YUMSUK
JOURNAL OF PURE AND APPLIED SCIENCES

A SYSTEMATIC LITERATURE REVIEW ON SOFTWARE-DEFINED NETWORKING CHALLENGES

Umar Abdullahi Umar[1*] and Aminu Malam Ya'u[2]
[1,2]Department of Computer Science, Sule Lamido University, Kafin Hausa, Nigeria
[*]Corresponding author e-mail: umar.abdullahi@slu.edu.ng[1]

Abstract:

Software-defined networking is a new networking innovation which provide programmable and centralized network control by separating control plane from data plane. This systematic literature review summarizes Software-defined networking challenges, applications and key development. The review covers security issues faced by software define networking and challenges faced by software defined networking. The problems faced by Software-defined networking that led to conduct this literature review can be one the following: Centralized controller vulnerabilities, insecure communication channels, flow table manipulation and switch vulnerabilities. While Software-defined networking offers benefits like flexibility, automation, and easier network management, it faces issues like lack of standardization, new security risks from centralized control, performance concerns due to increased abstraction, and complexity in large-scale deployments. The review highlights controller scalability, optimal placement, and coordination as critical for large networks. Other key challenges include guaranteeing Quality of Service, failover mechanisms, incremental deployment for legacy network transition, and the need for advanced telemetry, machine learning capabilities, and organizational change management. Ongoing research focused on these multifaceted issues shows promise in developing holistic solutions through innovations in controller architectures, distributed state management, security mechanisms, virtualization techniques, and hardware accelerators. As standards emerge and adoption grows, Software-defined networking has immense potential to become the basis for dynamic and intelligent networks of the future.

Keywords: Software-Defined Networking, Security, and Controller

INTRODUCTION

Software-defined networking (SDN) is an innovative network structure that incorporates a centralized control mechanism. SDN has demonstrated its effectiveness in enhancing both network performance and security (Tung *et al.*, 2020). The latest progress in cloud-based information and communications services offers a solution to address the limitations of scalability and complicated maintenance found in traditional networks. Software Defined Networking (SDN) has emerged as a promising approach to overcome these limitations while providing adaptable network management. In particular, SDN separates the control plane from the data plane, resulting in the abstraction of lower-level functions. This enables more effective network management and utilization(Abuarqoub, 2020). Software-defined networking (SDN) is a networking technology that separates the control plane and the data plane. This separation enables network applications to have access to a global network topology and the ability to customize packet forwarding rules. SDN has numerous innovative applications in various networking domains, including 5G, Internet of Things (IoT), and data center networks. However, the programming model used in SDN, specifically the match-action programming model represented by OpenFlow/Protocol Oblivious Forwarding (POF), has limitations. The OpenFlow/POF programming model can only process certain types of data, such as packets and metadata. This restricts its ability to support future network applications that require more complex data types and processing capabilities. Therefore, there is a need for more

advanced programming models and protocols that can provide greater flexibility and support for emerging network applications(Jing *et al.*, 2021). The concept of Software-Defined Networking (SDN) has brought significant transformations to the conventional network model by separating network operations from physical hardware and promoting logically centralized network control. This approach enhances network programmability and strengthens security by enabling a comprehensive overview of the entire network and efficient management by a centralized controller. Consequently, SDN facilitates more effective monitoring of network traffic and identification of vulnerabilities. Furthermore, it simplifies the rapid deployment of new services with increased flexibility (Goud & Gidituri, 2022).

Software-defined networking (SDN) has become a versatile network structure that enables centralized and programmable control. While SDN has the potential to enhance network security supervision and policy implementation, the continuous challenge for experts lies in safeguarding SDN from advanced attacks. Current network forensics tools strive to identify and trace such attacks, but effectively connecting the cause and effect across both the control and data planes remains a complex task (Ujcich *et al.*, 2021). Software-Defined Networking (SDN) is an up-to-date approach that offers a foundation for deploying dependable, centrally controlled, and automated security solutions in both traditional and emerging networks, including IoT, cloud computing, 5G/6G mobile communication networks, and vehicular communications. In these intricate systems, relying solely on manual security operations can cause delays or hindrances in identifying, mitigating, and preventing the constantly evolving sophisticated threats (Yungaicela Naula *et al.*, 2022).

