IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Impact Factor: 7.301
Volume 3, Issue 2, February 2023

Navigating the Network the Evolution of SDN Data Planes

T. Aditya[1], A. David Donald[1], G. Thippanna[2], M. Mohsina Kousar[3], K. Rekha[3]
Ashoka Women's Engineering College, Dupadu, Andhra Pradesh, India[1,2,3]

Abstract: *Software-defined networking (SDN) is becoming more popular because it makes the data layer of a network easier to change and adapt. That's the driving force behind its development and maturation. The purpose of this research was to look into the history of SDN data planes and highlight the key innovations and advances that have led to their current state. We will discuss the evolution of data planes from hardware to software, the development of programmable data planes, and the positive effects of software-defined networking data planes on network performance, control, and management. We will also look into the challenges of deploying SDN data planes and provide advice for companies on how to make the most of this transition. Finally, this article's goal is to provide a comprehensive analysis of the data planes in software-defined networks (SDN) and their effects on the networking sector.*

Keywords: SDN, Data plane

I. INTRODUCTION

The data plane is crucial to any network design since it processes and transmits data packets and is responsible for managing network traffic. The advent of SDN has led to a dramatic change in the data plane, making it infinitely more malleable and programmable. Software-defined networks aim to provide software-based data planes, which grant more granular control and management over network traffic. In contrast to hardware-based data planes, SDN data planes can be dynamically adapted to support a wide range of network tasks and applications. Because of its programmability, businesses can better adapt to changing market conditions and customer demands.

Network managers can better control network policies and routing thanks to the consolidated view of network traffic provided by SDN data planes. The consolidated nature of this control makes it easier to locate and repair network issues and keep the network secure. Business owners need to be aware of the challenges involved in deploying SDN data planes before they can start reaping the benefits. In particular, ensuring that the legacy network infrastructure can integrate smoothly with the new software-based data planes during the switch to software-defined networking (SDN) requires careful planning and coordination. Following this, we will delve more deeply into the most salient features and advantages of SDN data planes. In this lesson, we will learn about the various data planes, the significance of programmability, and the benefits of software-defined networking (SDN) data planes for enhancing network performance, control, and management.

Furthermore, we will look into the challenges that come with deploying SDN data planes. There are a number of challenges that must be overcome in order to implement a newly planned network architecture successfully. These include the need for extensive planning and coordination, the risk of being tied to a single vendor, and the availability of competent employees. Finally, we'll talk about some of the latest trends and accomplishments in this rapidly developing field of network architecture and give some insights into the best practises for constructing SDN data planes.

The overarching goal of this study is to provide an all-encompassing overview of SDN data planes. The purpose of this article is to shed light on the salient aspects of SDN data planes, including their advantages and disadvantages, as well as the issues that must be addressed by enterprises before they can successfully adopt this technology. By spreading this knowledge, we hope to help organisations make well-informed decisions about their networks' underlying infrastructure and unlock the full potential of SDN data planes for reaching their goals.

II. DATA PLANE FUNCTIONS

The data plane is responsible for forwarding network packets and managing network traffic. In traditional networking, this is done by hardware-based devices such as switches and routers. However, in SDN, the data plane is implemented using software-defined switches that run on commodity hardware. These switches can be programmed and configured to support a variety of network functions and applications.

Here are some of the key functions of the SDN data plane:

1. **Packet forwarding:** The data plane forwards network packets from one switch to another based on the rules defined by the control plane.
2. **Traffic engineering:** The data plane can be programmed to route traffic along specific paths in the network, enabling more efficient use of network resources.
3. **Quality of Service (QoS) management:** The data plane can prioritize network traffic based on specific applications, ensuring that mission-critical applications receive the necessary network resources.
4. **Network monitoring:** The data plane can collect and report network statistics, such as packet loss and latency, allowing network administrators to monitor network performance.
5. **Load balancing:** The data plane can distribute network traffic across multiple paths, ensuring that no single path becomes overloaded.
6. **Security:** The data plane can be programmed to enforce network security policies, such as access control and encryption, to protect against unauthorized access and data breaches.
7. **Network virtualization:** The data plane can be used to create virtual networks that run on top of the physical network, enabling greater flexibility and agility in network design.

The SDN data plane plays a critical role in enabling the flexibility and programmability of SDN networks. By providing a software-based, programmable data plane, SDN enables organizations to build more agile, efficient, and secure networks that can meet the demands of today's rapidly changing business environment.

