REVIEW OF FIREWALL APPLICATIONS IN MULTI-CONTROLLER-BASED SOFTWARE-DEFINED NETWORKS

Dipak Khatri[1]*, Bishnu Prasad Gautam[2], Kazuhiko Sato[1]

[1] Muroran Institute of Technology, 27-1 Mizumoto-cho, Muroran, Hokkaido 050-8585, Japan

[2] Kanazawa Gakuen University, 10 Suemachi, Kanazawa, Ishikawa 920-1392, Japan

Abstract

A firewall is the defensive guard of a network. It is an application that has sets of rules configured on it to prevent the network from unwanted viruses and attacks. Multi-controller techniques ensure that the firewall applications in a network function consistently. The working mechanism of a software-defined network (SDN) is based on the control and data planes. The data plane forwards packets to the targeted destinations. The control plane is the brain of an SDN. It plays a vital role in policy creation and its implementation according to defined rules or by creating new rules. Researchers around the world are working on the development of firewall applications to mitigate attacks and critical data losses (data theft and data loss) in SDNs. During this review study, we discovered that almost every research on SDN firewall applications has used a threshold limit for the number and size of packets. Conversely, very little research has been conducted on the multi-controller approach. However, there are the disadvantages of identifying packet information on a layer basis and a lack of network availability. Without packet-type information, it is difficult to identify an attacker. We compared and analyzed different methods of firewalls for securing SDNs by studying approximately 75 different studies related to SDN security. Furthermore, we provided a detailed overview and techniques for SDN protection.

Keywords: *Software-Defined Network, Multi-Controller, Firewall Application, Packet, Attack, Data plane, Control plane*

1. INTRODUCTION

A software-defined network (SDN) is a novel technology in the field of computer networking. The term “SDN” is defined as a collection of sets of programs that determine the functionality of a network system. Advances in communication technology have resulted in a large number of internet users. With the advancement of technology in the field of communication, users can communicate between miles of distance within milliseconds. The Internet is widely used by various organizations, which has created a large volume of data traffic. Only a well-managed network system can handle such large data. A well-managed network is a network system that has various services such as flexible, efficient, and manageable networking solutions that can meet the demands of dynamic businesses. Monitoring the massive data in a network system is a critical task in this era. Most developed countries are adopting the SDN architecture to manage the huge flow of data. The traditional base network architecture is being replaced by the SDN network architecture in consideration of its complexity of configuration. For instance, if an error is detected in a traditional network, a network engineer must troubleshoot each device for errors. Conversely, the SDN network architecture has a controller, which is the brain of the network and handles the whole network according to a set of programs.

An SDN is a framework that divides operations into control and data plane operations. SDN switches perform data plane functions, which help forward packets to the next device. Control plane functions are managed by the controller. The primary function of the controller is to determine the route for packet forwarding and update the flow table in SDN switches. The OpenFlow protocol facilitates communication between the control and data planes. In the SDN architecture, applications provide dynamic instructions to the controller, and the controller updates these instructions to the data plane according to programs configured in the applications. Firewalls are a key requirement for network security, and firewall applications are a set of programs created on different types of SDN controllers

(frameworks). They deny and permit services between trusted and untrusted networks to prevent unauthorized access and protect against cyber threats and data breaches. When a single-controller fails, firewall applications in multi-controller architecture ensure the availability of security services in the network. The deployment of multi-controller-based SDN architecture, where multiple controllers are used to manage a large volume of data on a single network, has further overcome the security challenges faced by traditional firewalls.

This literature review focuses on the current developments in firewall applications for multi-controller-based SDNs. With firewalls playing such a crucial role in securing these types of networks, it is important to assess various firewall options available and how they perform. This review will analyze recent research published within the last seven years to provide a comprehensive understanding of the advancements in firewall applications for multi-controller-based SDNs. We aim to provide useful insights for researchers, professionals, and anyone with an interest in the field by providing a comprehensive overview of firewall applications in multi-controller-based SDNs. In addition, this study proposes directions for future research and identifies areas that need improvement to further advance the field.

This paper is organized into seven sections. The first and second sections introduce and discuss SDNs, firewall applications, and the review process, respectively. In Section 3, we discuss the overview, types, and classification of firewall applications in SDNs, and Section 4 compares the single-controller and multi-controller architectures of SDNs. Section 5 evaluates the performance of firewall applications in multi-controller-based SDNs.. The sixth section focuses on the challenges and limitations of firewall applications in multi-controller-based SDNs. In the final section, we conclude the review and provide future research directions (Figure 1).

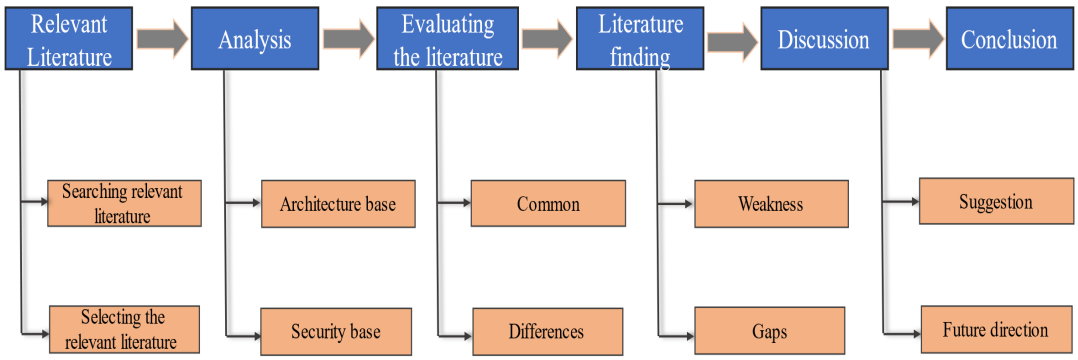

Fig. 1. The workflow of the review process

2. THE REVIEW PROCESS

In this review, we have compiled various pertinent studies on firewall applications in multi-controller-based SDNs from different sources such as conference proceedings, academic databases, and relevant journals. The papers were chosen according to the publication date and relevance to the topic. Our analysis of the review focused on network architecture and security. We compared and contrasted the selected studies to identify similarities and differences. In addition, we identified weaknesses and gaps in the existing literature and discussed our findings to suggest future research directions and fill these gaps in the understanding of firewall applications in multi-controller SDNs. To conclude, we provide a summary of the most commonly utilized firewall applications in multi-controller SDNs.

