Software-Defined Networking (SDN): A Review

1st Quadri Waseem
Faculty of Computing
Universiti Malaysia Pahang
Pekan, Malaysia
qwaseem@analyticray.com

2nd Wan Isni Sofiah Wan Din
Faculty of Computing
Universiti Malaysia Pahang
Pekan, Malaysia
sofiah@ump.edu.my

3rd Afrig Aminuddin*
Faculty of Computer Science
Universitas Amikom Yogyakarta
Sleman, Indonesia
afrig@amikom.ac.id
*corresponding author

4th Muzammil Hussain Mohammed
College of Computers and Information Technology
Taif University
Taif, Saudi Arabia
m.muzammil@tu.edu.sa

5th Rifda Faticha Alfa Aziza
Faculty of Computer Science
Universitas Amikom Yogyakarta
Sleman, Indonesia
rifda@amikom.ac.id

2022 5th International Conference on Information and Communications Technology (ICOIACT) | 978-1-6654-5140-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICOIACT55506.2022.9972067

***Abstract*—The Internet of Everything (IoE) connects millions of machines, vehicles, nodes, smoke detectors, watches, glasses, webcams, and other devices to the internet. These entities need the proper guidance and control for expected performance. There is always a need to manage their networks for better performance properly. However, managing all these entities is not easy; it is always a big concern. All types of network architectures are getting enhanced daily, and the traditional network management process becomes more complex, especially rendering the performance during technology and entity modifications. Software-Defined Networking (SDN) is extensively used in all types of networks, especially in future network technologies (IoT, IoV, 6G, AI, etc.) to tackle such types of concerns and issues. However, as with any new phrase or paradigm, no clear description of this technology has emerged yet, which will give a complete understanding of SDN, from basic terminology to its management capabilities. The contribution of this research article is a significant step forward in understanding the basics of SDN. This research article proposes a detailed review of SDN in the form of history, overview, architecture, benefits, services, trends, application, features, and challenges.**

Keywords—Software-Defined Networking; future technology; real-time data capabilities; SDN-based management; SDN applications

I. Introduction

Software-Defined Networking (SDN) is a network design that reduces network stiffness and provides ease in network management. Because of its compatible structure, SDN's network behavior is more flexible and adaptable to each enterprise, university, or user group's needs. Software-Defined Networking is not a brand-new or revolutionary concept; it grew out of contributions, ideas, and discoveries made in the field of research networking. In [1], the evaluation of SDN has been mentioned starting from the Active Networks (the mid-90s to early 2000), followed by the control plane separation (2001–2007), and then, at last, the existence of OpenFlow API and NOS (2007–2010). These three evolutionary achievements are considered the three major stages in the evolution of SDN. Hence, the current centralized design allows crucial network data to be collected and used to dynamically modify and adapt various network-related rules with ease for better performance [2].

Traditional IP networks, despite their widespread adoption, are incredibly complex and challenging to operate in terms of management. Various issues, such as policy enforcement across a wide range of devices, high connectivity, fault tolerance, complex traffic isolation, robustness application, user-aware routing, and so on, have resulted in the creation of overlapping mechanisms at various network layers, making management more complex and potentially exposing security loopholes. [3]. The network transport protocols and distributed control that are used in today's communication networks also present a slew of operational challenges in traditional networks.

The tight coupling of networking devices, hardware, and software has also hampered the development of these technologies or solutions. Furthermore, as connection rates increase, the entire transmission mechanism (packet forwarding) becomes hardware-centric, delegating management or network administration to opt for a software program. Because a server always has more significant processing and memory resources than a single network device [2]. Hence, this consideration has generated a need to develop the OpenFlow Protocol for the network community.

The OpenFlow protocol [4] is now the most extensively used research community and has laid the foundation for many initiatives. Software-Defined Networking (SDN) adjusts specific changes in today's networks. The data and control planes must be decoupled or separated to enable self-development and evolution. Second, SDN gives the network a centralized control plane with a global view. Finally, SDN establishes open communication channels between the control and data planes. The administrator can configure and manage network resources using the controller's accessible APIs (proprietary or open)[2] so that the user needs can be timely served with perfection.

II. SDN Architecture and its Pillars

SDN possesses a new architecture based on five pillars: (i) separation of the control and data planes, (ii) flow abstraction flexibility and agility, (iii) network programmability, (iv) vendor neutrality, and (v) central control and manageability.

