Scientific Research Publishing

Engineering, 2025, 17(1), 19-29
https://www.scirp.org/journal/eng
ISSN Online: 1947-394X
ISSN Print: 1947-3931

Software Defined Networks: Strengths, Weaknesses, and Resilience to Failures

Wendnéso Aïda Ouedraogo Rakissaga[1], Hamidou Harouna Omar[2], Pegdwindé Justin Kouraogo[1]

[1]LAMI, Joseph Ki-Zerbo University, Ouagadougou, Burkina-Faso
[2]ISIG, Aube Nouvelle University, Ouagadougou, Burkina-Faso
Email: aida.rakissaga@ujkz.bf, hamidou.oh@gmail.com, kouraogo@gmail.com

How to cite this paper: Ouedraogo Rakissaga, W.A., Omar, H.H. and Kouraogo, P.J. (2025) Software Defined Networks: Strengths, Weaknesses, and Resilience to Failures. *Engineering*, **17**, 19-29.
https://doi.org/10.4236/eng.2025.171002

Received: December 3, 2024
Accepted: January 14, 2025
Published: January 17, 2025

Abstract

This article examines the architecture of software-defined networks (SDN) and its implications for the modern management of communications infrastructures. By decoupling the control plane from the data plane, SDN offers increased flexibility and programmability, enabling rapid adaptation to changing user requirements. However, this new approach poses significant challenges in terms of security, fault tolerance, and interoperability. This paper highlights these challenges and explores current strategies to ensure the resilience and reliability of SDN networks in the face of threats and failures. In addition, we analyze the future outlook for SDN and the importance of integrating robust security solutions into these infrastructures.

Keywords

Software Defined Networking (SDN), SDN Architecture, Fault Tolerance, Network Security, Programmability, Interoperability, Communication Infrastructures

1. Introduction

Software-defined networking (SDN) has emerged as a revolutionary architecture that promises to transform how networks are designed, managed, and secured. Unlike traditional network infrastructures, where hardware and software are tightly integrated, SDN separates the control plane from the data plane, enabling centralized and programmatic management of network resources. This flexible and dynamic architecture fosters innovation and network performance optimization while introducing unique challenges, particularly security.

McKeown's pioneering paper on the topic in 2009 [1], SDNs have seen increasing adoption across industries, from data centers to service provider networks. A

fundamental aspect of SDNs is the use of the OpenFlow protocol, which allows controllers to remotely manage network hardware [2]. Many studies, such as that of Kreutz *et al.* [3], have highlighted the benefits of SDNs, including their ability to improve the responsiveness and efficiency of network operations.

However, this architecture is not free of challenges, especially regarding security. SDN networks are exposed to a wide range of threats, including DDoS attacks, which can target the central controller and compromise the integrity and availability of the network [4] [5]. Several works have examined techniques for detecting and mitigating these attacks, highlighting the need for robust strategies to protect SDN controllers [6] [7].

SDN security research has evolved to include diverse solutions, ranging from intrusion detection systems to adaptive defense mechanisms [8] [9]. Recent studies have focused on optimizing controller placement to improve resilience and fault tolerance, integrating advanced optimization algorithms to balance load and minimize latency [10] [11].

Fault tolerance has become a crucial aspect in the design of SDN systems, due to their reliance on centralized controllers. Research such as Hsieh *et al.* [12] and Kandoi [13] focus on the importance of establishing backup and redundancy mechanisms to ensure service continuity even in the event of a failure.

In this context, the objective of this study is to explore and evaluate the security challenges associated with SDN networks, focusing on DDoS attacks and proposing innovative solutions to enhance the security and resilience of these systems. The results of this research will contribute to enriching the body of knowledge on SDN and propose practical strategies for the secure implementation of this promising technology.

This article is structured as follows: first, we will review the SDN architecture and its key components. Then, we will discuss the strengths and weaknesses of this technology, followed by an analysis of the specific challenges it faces. Finally, we will address the issue of fault tolerance in SDN networks, presenting protection and recovery strategies, before concluding with prospects for improvement.

2. State of the Art on SDN Architecture and Fault Tolerance

With the emergence of software-defined networks (SDN), the way network infrastructures are designed and managed has evolved considerably. SDN architecture has become an important research topic, with an emphasis on network flexibility, programmability, and resilience. Several works have thus addressed the issue of SDN architecture [4] [5].