The network session limitations faced by Industrial Internet of Things (IIoT) applications are distinct and demanding. These constraints require a significant degree of adaptability, allowing the system to evaluate the impact of an event and make necessary network adjustments. In contrast to conventional networks, Software Defined Networking (SDN) separates the control and data planes to facilitate programmable network configuration. This approach is well-suited for accommodating the specific needs of smart cities, which have a significant impact on the system. However, reliability remains a challenge in implementing SDN for this purpose (Babbar *et al.*, 2021). There is a significant focus on analyzing and enhancing network traffic processes in the fields of network management and multimedia data mining. In the context of Software Defined Networks (SDN), security has emerged as the most challenging aspect, relying on a centralized and programmable controller. As a result, monitoring network traffic plays a crucial role in detecting and exposing intrusion anomalies within the SDN environment (Alshammri *et al.*, 2022).

Software-Defined Networking (SDN) is a modern networking architecture that has emerged as a promising approach for managing large-scale and complex networks. It allows network administrators to manage and control network traffic through a centralized controller, which separates the control plane from the data plane. One of the most widely used SDN protocols is the OpenFlow protocol, which enables communication between the controller and the network devices(Khalil Al Dulaimi *et al.*, 2018).However, as SDN continues to gain popularity, concerns have been raised about the security vulnerabilities of the OpenFlow protocol. Despite all the efforts to develop security mechanisms for OpenFlow, many potential vulnerabilities still exist that could be exploited by attackers to compromise the security and integrity of the network(Onyema *et al.*, 2022).

An efficient and reliable intrusion detection system (IDS) is essential for protecting computer networks against cyber-attacks. In recent years, Software-Defined Networking (SDN) has emerged as a promising approach to network management, offering centralized control and programmability of network functions. SDN enables flexible and dynamic network configurations that can improve the efficiency and security of network operations. However, designing an efficient IDS framework in SDN faces several challenges, including the need for real-time detection and response to attacks, the complexity of SDN architectures, and the need to balance security requirements with network performance. Therefore, researchers and practitioners have explored various techniques and

strategies to develop efficient IDS frameworks for SDN-based networks (Alshammri *et al.*, 2022). Network security is a critical concern in the modern era of interconnected systems and pervasive connectivity. Traditional network architectures face limitations in adapting to dynamic security threats and evolving network requirements. To overcome these limitations, Software Defined Networking (SDN) has emerged as a promising paradigm for enhancing network security (Ajiya Ahmad *et al.*, 2021).

In figure 1 below, the key principle of SDN is the separation of the data plane and control plane in networking devices. This differs from traditional networking where both planes are bundled together on devices like switches and routers. The SDN architecture consists of the following layers: Application Layer: This is the topmost layer which includes SDN applications that communicate their network requirements and desired policies down to the lower layers. Example SDN apps include traffic engineering, load balancing, firewalls, intrusion detection etc. Control Layer: This layer has the SDN controller which maintains a global view of the overall network. The controller manages flow control and routing decisions for the network based on the applications requirements. It uses southbound APIs to communicate with the infrastructure layer below. Popular SDN controller software includes Open Daylight, ONOS, Floodlight etc. Infrastructure Layer: This bottom layer consists of the physical SDN-enabled networking devices like switches, routers, gateways. The control plane is removed from these devices, and they provide a simple forwarding hardware that can be programmed via an open interface. OpenFlow is the most common protocol used between the controller and data plane. Management Layer: This layer provides the operations and business support systems needed to manage, orchestrate and monitor the physical and virtual network resources. Data Plane: The data plane in the physical networking devices handles all the forwarding/switching of data packets, guided by the decisions provided by the control plane. This allows for rapid forwarding in the data plane using ASICs/hardware, while the control logic is implemented in software in the control layer. The key benefits of this architecture include logical centralization of control, abstraction of underlying network infrastructure, programmability of network behavior, and separation of forwarding hardware from control logic. The controller has a holistic view of the network which simplifies automation, visualization and management. Adding new features is easier via software programs rather than configure changes.

