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

SDN Application Plane Where Intelligence Meets Networking

T. Aditya[1], A. David Donald[1], G. Thippanna[2],M.Mohsina Kousar[3], C. Madilety[3]

Ashoka Women's Engineering College, Dupadu, Andhra Pradesh, India[1,2,3]

Abstract: *The software-defined networking (SDN) paradigm has revolutionized the way we think about network management and control. The SDN application plane is a critical component of this paradigm, where intelligent applications are leveraged to drive network behavior and optimize network performance. In this abstract, we explore the role of the SDN application plane in bridging the gap between traditional network management and the intelligent, dynamic network of the future. We discuss the challenges and opportunities of deploying SDN applications, including the need for standardized APIs and the importance of intelligent analytics and machine learning techniques. Finally, we highlight the transformative potential of the SDN application plane, from enabling new applications and services to improving network security and resilience. By bringing intelligence and networking together in new and innovative ways, the SDN application plane promises to revolutionize the way we design, manage, and secure our networks.*

Keywords: SDN, Application Plane

I. INTRODUCTION

Software-defined networking (SDN) is an emerging paradigm that is transforming the way we design, manage, and secure our networks. At the heart of this paradigm is the SDN application plane, which provides a flexible and programmable interface for managing network behavior and optimizing network performance. Unlike traditional networking approaches that rely on manual configuration of network devices, the SDN application plane allows intelligent applications to drive network behavior and dynamically adapt to changing network conditions.

The SDN application plane provides a standardized interface for developing and deploying intelligent applications that can leverage network data and analytics to optimize network performance. These applications can be developed by network operators, third-party developers, or even end-users, and can range from simple traffic management and load balancing to complex security and resilience applications.

One of the key benefits of the SDN application plane is its ability to separate the control and data planes of the network. By centralizing control in a software-based controller, the SDN application plane enables network operators to dynamically configure and manage network resources without having to touch individual network devices. This centralization also allows for greater automation and orchestration of network resources, reducing operational complexity and improving network agility.

Key Features and Benefits of SDN Application Plane:

1. **Centralized control:** The SDN application plane enables centralized control of the network through a software-based controller. This allows network operators to configure and manage network resources from a single location, rather than having to manually configure individual network devices.
2. **Programmability:** The SDN application plane is highly programmable, allowing for the development and deployment of intelligent applications that can dynamically adapt to changing network conditions and optimize network performance.
3. **Standardized APIs:** The SDN application plane provides standardized APIs for developing and deploying applications, making it easier for third-party developers to create and integrate new applications into the network.
4. **Improved network agility:** By centralizing control and enabling programmability, the SDN application plane improves network agility, allowing network operators to quickly adapt to changing network conditions and optimize network performance.

DOI: 10.48175/IJARSCT-8529

5. **Increased automation:** The SDN application plane enables increased automation and orchestration of network resources, reducing operational complexity and improving network efficiency.

Challenges and Opportunities of SDN Application Plane:

1. **Lack of standards:** Despite the availability of standardized APIs, there is still a lack of standardization in the SDN application plane, which can make it difficult to develop and deploy applications across different vendors and platforms.
2. **Security concerns:** The centralized control and programmability of the SDN application plane can also create new security risks, such as the potential for unauthorized access or the manipulation of network behavior by malicious actors.
3. **Complexity:** The SDN application plane can be complex to deploy and manage, requiring specialized knowledge and expertise to develop and deploy applications effectively.
4. **Integration with existing networks:** The SDN application plane must also be integrated with existing networks, which can be challenging and require careful planning and coordination.

Despite these challenges, the SDN application plane represents a major step forward in network management and control, enabling intelligent, dynamic networks that can adapt to changing network conditions and optimize network performance in real-time. As the SDN paradigm continues to evolve, the SDN application plane will likely play an increasingly important role in shaping the future of networking.

II. NORTHBOUND INTERFACE

The northbound interface (NBI) is a critical component of software-defined networking (SDN) architecture, providing a standardized interface between the SDN controller and the applications that use it. The NBI is responsible for exposing the capabilities of the SDN controller to applications, allowing them to configure and manage network behavior and performance.The NBI provides a standard set of APIs for developers to create applications that interact with the SDN controller, enabling them to develop innovative solutions that leverage the flexibility and programmability of SDN. This allows for greater automation and orchestration of network resources, as well as the development of new applications and services that can optimize network performance and improve security and resilience.

