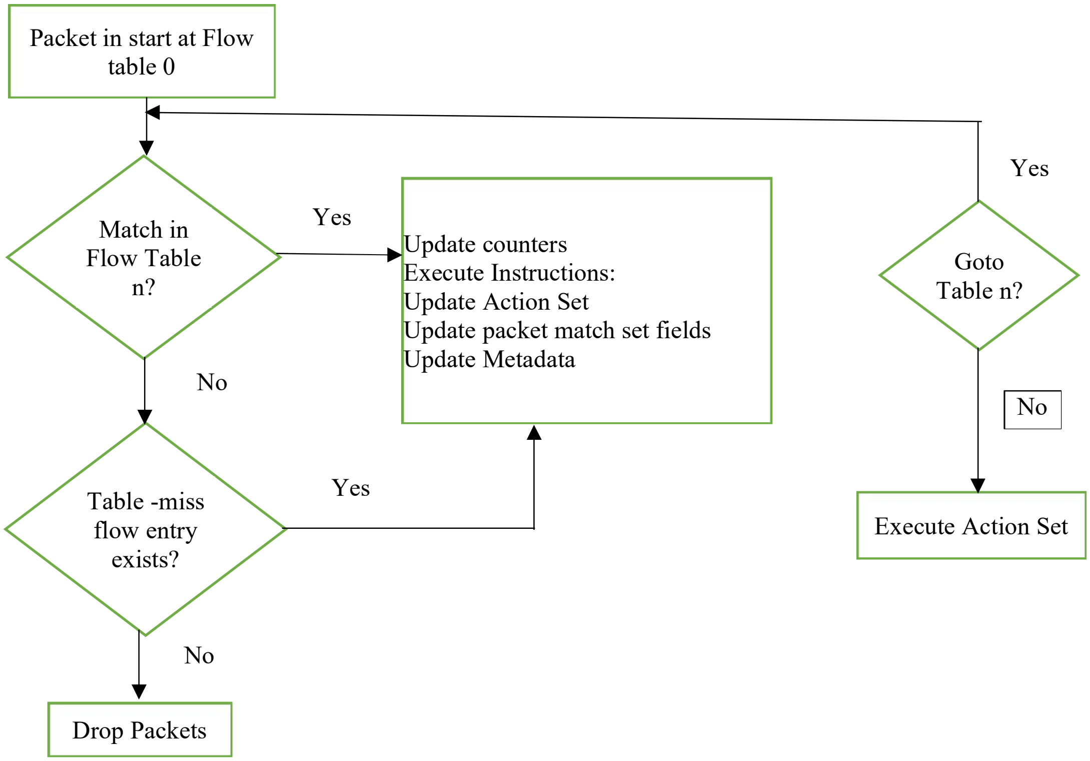

Figure 4. Detail of packet flow through an OpenFlow switch

Dedicated OpenFlow switches, also known as "type-0" switches, and OpenFlow-enabled switches are the two types of OpenFlow switches. A "type-0" switch is a basic switch device that forwards packets in accordance with rules that are established remotely. A conventional switch with the addition of a flow table, security channel, and OpenFlow protocol capability is called an OpenFlow-enabled switch.

SDN APPLICATION IN VARIOUS FIELD

The Internet of Things:

SDN and IOT coming together creates fascinating new platforms. SDN is capable of handling the massive amounts of data coming from the device that is being connected to the Internet of Things in an elegant manner and distributing traffic [38]. To manage the data coming from the network, SDN employs segmentation. It separates the entire Internet of Things network into manageable chunks, each of which may be managed by a separate controller to ensure optimal network performance. Because SDN-linked IOT networks provide a global network view, they are superior for security concerns.

Additional Networking Hardware:

By modifying the centralised approach, they put out an architecture based on the SDN concept to address the fragmentation issues with the home network. It suggested a brand-new kind of gadget that implements home networking devices according to the inclination for multimedia applications. This suggested technology offers a high degree of control and configuration flexibility, assisting users in relying less on manual configuration by many users and more on software applications.

Cloud-Based Computing:

The network has leaked an excessive amount of data. It gives rise to the idea of clouds; clouds are created by networks. Figure 5 represents the applications may need to be modified before being stored in the cloud due to the significant amount of space required for their storage. SDN enables it with the aid of a centralised controller, which is set up using protocols pertaining to software.