During our review, we assessed 72 papers related to firewall applications used in controllers. Based on our selection criteria, including publication in peer-reviewed journals or conference proceedings, and with a focus on firewall applications used in controllers specific to multi-controller in SDNs, we finalized a set of 21 papers. Among the finalized papers, we observed a significant increase in the

number of publications related to both multi-controller and firewall applications. Our search criteria, which included keywords such as "SDN," "Firewall Applications," "Controllers," and "hacking," allowed us to narrow down our selection to papers that were most relevant to our research focus. Overall, our review provides a comprehensive understanding of the current state of research on firewall applications used in controllers, particularly in the context of multi-controller in SDNs. By analyzing the finalized papers, we could gain insights into the latest trends and developments in the field, which can be used to inform future research efforts (Figure 2).

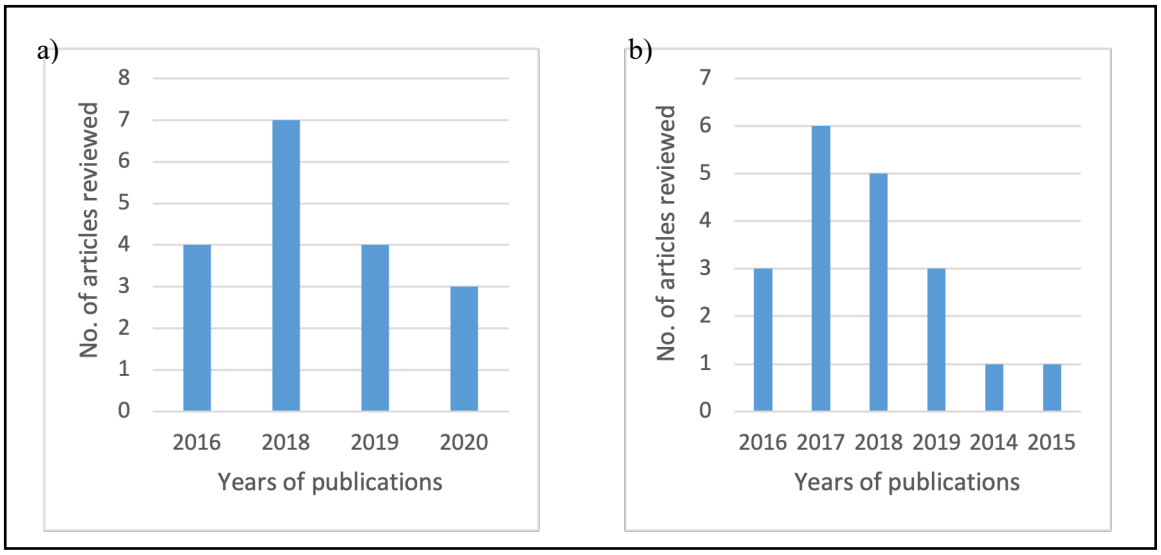

Fig. 2. The number of papers reviewed between the year by a) multi-controller and b) firewall applications

3. OVERVIEW OF FIREWALL APPLICATIONS

In recent years, there has been growing research interest in the development of firewall applications for multi-controller-based SDNs. Several studies have been conducted to assess the performance, security, and scalability of firewall applications in this context.

A firewall is designed to operate at multiple layers of the transmission control protocol/internet protocol (OSI/TCP/IP model) in a network. More precisely, it can filter packets on the transport and application layers of the TCP/IP model. Filtering at the transport layer involves considering the source and destination ports of packets. In contrast, filtering at the application layer can be achieved by examining various parameters, such as IP addresses, ports, or MAC addresses of the source/destination [1]. The experimental design conducted in this study used a single-controller architecture. However, note that a firewall does not currently incorporate an intrusion detection system (IDS), which could be a valuable feature to consider for future development. Furthermore, the scope of a firewall is confined to the transport and application layers of the TCP/IP model, and it does not cover the remaining layers of the model.

Azka et al. [2] discussed various methods for improving SDN security, including the use of security applications, secure application development frameworks, and security policies. One such framework, FRESCO, provides a secure environment for developing applications and includes a library of reusable security modules. To enforce security policies, tools such as FLOVER and NICE can be used to verify the flow and check for errors in the OpenFlow application code. An SDN firewall implements network segmentation and segregation according to security standards, using filtering techniques such as temporal and spatial filtering to minimize the vulnerability of industrial control devices. This firewall can be easily configured by network administrators through application-oriented white lists, making network management tasks easier.

Tsuchiya et al. [3] implemented an SDN firewall prototype, tested its viability, and found it to be effective in counter-attacks and configuring flat networks into virtual local area network segments. The SDN firewall used in their study provides improved security features, performance, and flexibility compared with commercially available application firewalls. The recommended prototype implementation required the implementation of several features to harness the full potential of the proposition and leverage its benefits to the fullest extent possible. A new SDN firewall system was proposed, which uses the OpenFlow protocol [4]. It has state tables and state transition rules added to the SDN switch and controller for state inspection. The firewall is tested on Floodlight and Open vSwitch, and the results show that it offers better performance than traditional packet-filtering firewalls, including recognition of packet types and fine-grained access control. The state-based firewall has a shorter delay time and improved overall performance than packet-filtering firewalls. A stateful distributed firewall (SDFW) system was proposed for a multi-tenant cloud network using an SDN [5]. This system addresses the challenge of implementing stateful firewalls in an SDN environment by tracking packet and flow states in the data plane. This system shows scalable security against data plane attacks with a minimal performance impact of a 1.6% reduction in the network bandwidth, and its limitations are limited defense against layer 4 security attacks and a lack of protection against application layer security attacks.