In addition to the functions mentioned above, SDN data planes also offer other benefits that traditional hardware-based data planes do not. Some of these benefits include:

1. **Centralized control:** SDN data planes offer a centralized view of network traffic, allowing administrators to manage network policies and routing more efficiently. This centralized control also makes it easier to troubleshoot network issues and ensure network security.
2. **Scalability:** The software-based nature of SDN data planes enables them to scale more easily and cost-effectively than traditional hardware-based data planes. This scalability makes it easier for organizations to expand their networks as their business needs grow.
3. **Programmability:** The programmability of SDN data planes enables organizations to create custom network functions and applications that are tailored to their specific business needs. This flexibility allows organizations to respond quickly to changing business requirements and enables them to innovate and differentiate themselves from their competitors.
4. **Openness:** SDN data planes are typically based on open standards, which means that they are not tied to specific vendors or proprietary technologies. This openness makes it easier for organizations to integrate SDN data planes into their existing networks and enables greater collaboration and interoperability among different vendors and technologies.
5. **Lower cost:** SDN data planes are typically less expensive than traditional hardware-based data planes, as they can be implemented using commodity hardware and open-source software. This lower cost makes it easier for organizations to adopt SDN technology and reap the benefits of a more flexible and programmable network infrastructure.

The SDN data plane is a critical component of the SDN architecture, enabling organizations to build more agile, efficient, and secure networks. By offering centralized control, scalability, programmability, openness, and lower cost, SDN data planes provide organizations with the tools they need to meet the demands of today's rapidly changing business environment.

III. SDN DATA PLANE PROTOCOLS

SDN data plane protocols are used to facilitate communication between the control plane and the data plane in an SDN network. These protocols are responsible for establishing and maintaining the rules and policies that govern the forwarding of network traffic within the network. Some of the most commonly used SDN data plane protocols include:

1. **OpenFlow:** OpenFlow is the most widely used SDN data plane protocol. It is a standardized protocol that enables the communication between the control plane and the data plane in an SDN network. OpenFlow is designed to provide a programmable and flexible data plane, allowing network administrators to manage network traffic more efficiently.
2. **NETCONF:** NETCONF is a protocol used for network configuration management. It is used to configure and manage network devices, including SDN switches, routers, and firewalls. NETCONF enables network administrators to configure network devices using a standardized set of data models, making it easier to manage network configuration across different devices.
3. **BGP-LS:** BGP-LS is a protocol used for the distribution of link-state and topology information in an SDN network. It enables network administrators to distribute network topology information to the SDN controller, allowing the controller to make routing decisions based on up-to-date network topology information.
4. **P4:** P4 is a programming language that is used to specify the forwarding behavior of an SDN data plane. It provides a high degree of flexibility and programmability, enabling network administrators to customize the forwarding behavior of the data plane based on the specific needs of their network.
5. **OF-Config:** OF-Config is a protocol used to manage the configuration of OpenFlow-enabled devices in an SDN network. It enables network administrators to manage the configuration of OpenFlow-enabled devices using a standardized set of data models, making it easier to manage network configuration across different devices.
6. **VXLAN:** VXLAN (Virtual Extensible LAN) is a protocol used for overlay network virtualization. It enables the creation of virtualized network segments that can span multiple physical networks, allowing network administrators to create virtual networks that are isolated from each other. VXLAN can be used with SDN data planes to create virtual networks that can be managed more efficiently and securely.
7. **MPLS:** MPLS (Multiprotocol Label Switching) is a protocol used for traffic engineering and network virtualization. It is commonly used in traditional networks, but it can also be used with SDN data planes to enable more efficient traffic routing and network segmentation. MPLS can be used to create virtual networks that are optimized for specific traffic types, enabling network administrators to prioritize critical traffic and ensure the best possible performance for their applications.
8. **GRE:** GRE (Generic Routing Encapsulation) is a protocol used for tunneling network traffic. It enables the creation of virtual tunnels that can be used to encapsulate network traffic and route it across different networks. GRE can be used with SDN data planes to create virtual tunnels that can span multiple physical networks, enabling more efficient and secure network communication.