In an SDN paradigm, the routing devices are intended to forward packets, with network control logic located at the network controller/operating system. Second, the centralized controller can configure, manage, secure, and optimize network resources in real-time [5]. Furthermore, network programmability is enabled via a centralized controller, allowing for program-based network management. The network traffic can be dynamically changed using a network abstraction and a global network view. Open standards and vendor neutrality are promoted by SDN, which encourages more innovation and acceptance for future modifications/changes. To control the forwarding elements,

978-1-6654-5140-6/22/$31.00 ©2022 IEEE

the controller employs the OpenFlow protocol. For communication between forwarding devices, the OpenFlow protocol is used; moreover, the controllers [6] have grown in popularity, with installations at various scales. Besides advantages, various technical, financial, and business challenges exist [7],[8].

III. Benefits Offered by SDN

Compared to traditional network devices, SDN forwarding devices (switches) become more straightforward and less expensive due to the centralization of the network controller. The management and configuration of the network have also been simplified [9]. To respond to new business requirements, SDN may be adjusted faster than present network architectures. SDN controls and enhances the network performance in today's world [10]. Besides that, any new application, protocol, or policy can be quickly installed by running an application on the forwarding device controller. Paper [11] has highlighted the benefits of SDN, which we have subdivided into five main subcategories:

A. Content delivery

One of the main benefits of SDN is that it allows regulated data flow (Traffic) smoothly. Implementing Quality of Services (QoS) becomes more accessible with the ability to automate and direct data traffic features of SDN [12]. Administrators now need to put less effort into efficient traffic management using SDN.

B. Cost-Effectiveness

Existing hardware can be reused to follow the commands of an SDN controller, enabling the use of more cost-effective hardware with a more significant impact. Due to automation in central controlling, the least effort can save time and affect the overall performance cost-effectively.

C. Centralization

SDN provides a consolidated view of an organization's whole network, making corporate administration and provisioning more efficient [13]. The centralized control makes controlling easier and more powerful in distributed environments.

D. Management

By deploying SDN, information technology teams can alter small network configurations without affecting the whole network. [11]. The management becomes simple and fewer efforts are utilized due to the various protocols and flexibility provided by the centralized control.

Content Delivery
Other Advantages
Cost Effective
Management
Centraliza tion

Fig. 1. Benefits offered by SDN

E. Other Advantages

Integration of multiple applications becomes easier, like (load balancing and routing programs) [14],[15]. The integrating modes of software and existing /new hardware are comfortable and are flexible for changes. SDN fosters innovation by allowing businesses to quickly deploy new services and apps to generate new revenue streams and increase network value [16],[17]. Other unique advantages [18] are acting as a trend in the world of IOE (Internet of Everything). Fig. 1 presents the benefits offered by SDN, which we have subdivided into five main subcategories as:

IV. Services and Current Trend of SDN

The Software-Defined Networking (SDN) paradigm [19] offers the capacity to define the network and thus permits the incorporation of auto and dynamic control approaches by separating hardware (data plane) and software (control plane) symmetrically and allowing their independent evolution. SDN attempts to centralize network control, allowing for enhanced visibility and flexibility in network management and performance optimization. Compared to the overlay network substitutes, SDN can effectively control and transmit data across a whole public network connection, not just on a small number of nodes. Furthermore, SDN relieves network operators from time-consuming efforts to construct a suitable overlay network for a particular use case. It includes a built-in programmatic architecture for hosting centralized security and access applications that meets today's IoT (Internet of Things) high demand needs [20] to ensure users' flawless quality of experience (QoE) in a best-managed way.

Along with the excitement of following trends, there are various concerns and doubts about the widespread adoption of SDN networks. Physical centralization of the control plane in a single programmable software component, such as a controller, has various limitations in terms of scalability, availability, reliability, and feasibility of SDN implementation. Moreover, the belief in the control plane as a distributed architecture [21] has become unavoidable, with multiple SDN controllers managing the entire network while preserving a logically centralized network perspective. In this regard, the networking community debated the best strategy to build distributed SDN designs and considered the unique issues of such distributed systems based on their requirements and future trends. Consequently, numerous SDN solutions are being investigated, and various SDN projects have emerged. Each suggested SDN controller platform uses a unique architectural design approach based on multiple considerations, comprising the traits of interest, performance targets, the development, and the deployment of SDN use cases to overcome the problems of several competing challenges. Although, despite the significant level of interest in SDN, it is still in its early stages of adoption in the industrial environment. Before the technology matures and the standardization efforts pay off, allowing SDN's full promise to be realistic and viable, there is a long road ahead. Currently, the need is emphasized to thoroughly examine the suggested SDN solutions to anticipate the future trends that will drive future research of SDN [22].