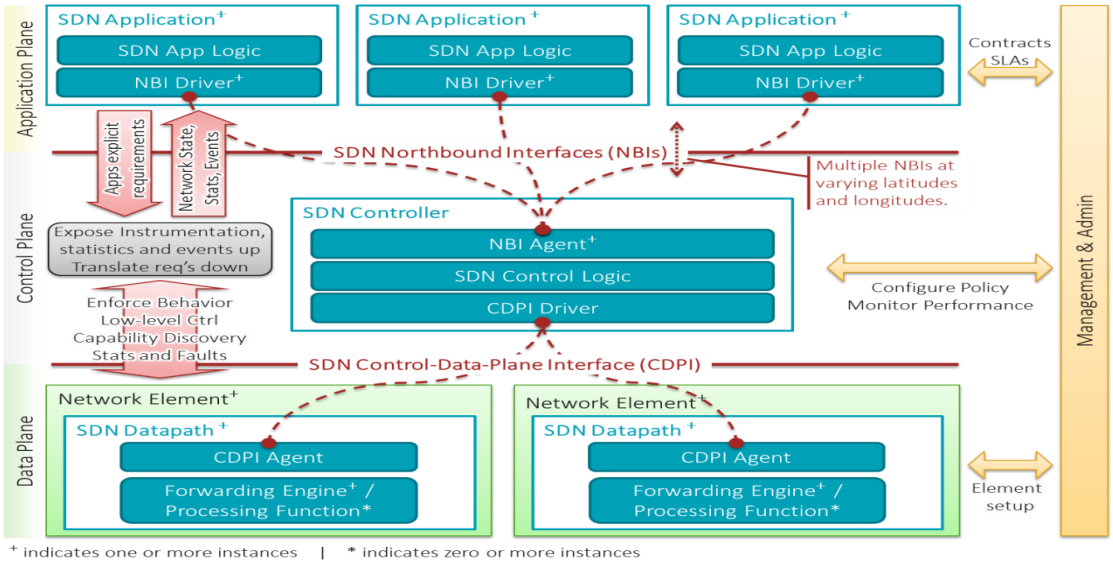

Figure 1: Architecture of SDN(Wikipedia contributors, 2023)

2. Literature Review: Software Defined Networking

SDN is a novel design that simplifies and accelerates the management of extensive networks by separating the control plane and the data plane. This entails relocating the control logic from the hardware level of the network to a centralized management level. To manage the network effectively, the OpenFlow Discovery Protocol (OFDP) is frequently employed to identify the network topology in the data plane and transmit it to the control plane. However, OFDP has performance drawbacks, such as excessive messaging between the control and data planes, which overloads the SDN-Controller. Because the accuracy of network topology information in the control plane is crucial for the application layer, it is necessary to obtain precise data plane network topology information. Consequently, we will focus on performance issues after presenting an overview of topology discovery protocols (Wazirali & Ahmad, 2021).

Software defined networking (SDN) is an emerging technology in the field of computer networking. It involves the implementation of a centralized program known as the SDN Controller, which provides a unified control mechanism for the entire network. By introducing this central point of control, SDN offers an effective solution to improve the performance of Internet of Things (IoT) networks and overcome the limitations that currently exist in such networks. With SDN, the network can be managed and optimized as a whole, rather than requiring each device to be individually configured. This enables efficient use of network resources and facilitates the implementation of advanced network management strategies, such as traffic prioritization and load balancing. Overall, SDN has the potential to enhance the scalability, security, and reliability of IoT networks, making it a promising technology for the future of computer networking (Hayajneh *et al.*, 2020).

The popularity of the Internet has created a demand for computer networks that are agile and flexible. However, traditional networking systems are not capable of meeting the current computing needs. The use of manually-configured proprietary devices lead to error-prone situations, and they are unable to fully utilize the potential of the physical network infrastructure. This has resulted in a shift in the networking industry towards Software Defined Networking (SDN). The use of an SDN platform offers several advantages, such as programmability, task virtualization, and easy network management. The POX Controller is an open-source OpenFlow SDN Controller based on Python. It is primarily used for fast development and prototyping of new network applications and comes pre-installed with the Mininet virtual machine. The POX Controller can transform dumb OpenFlow devices into hub, switch, load balancer, and firewall devices(Ngo *et al.*, 2020).