Table 1. Overview of advantages and disadvantages of Firewall applications

Authors	Methods	Advantages	Disadvantages
Badotra and Singh [1]	TCP/IP layer filtering	Counter-attack and VLAN configuration	Limited security scope
Azka et al. [2]	Security enforcement applications	Enhanced firewall features	Prototype incomplete
Yang and Ling [4]	OpenFlow firewall system	Advanced packet filtering	Not mentioned
Shirali-Shahreza and Ganjali [7]	FleXight SDN firewall	Effective and usable with low overhead	Challenging netFlow creation
Othman et al. [9]	POX firewall application	Policy-based packet filtering	OpenFlow v1.0 stateless

Zope et al. [6] introduced Floodlight as a framework that provides a network visibility graphical user interface and implemented it as a firewall and a Load Balancer module using a REST API, which can be monitored and control network elements. Conversely, in a recent context, the use of Internet of Things (IoT) resources has increased, posing significant security risks. Shirali–Shahrez and Ganjali [7] introduced an SDN firewall that used the FleXight platform and was very effective in identifying all attackers and 99% of vulnerable victims with very low overhead. However, creating NetFlow-like logs of network events on an enterprise network will be a challenging task. Pena and Yu [8] explored possible techniques for enhancing security in an SDN by developing a firewall prototype. The prototype leverages the features of OpenFlow, an open standard in SDNs, and was tested on a simulated network using Mininet. The results of the tests were positive, showing that the firewall prototype functions fully in a distributed configuration without any impact on network latency. Overall, the results from their study demonstrate the potential for using SDNs to improve security in networks. Othman et al. [9] Implemented firewall functionalities in SDNs by writing a firewall application that runs on the POX controller, which detects data traffic on a layer basis, including layers 2, 3, and 4 (Table 1). The firewall filters packets according to their headers and matches them against

predefined security policies. The drawback of this application is that it does not keep track of the state of connections, making it stateless because of the limitations of OpenFlow version 1.0.

The framework presented by Nife and Kotulski [10] suggested enhancing the security of SDNs by adding a security application to the controller as an extension. By leveraging the benefits of OpenFlow protocol messages, the aforementioned application examines particular flows in the network and transmits prompt security directives of the network. The proposed framework employs stateful packet-filtering by utilizing tables used to track each flow in the network, which can be easily programmed and configured through the central controller. Ajaeiya et al. [11] introduced a flow-based firewall that detects abnormal behavior of traffic flows by classifying them according to collected features (Figure 3). This approach differs from classical network-based intrusion detection and network and system management systems, as it leverages the statistics collected by OpenFlow switches to perform intrusion detection without requiring information from network hosts. The proposed approach provides an efficient and effective way to detect potential security threats, such as denial-of-service and hypertext transfer protocol brute force attacks, in the network.

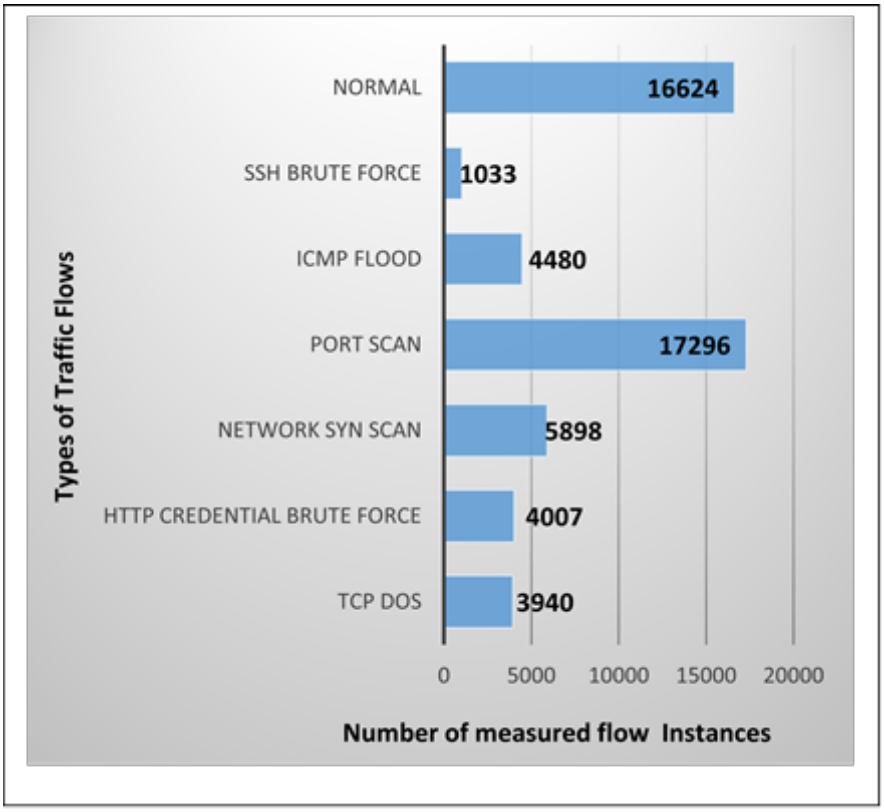

Fig 3. Classification of traffic flows from measurements [11]

Taylor et al. [12] aimed to improve the potential of SDNs for enterprise security by shifting the SDN agent to end-hosts, enabling in-depth host context and precise management of network flows. Network operators can create policies using data about the user and program for reference, leading to improved scalability and the ability to handle many network flows (Figure 4).

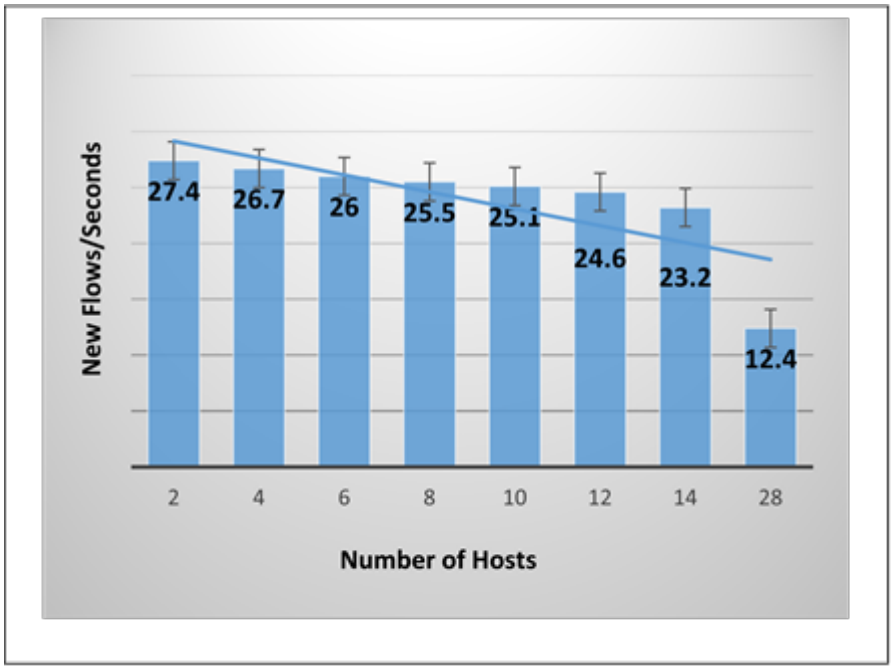

Fig. 4. New flow rate of host in seconds [12]