These SDN data plane protocols provide organizations with a range of options for configuring and managing their SDN networks. By enabling centralized control, programmability, and flexibility, these protocols enable organizations to build more agile, efficient, and secure networks that can meet the demands of today's rapidly changing business environment.

IV. OPEN FLOW LOGICAL NETWORK DEVICE

In SDN (Software-Defined Networking), an OpenFlow logical network device refers to a virtual switch or router that is created and controlled by the SDN controller using the OpenFlow protocol. Unlike traditional network devices, OpenFlow logical network devices are not physical devices but rather virtual representations of network devices that exist within the SDN controller's software.

The OpenFlow protocol enables network administrators to define and manage the forwarding behavior of the logical network device through a centralized control plane. The logical network device can be programmed to forward traffic based on a variety of criteria, including source and destination IP addresses, MAC addresses, transport protocols, and application-layer information.

IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Volume 3, Issue 2, February 2023

Impact Factor: 7.301

One of the key benefits of using an OpenFlow logical network device is its ability to provide a flexible and programmable data plane that can be customized based on the specific needs of the network. Network administrators can define and manage the forwarding behavior of the logical network device through a variety of tools and interfaces, including the OpenFlow protocol, programming languages like P4, and network management software.

The OpenFlow logical network device plays a critical role in enabling the programmability and flexibility of SDN networks. By providing a virtual representation of network devices that can be controlled through a centralized control plane, it enables organizations to build more agile, efficient, and secure networks that can adapt to the demands of today's rapidly changing business environment.

In addition to its programmability and flexibility, the OpenFlow logical network device also offers several other benefits for SDN networks. These include:

1. **Centralized management:** Because the logical network device is controlled through a centralized SDN controller, network administrators can manage the device's behavior and configuration from a single location. This can greatly simplify network management tasks, especially in large and complex networks.
2. **Improved network visibility:** The OpenFlow protocol provides real-time visibility into the traffic flowing through the logical network device, enabling network administrators to monitor and analyze network traffic in greater detail. This can help identify and troubleshoot network performance issues more quickly.
3. **Better security:** The programmability of the OpenFlow logical network device allows for more granular control over network traffic, which can help improve network security. Network administrators can define and enforce security policies at the logical network device level, enabling more precise control over which devices and applications can access the network.
4. **Scalability:** Because the OpenFlow logical network device is a virtual device, it can be easily scaled up or down to meet changing network demands. This enables organizations to expand their network capacity more easily and cost-effectively than with traditional physical devices.

The OpenFlow logical network device is a critical component of SDN networks, providing a flexible, scalable, and programmable data plane that can be customized to meet the unique needs of each organization. As SDN continues to evolve, the role of the logical network device is likely to become even more important, enabling organizations to build networks that are more efficient, secure, and resilient.

4.1 Flow Table Structure

In SDN (Software-Defined Networking), the flow table is a key component of the data plane that determines how network traffic is processed and forwarded through the network. The flow table is essentially a set of rules that are programmed by the SDN controller to define how the switch should handle different types of network traffic.

The structure of the flow table can vary depending on the specific implementation of the SDN network, but generally consists of the following components:

1. **Match fields:** These are the fields that the switch uses to match incoming traffic to a specific flow entry in the flow table. These fields can include a variety of criteria such as source and destination IP addresses, MAC addresses, transport protocols, and application-layer information.
2. **Actions:** Once the switch has identified a flow entry that matches the incoming traffic, it uses the actions specified in the flow entry to determine how to process and forward the traffic. These actions can include forwarding the traffic to a specific port or interface, modifying the packet header, dropping the packet, or sending the packet to the controller for further processing.
3. **Counters:** The flow table may also include counters that keep track of various statistics related to the traffic that is processed by the switch. These statistics can include the number of packets or bytes processed by the flow entry, as well as the number of packets or bytes that were dropped or sent to the controller for further processing.

In addition to these basic components, the flow table may also include other advanced features such as timers, priorities, and idle timeouts. Timers can be used to control how long a flow entry remains active in the flow table before it is deleted, while priorities can be used to specify the order in which flow entries are processed by the switch. Idle

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Impact Factor: 7.301
Volume 3, Issue 2, February 2023

timeouts can be used to specify how long a flow entry remains active in the flow table if no traffic matches its match fields.