The rise of the Internet of Things (IoT) has led to a surge in demand for embedded devices that enable sensors and actuators to interact autonomously while offering smart services. However, these IoT devices are limited in terms of computation, storage, and network capacity, which makes them vulnerable to hacking and compromise. To address this issue, it is necessary to develop scalable security solutions that are optimized for the IoT ecosystem. One promising paradigm that could aid in achieving secure development of IoT is Software Defined Networking (SDN). SDN is an essential component of the fifth generation of mobile systems (5G) and has the potential to detect and mitigate Denial of Service (DoS) and Distributed DoS (DDoS) threats. By using SDN, it is possible to engineer security solutions that can provide scalable, robust, and secure network communication for IoT devices, even with their limited capabilities(Galeano-Brajones *et al.*, 2020).The growing usage of Internet of Things (IoT) devices can lead to these issues: a rise in the controller's processing workload, and a lack of space in the switches' flow table to handle the flow entries. These problems can cause unwanted network behaviors and unstable network performance, particularly in large-scale networks. To address these issues and to enhance flow table management, decrease the controller's processing workload, and mitigate security threats and vulnerabilities in both controllers and switches, several solutions have to be suggested(Isyaku *et al.*, 2020).

The vulnerability of Software Defined Networking (SDN) control platforms to Denial of Service (DoS)

and Distributed DoS (DDoS) attacks, while SDN offers benefits such as separation of the control and data planes and centralized control for improved network management, the centralized control also exposes the network to security challenges, particularly in the form of DoS and DDoS attacks. The use of Machine Learning (ML) techniques as a solution to counter DoS and DDoS attacks in SDN. Leveraging ML's capabilities in fingerprinting security vulnerabilities. The ML techniques are evaluated in a practical setup, where the SDN controller is deliberately exposed to DDoS attacks. The purpose of this evaluation is to draw important conclusions regarding the effectiveness of ML-based security measures for future communication networks(Ahmad and Harjula, 2022).

Nowadays, security threats in Software Defined Network (SDN) architectures are like traditional networks but exhibit different characteristics. The emergence of SDN has introduced new dimensions to these threats. For instance, a denial-of-service attack targeting a centralized controller responsible for managing a large network with multiple network devices (e.g., routers, switches) can have more devastating consequences compared to an attack focused on a single router. A compromised SDN controller can provide a hacker with control over the entire network, while a compromised router would only disrupt traffic routed through it. As a result, SDN faces new security challenges, particularly in securing the SDN architecture itself. Securing SDN involves addressing challenges across multiple layers and programming interfaces within the three-layer architecture, posing significant hurdles. With the progressive deployment of SDN, these security challenges are expected to increase (Maleh Y., Qasmaoui Y., El Gholami K., Sadqi Y., 2022).

The literature review covers multiple papers discussing various challenges and effects of Software-Defined Networking (SDN). Key challenges highlighted include standardization and integration issues, security vulnerabilities, scalability concerns, performance impacts, complexity in deployment, reliability risks, QoS guarantees, integration with legacy systems, operational challenges, and organizational resistance. Effects of SDN discussed include improved network flexibility, centralized control, programmability, and potential for enhanced security and network management. The selection seems to prioritize recent publications (mostly from 2020-2022) to capture the latest developments in the field. The review includes at least 7-8 papers that specifically address security issues in SDN. These papers discuss various security challenges such as centralized controller vulnerabilities, communication channel security, flow table manipulation, and DoS/DDoS attacks. The review covers various aspects of SDN security, including threats to different components of the SDN architecture, potential attack vectors, and proposed mitigation strategies. It also touches on the need for ongoing research in areas such as controller security, secure communication protocols, and advanced threat detection mechanisms.

3. Security Issues in Software Defined Networking

Software-Defined Networking (SDN) is a paradigm that separates the control plane from the data plane in networking. It centralizes the network control and enables administrators to manage network services through abstracted software interfaces. While SDN offers many advantages such as flexibility, scalability, and automation, it also introduces several security challenges that need to be addressed. Let's explore some of the key security challenges in Software-Defined Networking:

3.1. Centralized Control:

In SDN, the control plane is centralized, meaning all network decisions are made from a single controller. If the controller is compromised, the entire network could be at risk. Attackers may attempt to gain unauthorized access to the controller to manipulate network policies, divert traffic, or perform other malicious activities.