3.1. Types of Firewall Applications

Stateful firewalls in the SDN track the state of network connections to determine which packets are allowed or blocked based on predetermined rules. This approach is more effective than traditional firewalls, which only assess individual packets and can lead to false positives or negatives. Stateful firewall applications in SDNs are a critical component of modern network security infrastructure. In traditional networks, firewalls are deployed as standalone devices; however, in SDNs, they can be integrated into the network architecture to provide a more flexible and efficient approach to network security. One advantage of stateful firewalls in SDNs is that they can be programmed to enforce policies based on user or device identity, as well as the type of traffic. This means that access control policies can be enforced more granularly, reducing the risk of unauthorized access. Stateful firewalls in SDNs can also be used to enforce policies across multiple network domains, such as public and private clouds or remote branches. This ensures consistent security policies across the entire network, regardless of location. Stateful firewall applications are also called dynamic packet-filtering applications. This type of firewall operates at L3, L4, and L5 and monitors the state of the TCP connection as well as a sequence number. Krongbaramee and Somchit [13] proposed the implementation of a stateful firewall in the data plane of an SDN using Open vSwitch and the OpenFlow protocol version 1.5.0. The proposed implementation utilizes the learn action in Open vSwitch to modify the rules in the SDN switch, allowing the SDN stateful firewall to be implemented without additional modifications to both the control and data planes. The authors suggest detailed and in-depth studies on the performance of the SDN stateful firewall with more complex network topologies and various security attacks to assess its accuracy. Flow tracker, an innovative firewall solution that strives to uphold the precision and effectiveness of a firewall while minimizing the burden on the communication and processing of controllers between the data and control planes [14]. The proposed solution includes a novel approach for installing selective flow control rules based on topology learning through machine learning techniques, adaptive monitoring of connection states for TCP and user datagram protocol connections, and non-interfering connection tracking with minimized end-to-end delay.

Stateless firewall applications, also known as static packet-filtering or packet-filtering, resemble the access control list of traditional networks, where permit and deny rules are established on the basis of

source and destination IP address, port number, and protocol number. In the context of SDN, effective security measures are essential to address emerging and existing security threats. Abubakar and Pranggaono [15] highlighted the vulnerability of the SDN architecture to attacks and proposed the integration of an IDS to mitigate these risks. To provide scalable threat detection, the Snort IDS is deployed, and a flow-based IDS model with machine learning is developed. Their study also utilizes pattern recognition due to its high accuracy rate compared with other neural network models. Maldonado–Lopez et al. [16] presented the implementation of FireWell, an Alloy-based framework that models firewall policies as formal predicates to identify and prevent conflicts in firewall policies, which was tested using the Floodlight controller and firewall application (Figure 5). This article serves as a valuable resource for network administrators seeking to enhance the security of their SDN architecture.

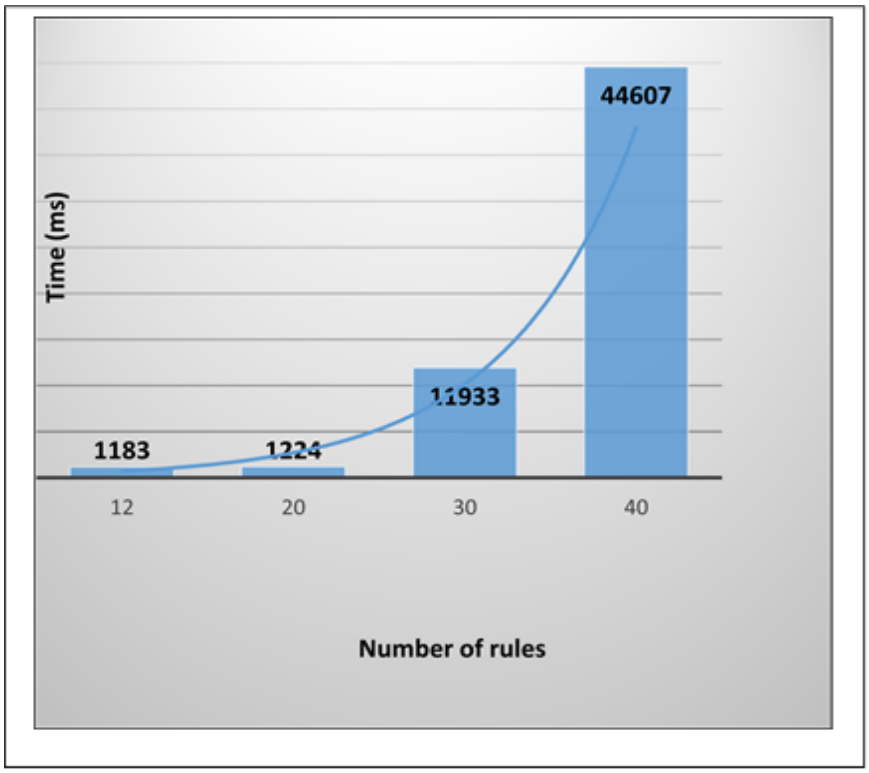

Fig. 5. Rule count's millisecond timing [16]

4. MULTI-CONTROLLER IN SDN

SDN controllers are the "brain" of SDNs and play a crucial role in managing network behavior using SDN protocols such as OpenFlow. Multi-controller architecture is a popular approach that uses different controllers to manage different parts of an SDN, improving the ability of the network to tolerate faults [17]. To address issues in SDNs, an efficient load-balancing scheme called the reliable and load balance-aware multi-controller deployment strategy has been proposed [18]. It utilizes the controller placement selection algorithm to explore reliable controller placements and multiple domain partition algorithms to allocate switches based on node attractability and the controller load-balancing rate. The communicating sequential process model is used to represent the routing service of SDNs under multi-controller architectures, and the model checker, the Process Analysis Toolkit, is used to verify the properties of the model, including deadlock freeness, consistency, and fault tolerance [19]. In addition, a new SDN architecture called the distributed rule store has been proposed by Wang et al. [20], which distributes flow rules to multiple controller instances, periodically checks rule consistency, and shows high performance in maintaining a distributed rule store. Furthermore, a dynamic clustering

algorithm has been presented to balance the load among distributed controllers in SDNs [21]. It utilizes two levels of controllers, namely, distributed controllers and a master controller, to manage and perform clustering based on the workload of the distributed controllers. This approach leads to better resource usage, reliability, and resilience. The use of a multi-controller architecture in an SDN can provide many benefits in terms of scalability, availability, and fault tolerance and is an important consideration for large-scale network deployments.