V. SDN Application

SDN provides the capacity to change network behaviour based on user needs. To put it another way, SDN does not solve any specific problem but provides a more versatile tool for network administration. The following are some of these applications of real-time. In paper [2], applications of SDN have been highlighted, and we have subdivided them into seven main subcategories:

A. *Home Networking*

With the emergence of the Internet of Things (IoT), home networking has come up with a new type of ease in our daily lives. Due to many customers and devices linked to the same point, managing devices and network resources in home networks is a significant difficulty (usually a base station). The real-time implementation can be found in [23],[24].

B. *Security-Based*

In today's world, security-based applications are a trend. The network's overall vision cannot be realistic till the system is not secure. All countermeasures are useless when the host is hacked; this security cannot be reliant on the host only; there are other factors also. The real-time implementation can be found in [25],[26].

C. *Virtualization*

Virtualization in networks is comparable to operating system virtualization, enabling many operating systems to share hardware resources. With network virtualization, numerous virtual networks with their topology and routing logic can run on the same infrastructure. The real-time implementation can be found in [27],[28].

D. *Mobile Networks*

All network-based limits are applied to mobile carrier communications infrastructure. Carriers adhere to industry standards and procedures, such as the 3rd Generation Partnership Project (3GPP), and vendor-specific customizations are prominent. With SDN's flow-based idea, the SDN can now be easily applied to mobile users and applications. The real-time implementation can be found in [29],[30].

E. *Multimedia*

Multiple online multimedia services, such as real-time broadcasts, necessitate a high level of information accessibility and the demands efficiency of the internet infrastructure. To achieve efficient performance, various types of multimedia should be effectively controlled for maintenance purposes of communication. The real-time implementation can be found in [31],[32].

F. *Reliability and Recovery*

One typical issue in basic network management is recovering from a connection failure. Because of the node's limited information, the path must be recalculated, affecting the convergence time. When the controller receives a failure signal, it utilizes the restoration mechanism to hunt for an alternative path. Meanwhile, the system envisions a failure and generates a backup solution ahead of time for security protection [2]. The real-time implementation can be found in [33],[34].

G. *Other Future Applications*

The data distributed in connected entities like autonomous cars, machines, and various electronic devices are considered a future SDN application. The management process in these entities includes the safety primitives and other efficiency-related factors like cost, which are also a crucial part of SDN implementation. These devices are always dependent and require proper communication and maintenance for better efficiency. The paper [11] has mentioned such future applications of SDN, for example, rural connection, data center upgrading, etc., in detail. The real-time implementation can be found in [35-37]. Fig. 2 presents the SDN applications, which we have subdivided into seven main subcategories as:

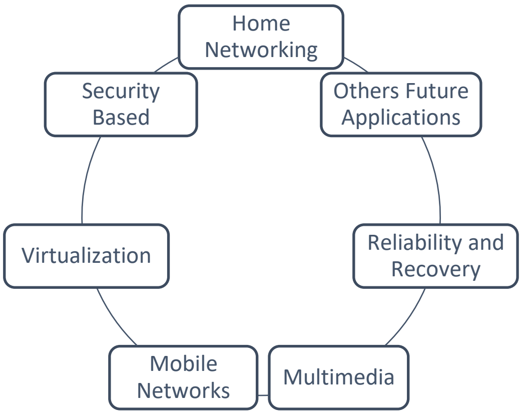

Fig. 2. SDN Applications

VI. SDN Features/Parameters

As discussed in previous sections, SDN is a flexible and effective network control and management tool. Scientists are working harder to demonstrate how a future SDN might gain from aligning SDN's unique features to various use cases in determining whether an SDN adopter is a suitable solution for a particular use case. In paper [42], the features of SDN have been identified as:

A. Programmability:

It is a key feature of SDN, and it's what drives the majority of SDN use cases. This enables control plane innovation using regular application development approaches, allowing network adaptation based on a particular setup or condition. Software and hardware integration is feasible, and all credit goes to the programmability feature of SDN.

B. *Protocol Independence*

SDN may control or function alongside various networking technologies and protocols at many network layers due to its protocol freedom. This feature allows the transition from old to new products and the use of a separate network protocol stack for each program. This feature eliminates the cost issues arising during each major change at the modification level.

C. *Dynamic Nature*

This SDN feature is defined by the ability to dynamically and near-real-time alter network settings. Due to the utilization of timeframes, dynamic reconfiguration is feasible in SDN. This might range from wide-area networks to local networks ranging with only a few daily changes to data center networks, where virtual machines are continuously created or moved, and where network connectivity must be established within minutes, if not seconds.