3.2. Controller Security:

SDN controllers are critical components in the SDN architecture. They are responsible for managing network devices and enforcing network policies. If an attacker gains control of the controller, they can manipulate network configurations and cause disruptions or unauthorized access. Therefore, securing the controller infrastructure, applying strong access controls, and keeping the controller software

up-to-date are essential to prevent such attacks.

3.3. Communication Security:

SDN relies heavily on communication between the controller and network devices. This communication needs to be secure to prevent eavesdropping, man-in-the-middle attacks, or tampering. Encrypting the communication channels and using secure authentication mechanisms are crucial to protect against these threats.

3.4. SDN Northbound and Southbound APIs:

SDN networks use northbound APIs to communicate with higher-level applications and southbound APIs to interact with network devices. Both of these APIs can be potential attack vectors. Insecure APIs may allow attackers to manipulate network behavior or inject malicious commands. Proper authentication, authorization, and input validation must be enforced on these APIs to mitigate such risks.

3.5. Virtualization Vulnerabilities:

SDN often involves network virtualization, where multiple virtual networks or overlays run on the same physical infrastructure. Vulnerabilities in the virtualization layer could lead to unauthorized access to other virtual networks or resource isolation issues. Regularly updating and patching virtualization software is vital to minimize these risks.

3.6. Denial-of-Service (DoS) Attacks:

SDN controllers can become targets of DoS attacks, which overwhelm the controller's resources and cause network disruptions. Additionally, SDN enables new attack vectors for DoS, such as exploiting flow table exhaustion or control message flooding. Implementing rate limiting, flow table management, and traffic filtering mechanisms can help mitigate these attacks.

3.7. Flow Table Security:

The flow table is a fundamental component in SDN switches that maintains forwarding rules. If an attacker gains unauthorized access to the flow table, they can manipulate traffic routing and potentially bypass security policies. Strong access controls and regular flow table validation are essential to protect against such attacks.

3.8. Misconfigurations and Policy Errors:

The flexibility of SDN can lead to misconfigurations or policy errors that result in security vulnerabilities. For instance, misconfigured network policies might inadvertently expose sensitive data or services. Proper auditing, monitoring, and validation of network policies can help prevent these issues.

3.9. Insider Threats:

While not exclusive to SDN, the centralized control and management of SDN networks increase the potential impact of insider threats. Malicious or negligent insiders with access to the controller may abuse their privileges to compromise the network.

To address these security challenges, organizations implementing SDN should adopt a comprehensive security approach:

- Secure Development and Configuration: Implement secure coding practices, follow security guidelines for SDN components, and ensure proper configuration management.
- Access Control and Authentication: Enforce strong access controls to SDN components and use multi-factor authentication for critical access.
- Encryption: Encrypt communication channels between the controller and network devices to prevent unauthorized access and data interception.
- Network Monitoring and Logging: Deploy robust monitoring and logging solutions to detect suspicious activities, anomalous behavior, and potential security breaches.
- Regular Updates and Patches: Keep all SDN components, including the controller and network devices, up-to-date with security patches and firmware updates.
- Security Audits and Penetration Testing: Regularly conduct security audits and penetration testing to identify vulnerabilities and weaknesses in the SDN infrastructure.

By addressing these security challenges and implementing best practices, organizations can enhance the security posture of their SDN deployments and leverage the benefits of this innovative networking paradigm while mitigating potential risks.