4.1. Controller Placement in SDN Architecture

An SDN separates network control from data forwarding, and the placement of controllers is a crucial aspect that can significantly affect network performance and scalability. The controller placement problem involves finding optimal locations to install SDN controllers in a network to ensure network efficiency and effectiveness while minimizing deployment costs. Various algorithms have been proposed to solve the controller placement problem in SDNs, considering factors such as network topology, traffic volume and distribution, and network reliability requirements. In this context, Chen et al. [22] proposed an approach called "community detection controller deployment" that considers the network topology of multiple communities and selects suitable positions for placing controllers in each community. The method aims to achieve a state of topological equilibrium partition by equalizing the number of switches handled by each controller and modifying the count of nodes present in the communities. Experimental results on real network topologies show that the proposed method effectively balances the load of the controller while keeping latency low. However, they suggested that minimizing costs and maximizing network resilience are also crucial factors to consider during controller deployment.

In addition, in SDNs, inter-controller communications and switch controllers are both important aspects of the control plane; however, most research has focused on switch controllers. In this regard, a novel Steiner tree-based inter-controller latency model has been proposed by Das and Gurusamy [23] to optimize the network state synchronization cost before and after single-link failures. A multi-objective integer linear programming formulation is presented to determine the associated controller placement. The approach achieves an optimal post-failure network synchronization cost in approximately 70% of cases and requires a lower post-failure network reconfiguration cost than the optimal approach. Furthermore, in the wide area network architecture, the controller placement problem is an essential issue of SDNs. Qi et al. [24] proposed a placement algorithm called Modified Density Peak Clustering (MDPC) to partition a network into sub-networks based on latency. The performance of this approach is tested on the Internet2 OS3E, and the results show that MDPC effectively reduces propagation latency, particularly in terms of average latency. The placement of SDN controllers in edge network architectures was investigated by Qin et al. [25], where a methodology for the optimal placement of controllers and node assignments is developed and found to be more effective than other existing methods (Table 2).

Table 2. Advantages and disadvantages of multi-controller architecture

Authors	Methods	Advantages	Disadvantages
Hu et al. [18]	RLMD Controller deployment	Model-based workload balancing	OpenDayLight limitation
Xiao et al. [19]	CSP Model verification	DoS Resilience, disclosure risk	Fake path Detection
Muthanna et al. [21]	Dynamic controller clustering	Load balancing, fault tolerance	Two-level controller limitation
Chen et al. [22]	Community detection deployment	Low latency load balancing	Cost-resilience controller deployment
Das and Gurusamy [23]	Latency-based placement algorithm	Reduced propagation latency	WAN architecture limitation

Moreover, Sminesh et al. [26] proposed a modified affinity propagation clustering algorithm to partition a network and identify candidate exemplars for SDN controller placement. Simulation results on standard network topologies demonstrate that the method minimizes inter-controller latency and improves controller imbalance, resulting in the optimal number and placement of SDN controllers. In addition, Tao et al. [27] proposed a new approach for optimizing the performance of SDNs by minimizing the flow request cost and achieving load-balancing among controllers. The simulation results show that the proposed controller layout scheme reduces flow request costs and ensures load-balancing among controllers. In conclusion, ongoing research in multi-controller placement in SDNs is focusing on exploring additional mechanisms for controller clustering and accounting for factors such as interference and congestion in the network. The proposed approaches and algorithms can contribute to the effective deployment of SDN controllers and improve network performance and scalability.

4.2. Method of Using Multi-Controller in SDN

The method of using a multi-controller architecture in an SDN depends on the specific needs of the network. For instance, the size, complexity, and geographic distribution of the network determine how controllers are deployed. The desired level of fault tolerance and the need for fine-grained control over network policies are also important considerations. Therefore, various studies have proposed different methods based on the unique requirements of the network.

Wang et al. [28] proposed a novel approach to enhance the scalability of multi-domain and multi-vendor networks using an SDN architecture. The proposed architecture involves a multi-controller that facilitates effective coordination among different controllers. The authors validated their approach by conducting experiments involving three different vendors, demonstrating its effectiveness in maintaining consistent network views and providing end-to-end services. Deng et al. [29] presented a novel method for selecting inter-domain transmission paths in multi-controller-based SDNs. The authors proposed a collaborative model for multiple controllers, which involves segmenting the network into several domains and synchronizing the topology via joint efforts. The proposed approach uses a path-selection algorithm to generate low-latency and high-throughput transmission paths based on the global topology and user requests. Simulation experiments have confirmed the validity of this approach, showing that it accurately handles requests and generates optimal transmission paths within a short time. Furthermore, this approach reduces the SDN controller load and enhances transmission efficiency. Wang et al. [30] presented a new approach for deploying multiple controllers in an SDN

using a Particle Swarm Optimization (PSO) algorithm. The proposed deployment method was designed on the basis of PSO with dynamic parameters that optimize the average transmission delay of the switch controller, ensuring effective deployment in a continuous two-dimensional space. The authors validated the proposed method by conducting experiments that demonstrate its effectiveness in reducing the average transmission delay of the switch controller by 10%. Moreover, the proposed method enhances the global search capability and convergence accuracy of the PSO algorithm, enabling it to optimize the deployment of multiple controllers in SDNs. These findings are expected to contribute to the development of efficient SDN architectures for future networks.

