www.ijmserh.com

Volume 4, Issue 3, July 2016

ISSN: 2349-6037

IJMSERH International Journal of Multidisciplinary and Scientific Emerging ResearcH (IJMSERH)

Introduction to Software-Defined Networking (SDN)

Geetha Nagarajan, Kanishka R.G

Department of CSE, SA Engineering College, Chennai, India

ABSTRACT: Software-Defined Networking (SDN) is a revolutionary paradigm that decouples the control plane from the data plane in networking, allowing centralized management and dynamic configuration. This paper provides an introduction to SDN, its architecture, components, and benefits. It also explores existing literature, methodologies for implementing SDN, its background, and potential future research directions. The findings suggest that SDN enhances network flexibility, programmability, and security while reducing operational costs.

KEYWORDS: Software-Defined Networking, SDN Architecture, Network Management, Network Virtualization, Control Plane, Data Plane

I. INTRODUCTION

Traditional networking architectures are often rigid, complex, and difficult to manage. Software-Defined Networking (SDN) introduces a paradigm shift by separating the control plane from the data plane, enabling more efficient network management. SDN allows centralized control, dynamic network configuration, and enhanced security. This paper aims to introduce SDN, discuss its fundamental concepts, and explore its impact on modern networking.

II. LITERATURE REVIEW

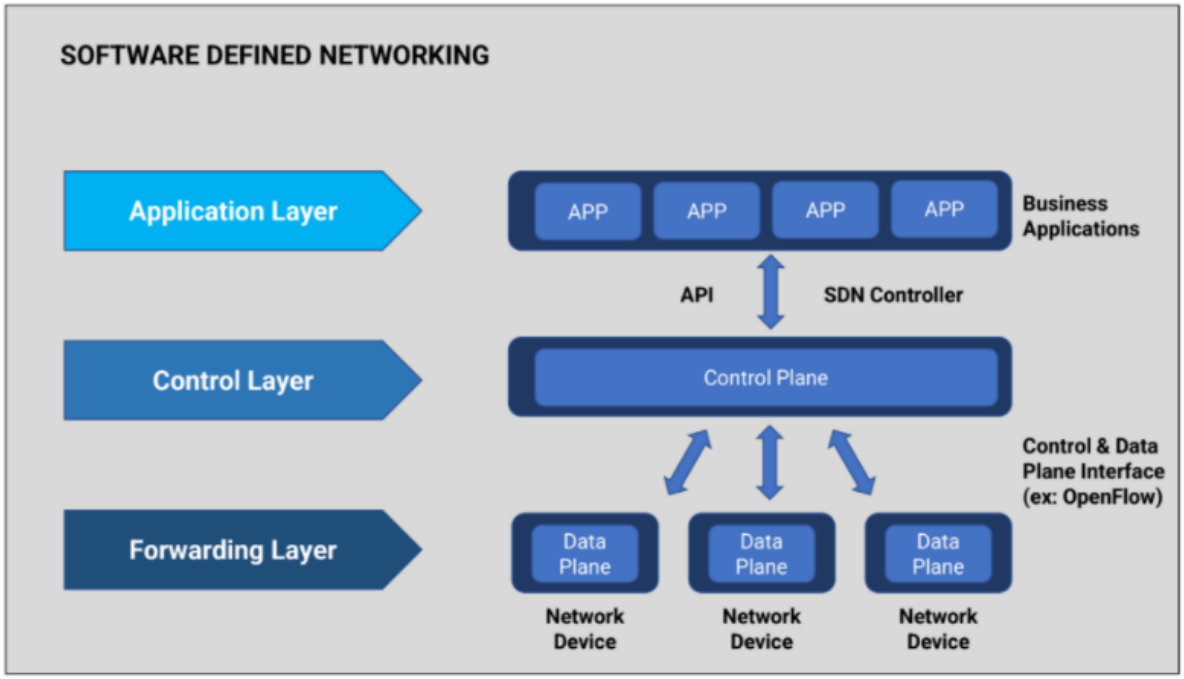
Numerous studies have highlighted the advantages and challenges of SDN. McKeown et al. (2008) introduced the OpenFlow protocol as a foundational component of SDN, enabling programmability in network devices. Nunes et al. (2014) provided a comprehensive survey on SDN, emphasizing its impact on network management and security. Recent works have focused on SDN applications in cloud computing, Internet of Things (IoT), and cybersecurity. However, challenges such as scalability, interoperability, and security remain significant research areas.