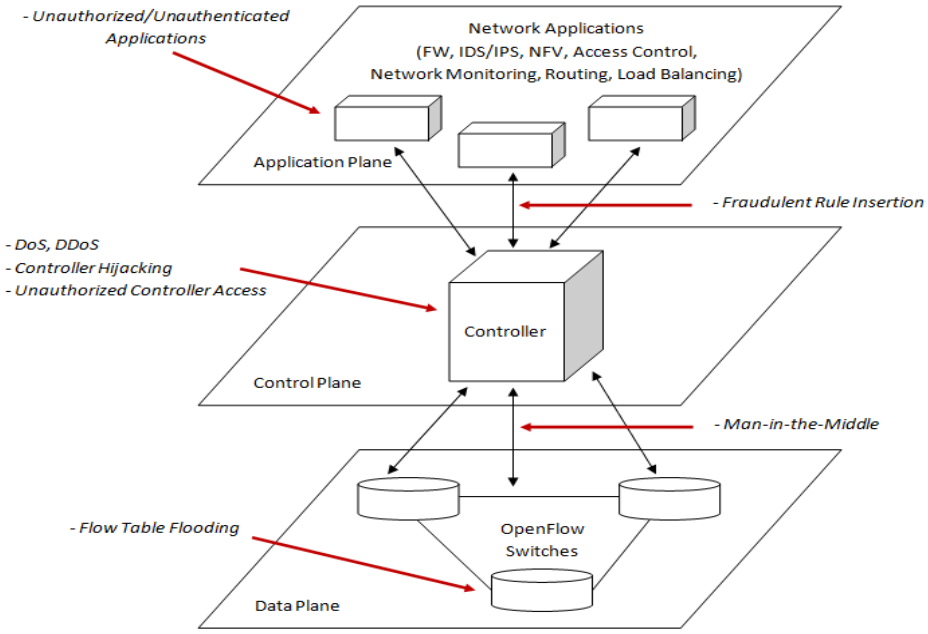

Figure 2: Security threats and attacks surface in SDN ((Akbaş *et al.*, 2016))

In figure 2 above, SDN security threats encompass the risk of controller compromise, insecure SDN components, malicious flow installation, DoS attacks, MitM attacks, flow table exhaustion, policy manipulation, insider threats, misconfiguration, and data plane attacks. The attack surface includes vulnerabilities in the interfaces, communication channels, management protocols, and APIs between the controller and devices, as well as the flow table management mechanisms, northbound APIs, policy management interfaces, privileged interfaces, configuration mechanisms, and data plane interfaces. Mitigating these threats requires employing secure development practices, regular security assessments, network monitoring, access controls, encryption, strong authentication, and promoting security awareness among staff.

4. Challenges of Software-Defined Networking

Software-defined networking (SDN) takes a groundbreaking approach to network management that streamlines infrastructure and enhances adaptability and responsiveness. However, materializing the complete potential of SDN requires tackling some significant obstacles.

4.1. Standardization and Integration: The absence of standardized interfaces and protocols leads to integration and interoperability challenges between components from various vendors. Industry-wide standardization is imperative to avert vendor lock-in and ensure seamless integration of diverse solutions.

4.2. Security: Centralized control introduces risks of single-point failures and new attack vectors. Robust security measures like encrypted channels, access controls, and sandboxing are vital for safeguarding controllers and ensuring secure controller-switch communications. Moreover, programming errors can inadvertently introduce vulnerabilities in dynamic, programmable environments.

4.3. Scalability: As networks expand and traffic increases, scaling controller functionality and flow rules without compromising performance becomes problematic. Controller placement strategies, clustering, and horizontal scaling techniques are critical for large-scale deployments.

4.4. Performance: The additional abstraction layer can raise latency concerns and degrade network performance. Solutions like efficient caching, optimized rule distribution, and fast packet processing in data plane hardware are key for maintaining low latency.

4.5. Complexity: Despite simplifying management, initial SDN deployment and configuration can be complicated for traditional

network admins. Personnel may require developing competencies in scripting and SDN paradigms. Moreover, distributed states and abstractions make debugging tricky.

4.6. Reliability: Centralized control introduces availability and reliability concerns. Redundancy protocols, failover mechanisms, and distributed/hierarchical control planes are essential for continuous operation during outages.

4.7. QoS: Guaranteeing critical traffic prioritization while preventing congestion is complicated across dynamic, heterogeneous environments. Careful capacity planning, traffic engineering, and real-time monitoring are required for effective QoS.

4.8. Integration: Migrating legacy infrastructures or integrating them with SDN networks is challenging due to differences in control paradigms and configurations. Incremental deployment through overlays and gateways facilitates gradual transitions.

4.9. Operations: Continuous monitoring, rapid adaptation to changing conditions, traffic forecasting, and root cause analysis become imperative. Advanced telemetry, machine learning-based analytics and automation are critical for these tasks.