2.1. SDN Architecture

The SDN architecture as presented in **Figure 1** is composed of three parts:

- Application Layer: Located at the top of the architecture, this layer manages all business and security applications. It includes essential software services

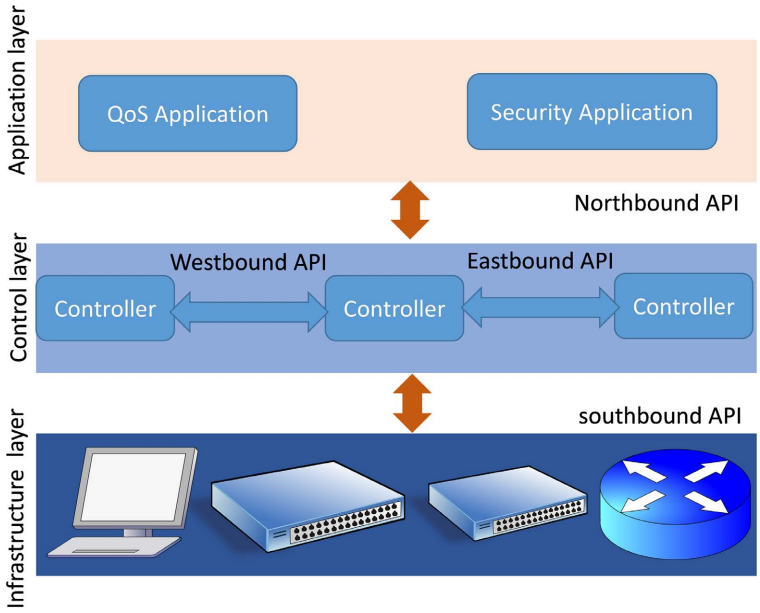

Figure 1. SDN network architecture.

such as routing, quality of service, load balancing, as well as intrusion detection and prevention systems (IDS/IPS), and mobility management.

- Control Layer: This layer includes the Network Operating System (NOS), also known as the network controller. The controller, which operates logically in a centralized manner, is responsible for managing the entire network and making decisions about routing, forwarding, and dropping packets through programming. It operates in a logically centralized and physically distributed environment, with east-west and north-south interfaces for communication.
- Data Layer: Its main function is to forward packets according to the policies and rules defined by the controller. It consists of physical network devices such as switches, routers, and access points, as well as virtual switches (OpenvSwitch, Indigo, Pica8, Nettle, OpenFlow, etc.).

2.2. Maintaining Fault Tolerance in SDN

Fault tolerance in Software-Defined Networking (SDN) is a fundamental requirement for ensuring the continuity and resilience of network services in the presence of failures [14] [15]. It can be defined as the network's ability to maintain its operational functionality despite disruptions. This involves the integration of several key components: the data plane, which must remain resilient to failures in links and switches; the control plane, which must be continuously available to perform centralized management and decision-making; and the management plane, which relies on software with minimal vulnerabilities to ensure smooth operation.

Fault tolerance strategies in SDN networks are typically classified into two broad categories: proactive and reactive approaches. Proactive approaches seek to prevent failures before they impact the system, often through strategies such as the strategic placement of controllers [16]-[18], switch migration [17], and the pre-configuration

of backup paths within the network [19] [20]. Reactive methods, on the other hand, address faults once they have occurred. A notable example of reactive defense is Moving Target Defense (MTD), which dynamically alters the network's attack surface to prevent attackers from exploiting accumulated knowledge [21].

SDN's inherent flexibility and programmability offer distinct advantages in terms of dynamic reconfiguration and network management. However, these benefits necessitate the integration of robust fault tolerance mechanisms to ensure the availability and security of the network in the event of failures. Table 1 provides an overview of key contributions to fault tolerance in SDN, including both proactive and reactive approaches.

While SDN offers substantial gains in network flexibility and programmability, the need for fault-tolerant mechanisms is critical for maintaining service reliability, particularly in large-scale networks. Ongoing research continues to explore novel strategies for improving both the resilience and security of SDN networks, aiming to mitigate vulnerabilities while enhancing performance and scalability.

Table 1. Overview of fault tolerance approaches in SDN networks.