D. *Granularity*

Networking encompasses multiple protocol layers as well as different degrees of data flow aggregations. On both aggregate and policy layers, SDN allows traffic flow management at various levels of granularity. Massive MPLS tunnels in data centers can be seen connecting to a single

Transmission Control Protocol (TCP) link in a residential LAN (LAN). This granularity is necessary for adaptability and the ability of the control plane to operate at various levels.

E. Elasticity

The elasticity characteristic of SDN refers to the capacity of the SDN network control and data planes to adjust their resource usage depending on the capabilities required. Because controllers are programs, they may be developed and coordinated across multiple physical or virtual hosts with relative convenience by utilizing a distributed or hierarchical method. As a result, the control plane can respond to traffic mix and volume changes and adjust its expansion based on the current load [42]. Fig. 3 presents the SDN features, which have been sub-divided into five main subcategories as:

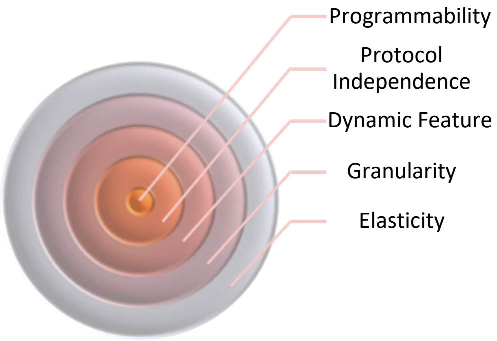

Fig. 3. Features of SDN

VII. Challenges of SDN Technology

As a production network technology, SDN offers various benefits. However, some critical hurdles in terms of scalability, security, dependability, and other elements must be addressed before the implementation for the future feasibility of commercial exploitation.

SDN paradigm faces multiple hurdles, including technology, finance, and business problems. These issues have hampered the acceptance of SDN as a full deployment in future technologies. Many technical problems must be answered, such as incorporating the centralized controller's adaptability, robustness, and endurance without introducing a single point of failure. SDN controller is an appealing target in terms of security and threats due to its central control mechanism. As the network expands, more than a single SDN controller can be added to control the high load as a temporary solution [7]. In the absence of a safe and trustworthy controller, attackers can change the behavior of the underlying network by manipulating the controller program. A high focus on security is required, and a basic need for making SDN acceptable on a big scale is also needed. So far, there have been a few talks about SDN security in the business and innovation communities. There are possible weaknesses throughout the SDN platform. Authentication and authorization techniques, for example, have been questioned for their ability to allow numerous businesses to use network resources while yet ensuring that these resources are adequately protected [8]. Financial obstacles include the significant budget commitment required for SDN deployment, which most firms cannot afford. This new paradigm necessitates the creation of a new set of rules and regulations by the network administrator and the paradigm's application for new hardware devices' security. There are currently no well-tested, production-ready techniques for dealing with these legitimate issues. The answer can be a complete policy adoption or a gradual rollout. In terms of business challenges, the shift could cause end-user services to be disrupted. Network operators must develop the confidence to choose the new paradigm over the tried-and-true traditional options. [3].

Besides these technical and commercial challenges, other implementation challenges must be addressed urgently before planning to implement SDN on a large scale and already existing setups. Paper [2] has highlighted the challenges of SDN, which we have subdivided into four main subcategories

A. Reliability

The data and control planes are separated, allowing for autonomous development and evolution. The rate at which packets are processed in the data plane is determined by the hardware installed, such as Application-Specific Integrated Circuits (ASIC), Application Specific Standard Products (ASSP), multicore CPU/GPP, or Field Programmable Gate Array (FPGA). On the other hand, the control plane's performance is mainly determined by the hardware and the Network Operating system-NOS -NOS (Beacon, POX, and Floodlight) [2]. Bad performance from one of the two planes can lead to severe issues such as packet loss or slowness and inaccurate network denial-of-service behavior (DDoS). As a result, SDN elements (software and hardware) should balance performance, reliability, affordability, and simplicity deployment for future real-time implementation [2].

B. Scalability

OpenFlow uses hardware resources such as widespread flow tables in real-world networks. On the other hand, SDN can be extended beyond flow tables to take advantage of extra hardware resources [38]. A recent open topic is integrating and investigating new features between the control and data planes. SDN innovation can combine and conveniently use apps like encryption, assessment, and network traffic classification as well as multiple systems like middleboxes and customized packet processors. The network's design and volumetric efficiency-based performance are the determining variables for the number and position of controllers, according to the study mentioned in [39].