One of the key benefits of the NBI is that it enables third-party developers to create and integrate applications into the SDN architecture, regardless of the specific controller or hardware platform being used. This promotes interoperability and innovation, as developers are not restricted to a particular vendor or hardware platform.

There are several key components of the NBI, including:

1. **Application Programming Interfaces (APIs):** These are the interfaces that enable developers to create and integrate applications into the SDN architecture.
2. **Service Abstraction Layer (SAL):** This layer provides a standard interface for the SDN controller to interact with network devices and services.
3. **Information Model:** This model defines the structure and format of the data exchanged between the SDN controller and applications.

There are several types of northbound interfaces, including RESTful APIs, OpenFlow, and OVSDB. Each type of interface has its own strengths and weaknesses, and the choice of interface will depend on the specific needs of the network and the applications that will be using it.

RESTful APIs are a popular type of northbound interface because they are easy to use and provide a standardized interface for developers to create applications that interact with the SDN controller. RESTful APIs use HTTP requests to exchange information between the SDN controller and applications, making it easy for developers to create applications that can be used across different platforms and devices.

OpenFlow is another type of northbound interface that is commonly used in SDN architectures. OpenFlow enables the SDN controller to control the behavior of network switches and routers, allowing for greater flexibility and control over network resources. OpenFlow also provides a standardized interface for developing applications that interact with the SDN controller, making it easier for developers to create new applications and services.

DOI: 10.48175/IJARSCT-8529

OVSDB is a northbound interface that is specifically designed for use with Open vSwitch (OVS), a popular software-based switch that is commonly used in SDN architectures. OVSDB provides a standardized interface for managing and configuring OVS switches, allowing for greater flexibility and control over network resources.

The northbound interface is a critical component of the SDN architecture, enabling greater flexibility, programmability, and control over network resources. By providing a standardized interface between the SDN controller and applications, the NBI promotes interoperability and innovation, allowing developers to create new applications and services that can optimize network performance and improve security and resilience. As the SDN paradigm continues to evolve, the NBI will continue to play a key role in shaping the future of networking.

III. NETWORK APPLICATIONS

The application plane in SDN refers to the layer where network applications run. It is the layer that sits on top of the control plane and interacts with the underlying network infrastructure through the data plane.

In SDN, network applications can be developed and deployed independently from the underlying hardware and network devices. This allows for more flexibility and agility in network management and enables network operators to quickly adapt to changing business requirements.

Examples of network applications that can run on the application plane in SDN include:

1. **Network security applications:** These applications can monitor network traffic, detect and prevent security threats, and enforce security policies.
2. **Traffic engineering applications:** These applications can optimize network traffic flow, reduce congestion, and ensure efficient use of network resources.
3. **Quality of Service (QoS) applications:** These applications can prioritize network traffic based on specific requirements, such as bandwidth, latency, and packet loss.
4. **Network monitoring and analytics applications:** These applications can collect and analyze network data to identify performance issues, troubleshoot network problems, and provide insights for network optimization.

The application plane in SDN also provides a level of abstraction that simplifies network management and reduces complexity. Network administrators can use standard programming interfaces to develop, deploy, and manage network applications, instead of relying on proprietary interfaces and command-line interfaces (CLIs) of network devices.

In addition, the application plane in SDN enables network operators to automate network operations and streamline network workflows. By integrating network applications with orchestration and automation tools, network administrators can simplify and accelerate network provisioning, configuration, and monitoring.

Overall, the application plane in SDN plays a critical role in enabling network innovation and agility, simplifying network management, and improving network performance and efficiency. As SDN continues to evolve, we can expect to see more powerful and sophisticated network applications that take advantage of the flexibility and programmability of the SDN architecture.

IV. USER INTERFACE

The user interface (UI) in SDN refers to the layer of the SDN architecture that allows network administrators to interact with and manage the network. The UI provides a graphical interface that allows administrators to configure, monitor, and troubleshoot the network.