Fang et al. [31] presented a novel approach to address the controller deployment problem in an SDN through the concept of network partitioning. A multi-controller cooperative model based on a Field Programmable Gate Array (FPGA) is designed, and an improved k-means algorithm is proposed to model the network partition problem and reduce latency between controllers and switches. Experimental results on both FPGA and software platforms demonstrate the feasibility and effectiveness of the proposed algorithm, achieving a significant acceleration ratio of 14 when run on the FPGA platform. However, their study is limited to the latency performance index, and further work is required to optimize the controller deployment solution to meet load balance requirements.

4.3. Single and Multi-Controller Architecture

The SDN controller is a key component of the SDN architecture, which is responsible for managing network devices and providing network services. The controller can be deployed in either a single or multi-controller architecture. A single controller is also known as the default SDN architecture (Figure 6). Single-controller SDN architecture consists of a single controller that is responsible for managing the entire network. This architecture is simple to deploy and manage, but it can suffer from performance and scalability issues, especially in large and complex networks [32].

Multi-controller SDN architecture consists of multiple controllers that collaborate to manage the network (Figure 7). This architecture can improve performance, scalability, and reliability, as well as provide fault tolerance and load-balancing. However, it is more complex to deploy and manage, as the controllers need to coordinate with each other [33]. Single-controller SDN architecture may suffer from network congestion and single point of failure issues, whereas multi-controller SDN architecture can distribute traffic load among controllers, provide redundancy and fault tolerance, and improve performance. In terms of scalability, single-controller SDN architecture can have limitations, whereas multi-controller SDN architecture can improve scalability by dividing the network into multiple domains managed by separate controllers [34].

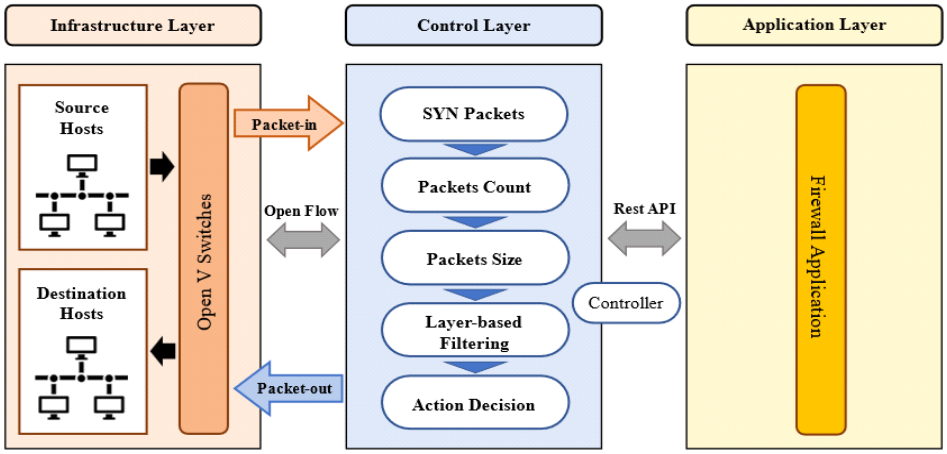

Fig. 6. Single-controller-based SDN Architecture

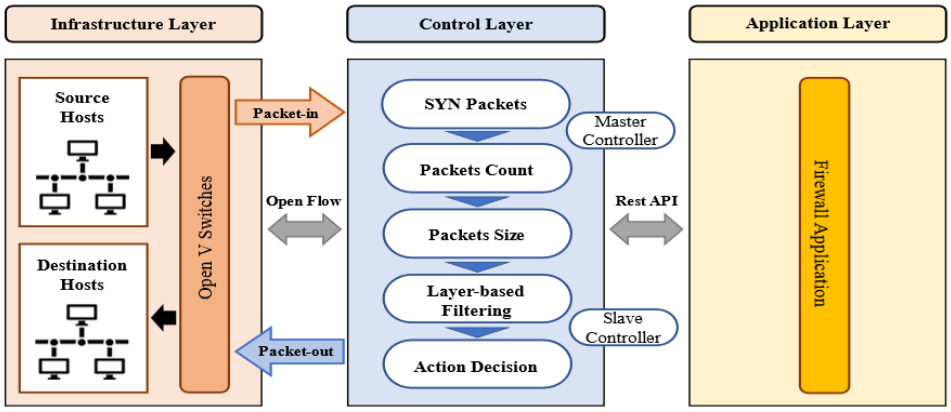

Fig. 7. Multi-controller-based SDN Architecture

5. PERFORMANCE

The evaluation of firewall applications in multi-controller-based SDNs is a research area, where several studies have been conducted to improve the scalability, reliability, and resource utilization of firewall systems in multi-controller SDN environments. Tran and Ahn [11] proposed a new firewall rule deployment strategy that leverages multiple controllers to improve the scalability and reliability of the system, and the results showed that the proposed strategy outperformed traditional centralized and distributed firewall systems in terms of scalability and reliability. Another approach for improving the performance of distributed firewall systems in multi-controller SDN environments is using deep learning-based techniques to learn the characteristics of network traffic and identify malicious traffic patterns in real time [35]. Furthermore, the performance of a distributed firewall system in a multi-controller SDN environment can be evaluated using an adaptive rule allocation strategy, and researchers have proposed a dynamic rule allocation strategy that adjusts the number of firewall rules assigned to each controller based on the network traffic load [36]. In summary, the evaluation of firewall applications in multi-controller-based SDNs is an ongoing research topic, and researchers continue to propose various approaches for improving the performance of firewall systems in multi-controller SDN environments.

6. DIRECTIONS AND FUTURE WORK

In terms of future research directions, several areas require further exploration. First, there is a need to develop more advanced firewall applications that can handle a larger volume of data traffic in multi-controller-based SDN environments. Second, research is required to develop techniques for optimizing the resource utilization of firewall applications in multi-controller-based SDNs. Third, further research is required to investigate the impact of different network topologies on the performance of firewall applications in multi-controller-based SDNs. Fourth, there is a need to develop more efficient methods for identifying and mitigating various types of cyber threats and attacks in multi-controller-based SDN environments [37]. Finally, research is required to explore the potential of using machine learning and other advanced technologies to improve the performance of firewall applications in multi-controller-based SDNs. These directions and future work should be considered by researchers when writing journal publications related to firewall applications in multi-controller-based SDNs.