Another important aspect of the flow table structure is the ability to support multiple flow tables. In more complex SDN networks, it may be necessary to use multiple flow tables to achieve greater flexibility and control over network traffic. Multiple flow tables allow network administrators to define different policies and rules for different types of traffic or applications, enabling more granular control over the network.

In addition to the flow table structure itself, the process of programming and managing the flow table is also critical for the proper functioning of SDN networks. The process of programming the flow table involves specifying the match fields, actions, and other parameters for each flow entry, as well as setting timers, priorities, and other advanced features as needed.

Managing the flow table involves monitoring network traffic and making adjustments to the flow table as needed to ensure optimal network performance. This may involve modifying existing flow entries, adding new flow entries, or deleting flow entries that are no longer needed.

The flow table structure is a fundamental component of the SDN data plane, enabling network administrators to define customized network policies and rules that meet the unique needs of their organizations. By providing a flexible and programmable framework for processing and forwarding network traffic, the flow table helps to improve network performance, security, and efficiency in SDN networks.

4.2 Flow Table Pipeline

The flow table pipeline is another important concept in the SDN data plane. It refers to the sequence of processing stages that a packet must pass through as it traverses the network. Each stage of the pipeline corresponds to a specific flow table, with each table containing a set of rules that define how packets should be processed and forwarded.

The flow table pipeline typically consists of three main stages: the ingress pipeline, the processing pipeline, and the egress pipeline. The ingress pipeline is responsible for processing incoming packets as they enter the network, while the egress pipeline is responsible for processing outgoing packets as they leave the network. The processing pipeline sits between the ingress and egress pipelines and is responsible for routing packets through the network based on the rules defined in the flow tables. The flow table pipeline is typically implemented using a series of hardware and software components, including switches, routers, and other network devices. Each device in the network may have its own set of flow tables and associated processing pipeline, allowing for distributed processing of network traffic.

The flow table pipeline plays a critical role in enabling the flexibility and programmability of SDN networks. By allowing network administrators to define customized flow tables and processing pipelines, SDN networks can be tailored to meet the unique needs of different organizations and applications. This can help to improve network performance, security, and efficiency, while also reducing the complexity and costs associated with traditional network architectures. To further elaborate, the flow table pipeline operates by inspecting each incoming packet against the rules defined in the ingress flow table. If a match is found, the packet is forwarded to the next stage in the pipeline, which typically involves additional processing or modification of the packet header. This process continues until the packet reaches the egress pipeline, where it is matched against the rules defined in the egress flow table and forwarded out of the network.

The flow table pipeline is highly flexible and customizable, allowing network administrators to define multiple flow tables with different priorities and associated processing logic. For example, an organization might define a flow table for high-priority traffic that receives preferential treatment over other types of traffic, or a flow table that directs traffic to specific network resources based on the application or service being used. Another important aspect of the flow table pipeline is the ability to support dynamic updates and modifications to the flow tables in real-time. This allows network administrators to respond quickly to changing network conditions and traffic patterns, and to implement new policies and rules as needed to optimize network performance and security.

The flow table pipeline is a key element of the SDN data plane architecture, enabling network administrators to define customized processing logic and rules for handling network traffic. By providing a flexible and programmable framework for processing and forwarding packets, the flow table pipeline helps to improve network performance, efficiency, and security in SDN networks.

DOI: 10.48175/IJARSCT-8525

4.3 Use of Multiple Tables

the use of multiple flow tables is an important feature of the SDN data plane. In more complex networks, it may be necessary to use multiple tables to achieve greater flexibility and control over network traffic. This allows network administrators to define different policies and rules for different types of traffic or applications, enabling more granular control over the network.

One common use case for multiple flow tables is traffic classification. By using a dedicated flow table for traffic classification, network administrators can apply different policies and rules based on traffic attributes such as source IP address, destination IP address, port numbers, and protocol type. For example, network administrators can use a flow table to direct traffic from specific applications to specific servers, or to apply different security policies to traffic from different sources.

Another use case for multiple flow tables is Quality of Service (QoS) management. By using separate flow tables for high-priority and low-priority traffic, network administrators can ensure that critical traffic receives preferential treatment over less important traffic. This can help to improve network performance and reduce the risk of congestion and packet loss.In addition to traffic classification and QoS management, multiple flow tables can also be used for load balancing, security policy enforcement, and other advanced network functions. By providing a flexible and customizable framework for handling network traffic, the use of multiple flow tables helps to improve network performance, security, and efficiency in SDN networks.