The SDN UI typically includes a dashboard that provides an overview of the network topology, including network devices, links, and services. From the dashboard, administrators can drill down into specific devices and services to view detailed information, such as device status, interface statistics, and flow tables.

The SDN UI also provides tools for configuring and managing network services, such as routing, switching, and security. Administrators can use the UI to create and modify network policies, configure network services, and monitor network performance.

The UI in SDN can be accessed through a web-based portal, a command-line interface (CLI), or a dedicated software application. Some SDN controllers also provide APIs that allow administrators to interact with the controller and network through custom-built applications.

The UI in SDN also plays an important role in enabling network automation and orchestration. By integrating the UI with automation and orchestration tools, network administrators can simplify and accelerate network management tasks and reduce the risk of errors.

For example, an SDN UI could be used to configure a network service, such as a firewall policy. Once the policy is defined, an automation tool could be used to deploy the policy across the network, ensuring consistent configuration and reducing the risk of errors.

The SDN UI can also be used to visualize network performance and troubleshoot network issues. Network administrators can use the UI to view real-time network statistics, such as traffic volumes, latency, and packet loss. They can also use the UI to identify network anomalies, such as traffic spikes or unexpected traffic patterns, and take corrective action to mitigate the issue.

The SDN UI is a critical component of the SDN architecture that enables network administrators to manage complex networks with ease and efficiency. As SDN continues to evolve, we can expect to see more powerful and intuitive UIs that make it even easier for administrators to manage and automate their networks.

V. NETWORK SERVICES ABSTRACTION LAYER

The Network Services Abstraction Layer (NSAL) in SDN is a component of the SDN architecture that provides a standardized interface for network services. NSAL abstracts the complexity of network services and presents a uniform interface that can be used by network applications.

NSAL provides a set of APIs that allow network applications to interact with network services in a consistent and standardized manner. These APIs abstract the underlying network hardware and protocols and provide a high-level view of network services, such as routing, switching, and security.

NSAL enables network applications to be developed independently from the underlying network infrastructure. This makes it easier to develop and deploy network applications and provides greater flexibility in network management.NSAL also enables network applications to be dynamically reconfigured in response to changing network conditions. For example, an application that provides traffic engineering services can use NSAL to dynamically adjust network traffic flows in response to changes in network traffic.

NSAL also provides a level of abstraction that allows network operators to manage network services independently from the underlying network hardware. This makes it easier to manage complex network topologies and enables network operators to scale their networks more efficiently.

Another benefit of NSAL is that it enables network applications to be developed using standard programming languages and APIs. This makes it easier for developers to create network applications and enables network applications to be integrated with other IT systems, such as cloud computing platforms, data analytics tools, and automation frameworks.

NSAL also provides a platform for network innovation by enabling the development of new and innovative network services. For example, NSAL can be used to develop network services that leverage artificial intelligence (AI) and machine learning (ML) technologies to optimize network performance and security.

NSAL is a critical component of the SDN architecture that enables network innovation and agility, simplifies network management, and improves network performance and efficiency. As SDN continues to evolve, we can expect to see more powerful and sophisticated network services that take advantage of the flexibility and programmability of NSAL.

VI. ABSTRACTIONS IN SDN

Abstractions in SDN refer to the layers of the SDN architecture that provide simplified and standardized views of network resources and services. These abstractions enable network administrators and developers to manage and program networks more easily and efficiently.

Some of the key abstractions in SDN include:

1. **Control plane abstraction:** The control plane abstraction separates the network control functions from the data forwarding functions. This allows network administrators to centrally control and manage the network, rather than configuring each network device individually.

DOI: 10.48175/IJARSCT-8529

2. **Data plane abstraction:** The data plane abstraction separates the forwarding logic from the underlying network hardware. This allows network administrators to program the forwarding behavior of the network devices using software-defined rules, rather than relying on proprietary hardware.
3. **Application plane abstraction:** The application plane abstraction provides a standardized interface for network applications to interact with the network services. This enables network applications to be developed independently from the underlying network hardware and protocols.
4. **Network services abstraction:** The network services abstraction provides a uniform view of network services, such as routing, switching, and security. This enables network applications to interact with network services in a consistent and standardized manner.