C. Security

Another crucial aspect that must be considered is SDN security. All network apps may not have the same user access [40]. However, to gain access to network resources, profiles must be assigned, authentication must be accomplished, and authorization must be issued appropriately. TLS (Transport Layer Security) can also be used as an authentication method between the switch and the controller in OpenFlow. However, there are no defined security criteria for many controller systems communicating data with each other and switches. Furthermore, OpenFlow demonstrates that an unidentified packet can be sent to the controller, which can be seriously affected by the virus. Paper [11] has expanded on several security concerns regarding SDN. The developments of SDN applications that increase network security and the security of the SDN infrastructure are discussed for future SDN implementation.

D. Other Technical Issues

It's also challenging to move from traditional network topologies to SDN-based designs. It is impossible to completely overhaul the communications infrastructure despite the emergence of the OpenFlow-compatible network feature. Mechanisms, protocols, and interfaces that allow both architectures to coexist throughout the transition period are required for implementation. The Open Networking Foundation (ONF) has presented the IF Config Protocol [41] as a first step in customizing OpenFlow devices, and also substantial efforts are underway for its large-scale implementations. Similarly, the European Telecommunications Standards Institute (ETSI), the Internet Engineering Task Force (IETF) Forwarding, and the Control Element Separation Working Group (ForCES) collaborate on regulating interfaces for SDN deployment. [2]. Fig. 4 presents the challenges of SDN technology, which we have sub-divided into four main subcategories as:

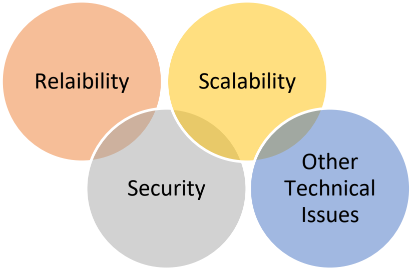

Fig. 4. SDN Challenges

As the demand for real-time applications grows, scaling current networks while maintaining dependability and security without sacrificing performance has become difficult. Due to the availability of multiple technologies and partners, traditional network strategies frequently try to improve service quality for several services or for a portion of networks. While aiming for optimal performance, these approaches rely on local data and ignore cross-layer considerations, resulting in sub-optimal results. The challenges with network management are numerous, and the connectivity alterations that must be undertaken are unexpected.

VIII. Latest Literature Survey

We have seen a lot of recent works in the field of SDN. The SDN has been utilized in almost all types of future networks and future technologies. Some of the recent works related to various uses of SDN include enhancement in load balancing [43],[44], network management[45], security[46]-[51] etc. Additionally, SDN has been used in various types of domains like 5G architectures [52], Blockchain[53], Wireless Sensor Networks (WSN)[54], IoT [55],[56], and many other related technologies.

IX. Conclusion

This research article outlines and surveys the SDN and its overview in the form of history, architectural overview, benefits, services, future trends, applications, and SDN challenges. We have divided the body of information into SDN and its management to comprehend the essential technologies complexities better. We have offered a thorough lesson on the idea, system design, trend, and applications for each element individually. Besides giving a relative benefit, there is still a slew of unsolved issues that require additional exploration from the standpoint of essential approaches and advanced solutions for using SDN in the future for future internet and its related technologies. In the future, we plan to survey the SDN-related applications and domains where SDN can enhance and improve the efficiency of their basic architectures.

Acknowledgment

This research was supported by Universiti Malaysia Pahang through the Research Grant Scheme RDU192622.

References

[1] Li, C-S., Brad L. Brech, Scott Crowder, Daniel M. Dias, Hubertus Franke, Matt Hogstrom, David Lindquist et al. “Software defined environments: An introduction.” IBM Journal of Research and Development 58, no. 2/3 (2014): 1-1.

[2] Valdivieso Caraguay, Ángel Leonardo, Alberto Benito Peral, Lorena Isabel Barona Lopez, and Luis Javier Garcia Villalba. “SDN: Evolution and opportunities in the development IoT applications.” International Journal of Distributed Sensor Networks 10, no. 5 (2014): 735142.

[3] Sinha, Yash, and K. Haribabu. “A survey: Hybrid sdn.” Journal of Network and Computer Applications 100 (2017): 35-55.

[4] McKeown, Nick, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. “OpenFlow: enabling innovation in campus networks.” ACM SIGCOMM computer communication review 38, no. 2 (2008): 69-74.