4.10. Organizational: Technological transitions bring cultural and organizational shifts. Stakeholder resistance, lack of expertise, and apprehensions about new technologies can impede adoption. Change management and skills development are key.

While significant, ongoing R&D is resolving many of these challenges as SDN evolves. Widespread adoption is expected to further surmount these hurdles and enable more responsive, dynamic networks.

5. Future Research Directions

The future research problems in software defined networking include: how to build large scale SDN networks with millions of flows and devices. Current SDN controllers have limitations in the number of flows they can handle. More research is needed on distributed SDN controller architectures and coordination between controllers to handle very large networks. Optimization of controller placement in large networks. As networks grow bigger, determining the optimal number of controllers and where to place them geographically becomes critical. More work is needed on algorithms for dynamic controller placement. Development of standard high-level programming languages and abstractions for SDN. This will make it easier for network operators to program the network. Debugging and troubleshooting tools for SDN networks. Networks are complex and bugs/errors can occur, so tools are needed to quickly identify and fix issues. Securing the controller and communication channels from attacks. The centralized controller is a prime target. Mechanisms to provide reliability, fault-tolerance and fast failover in SDN networks. Networks must remain operational under various failure scenarios. Effective mapping of virtual networks and network slices onto the physical infrastructure. This includes optimization of resource allocation. Isolation and customization of network slices for different applications and users. Transition strategies from legacy networks to SDN, and operation of hybrid SDN/legacy networks. A gradual transition is needed in most cases. Interoperability between SDN and traditional control planes. Protocols and interfaces for hybrid operation. Using machine learning to optimize traffic engineering, load balancing, anomaly detection and other network functions in an SDN architecture. Automated network management using reinforcement learning and neural networks for dynamic network control.

6. Conclusions

Software defined networking (SDN) has emerged as a transformational new network architecture that decouples the control and data planes, enabling centralized and programmable control. The literature review summarizes key developments, applications and remaining challenges of SDN across security, scalability, reliability, performance, and other aspects. While SDN promises many benefits like flexibility, automation, and easier network management, realizing its full potential requires overcoming non-trivial challenges. Standardization across various components is still lacking, posing integration issues. Centralized control introduces single-point failures and new attack surfaces demanding

rigorous security. Performance and latency concern due to increased abstraction have to be 2sorted out through caching, optimized rules and faster data plane processing. Large scale deployments face controller scalability and placement issues. Despite simplifying management, initial configuration complexity persists. Guaranteeing QoS and failover remains tricky. Incremental deployment strategies are needed for migrating legacy networks. Continuous monitoring, root cause analysis and traffic engineering necessitate advances in ML driven analytics, automation and control. As SDN continues maturing, ongoing research and innovation focused on these open challenges show promise. Growing adoption should facilitate developing standards and best practices. Controller architectures, distributed state management, security mechanisms, high-level abstractions, virtualization techniques, and hardware accelerators are areas seeing focused improvement. With holistic solutions to these multifaceted challenges, SDN is poised to become the norm for dynamic, flexible and highly automated network architectures of the future.

References

Ahmad, E., Harjula, M. Y. and I. A. (2022). Evaluation of Machine Learning Techniques for Security in SDN. *IEEE Globecom Workshops (GC Wkshps, Taipei, Taiwan*, 1–6. Abuarqoub, A. (2020). A review of the control plane scalability approaches in software defined networking. *Future Internet*, *12*(3). https://doi.org/10.3390/fi12030049

Ajiya Ahmad, A., Boukari, S., Musa Bello, A., Alhaji Madu, M., and Gimba, adatu. (2021). A Review on Software Defined Network (SDN) Based Network Security Enhancements. *Quest Journals Journal of Software Engineering and Simulation*, *7*(9), 2321–3809. www.questjournals.org

Akbaş, M. F., Karaarslan, E., and Güngör, C. (2016). A Preliminary Survey on the Security of Software-Defined Networks. *International Journal of Applied Mathematics, Electronics and Computers*, *4*(Special Issue-1), 184–184. https://doi.org/10.18100/ijamec.270088

Alshammri, G. H., Samha, A. K., Hemdan, E. E. D., Amoon, M., and El-Shafai, W. (2022). An Efficient Intrusion Detection Framework in Software-Defined Networking for Cybersecurity Applications. *Computers, Materials and Continua*, *72*(2), 3529–3548. https://doi.org/10.32604/cmc.2022.025262