By providing these abstractions, SDN simplifies network management and enables network innovation and agility. SDN also provides a level of flexibility and programmability that traditional networks lack, making it easier to adapt to changing network requirements and conditions.

VII. FRENETIC

In the context of SDN (Software-Defined Networking), "frenetic" usually refers to the Frenetic Network Controller, which is an open-source controller for SDN networks.The Frenetic Network Controller is built using the Frenetic programming language, which is a high-level language designed specifically for programming SDN controllers. It provides a simple and concise syntax for expressing network policies and allows for easy composition and decomposition of network functions.

One of the key features of the Frenetic Network Controller is its ability to handle dynamic network conditions in real-time. It can automatically detect changes in the network topology and adjust network policies accordingly, without requiring manual intervention.

Some additional features of the Frenetic Network Controller include:

1. **Modular architecture:** The controller is designed to be modular, which allows for easy integration of new network applications and services. This modularity also makes it easy to update and maintain the controller over time.
2. **Scalability:** The Frenetic Network Controller is designed to scale to handle large and complex networks. It uses efficient algorithms and data structures to optimize the use of system resources.
3. **Open-source:** The Frenetic Network Controller is an open-source project, which means that it is freely available for anyone to use, modify, and distribute. This makes it a popular choice for researchers, developers, and network operators who want to experiment with and contribute to the development of SDN.
4. **Extensive documentation:** The Frenetic project provides extensive documentation, including user guides, tutorials, and API references, to help users get started with the controller and the Frenetic programming language.

The Frenetic Network Controller provides a powerful and flexible platform for building and managing SDN networks. Its features and capabilities make it a popular choice for a wide range of applications, from small-scale experimental networks to large-scale production networks.

VIII. FRENETIC- TRAFFIC ENGINEERING MEASUREMENT AND MONITORING SECURITY

The Frenetic Network Controller can be used to implement traffic engineering, measurement, monitoring, and security functions in SDN networks.

1. **Traffic engineering**: With the Frenetic Network Controller, network operators can easily manage traffic flow in their networks by setting policies to prioritize certain types of traffic, reroute traffic around congested links, or balance traffic across multiple paths. This can improve network performance and reduce congestion.
2. **Measurement and monitoring**: The Frenetic Network Controller can collect and analyze data on network traffic, such as packet counts, flow statistics, and network performance metrics. This can help network operators identify and troubleshoot network issues, monitor network usage, and optimize network performance.

3. **Security**: The Frenetic Network Controller can be used to enforce security policies in SDN networks, such as blocking traffic from known malicious sources, detecting and preventing network attacks, and isolating compromised devices. Additionally, Frenetic provides a secure programming environment to prevent attacks such as buffer overflow, code injection, and other common vulnerabilities.

Frenetic provides a flexible and powerful platform for implementing a wide range of traffic engineering, measurement, monitoring, and security functions in SDN networks. By leveraging the Frenetic programming language and the modular architecture of the Frenetic Network Controller, network operators can customize and extend the controller to meet their specific needs.

IX. DATA CENTRE NETWORKING

Data centre networking refers to the networking infrastructure that connects the servers, storage, and other resources in a data centre. In traditional data centre networks, switches and routers are used to connect the various components of the data centre. However, with the rise of cloud computing and virtualization, software-defined networking (SDN) has become an increasingly popular approach to data centre networking.

SDN provides several benefits for data centre networking, including:

1. Centralized management: With SDN, the network control plane is decoupled from the data plane, allowing network administrators to centrally manage and configure the network. This makes it easier to provision, monitor, and troubleshoot the network.
2. Programmability: SDN provides a programmable interface for configuring and controlling the network, which allows network administrators to automate network operations and customize the network to meet their specific needs. This can improve network agility and reduce operational costs.
3. Traffic engineering: SDN provides granular control over network traffic, allowing network administrators to implement traffic engineering functions such as load balancing, QoS, and path optimization. This can improve network performance and reduce congestion.
4. Security: SDN provides a number of security features, such as flow-based access control and network segmentation, which can help protect the network from attacks and unauthorized access.

Some of the popular SDN technologies used in data centre networking include OpenFlow, VXLAN, and Cisco ACI. These technologies provide a range of capabilities for implementing SDN-based data centre networks, including virtualization, network slicing, and micro-segmentation.