[5] Kim, Hyojoon, and Nick Feamster. “Improving network management with software defined networking.” IEEE Communications Magazine 51, no. 2 (2013): 114-119.

[6] McKeown, Nick, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. “OpenFlow: enabling innovation in campus networks.” ACM SIGCOMM computer communication review 38, no. 2 (2008): 69-74.

[7] Sezer, Sakir, Sandra Scott-Hayward, Pushpinder Kaur Chouhan, Barbara Fraser, David Lake, Jim Finnegan, Niel Viljoen, Marc Miller, and Navneet Rao. “Are we ready for SDN? Implementation challenges for software-defined networks.” IEEE Communications Magazine 51, no. 7 (2013): 36-43.

[8] Ahmad, Ijaz, Suneth Namal, Mika Ylianttila, and Andrei Gurtov. “Security in software defined networks: A survey.” IEEE Communications Surveys & Tutorials 17, no. 4 (2015): 2317-2346.

[9] Jimson, Emilia Rosa, Kashif Nisar, and Mohd Hanafi bin Ahmad Hijazi. “Bandwidth management using software defined network and comparison of the throughput performance with traditional network.” In 2017 International Conference on Computer and Drone Applications (IConDA), pp. 71-76. IEEE, 2017.

[10] Xia, Wenfeng, Yonggang Wen, Chuan Heng Foh, Dusit Niyato, and Haiyong Xie. “A survey on software-defined networking.” IEEE Communications Surveys & Tutorials 17, no. 1 (2014): 27-51.

[11] Nisar, Kasif, Ian Welch, Rosilah Hassan, Ali Hassan Sodhro, and Sandeep Pirbhulal. “A survey on the architecture, application, and security of software defined networking.” Internet of Things (2020): 100289.

[12] Yu, Changhe, Julong Lan, Zehua Guo, and Yuxiang Hu. “DROM: Optimizing the routing in software-defined networks with deep reinforcement learning.” IEEE Access 6 (2018): 64533-64539.,

[13] Ventre, Pier Luigi, Mohammad Mahdi Tajiki, Stefano Salsano, and Clarence Filsfils. “SDN architecture and southbound APIs for IPv6 segment routing enabled wide area networks.” IEEE Transactions on Network and Service Management 15, no. 4 (2018): 1378-1392.

[14] Kreutz, Diego, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. “Software-defined networking: A comprehensive survey.” Proceedings of the IEEE 103, no. 1 (2014): 14-76.

[15] Casado, Martin, Nate Foster, and Arjun Guha. “Abstractions for software-defined networks.” Communications of the ACM 57, no. 10 (2014): 86-95.

[16] Saadon, Guy, Yoram Haddad, and Noemie Simoni. “A survey of application orchestration and OSS in next-generation network management.” Computer Standards & Interfaces 62 (2019): 17-31.

[17] Zhang, Yuan, Lin Cui, Wei Wang, and Yuxiang Zhang. “A survey on software defined networking with multiple controllers.” Journal of Network and Computer Applications 103 (2018): 101-118.

[18] Herrera, Juliana Arevalo, and Jorge E. Camargo. “A survey on machine learning applications for software defined network security.” In International Conference on Applied Cryptography and Network Security, pp. 70-93. Springer, Cham, 2019.

[19] Feamster, Nick, Jennifer Rexford, and Ellen Zegura. “The road to SDN: an intellectual history of programmable networks.” ACM

SIGCOMM Computer Communication Review 44, no. 2 (2014): 87-98.

[20] Li, Yuhong, Xiang Su, Jukka Riekki, Theo Kanter, and Rahim Rahmani. “A SDN-based architecture for horizontal Internet of Things services.” In 2016 IEEE International Conference on Communications (ICC), pp. 1-7. IEEE, 2016.

[21] Canini, Marco, Daniele De Cicco, Petr Kuznetsov, Dan Levin, Stefan Schmid, and Stefano Vissicchio. "STN: A robust and distributed SDN control plane." (2014).

[22] Bannour, Fetia, Sami Souihi, and Abdelhamid Mellouk. “Distributed SDN control: Survey, taxonomy, and challenges.” IEEE Communications Surveys & Tutorials 20, no. 1 (2017): 333-354.

[23] Wang, Song, Karina Mabell Gomez, Kandeepan Sithamparanathan, and Paul Zanna. “Software defined network security framework for IoT based smart home and city applications.” In 2019 13th International Conference on Signal Processing and Communication Systems (ICSPCS), pp. 1-8. IEEE, 2019.