7. CHALLENGES AND LIMITATIONS OF FIREWALL APPLICATIONS IN MULTI-CONTROLLER-BASED SDNS

In multi-controller-based SDN environments, firewall applications are critical components of network security. However, several challenges are associated with their implementation. One significant challenge is scalability, as managing and configuring firewall applications on multiple controllers can be time-consuming and may affect network performance. Furthermore, centralized control of firewall applications creates a single point of failure, which could lead to network vulnerabilities if the controller fails or is compromised. Limited visibility and the introduction of performance overhead are also significant challenges associated with firewall applications in multi-controller-based SDNs. Furthermore, the lack of standardization and limited interoperability can make it challenging to integrate different firewall applications seamlessly and implement end-to-end security policies across the entire network infrastructure.

In addition to these challenges, several limitations are associated with firewall applications in multi-controller-based SDNs. One limitation is the dependence on flow rules to enforce security policies, which can limit the ability to detect new threats not already defined in the policy. The performance overhead associated with processing complex security policies can also affect network performance. Furthermore, the need for customized firewall applications for different SDN environments can be time-consuming and costly. Another limitation is the lack of standardization across different firewall applications, resulting in interoperability issues with legacy network devices that do not support SDNs. This limitation can also limit the ability of organizations to implement end-to-end security policies. Finally, the limited support for legacy applications and protocols in firewall applications can create vulnerabilities in the network, making it challenging to provide comprehensive network security.

8. CONCLUSION

In conclusion, the research and development in the field of firewall applications for SDNs have made significant progress over the years. The emergence of SDN technology has provided new opportunities for enhancing network security by offering improved performance, flexibility, and security features compared with traditional firewalls. This literature review provides a comprehensive overview of firewall applications in multi-controller-based SDNs, including an introduction to SDNs and firewall applications, a classification of firewall applications in SDNs, a comparison of single-controller and multi-controller architectures, performance evaluation, challenges, limitations, and future research directions. This review has demonstrated that stateful firewalls that track the state of network connections and distributed firewalls for multi-tenant cloud networks are among the most effective firewall applications in SDNs. These applications can be programmed to enforce policies based on user or device identity, as well as the type of traffic, and can enforce policies across multiple network domains, ensuring consistent security policies across the entire network, regardless of location. This review has also identified some limitations and challenges that need to be addressed, such as

scalability issues, the need for advanced programming skills, and the lack of standardization. Overall, this review highlights the importance of further research and development in the field of firewall applications for SDNs. Future studies should focus on addressing the limitations and challenges identified in this review, as well as exploring new approaches for enhancing network security in various contexts. By providing a comprehensive understanding of the advancements made in firewall applications for multi-controller-based SDNs, this review aims to provide useful insights for researchers, professionals, and anyone with an interest in the field.

ACKNOWLEDGEMENTS

We would like express our sincere appreciation to Enago (www.enago.jp) for the English language review of this manuscript. Finally, we declare that all authors have no conflicts of interest.

REFERENCES

1. Badotra, S, Singh, J 2019, "Creating firewall in transport layer and application layer using software-defined networking", *Springer*, vol. 32. https://doi.org/10.1007/978-981-10-8201-6_11
2. Azka, W, Sathiya, R, Geetha, A 2017, "A survey of applications and security issues in software defined networking", *International Journal of Computer Network and Information Security*, vol. 3, pp. 21-28.
3. Akihiro, T, Franciso, F, Ichiro, Angel, O, Raul, P 2018, "Software-defined networking firewall for industry 4.0 manufacturing systems", *Journal of Industrial Engineering and Management (JIEM),*ISSN 2013-0953, OmniaScience, Barcelona, vol. 11, no. 2, pp. 318-333. https://doi.org/10.3926/jiem.2534
4. Yang, H, Ling, Lu 2019, "Research of inspection firewall fine-grained access control on SDN state", *Journal of Physics: Conference Series*, vol. 1395, no. 012014.
5. Chowdhary, A, Huang, D, Alshamrani, A, Sabur, A 2018, "SDFW: SDN-based stateful distributed firewall", *arXiv*. https://doi.org/10.48550/arXiv.1811.00634
6. Zope, N, Pawar, S, Saquib, Z, 2016, "Firewall and load balancing as an application of SDN", *In Proceedings of 2016 Conference on Advances in Signal Processing (CASP)*, Pune, June 9-11
7. Shirali-Shahreza, S, Ganjali, Y, 2018, "Protecting home user devices with an SDN-based firewall", in *IEEE Transactions on Consumer Electronics*, vol. 64, no. 1, pp. 92-100. Doi:10.1109/TCE.2018.2811261
8. Pena, JGV, Yu, WE, 2014, "Development of a distributed firewall using software-defined networking technology", *4th IEEE International Conference on Information Science and Technology*, Shenzhen, China, pp. 449-452. Doi:10.1109/ICIST.2014.6920514
9. Othman, WM, Chen, H, Al-moalmi, A, Hadi, AN 2017, "Implementation and performance analysis of SDN firewall on POX controller", *9th IEEE International Conference on Communication Software and Networks (ICCSN)*, Guangzhou, China, pp. 1461-1466. Doi:10.1109/ICCSN.2017.8230351
10. Nife, F, Kotulski, Z 2017, "Multi-level stateful firewall mechanism for software-defined networks", Computer Networks, vol. 718, pp. 271-286. DOI:10.1007/978-3-319-59767-6_22
11. Ajaeiya, G, Adalian, N, Elhajj, IE, Kayassi, A, Chehab, A 2017, "Flow-based intrusion detection system for SDN, *2017 IEEE Symposium on Computers and Communications (ISCC)*, Heraklion, Greece, pp. 787-793. Doi:10.1109/ISCC.2017.8024623