Some additional details on data centre networking:

1. **Network architecture:** Data centre networks are typically designed using a hierarchical network architecture, consisting of three layers: access, aggregation, and core. The access layer connects end devices, such as servers and storage, to the network, while the aggregation layer connects access switches to the core network. The core network provides high-speed connectivity between different parts of the data centre.
2. **Network virtualization:** Virtualization technologies, such as VMware vSphere, are commonly used in data centres to improve resource utilization and flexibility. Virtualization allows multiple virtual machines (VMs) to share a single physical server, and network virtualization extends this concept to the network layer. Network virtualization technologies, such as VXLAN, allow multiple virtual networks to be overlaid on a physical network, enabling greater network agility and flexibility.
3. **Storage networking:** Storage networking is a critical component of data centre networking, as it provides connectivity between servers and storage devices. Fibre Channel and iSCSI are two popular storage networking protocols used in data centres. Fibre Channel provides high-speed, low-latency connectivity between servers and storage devices, while iSCSI uses standard Ethernet networks to provide block-level storage access.
4. **Network management:** Network management in data centre networks can be challenging due to the complexity of the network infrastructure and the large number of devices and services running in the data centre. Network management tools, such as network monitoring and configuration management tools, can help simplify network operations and improve network performance and availability.

Data centre networking is a complex and critical component of modern data centres. SDN provides a flexible and powerful approach to building and managing data centre networks, and virtualization and storage networking technologies are commonly used to improve resource utilization and flexibility. Effective network management is essential for ensuring the performance, availability, and security of data centre networks.

X. MOBILITY AND WIRELESS

Mobility and wireless technologies have revolutionized the way we access information and communicate with one another. Wireless networks allow us to connect to the internet and other networks without the need for physical cables, providing greater mobility and flexibility. Mobile devices, such as smartphones and tablets, have become ubiquitous in today's society, enabling us to stay connected and productive wherever we go.

Some of the key concepts related to mobility and wireless technologies include:

1. **Wireless networking:** Wireless networking refers to the use of radio waves to connect devices to a network. Wi-Fi is a popular wireless networking technology used in homes, businesses, and public spaces. Wi-Fi provides high-speed wireless connectivity to the internet and other networks, allowing users to access information and communicate with one another.
2. **Mobile computing:** Mobile computing refers to the use of mobile devices, such as smartphones and tablets, to access information and perform tasks. Mobile devices are equipped with wireless connectivity technologies, such as Wi-Fi and cellular data, allowing users to stay connected to the internet and other networks while on the go.
3. **Wireless security:** Wireless networks are vulnerable to a range of security threats, such as unauthorized access, data interception, and malware attacks. Wireless security technologies, such as WPA2 encryption and MAC address filtering, can help protect wireless networks from these threats.
4. **Mobile device management:** Mobile device management (MDM) refers to the process of managing and securing mobile devices, such as smartphones and tablets, used in an organization. MDM solutions can help organizations manage device configurations, enforce security policies, and track device usage.
5. **Wireless sensor networks:** Wireless sensor networks are a type of wireless network used to collect data from sensors distributed throughout an environment. Wireless sensor networks are commonly used in industrial, environmental, and healthcare applications, allowing users to monitor and control a range of parameters, such as temperature, humidity, and air quality.
6. **Mobile application development:** Mobile application development refers to the process of creating software applications that run on mobile devices, such as smartphones and tablets. Mobile applications can be developed for a variety of purposes, such as gaming, social networking, productivity, and entertainment.
7. **Wireless standards:** Wireless standards are a set of guidelines that define how wireless devices communicate with one another. Some of the popular wireless standards include Wi-Fi (IEEE 802.11), Bluetooth (IEEE 802.15.1), and cellular (3G, 4G, and 5G).
8. **Wireless access points:** Wireless access points (WAPs) are devices that allow wireless devices to connect to a network. WAPs are typically used in wireless networks to provide coverage over a specific area, such as a room or a building.
9. **Wireless range and coverage:** Wireless range and coverage refer to the distance and area over which wireless signals can be transmitted and received. The range and coverage of wireless networks can be affected by a range of factors, such as the strength of the wireless signal, the number of obstacles in the environment, and the type of wireless technology being used.
10. **Wireless spectrum:** Wireless spectrum refers to the range of frequencies used for wireless communications. Different wireless technologies use different parts of the wireless spectrum, and the use of the wireless spectrum is regulated by national and international agencies to prevent interference between different wireless technologies.