III.METHODOLOGY

To implement SDN effectively in an organization or a network infrastructure, it is essential to follow a systematic methodology. This methodology generally includes the following steps:

1. **Network Assessment:** Begin by evaluating the existing network infrastructure. This includes reviewing current network topologies, performance, hardware limitations, and scalability concerns. Identifying pain points such as high costs, limited flexibility, and network inefficiencies helps in determining the areas where SDN can provide the most benefit.
2. **Define Objectives and Use Cases:** Establish clear objectives for SDN implementation. These might include improving network agility, reducing operational costs, or improving network automation. It's also important to identify specific use cases like data center networking, multi-cloud environments, or enterprise campus networks where SDN can be applied.
3. **Choose the Right SDN Architecture:** There are various SDN architectures to choose from, such as **OpenFlow**, **ONF-based architectures**, and proprietary systems. Deciding on the appropriate architecture depends on the requirements for flexibility, compatibility with existing infrastructure, and the desired level of control.
4. **Select SDN Controllers and Tools:** Choose the SDN controllers that best meet the needs of the organization. Popular open-source SDN controllers include **OpenDaylight**, **Ryu**, and **Floodlight**. Additionally, there are commercial SDN controllers, such as **Cisco ACI** and **VMware NSX**, which offer more specialized features.
5. **Network Virtualization:** Once the SDN controller is selected, the next step is to implement network virtualization. This involves creating virtual networks that run on top of physical infrastructure, offering greater flexibility, scalability, and isolation. Virtualization helps decouple the network architecture from physical hardware.

6. **Integration and Testing:** With SDN components in place, it is essential to integrate the SDN network with existing network infrastructure and perform thorough testing. This includes checking for performance issues, ensuring interoperability, and validating that SDN applications can control the traffic flow as intended.
7. **Deployment and Monitoring:** After successful testing, deploy the SDN solution across the organization or network. Once deployed, monitoring becomes a key part of SDN management to ensure that the network is performing optimally. Network administrators use SDN tools to track traffic patterns, adjust policies, and quickly troubleshoot issues as they arise.
8. **Continuous Optimization:** The final step is continuous optimization. SDN allows dynamic changes to be made to the network in real-time, so regular monitoring and optimization are critical. Over time, SDN deployments can be fine-tuned based on new use cases, performance metrics, and evolving organizational needs.



Architecture of Software Defined Networking (SDN):

IV. BACKGROUND ON SDN

SDN consists of three primary layers:

- **Application Layer:** Contains network applications that dictate network behavior.
- **Control Layer:** Includes SDN controllers that manage data flows.
- **Infrastructure Layer:** Comprises network devices such as switches and routers that handle packet forwarding. SDN relies on the OpenFlow protocol to facilitate communication between these layers, enhancing network agility and automation.

Benefits of SDN:

- **Centralized Control:** SDN centralizes network management, enabling easier configuration, monitoring, and troubleshooting.
- **Flexibility and Programmability:** Administrators can program the network to meet specific demands without needing to configure individual devices.

- **Cost Reduction:** By using standardized hardware and open protocols, SDN reduces reliance on expensive proprietary network devices.
- **Scalability:** SDN makes it easier to scale the network, especially in environments that require flexibility such as cloud computing.
- **Automation:** SDN enables automation of many network management tasks, such as load balancing, traffic routing, and network monitoring.

Challenges of SDN:

- **Security Risks:** The centralized nature of SDN means that compromising the SDN controller can jeopardize the entire network.
- **Complexity:** Implementing SDN may require significant changes to network architecture and may involve a steep learning curve for administrators.
- **Interoperability:** Integrating SDN into existing networks with legacy devices can pose challenges in terms of compatibility.

V. CONCLUSION

SDN represents a transformative approach to networking, offering flexibility, programmability, and cost efficiency. While it provides numerous benefits, challenges such as security risks, implementation complexity, and scalability issues must be addressed. Future advancements in artificial intelligence (AI) and machine learning (ML) are expected to enhance SDN's capabilities.

VI. FUTURE WORK

Future research should focus on improving SDN security, enhancing scalability for large-scale deployments, and integrating AI-driven network management. Additionally, further studies on SDN applications in emerging technologies like 5G, edge computing, and IoT will be valuable.

REFERENCES

1. McKeown, N., et al. (2008). "OpenFlow: Enabling Innovation in Campus Networks." ACM SIGCOMM Computer Communication Review, 38(2), 69-74.
2. Nunes, B. A. A., et al. (2014). "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks." IEEE Communications Surveys & Tutorials, 16(3), 1617-1634.
3. Kreutz, D., et al. (2015). "Software-Defined Networking: A Comprehensive Survey." Proceedings of the IEEE, 103(1), 14-76.
4. Mohit, Mittal (2013). The Rise of Software Defined Networking (SDN): A Paradigm Shift in Cloud Data Centers. International Journal of Innovative Research in Science, Engineering and Technology 2 (8):4150-4160.
5. Pavan Reddy, Vaka (2012). Zero-Day Vulnerabilities. International Journal of Innovative Research in Science, Engineering and Technology 1 (2):318-322.
6. Hu, F., et al. (2014). "A Survey on SDN and OpenFlow: From Concept to Implementation." IEEE Communications Surveys & Tutorials, 16(4), 2181-2206.