[24] Rbii, Emna, and Imen Jemili. “Leveraging SDN for Smart City Applications Support.” In International Workshop on Distributed Computing for Emerging Smart Networks, pp. 95-119. Springer, Cham, 2020.

[25] Park, Younghee, Hongxin Hu, Xiaohong Yuan, and Hongda Li. “Enhancing Security Education Through Designing SDN Security Labs in CloudLab.” In Proceedings of the 49th ACM Technical Symposium on Computer Science Education, pp. 185-190. 2018.

[26] Varadharajan, Vijay, Kallol Karmakar, Uday Tupakula, and Michael Hitchens. “A policy-based security architecture for software-defined networks.” IEEE Transactions on Information Forensics and Security 14, no. 4 (2018): 897-912.

[27] Malandrino, Francesco, Carla-Fabiana Chiasserini, and Claudio Casetti. “Virtualization-based evaluation of backhaul performance in vehicular applications.” Computer Networks 134 (2018): 93-104.

[28] Boudi, Abderrahmane, Ivan Farris, Miloud Bagaa, and Tarik Taleb. “Lightweight virtualization based security framework for network edge.” In 2018 IEEE Conference on Standards for Communications and Networking (CSCN), pp. 1-6. IEEE, 2018.

[29] Zacarias, Iulisloi, Janaina Schwarzrock, Luciano P. Gaspary, Anderson Kohl, Ricardo QA Fernandes, Jorgito M. Stocchero, and Edison P. De Freitas. “Employing SDN to control video streaming applications in military mobile networks.” In 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), pp. 1-4. IEEE, 2017.

[30] Du, Ping, and Akihiro Nakao. “Deep learning-based application specific RAN slicing for mobile networks.” In 2018 IEEE 7th International Conference on Cloud Networking (CloudNet), pp. 1-3. IEEE, 2018.

[31] Barakabitze, Alcardo Alex, Lingfen Sun, Is-Haka Mkwawa, and Emmanuel Ifeachor. “A novel QoE-centric SDN-based multipath routing approach for multimedia services over 5G networks.” In 2018 IEEE International Conference on Communications (ICC), pp. 1-7. IEEE, 2018.

[32] Abdulkadir, Ibrahim, Ahmed Al-Jawad, Purav Shah, Quoc-Tuan Vien, Mehmet Fatih Tuysuz, and Ramona Trestian. “AROMA: An adapt-or-reroute strategy for multimedia applications over SDN-based wireless environments.” In 2019 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), pp. 1-6. IEEE, 2019.

[33] Ren, Xiaodong, Gagangeet Singh Aujla, Anish Jindal, Ranbir Singh Batth, and Peiying Zhang. “Adaptive recovery mechanism for SDN controllers in Edge-Cloud supported FinTech applications.” IEEE Internet of Things Journal (2021).

[34] Seliuchenko, Marian, Mykola Beshley, Marian Kyryk, and Mykhailo Zhovtonoh. “Automated Recovery of Server Applications for SDN-Based Internet of Things.” In 2019 3rd International Conference on Advanced Information and Communications Technologies (AICT), pp. 149-152. IEEE, 2019.

[35] Kundrát, Jan, Josef Vojtěch, Pavel Škoda, Rudolf Vohnout, Jan Radil, and Ondřej Havliš. "Yang/netconf roadm: Evolving open dwdm toward sdn applications." Journal of Lightwave Technology 36, no. 15 (2018): 3105-3114.

[36] Herrera, Juliana Arevalo, and Jorge E. Camargo. “A survey on machine learning applications for software defined network security.” In International Conference on Applied Cryptography and Network Security, pp. 70-93. Springer, Cham, 2019.

[37] Nguyen, Quang Huy, Ngoc Ha Do, and Hai-Chau Le. “Development of a QoS Provisioning Capable Cost-Effective SDN-based Switch for IoT Communication.” In 2018 International Conference on Advanced Technologies for Communications (ATC), pp. 220-225. IEEE, 2018.

[38] Caraguay, Angel Leonardo Valdivieso, Lorena Isabel Barona Lopez, and Luis Javier Garcia Villalba. "Evolution and challenges of software defined networking." In 2013 IEEE SDN for Future Networks and Services (SDN4FNS), pp. 1-7. IEEE, 2013.

[39] Heller, Brandon, Rob Sherwood, and Nick McKeown. “The controller placement problem.” ACM SIGCOMM Computer Communication Review 42, no. 4 (2012): 473-478.