Reference	Layer Studied	Approach	Contribution	Benefits	Disadvantages
[16] [17]	Control Plane	Proactive: Controller Placement	Strategic placement of controllers to minimize latency and maximize resilience.	Improved network efficiency and reliability	Complexity in optimal placement algorithms
[17]	Data Plane	Proactive: Switch Migration	Enables seamless migration of switches to mitigate risks of failure.	Enhances system flexibility and failure response	High resource consumption during migration
[19] [20]	Data Plane	Proactive: Backup Paths	Pre-configures backup paths to ensure continuity in case of link or switch failures.	Reduces downtime and increases fault tolerance	Increased overhead due to pre-configured paths
[21]	Data Plane	Reactive: Moving Target Defense (MTD)	Dynamically modifies the network attack surface to invalidate attacker information.	Reduces vulnerability to targeted attacks	Complexity in maintaining dynamic updates
[22]	Control Plane	Protection and Recovery	Combines SCIT (Self-Cleansing Intrusion Tolerance) with MTD to enhance intrusion tolerance.	Reduces attack success rates	High complexity in parameter settings
[23]	Control Plane	Recovery	Introduces a Stationary Agent (SA) into the SDN controller to improve communication and fault tolerance.	Ensures reliable and effective communication	Limited scalability for large SDN networks
[24]	Control Plane	Recovery	Proposes the General Multi-Controller Dynamic Agreement (GDMCA) protocol for Byzantine fault tolerance.	Reduces the number of required controllers	High overhead due to message exchange

3. Methodology

This study aims to explore the challenges and solutions related to security in

software-defined networks (SDN), particularly with regard to distributed denial of service (DDoS) attacks and fault tolerance. The adopted methodology is based on a literature search approach, a comparative analysis, and a proposal of solutions based on the results of previous studies.

3.1. Documentary Research

An extensive literature search was conducted consulting a variety of academic and technical sources to establish a solid theoretical framework. The selected articles address, among other things, the strong and weak points of SDN, the question of security, the comparison of the architecture of SDN to that of traditional networks, the threats and problems linked to the architecture of the research previous efforts aimed at finding security solutions.

The documentary research cover various aspects of SDN, including:

- The architecture and operation of SDN networks [1] [3] [4].
- Associated security challenges, including DDoS attacks and potential vulnerabilities in SDN controllers [5] [6] [12] [25].
- Existing approaches and techniques for attack detection and mitigation, as well as fault tolerance solutions [12] [21] [22] [26].

3.2. Comparative Analysis

To evaluate the effectiveness of different methods for detecting and mitigating DDoS attacks in SDN environments, a comparative analysis was conducted. This analysis includes:

- Identification of best practices: By reviewing solutions proposed in the literature, such as the use of distributed controllers to improve resilience [16] [19], as well as dynamic defense mechanisms [21].
- Assessment of strengths and weaknesses: Each approach was evaluated based on criteria such as complexity, scalability, and ability to adapt to different network topologies [3] [24] [27].

4. Results and Discussions: Strengths and Weaknesses of the SDN

In this section, we present the results of our study of the main strengths and limitations of Software-Defined Networking (SDN), resulting from the analysis of previous works. The identified strengths and weaknesses constitute the basis for considering future improvements and research directions.

4.1. SDN Forces

The results show that SDN has several significant advantages, as detailed in **Table 2**.

The results thus confirm that the programmability and centralization of SDN offer gains in terms of flexibility and simplified management, particularly useful in large-scale networks. However, these advantages are partially compromised by

vulnerabilities, notably due to the centralization of the controller.

4.2. Weaknesses of SDN

Our results also reveal several important technical challenges related to SDN architecture, summarized in Table 3.

The analyses highlight the need for scalable and security-enhanced solutions to address current SDN weaknesses, particularly in the face of security threats and interoperability issues. The identified weaknesses, although significant, pave the

Table 2. Advantages and research opportunities in SDN.

Strong point	Description	Impact observed	Weakness associated	Research opportunity	Reference
Separation of control and data plane	SDN decouples data flow management (data plane) from decision-making (control plane).	Significant improvement in network programmability and flexibility, enabling better responsiveness to configuration changes.	This separation introduces potential vulnerabilities between layers, exposing the network to increased risks of compromise.	Studies to strengthen security protocols between control and data planes to minimize the risk of compromise.	[6]
Centralization of control	The centralized controller simplifies overall network management and monitoring.	Reduced management and resource optimization time, facilitating centralized administration of network policies.	The centralized controller constitutes a single point of failure, posing a risk to service continuity.	Design of resilient distributed controllers to improve the robustness of SDN networks against critical failures.	[6]
Network programmability	Using APIs allows dynamic and customized configuration of network policies.	Increased flexibility in configuration management, facilitating updates and rapid response to incidents.	API management can introduce additional complexity and pose security risks, especially with insufficiently protected interfaces.	Development of adaptive security mechanisms for APIs to limit potential attack vectors and improve the interface.	[7]

Table 3. Technical challenges and research opportunities in SDN.