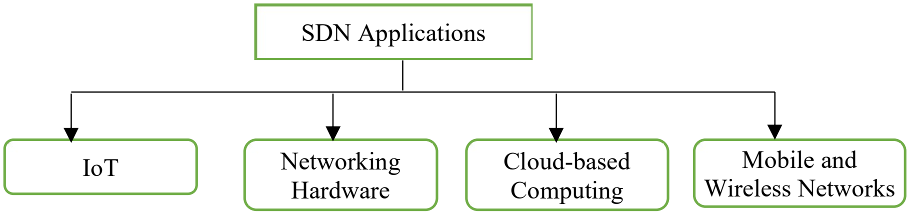

Figure 5. SDN controllers in various applications

Mobile and Wireless Networks:

SDWN is the term used to describe the contribution of SDN to wireless networks. The Open-flow suggested by the whitepaper has been researched by the researchers. Open-flow is based on an Ethernet switch and consists of a flow table that can have flow entries added or removed. By integrating Open-road, SDN deployment in wireless networks enables smooth transition across various wireless technologies. SDN facilitates the flow-centric paradigm, which by adding functionalities and making it programmable at higher layers, helps to solve the node migration problem10. The SDN architecture uses open-flow test-beds like WiMAX and WiFi and is shareable and open across various service providers.

TOOLS FOR SDN SIMULATION:

Numerous SDN simulators and emulators are available, such as fs-sdn, NS-3, EstiNet, and Mininet. Mininet is the most widely used and well-researched platform among them. A mininet can simulate many network topologies, including hosts, layer 2 switches, and layer 3 routers. Using Mininet, the researchers developed and optimised an SDN network [39]. The OpenFlow device and SDN network emulator is open source software. A mininet merely shows the network virtually rather than physically replacing the switches. A single workstation can be used to simulate the whole network with mininet.

Some of the popular SDN simulation tools, including W3, FatTire, and Fs-sdn, are described in the table. W3 was added to the SDN environment in order to troubleshoot problems. The Mininet tool's performance is being examined in order to emulate SDN. Numerous factors, such as altering the topology, adding more nodes, and regulating the behaviour of switches, are taken into account when analysing the emulation tool, and it is determined that the simulation environment is crucial to Mininet's performance. When accessible, Mininet is seen to require more RAM for the same topology. Tools for SDN testing and debugging have been examined tabularly, together with their benefits, drawbacks, supported versions, and license. The techniques listed above can also be used to identify security flaws such as data integrity or configuration attacks.

ADVANTAGE AND DISADVANTAGE

The controller, a centralised server, is where all control logic is located in the SDN architecture. By using an API, the controller controls the network, simplifying the process of configuring new features.

1.The centralised approach in SDN lowers equipment costs by doing away with the need to support several data plane standards and protocols.

2. It has been demonstrated that using energy-efficient algorithms may reduce energy consumption and increase the scalability of SDN networks.

3. High frequency requests are being delivered to the controller, resulting in high controller load, in order to achieve high link utilisation.

4. By dividing traffic among the available pathways, the SDN design skillfully balances the load and enables quick responses to a high number of data flows. When distributed systems are fully utilised, deployment costs are lowered.

5. Because SDN's flow table size is constrained, the switch becomes overwhelmed when a lot of requests are made. Packet loss or switch failure are the outcomes of this.It has been noted that using idle devices and redundant links wastes a significant amount of power and energy. By using an intelligent technique, SDN enables carbon footprint reduction.

6. To enhance control plane performance and lessen negative effects on the network, the controller's software must be updated on a regular basis.

7. SDN responds to scenarios with dynamic traffic more quickly. It offers improved load balancing and dynamic provisioning.

8. The Mininet simulation programme consumes a lot of RAM; in SDN, RAM utilisation is often quite low, but it also makes use of as much RAM as possible for the same topology.

CONCLUSION

This report summarises the state of the art in SDN security research and makes predictions for future directions. SDN provides more control over the network because of the presence of a central component, the controller. Network-connected hosts may be instantly identified and controlled. Network traffic can be accurately routed by any route, classified by any criteria, and new features may be deployed with the right authentication procedures in place. In the end, the network may be tailored to meet the needs. SDN gives you control over the network and eliminates the need for several protocols and their drawbacks. The fact that SDN is now implemented demonstrates that researchers have acknowledged the need for safe SDN solutions and have responded by offering frameworks or system checking solutions. Other implementations exist that served a similar function but were not covered in this survey. Some of these authentication problems can be resolved by the research presented here, but more work has to be done to adequately address these flaws.

REFERENCE

1. Mohammed, A.H., Khaleefah, R.M. and Abdulateef, I.A., 2020, A review software defined networking for internet of things. In *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1-8.