To implement multiple flow tables in an SDN network, switches must support the OpenFlow Multiple Tables feature. This feature allows switches to process packets through multiple flow tables in a pipeline fashion, with each flow table handling a specific stage of the packet processing pipeline.The pipeline structure of multiple flow tables is similar to the pipeline structure of a single flow table. Each flow table in the pipeline has its own set of match fields, actions, and other parameters that define how packets are processed and forwarded. Packets are processed through each flow table in the pipeline sequentially, with the results of each stage being passed on to the next stage for further processing.

One advantage of the pipeline structure is that it allows network administrators to define more complex packet processing policies and rules. For example, a network administrator could use one flow table to classify traffic based on the source and destination IP addresses, and then use another flow table to apply QoS policies based on the classification results. By breaking down packet processing into smaller stages, network administrators can define more granular policies and rules that better reflect the needs of their organization.

The use of multiple flow tables is an important feature of the SDN data plane, providing network administrators with a flexible and customizable framework for handling network traffic. By breaking down packet processing into smaller stages and allowing for more granular policies and rules, multiple flow tables help to improve network performance, security, and efficiency in SDN networks.

V. GROUP TABLE

In an SDN network, the group table is another key component of the data plane. The group table is used to implement group-based actions, which allow multiple actions to be applied to packets that match a specific set of criteria. Group-based actions are useful for implementing multicast, load balancing, and other advanced packet processing functions. The group table is similar to the flow table in structure, with each entry in the group table defining a group of actions that can be applied to packets. Each group table entry includes a group ID, a group type, and a set of action buckets. The group ID is used to identify the group table entry, while the group type specifies how the group should be used (e.g., for multicast or load balancing). The action buckets define the set of actions that should be applied to packets that match the group criteria.

The use of the group table is particularly useful for implementing multicast in SDN networks. Multicast allows network traffic to be sent from a single source to multiple destinations, reducing network congestion and improving efficiency. With the group table, network administrators can define a multicast group that includes a set of destination addresses. When a packet matches the multicast group criteria, the packet is replicated and sent to all destinations in the group. In addition to multicast, the group table can also be used for load balancing, where packets are distributed across multiple network paths to improve performance and reliability. By defining a group of network paths in the group table, network administrators can distribute packets across these paths using load balancing algorithms.

The group table is designed to work in conjunction with the flow table, allowing network administrators to define more complex network policies and rules that involve multiple actions. For example, a flow table entry can be set up to match a specific set of criteria and then forward the packet to a group in the group table. The group table can then define a set of actions that should be applied to the packet, such as replicating the packet and sending it to multiple destinations. In addition to multicast and load balancing, the group table can also be used for other advanced packet processing functions, such as packet filtering and traffic shaping. For example, a group table entry can be set up to filter out unwanted packets, such as packets with a specific source or destination address. The group table can also be used to apply different levels of service to different types of traffic, such as prioritizing real-time traffic over non-real-time traffic.

Like the flow table, the group table is programmable, which means that network administrators can define customized group-based actions to meet the unique needs of their organizations. This makes the group table an important tool for implementing complex network policies and rules in SDN networks.

The group table is an important component of the SDN data plane, providing network administrators with a flexible and programmable framework for implementing advanced packet processing functions. By allowing for group-based actions, the group table helps to improve network performance, scalability, and efficiency in SDN networks.

5.1 Open Flow Protocol

The OpenFlow protocol is a key component of the SDN data plane, providing a standardized way for SDN controllers to communicate with switches and routers in the network. The protocol defines the format and content of messages exchanged between the controller and the data plane devices, allowing the controller to program and manage the flow tables and group tables on these devices. The OpenFlow protocol is based on a client-server model, where the controller acts as the client and the data plane devices act as the servers. The protocol consists of several message types, including configuration messages, request messages, and reply messages. These messages are used to perform various functions, such as configuring flow tables, querying device capabilities, and reporting network statistics.

One of the key features of the OpenFlow protocol is its support for multiple versions. This allows for backward compatibility with older devices that may not support the latest version of the protocol, while also allowing for the development of new features and capabilities in future versions. The OpenFlow protocol has become a widely adopted standard in the SDN industry, with support from many major vendors and organizations. Its flexibility and programmability make it a powerful tool for implementing customized network policies and rules, while its standardized format ensures interoperability across different devices and platforms.