Challenge	Description	Impact observed	Research opportunity	Reference
Single point of failure	Centralizing the control plane exposes the network to increased vulnerability in the event of a controller failure.	Increased risk of cascading failures, which can seriously affect network availability and reliability.	Multi-controller architectures with redundancy and automatic failover mechanisms.	[3] [11]
Lack of standardization of open APIs	The lack of standards for open APIs hampers interoperability between SDN solutions.	Inter-device integration and communication, limiting possibilities for customization and scalability.	Development of standardized protocols to harmonize APIs and facilitate interoperability between SDN devices and infrastructures.	[5]
Vulnerability to DOS/DDOS attacks	OpenFlow switches are particularly susceptible to denial-of-service attacks, potentially saturating SDN controllers.	High risk of bottlenecks, limiting operational efficiency, and increasing network latency.	Designing new attack detection and mitigation strategies, including the integration of AI-based systems to better identify and respond to threats.	[27] [28]

way for research focused on resilient and secure solutions for next-generation SDN networks.

5. Results and Discussions: Strengths and Weaknesses of the SDN

The analysis of the strengths and weaknesses of the SDN architecture has identified key elements that contribute to its performance, but also limitations that hinder its large-scale adoption in sensitive environments. This section explores the implications of the results obtained and the main challenges that SDN must face to become a robust and scalable network solution.

5.1. Discussion

The distinctive advantages of SDN—including separation of the control plane and data plane, centralized management, and network programmability—represent fundamental changes over traditional networks. These features provide increased flexibility [3] to adapt and optimize network behavior, meeting dynamic needs such as bandwidth management and threat isolation. In particular, the ability to easily isolate a compromised host in real-time exemplifies the proactive security improvements enabled by SDN.

However, the identified weaknesses reveal substantial concerns about the security and reliability of this architecture. The centralization of the control plane, while advantageous for visibility and management, creates a single point of failure that threatens the resilience of the network in the event of a controller failure or attack. This vulnerability could have serious consequences, especially in critical infrastructures where service continuity is essential. Multi-controller solutions [11] could offer an answer, but they in turn pose challenges in terms of management complexity and inter-controller communication.

Programmability and the use of open APIs, while promoting interoperability and customization, also introduce potential vulnerabilities that can be exploited by sophisticated attacks. Secure API management and the development of common standards are therefore necessary to minimize vulnerability risks and promote more secure adoption of SDN in multi-vendor environments.

5.2. Challenges

Adopting SDN on a broader scale requires solving several technical and organizational challenges:

- Resilience and Fault Tolerance: The centralization of the control plane in SDN leads to a risk of cascading failure in the event of an incident at the controller level. Implementing redundancy solutions, such as distributed controllers and multi-controller architectures, poses technical challenges related to the complexity of data synchronization and consistency between controllers, especially in multi-domain environments [11] [22] [26].
- Security against DOS/DDOS Attacks: DOS and DDOS attacks against OpenFlow

switches, which saturate controllers, pose a serious challenge to SDN security and stability. Developing advanced defense strategies, such as AI-based adaptive detection, can improve resilience, but this also requires high processing capabilities and constant monitoring of evolving attack patterns [25] [28].

- API Standardization and Interoperability: The lack of API standardization limits interoperability between devices and SDN solutions from different vendors. Implementing standardized API protocols would facilitate device integration, simplifying management and customization. However, achieving consensus among different industry players represents a coordination challenge on an international scale.
- Management Complexity in Multi-Controller Environments: While multi-controller solutions can mitigate single-point-of-failure issues, they introduce increased complexity in management and synchronization. Implementing reliable protocols for information sharing and load balancing between controllers is essential to avoid network fragmentation and maintain consistency in routing and flow management [16] [17].
- Programmability and Customization Challenges: SDN programmability, while beneficial for customizing and updating network policies, can be a source of vulnerabilities if API interfaces are not adequately secured. Designing secure API interfaces, combined with ongoing administrator education on the risks associated with customization, is crucial to maximizing security while maintaining the flexibility of SDN.