2. Javed, F., Afzal, M.K., Sharif, M. and Kim, B.S., 2018, Internet of Things (IoT) operating systems support, networking technologies, applications, and challenges: A comparative review. *IEEE Communications Surveys & Tutorials*, vol.*20*, no.3, pp.2062-2100.
3. Dwivedi, A.K., Dwivedi, M. and Kumar, M., 2023, Advances in Network Security: A Comprehensive Analysis of Measures, Threats, And Future Research Directions.
4. Fuentes-García, M., Camacho, J. and Maciá-Fernández, G., 2021, Present and future of network security monitoring. *IEEE Access*, vol.*9*, pp.112744-112760.
5. El Moussaid, N., Toumanari, A. and El Azhari, M., 2017, Security analysis as software-defined security for SDN environment. In *2017 Fourth International Conference on Software Defined Systems (SDS)*, pp. 87-92.
6. Farooq, M.S., Riaz, S. and Alvi, A., 2023, Security and Privacy Issues in Software-Defined Networking (SDN): A Systematic Literature Review. *Electronics*, vol.*12*, no.14, p.3077.
7. Shin, S., Xu, L., Hong, S. and Gu, G., 2016, Enhancing network security through software defined networking (SDN). In *2016 25th international conference on computer communication and networks (ICCCN)*, pp. 1-9.
8. Aziz, N.A., Mantoro, T. and Khairudin, M.A., 2018, Software defined networking (SDN) and its security issues. In *2018 International conference on computing, engineering, and design (ICCED)*, pp. 40-45. IEEE.
9. Ajaeiya, G.A., Adalian, N., Elhajj, I.H., Kayssi, A. and Chehab, A., 2017, Flow-based intrusion detection system for SDN. In *2017 IEEE Symposium on Computers and Communications (ISCC)*, pp. 787-793.
10. Mazhar, N., Salleh, R., Hossain, M.A. and Zeeshan, M., 2020, SDN based intrusion detection and prevention systems using manufacturer usage description: A survey. *International Journal of Advanced Computer Science and Applications*, vol.*11*, no.12.
11. Shu, Z., Wan, J., Li, D., Lin, J., Vasilakos, A.V. and Imran, M., 2016, Security in software-defined networking: Threats and countermeasures. *Mobile Networks and Applications*, vol.*21*, pp.764-776.
12. Iqbal, M., Iqbal, F., Mohsin, F., Rizwan, M. and Ahmad, F., 2019, Security issues in software defined networking (SDN): risks, challenges and potential solutions. *International Journal of Advanced Computer Science and Applications*, vol.*10*, no.10, pp.298-303.
13. Rouka, E., Birkinshaw, C. and Vassilakis, V.G., 2020, SDN-based malware detection and mitigation: The case of ExPetr ransomware. In *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, pp. 150-155.
14. Sharma, P.K. and Tyagi, S.S., 2019, Improving security through software defined networking (SDN): An SDN based model. *IJRTE*, vol.*8*, pp.295-300.
15. Kujur, P., Biswal, S.P. and Patel, S., 2022, Security Challenges and Analysis for SDN-Based Networks. *Software Defined Networks: Architecture and Applications*, pp.321-346.