[40] di Lallo, Roberto, Federico Griscioli, Gabriele Lospoto, Habib Mostafaei, Maurizio Pizzonia, and Massimo Rimondini. “Leveraging SDN to monitor critical infrastructure networks in a smarter way.” In 2017 IFIP/IEEE symposium on integrated network and service management (IM), pp. 608-611. IEEE, 2017.

[41] OpenFlow Management and Configuration Protocol (OF-Config 1.1.1), https://opennetworking.org/wp-content/uploads/2013/02/of-config-1-1-1.pdf

[42] Jarschel, Michael, Thomas Zinner, Tobias Hoßfeld, Phuoc Tran-Gia, and Wolfgang Kellerer. "Interfaces, attributes, and use cases: A compass for SDN." IEEE Communications Magazine 52, no. 6 (2014): 210-217.

[43] N. Handigol, S. Seetharaman, M. Flajslik, N. McKeown, and R. Johari, ‘‘Plug-n-serve: Loadbalancing web traffic using OpenFlow,’’ 2009.

[44] M. R. Çelenlioğlu and H. A. Mantar, “Energy aware adaptive resource management model for software - defined networking - based service provider networks,” IET Networks, vol. 10, no. 2, pp. 88–100, Jan. 2021, doi: 10.1049/ntw2.12006.

[45] Gudipati, D. Perry, L. E. Li, and S. Katti, “SoftRAN,” presented at the the second ACM SIGCOMM workshop, 2013, doi: 10.1145/2491185.2491207.

[46] M. Casado, T. Garfinkel, A. Akella, M. J. Freedman, D. Boneh,N. McKeown, and S. Shenker, “SANE: a protection architecture for enterprise networks,” in Proceedings of the 15th conference on USENIX Security Symposium - Volume 15, ser. USENIX-SS’06, Berkeley, CA,USA, 2006.

[47] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky and S. Uhlig, “Software-Defined Networking: A Comprehensive Survey,” in Proceedings of the IEEE, vol. 103, no. 1, pp. 14-76, Jan. 2015, doi: 10.1109/JPROC.2014.2371999.

[48] F. Ernawan, A. Aminuddin, D. Nincarean, M. F. A. Razak, and A. Firdaus, “Three Layer Authentications with a Spiral Block Mapping to Prove Authenticity in Medical Images,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 4, 2022, doi: 10.14569/IJACSA.2022.0130425.

[49] A. Aminuddin, “Android Assets Protection Using RSA and AES Cryptography to Prevent App Piracy,” *2020 3rd Int. Conf. Inf. Commun. Technol. ICOIACT 2020*, pp. 461–465, Nov. 2020, doi: 10.1109/ICOIACT50329.2020.9331988.

[50] A. Aminuddin and F. Ernawan, “AuSR2: Image watermarking technique for authentication and self-recovery with image texture preservation,” *Comput. Electr. Eng.*, vol. 102, p. 108207, Sep. 2022, doi: 10.1016/J.COMPELECENG.2022.108207.

[51] O. J. Ibrahim and W. S. Bhaya, “Intrusion Detection System for Cloud Based Software-Defined Networks,” J. Phys.: Conf. Ser., vol. 1804, no. 1, p. 012007, Feb. 2021, doi: 10.1088/17426596/1804/1/012007.

[52] A. Abdulghaffar, A. Mahmoud, M. Abu-Amara, and T. Sheltami, “Modeling and Evaluation of Software Defined Networking Based 5G Core Network Architecture,” IEEE Access, vol. 9, pp. 10179–10198, 2021, doi: 10.1109/access.2021.3049945.

[53] A. Rahman et al., “SmartBlock-SDN: An Optimized Blockchain-SDN Framework for Resource Management in IoT,” in IEEE Access, vol. 9, pp. 28361-28376, 2021, doi: 10.1109/ACCESS.2021.3058244.

[54] Q. Liu et al., “Cluster-based flow control in hybrid software-defined wireless sensor networks,” Computer Networks, vol. 187, p. 107788, Mar. 2021, doi: 10.1016/j.comnet.2020.107788.

[55] M. Rahouti, K. Xiong and Y. Xin, “Secure Software-Defined Networking Communication Systems for Smart Cities: Current Status, Challenges, and Trends,” in IEEE Access, vol. 9, pp. 12083-12113, 2021, doi: 10.1109/ACCESS.2020.3047996.

[56] N. Saha, S. BERA and S. Misra, “Sway: Traffic-Aware QoS Routing in Software-Defined IoT,” in IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 1, pp. 390-401, 1 Jan.-March 2021, doi: 10.1109/TETC.2018.2847296.