5.3. Research Perspectives

Future research should focus on hybrid solutions and multi-tier SDN architectures to ensure better fault tolerance and enhanced security. Furthermore, integrating adaptive defense mechanisms, such as Moving Target Defense, could enhance resilience against attacks while maintaining operational flexibility. Finally, efforts in API standardization could facilitate interoperability and encourage broader adoption of SDN in complex environments.

6. Conclusions

This article has explored in depth the advantages and limitations of SDN architecture, highlighting its key strengths such as flexibility, programmability, and centralized management. We analyzed how these features provide optimized resource management, dynamic control, and better threat response while facilitating automation and customization of network policies. However, our study also highlighted weaknesses, including the single point of failure introduced by control plane centralization, vulnerability to DOS/DDOS attacks, and interoperability challenges due to lack of API standardization.

Among the most significant findings, the analysis shows that while SDN networks are inherently more adaptive and responsive, their security and resilience remain improvable, especially in critical environments. Multi-controller solutions,

while promising, add additional complexity, and the risks associated with open API programming highlight the need for stronger security and standardization in the SDN ecosystem.

Future research opportunities include developing effective redundancy mechanisms to mitigate the risk of single points of failure, as well as improving defenses against DDOS attacks. Furthermore, implementing standardized protocols for SDN APIs could enhance interoperability and simplify the integration of different solutions. Finally, research on hybrid SDN architectures, which combine distributed controllers and enhanced programmability, could pave the way for more resilient, secure networks that are fit for tomorrow's demands.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Greene, K. (2009) Software-Defined Networking: Nick McKeown Believes That Remotely Controlling Network Hardware with Software Can Bring the Internet up to Speed. *Technology Review* (*Cambridge*, *Mass.*).

[2] Bruyere, M., *et al.* (2019) The Openflow Faucet Controller. JRES.

[3] Kreutz, D., Ramos, F.M.V., Esteves Verissimo, P., Esteve Rothenberg, C., Azodolmolky, S. and Uhlig, S. (2015) Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, **103**, 14-76. https://doi.org/10.1109/jproc.2014.2371999

[4] Choukri, I., *et al.* (2019) Software Defined Networking (SDN): State of the Art. *Conference on Connected Objects and Systems*, Casablanca.

[5] Shaghaghi, A., *et al.* (2018) Software-Defined Network (SDN) Data Plane Security: Issues, Solutions, and Future Directions. *Cluster Computing Journal*.

[6] Ubale, T. and Jain, A.K. (2020) Survey on DDoS Attack Techniques and Solutions in Software-Defined Network. In: Gupta, B.B., *et al.*, Eds., *Handbook of Computer Networks and Cyber Security*: *Principles and Paradigms*, Springer International Publishing, 389-419. https://doi.org/10.1007/978-3-030-22277-2_15

[7] Yan, Q., Yu, F.R., Gong, Q. and Li, J. (2016) Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges. *IEEE Communications Surveys & Tutorials*, **18**, 602-622. https://doi.org/10.1109/comst.2015.2487361

[8] Ubale, T. and Jain, A.K. (2018) Taxonomy of DDoS Attacks in Software-Defined Networking Environment. In: Singh, P.K., *et al.*, Eds., *Futuristic Trends in Network and Communication Technologies*, Springer, 278-291. https://doi.org/10.1007/978-981-13-3804-5_21

[9] Open Networking Foundation (2015) Open Networking Specifications 1.5.1, Vol. 3.

[10] Kandoi, R. and Antikainen, M. (2015) Denial-of-Service Attacks in OpenFlow SDN Networks. 2015 *IFIP/IEEE International Symposium on Integrated Network Management* (*IM*), Ottawa, 11-15 May 2015, 1322-1326. https://doi.org/10.1109/inm.2015.7140489

[11] Yao, G., Bi, J. and Guo, L.Y. (2013) On the Cascading Failures of Multi-Controllers in Software Defined Networks. 2013 21*st IEEE International Conference on Network Protocols* (*ICNP*), Goettingen, 7-10 October 2013, 1-2.

https://doi.org/10.1109/icnp.2013.6733624

[12] Singh, J. and Behal, S. (2020) Detection and Mitigation of DDoS Attacks in SDN: A Comprehensive Review, Research Challenges and Future Directions. *Computer Science Review*, **37**, Article ID: 100279. https://doi.org/10.1016/j.cosrev.2020.100279