Mobility and wireless technologies have had a significant impact on the way we live and work. Wireless networking, mobile computing, wireless security, mobile device management, wireless sensor networks, mobile application development, wireless standards, wireless access points, wireless range and coverage, and wireless spectrum are some of the key concepts related to mobility and wireless technologies. Understanding these concepts is essential for building and managing effective wireless networks and mobile applications.

XI. CONCLUSION

The application plane is a critical component of the Software-Defined Networking (SDN) architecture. It is responsible for managing the network applications and services that run on top of the SDN infrastructure. By abstracting the network services from the underlying hardware and providing a centralized control plane, the SDN application plane enables network administrators to manage and orchestrate network resources more efficiently and effectively.The SDN application plane provides a range of benefits, including simplified network management, improved network flexibility and agility, faster service deployment, and reduced operational costs. It also enables the development of innovative network applications and services that can be customized to meet the specific needs of different organizations and industries.

Some of the key SDN application plane technologies and protocols include OpenFlow, NETCONF, REST APIs, and northbound APIs. These technologies enable network administrators to program and control network services and applications in a more flexible and automated way, improving the overall efficiency and effectiveness of the network. Overall, the SDN application plane is a critical component of the SDN architecture, providing network administrators with the tools and capabilities they need to manage and orchestrate network services and applications more efficiently and effectively. As organizations continue to adopt SDN and embrace the benefits of network automation, the importance of the application plane will only continue to grow.

REFERENCES

[1]. Scott-Hayward, Sandra, Christopher Kane, and Sakir Sezer. "Operationcheckpoint: Sdn application control." In 2014 IEEE 22nd International Conference on Network Protocols, pp. 618-623. IEEE, 2014.

[2]. Karakus, Murat, and Arjan Durresi. "A survey: Control plane scalability issues and approaches in software-defined networking (SDN)." Computer Networks 112 (2017): 279-293.

[3]. Banse, Christian, and Sathyanarayanan Rangarajan. "A secure northbound interface for SDN applications." In 2015 IEEE Trustcom/BigDataSE/ISPA, vol. 1, pp. 834-839. IEEE, 2015.

[4]. Mekky, Hesham, Fang Hao, Sarit Mukherjee, Zhi-Li Zhang, and T. V. Lakshman. "Application-aware data plane processing in SDN." In Proceedings of the third workshop on Hot topics in software defined networking, pp. 13-18. 2014.

[5]. Tatang, Dennis, Florian Quinkert, Joel Frank, Christian Röpke, and Thorsten Holz. "SDN-Guard: Protecting SDN controllers against SDN rootkits." In 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 297-302. IEEE, 2017.

[6]. Kulkarni, Manasa, Bhargavi Goswami, and Joy Paulose. "Experimenting with scalability of software defined networks using pyretic and frenetic." In Computing Science, Communication and Security: Second International Conference, COMS2 2021, Gujarat, India, February 6–7, 2021, Revised Selected Papers, pp. 168-192. Cham: Springer International Publishing, 2021.

[7]. Azodolmolky, Siamak, Philipp Wieder, and Ramin Yahyapour. "SDN-based cloud computing networking." In 2013 15th international conference on transparent optical networks (ICTON), pp. 1-4. IEEE, 2013.

[8]. Francois, Frederic, and Erol Gelenbe. "Optimizing secure SDN-enabled inter-data centre overlay networks through cognitive routing." In 2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 283-288. IEEE, 2016.

[9]. Yan, Muxi, Jasson Casey, Prithviraj Shome, Alex Sprintson, and Andrew Sutton. "ÆtherFlow: Principled wireless support in SDN." In 2015 IEEE 23rd International Conference on Network Protocols (ICNP), pp. 432-437. IEEE, 2015.

[10]. Tomovic, Slavica, Milica Pejanovic-Djurisic, and Igor Radusinovic. "SDN based mobile networks: Concepts and benefits." Wireless Personal Communications 78 (2014): 1629-1644.