16. Shuvro, A.A., Khan, M.S., Rahman, M., Hussain, F., Moniruzzaman, M. and Hossen, M.S., 2023. Transformer Based Traffic Flow Forecasting in SDN-VANET. IEEE Access.
17. Wassie, G., Ding, J. and Wondie, Y., 2023. Traffic prediction in SDN for explainable QoS using deep learning approach. Scientific Reports, 13(1), p.20607.
18. Nyaramneni, S., Saifulla, M.A. and Mehra, S.S., 2020. Internet traffic prediction in sdn using rf and xgb. In Advances in Computational and Bio-Engineering: Proceeding of the International Conference on Computational and Bio Engineering, 2019, Volume 2 (pp. 153-159). Springer International Publishing.
19. Shukla, P.K., Maheshwary, P., Subramanian, E.K., Shilpa, V.J. and Varma, P.R.K., 2023. Traffic flow monitoring in software-defined network using modified recursive learning. Physical Communication, 57, p.101997.
20. Bhatia, J., Dave, R., Bhayani, H., Tanwar, S. and Nayyar, A., 2020. SDN-based real-time urban traffic analysis in VANET environment. Computer Communications, 149, pp.162-175.
21. Yuqing, Wang., Dingde, Jiang., Liuwei, Huo., Yong, Zhao. (2021). A New Traffic Prediction Algorithm to Software Defined Networking. Mobile Networks and Applications, Available from: 10.1007/S11036-019-01423-3.
22. Chen, J., Wang, Y., Ou, J., Fan, C., Lu, X., Liao, C., Huang, X. and Zhang, H., 2022, Albrl: Automatic load-balancing architecture based on reinforcement learning in software-defined networking. *Wireless Communications and Mobile Computing*, *2022*, pp.1-17.
23. Liu, W.X., Cai, J., Chen, Q.C. and Wang, Y., 2021, DRL-R: Deep reinforcement learning approach for intelligent routing in software-defined data-center networks. *Journal of Network and Computer Applications*, vol.*177*, p.102865.
24. Eyobu, O.S. and Edwinah, K., 2023, A Deep Learning-Based Routing Approach for Wireless Mesh Backbone Networks. *IEEE Access*.
25. Chen, J., Xiao, Z., Xing, H., Dai, P., Luo, S. and Iqbal, M.A., 2020, STDPG: A spatio-temporal deterministic policy gradient agent for dynamic routing in SDN. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1-6.
26. Younus, M.U., Khan, M.K., Anjum, M.R., Afridi, S., Arain, Z.A. and Jamali, A.A., 2020, Optimizing the lifetime of software defined wireless sensor network via reinforcement learning. *ieee access*, vol.*9*, pp.259-272.
27. Ferrazani Mattos, D.M. and Duarte, O.C.M.B., 2016, AuthFlow: authentication and access control mechanism for software defined networking. *annals of telecommunications*, vol.*71*, pp.607-615.
28. Miranda, C., Kaddoum, G., Bou-Harb, E., Garg, S. and Kaur, K., 2020, A collaborative security framework for software-defined wireless sensor networks. *IEEE Transactions on Information Forensics and Security*, vol.*15*, pp.2602-2615.
29. Raja, G., Anbalagan, S., Vijayaraghavan, G., Dhanasekaran, P., Al-Otaibi, Y.D. and Bashir, A.K., 2020, Energy-efficient end-to-end security for software-defined vehicular networks. *IEEE Transactions on Industrial Informatics*, vol.*17*, no.8, pp.5730-5737.

30. Singh, M., Aujla, G.S., Singh, A., Kumar, N. and Garg, S., 2020, Deep-learning-based blockchain framework for secure software-defined industrial networks. *IEEE Transactions on Industrial Informatics*, vol.*17*, no.1, pp.606-616.
31. Bhayo, J., Hameed, S. and Shah, S.A., 2020, An efficient counter-based DDoS attack detection framework leveraging software defined IoT (SD-IoT). *IEEE Access*, vol.*8*, pp.221612-221631.
32. Kipongo, J., Esenegho, E. and Swart, T., 2020, Efficient topology discovery protocol for software defined wireless sensor network. *Int J Electr Comput Eng (IJECE)*, vol.*9*, no.4.
33. Ong, L., 2017, ONF SDN architecture and standards for transport networks: Control architecture and network modeling I M2H. 1. In *2017 Optical Fiber Communications Conference and Exhibition (OFC)*, pp. 1-41.
34. Kaliyamurthy, N.M., Taterh, S., Shanmugasundaram, S., Saxena, A., Cheikhrouhou, O. and Ben Elhadj, H., 2021, Software-defined networking: An evolving network architecture—programmability and security perspective. *Security and Communication Networks*, *2021*, pp.1-7.
35. Hui, Z., Zhang, S., Shao, Y. and Chen, T., 2021, The Overview of SDN Architecture and its Practical Application with Improving Methods. *The Frontiers of Society, Science and Technology*, vol.*3,* no.1.
36. Yigit, B., Gur, G., Tellenbach, B. and Alagoz, F., 2019, Secured communication channels in software-defined networks. *IEEE Communications Magazine*, vol.*57*, no.10, pp.63-69.
37. Aliyu, A.L., Bull, P. and Abdallah, A., 2017, Performance implication and analysis of the OpenFlow SDN protocol. In *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 391-396.
38. Li, T., Chen, J. and Fu, H., 2019, Application scenarios based on SDN: an overview. In *Journal of Physics: Conference Series*, vol. 1187, no. 5, p. 052067.
39. Buzura, S., Peculea, A., Iancu, B., Cebuc, E., Dadarlat, V. and Kovacs, R., 2023, A hybrid software and hardware SDN simulation testbed. *Sensors*, vol.*23*, no.1, p.490.