[13] Joëlle, M.M. and Park, Y. (2018) Strategies for Detecting and Mitigating DDoS Attacks in SDN: A Survey. *Journal of Intelligent & Fuzzy Systems*, **35**, 5913-5925. https://doi.org/10.3233/jifs-169833

[14] Petroulakis, N.E., Spanoudakis, G. and Askoxylakis, I.G. (2017). Fault Tolerance Using an SDN Pattern Framework. *GLOBECOM* 2017—2017 *IEEE Global Communications Conference*, Singapore, 4-8 December 2017, 1-6. https://doi.org/10.1109/glocom.2017.8254082

[15] Aly, W.H.F. (2019) Generic Controller Adaptive Load Balancing (GCALB) for SDN Networks. *Journal of Computer Networks and Communications*, **2019**, Article ID: 6808693. https://doi.org/10.1155/2019/6808693

[16] Radam, N.S., Al-Janabi, S. and Shaker, K. (2022) Optimisation Methods for the Controller Placement Problem in SDN: A Survey. *Webology*, **19**, 3130-3149. https://doi.org/10.14704/web/v19i1/web19207

[17] Ramya, G. and Manoharan, R. (2020) Enhanced Optimal Placements of Multi-Controllers in SDN. *Journal of Ambient Intelligence and Humanized Computing*, **12**, 8187-8204. https://doi.org/10.1007/s12652-020-02554-2

[18] Singh, G.D., Tripathi, V., Dumka, A., Rathore, R.S., Bajaj, M., Escorcia-Gutierrez, J., *et al.* (2024) A Novel Framework for Capacitated SDN Controller Placement: Balancing Latency and Reliability with PSO Algorithm. *Alexandria Engineering Journal*, **87**, 77-92. https://doi.org/10.1016/j.aej.2023.12.018

[19] Lakhani, G. and Kothari, A. (2020) Fault Administration by Load Balancing in Distributed SDN Controller: A Review. *Wireless Personal Communications*, **114**, 3507-3539. https://doi.org/10.1007/s11277-020-07545-2

[20] Al-Tam, F. (2019) Fault Administration by Load Balancing in Distributed SDN Controller: A Review. *IEEE Access*, Springer.

[21] Narantuya, J., Yoon, S., Lim, H., Cho, J., Kim, D.S., Moore, T., *et al.* (2019) SDN-Based IP Shuffling Moving Target Defense with Multiple SDN Controllers. 2019 49*th Annual IEEE/IFIP International Conference on Dependable Systems and Networks—Supplemental Volume* (*DSN-S*), Portland, 24-27 June 2019, 15-16. https://doi.org/10.1109/dsn-s.2019.00013

[22] Sanoussi, N., Chetioui, K., Orhanou, G. and El Hajji, S. (2023) ITC: Intrusion Tolerant Controller for Multicontroller SDN Architecture. *Computers & Security*, **132**, Article ID: 103351. https://doi.org/10.1016/j.cose.2023.103351

[23] Mbodila, M. (2022) Towards Fault Tolerance Management Systems in SDN. *International Conference on Intelligent and Innovative Computing Applications*, Vol. 2022, 302-314. https://doi.org/10.59200/iconic.2022.033

[24] Hsieh, H.-C., Chiang, M.-L. and Chang, T.-Y. (2021) Improving the Fault-Tolerance of Software-Defined Networks with Dynamic Overlay Agreement. *Cluster Computing*, **24**, 2597-2614.

[25] Bawany, N.Z., Shamsi, J.A. and Salah, K. (2017) DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions. *Arabian Journal for Science and Engineering*, **42**, 425-441. https://doi.org/10.1007/s13369-017-2414-5

[26] Yamansavascilar, B., Baktir, A.C., Ozgovde, A. and Ersoy, C. (2020) Fault Tolerance in SDN Data Plane Considering Network and Application Based Metrics. *Journal of*

Network and Computer Applications, **170**, Article ID: 102780.
https://doi.org/10.1016/j.jnca.2020.102780

[27] Kreutz, D., Ramos, F.M.V. and Verissimo, P. (2013) Towards Secure and Dependable Software-Defined Networks. *Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, Hong Kong, 16 August 2013, 55-60. https://doi.org/10.1145/2491185.2491199

[28] Dover, J. (2013) A Denial of Service Attack against the Open Floodlight SDN Controller. No. Tech. Rep., Dover Networks.