Babbar, H., Rani, S., Singh, A., Abd-Elnaby, M., and Choi, B. J. (2021). Cloud based smart city services for industrial internet of things in software-defined networking. *Sustainability (Switzerland)*, *13*(16), 1–14. https://doi.org/10.3390/su13168910

Galeano-Brajones, J., Carmona-Murillo, J., Valenzuela-Valdés, J. F., and Luna-Valero, F. (2020). Detection and mitigation of DoS and DDoS attacks in iot-based stateful SDN: An experimental approach. *Sensors (Switzerland)*, *20*(3), 1–18. https://doi.org/10.3390/s20030816

Goud, K. S., and Gidituri, S. R. (2022). Security Challenges and Related Solutions in Software Defined Networks: A Survey. *International Journal of Computer Networks and Applications*, *9*(1), 22. https://doi.org/10.22247/ijcna/2022/211595

Hayajneh, A. Al, Bhuiyan, M. Z. A., and McAndrew, I. (2020). Improving internet of things (IoT) security with software-defined networking (SDN). *Computers*, *9*(1), 1–14. https://doi.org/10.3390/computers9010008

Isyaku, B., Soperi, M., Zahid, M., and Kamat, M. B. (2020). *Software Defined Networking Flow Table Management of OpenFlow Switches Performance and Security Challenges : A Survey*.

Jing, L., Chen, X., and Wang, J. (2021). Design and implementation of programmable data plane supporting multiple data types. *Electronics (Switzerland)*, *10*(21). https://doi.org/10.3390/electronics10212639

Khalil Al Dulaimi, L. A., Badlishah Ahmad, R., Yaakob, N., and Hussein, Q. M. (2018). A Secured OpenFlow Protocol Using Elliptic Curves Cryptographic for Software Defined Networks. *Journal of Physics: Conference Series*, *1019*(1). https://doi.org/10.1088/1742-6596/1019/1/012014

Maleh Y., Qasmaoui Y., El Gholami K., Sadqi Y., and M. S. (2022). *A comprehensive survey on*

SDN security: threats, mitigations, and future directions. (pp. 1–39).

Ngo, H. V, Tran, P. H. L., Nuclei, A. G., Hood, C. E., Barth, A. J., Ho, L. C., Defined, S., Controllers, N., Abdullah, A. F., and Salem, F. M. (2020). *POX Controller and Open Flow Performance Evaluation in Software Defined Networks (SDN) Using Mininet Emulator POX Controller and Open Flow Performance Evaluation in Software Defined Networks (SDN) Using Mininet Emulator*. https://doi.org/10.1088/1757-899X/881/1/012102

Onyema, E. M., Kumar, M. A., Balasubaramanian, S., Bharany, S., Rehman, A. U., Eldin, E. T., and Shafiq, M. (2022). A Security Policy Protocol for Detection and Prevention of Internet Control Message Protocol Attacks in Software Defined Networks. *Sustainability (Switzerland)*, *14*(19). https://doi.org/10.3390/su141911950

Tung, Y. H., Wei, H. C., Ti, Y. W., Tsou, Y. T., Saxena, N., and Yu, C. M. (2020). Counteracting UDP flooding attacks in SDN. *Electronics (Switzerland)*, *9*(8), 1–28. https://doi.org/10.3390/electronics9081239

Ujcich, B. E., Jero, S., Skowyra, R., Bates, A., Sanders, W. H., and Okhravi, H. (2021). Causal analysis for software-defined networking attacks. *Proceedings of the 30th USENIX Security Symposium*, 3183–3200.

Wazirali, R., and Ahmad, R. (2021). *applied sciences SDN-OpenFlow Topology Discovery : An Overview of Performance Issues*.

Wikipedia contributors. (2023). *Software-defined networking*.

Yungaicela Naula, N. M., Vargas Rosales, C., Pérez Díaz, J. A., and Zareei, M. (2022). Towards security automation in Software Defined Networks. *Computer Communications*, *183*(March 2021), 64–82. https://doi.org/10.1016/j.comcom.2021.11.014