The OpenFlow protocol also enables network administrators to implement traffic engineering, load balancing, and other advanced features by providing granular control over network traffic. It allows for the creation of dynamic network topologies and the efficient use of network resources, leading to improved network performance and efficiency.In addition to its benefits for network administrators, the OpenFlow protocol also provides several advantages for network security. By allowing for fine-grained control over network traffic, it makes it easier to implement security policies and detect and prevent network threats. It also enables the creation of virtual network slices and isolation of traffic between different network segments, improving overall network security.

The OpenFlow protocol is not without its challenges, however. Its centralized architecture, which relies on a single controller to manage the network, can be a single point of failure and a potential bottleneck for network traffic. In addition, the protocol's reliance on software-based switches can limit its scalability and performance in large-scale networks.To address these challenges, new technologies such as P4 (Programming Protocol-Independent Packet Processors) are being developed to enable more flexible and programmable data plane processing in SDN networks.

The OpenFlow protocol is a foundational technology for the SDN data plane, providing a standardized and flexible framework for programming and managing network devices. Its benefits for network performance, efficiency, and security have made it a widely adopted standard in the SDN industry, and it is likely to continue playing a critical role in the evolution of SDN technology.

VI. CONCLUSION

the SDN data plane is a critical component of software-defined networking that enables network administrators to program and manage network devices in a centralized and flexible manner. The flow table structure is a fundamental element of the data plane, allowing for the creation of customized network policies and rules that meet the unique needs of organizations. The use of multiple tables and the group table feature provides additional flexibility and control over network traffic, while the OpenFlow protocol provides a standardized framework for programming and managing network devices.The benefits of SDN technology, including improved network performance, efficiency, and security, have made it a widely adopted standard in the networking industry. As SDN continues to evolve, new technologies and standards will emerge, further enhancing the capabilities and benefits of the SDN data plane. Overall, the SDN data plane represents a major shift in networking technology, providing a more agile and responsive network infrastructure that is better suited to the dynamic needs of modern organizations. The SDN data plane is a critical component of software-defined networking that enables network administrators to create and manage highly customizable network policies and rules. As SDN technology continues to evolve, the data plane will remain a key focus for innovation and development, helping organizations to achieve greater agility, flexibility, and security in their network infrastructure.

REFERENCES

[1]. https://www.techopedia.com/definition/32318/data-plane#:~:text=In%20software%2Ddefined%20networking%20(SDN,Plane%2C%20Forwarding%20Plane%2C%20Carrier%20Plane
[2]. https://www.bmc.com/blogs/software-defined-networking/
[3]. https://www.sciencedirect.com/topics/computer-science/data-plane-function
[4]. https://www.geeksforgeeks.org/difference-between-control-plane-and-data-plane/
[5]. https://sdn.systemsapproach.org/intro.html
[6]. https://www.bmc.com/blogs/software-defined-networking/
[7]. https://www.sdxcentral.com/networking/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/inside-sdn-architecture/
[8]. https://noviflow.com/the-basics-of-sdn-and-the-openflow-network-architecture/
[9]. https://overlaid.net/2017/02/15/openflow-basic-concepts-and-theory/
[10]. https://noviflow.com/the-basics-of-sdn-and-the-openflow-network-architecture/
[11]. https://www.sciencedirect.com/topics/computer-science/flow-table-entry
[12]. Isyaku, Babangida, MohdSoperiMohd Zahid, MaznahBteKamat, Kamalrulnizam Abu Bakar, and Fuad A. Ghaleb. 2020. "Software Defined Networking Flow Table Management of OpenFlow Switches Performance and Security Challenges: A Survey" Future Internet 12, no. 9: 147. https://doi.org/10.3390/fi12090147
[13]. https://www.cs.rice.edu/~eugeneng/papers/SDS15.pdf
[14]. https://www.sciencedirect.com/topics/computer-science/multiple-flow-table
[15]. https://sdn.systemsapproach.org/intro.html
[16]. http://courses.washington.edu/ee565/handouts/chapter4.pdf
[17]. https://www.geeksforgeeks.org/architecture-of-software-defined-networks-sdn/
[18]. https://www.section.io/engineering-education/openflow-sdn/#:~:text=OpenFlow%20is%20the%20standard%20southbound,port%20statistics%20in%20network%20management.