12. Taylor, CR, MacFarland, DC, Smestad, DR, Shue, CA 2016, "Contextual, flow-based access control with scalable host-based SDN techniques", *The 35th Annual IEEE International Conference on Computer Communications*, San Fransico, CA, USA, pp.1-9. Doi:10.1109/INFOCOM.2016.7524498
13. Krongbaramee, P, Somchit, Y 2018, "Implementation of SDN stateful firewall on data plane using open vSwitch", *15th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Nakhonpathom, Thailand, 99. 1-5. Doi:10.1109/JCSSE.2018.8457354.
14. Tran, TV, Ahn, H 2016, "FlowTrack: A SDN stateful firewall solution with adaptive connection tracking and minimized controller processing", 2016 International Conference on Software Networking (ICSN) Jeju, Korea (South), pp.1-5. Doi:10.1109/ICSN.2016.7501925
15. Abubakar, A, Pranggaono, B 2017, "Machine learning based intrusion detection system for software define networks", *2017 Seventh International Conference on Engineering Security Technologies (EST)*, Canterbury, UK, 2017, pp. 138-143. Doi:10.1109/EST.2017.8090413
16. Maldonado-Lopez, FA, calle, E, Donoso, Y 2015, "Detection and prevention of firewall-rule conflicts on software-defined networking", 2015 7th International Workshop on Reliabble Networks Design and Modeling (RNDM), Munich, Germany, pp. 259-265, doi:10.1109/RNDM.2015.7325238
17. Hai, NT, Kim, D 2016, "Efficient load balancing for multi-controller in SDN-based mission-critical networks", 2016 IEEE 14th International Conference on Industrial Informatics (INDIN), Poitiers, France, pp. 420-425. Doi:10.1109/INDIN.2016.7819196
18. Hu, T, Yi, P, Zhang, J, Lan J 2018, "Reliable and load balance-aware multi-controller deployment in SDN", in *China Communications*, vol. 15, no. 11, pp. 184-198. doi: 10.1109/CC.2018.8543099.
19. Xiao, L, Zhu, H, Xiang, S, Vinh, PC 2018, "Modeling and verifying SDN under Multi-controller architectures using CSP", *Concurrency and Computation: Practice and Experience*, no. e5334.\
20. Wang, H, Zhang, P, Xiong, L, Liu, L, Hu, C 2016, "A secure and high-performance multi-controller architecture for software-defined networking", Frontiers of Information Technology & Electronic Engineering, vol. 17, no. 7, pp. 634-646
21. Muthanna, A, Ateya, AA, Makolkina, M, Vybornova, A, Markova, E, Gogol, A, Koucheryavy, A 2018, "SDN multi-controller networks with load balanced", no. 57, pp. 1-6. https://doi.org/10.1145/3231053.3231124
22. Chen, W, Chen, C, Jiang, X, Liu, L 2018, "Multi-controller placement towards SDN based on Louvain Heuristic algorithm", in IEEE Access, vol. 6, pp. 49486-49497. Doi:10.1109/ACCESS.2018.2867931
23. Das, T, Gurusamy, M 2020, "Controller Placement for Resilient Network State Synchronization in Multi-Controller SDN", in *IEEE Communications Letters*, vol. 24, no. 6, pp. 1299-1303, doi: 10.1109/LCOMM.2020.2979072.
24. Qi, Y, Wang, D, Yao, W, Li, H, Cao, Y 2019, "Towards Multi-Controller Placement for SDN Based on Density Peaks Clustering", *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, pp. 1-6, doi: 10.1109/ICC.2019.8761814.
25. Qin, Q, Poularakis, K, Isoifidis, G, Tassiulas, L, 2018, "SDN Controller Placement at the Edge: Optimizing Delay and Overheads", *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, Honolulu, HI, USA, 2018, pp. 684-692, doi: 10.1109/INFOCOM.2018.8485963.
26. Sminesh, CN, Kanaga, EGM, Sreejish, AG 2020, "A multi-controller placement strategy in software defined networks using affinity propagation", International Journal of Internet technology and Secured Transactions, vol. 10, pp. 229-253.

27. Tao, P, Ying, C, Sun, Z, Tan, S, Wang, P, Sun, Z 2018, "The controller placement of software-defined networks based on minimum delay and load balancing", *2018 IEEE 16th Intl Conf on Dependable, Automatic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress*, Athens, Greece, pp. 310-313. Doi:10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00059

28. Wang, J, Shou, G, Hu, Y, Guo, Z 2016, "A multidomain SDN scalability architecture implementation based on the coordinate controller", *2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, Chengdu, China, pp. 494-499, doi: 10.1109/CyberC.2016.100.

29. Deng, Y, Wang, Y, He, X, Feng, H 2018, "An Inter-Domain Transmission Path Selection Approach of Multi-Controller SDN Network Based on FPGA", *2018 14th International Conference on Computational Intelligence and Security (CIS)*, Hangzhou, China, pp. 66-70, doi: 10.1109/CIS2018.2018.00023.

30. Li, Y, Sun, W, Guan, S 2020, "A Multi-controller deployment method based on PSO algorithm in SDN environment", *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chongqing, China, pp. 351-355, doi: 10.1109/ITNEC48623.2020.9084702.

31. Fang, M, Wang, Ye, M 2019, "A Multi-Controller Deployment Method of SDN Network Based on FPGA", *2019 15th International Conference on Computational Intelligence and Security (CIS)*, Macao, China, 2019, pp. 262-266, doi: 10.1109/CIS.2019.00062.

32. Blial, O, Mamoun, MB, Benaini, R 2016, "An overview on SDN architectures with multiple controllers", Journal of Computer Networks and Communications, vol. 2016, no. 9396525

33. Hossein, A, Watts, MJ, Ahmadi, K 2019, "An overview of multi-controller architecture in software-defined networking", Journal of Computer Networks and Communications, vol. 2016, no. 9396525, pp. 1-8. http://dx.doi.org/10.1155/2016/9396525

34. Zhang, Y, Cui, L, Wang, W, Zhang, Y 2018, "A survey on software-defined networking with multiple controllers," Journal of Network and Computer Applications, vol. 103, no. 2018, pp.101-118

35. Cheng, Q, Wu, C, Zhou, H, Zhang, Y, Wang, R, Ruan, W, "Guarding the Perimeter of Cloud-Based Enterprise Networks: An Intelligent SDN Firewall", *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)"*, Exeter, UK, 2018, pp. 897-902, doi: 10.1109/HPCC/SmartCity/DSS.2018.00149.

36. Sahoo, KS, Panda, SK, Sahoo, S, Sahoo, B, Dash, R 2019, "Towards secure software-defined networks against distributed denial of service attack", *The Journal of Supercomputing*, vol. 75, pp. 4829-4874. https://doi.org/10.1007/s11227-019--02767-z

37. Slim, F, Guillemin, F, Gravey, A, Hadjadj-Aoul, Y 2017, "Towards a dynamic adaptive placement of virtual network functions under ONAP", 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV_SDN), Berlin, Germany, pp. 210-215. Doi:10.1109/NFV_SDN.8169880.