Quest Journals
Journal of Software Engineering and Simulation
Volume 9 ~ Issue 10 (2023) pp: 01-04
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

TITLE

AUTHOR

Abstract
This paper delves into the challenges and opportunities presented by Software Defined Networking (SDN) and explores the selection of an optimal SDN controller. The aim is to simplify network management, reduce implementation costs, and enhance network maintenance for large organizations. The paper also provides an introductory overview of Artificial Intelligence and its key domains, highlighting their integration within SDN.

***R**eceived 11 Oct., 2023; **R**evised 23 Oct., 2023; **A**ccepted 25 Oct., 2023 © The author(s) 2023.*
P**ublished with open access at **www.questjournals.org

I. Introduction

Over the last decade, the predominant topic in network administration has been Software Defined Networking (SDN) and the associated SDN Controllers. SDN has emerged as a solution to address the limitations of traditional network structures, including separate control planes for each network device, lack of centralized network visibility, and the need for additional Network Monitoring Systems. These traditional network structures often fall short in meeting the demands of contemporary IT requirements. SDN has garnered substantial attention due to its advantages, largely attributed to network virtualization. Key to SDN is the consolidation of the control and data planes, enabling a single software control program to manage various data plane elements. SDN leverages virtualization to optimize resource allocation, streamline IT procedures, and fine-tune applications and systems. By using virtualization, SDN efficiently scales network capacity without depleting resources, simplifying resource management.

A significant challenge in traditional networking is the existence of a separate control plane for each device, leading to distributed decision-making and processing. Additionally, the absence of centralized network visibility necessitates separate Network Monitoring Systems. SDN addresses these issues by adopting a centralized approach, employing a distinct controller and data plane, offering network visibility, programmable interfaces, high-speed networks, open interface compatibility, and access to source code. In SDN, the data and control planes coexist within the controller but function independently. The data plane handles traffic, while the control plane manages communication between the controller, network applications, decision-making, and action forwarding.

Notably, several SDN Controllers have emerged, including Beacon OpenFlow controller, NOX, POX, Nettle, OpenDayLight, FloodLight, and Ryu. These platforms have enabled scientists and researchers to develop applications such as Load Balancing, Network Virtualization, Energy Efficient Networking, Dynamic Access Control in Enterprise networks, and Virtual Machine Mobility, among others.

II. Software Defined Networking (SDN)

Software-Defined Networking is a highly discussed and researched topic, offering solutions to the limitations of traditional networking models. SDN's architecture separates the network plane and centralizes control within an SDN controller. This decoupling of the data and control planes serves various purposes.

The implementation of SDN in existing networks remains complex, but its achievements in the networking field continue to evolve. The delay in implementing SDN is primarily attributed to the complexities it introduces. Ongoing research and the deployment of industrial equipment aim to address challenges related to performance, scalability, security, and interoperability. Despite these challenges, the scalability and reliability of SDN justify its adoption.

The SDN architecture consists of three main layers: the application layer, the control layer, and the underlying infrastructure layer. Unlike traditional architecture with a separate control plane for each device, SDN centralizes the control plane in an isolated process called the controller, operating in the control layer. This allows applications running in the application layer to operate as if they were on a single logical network switch.

III. OpenFlow

The OpenFlow system, initially developed at Stanford and now overseen by the Open Networking Foundation (ONF), aligns with SDN's layered architecture. It sits between the control and forwarding planes, acting as a communication protocol. OpenFlow is typically implemented between SDN Controllers and OpenFlow-enabled switches, using flow tables to match traffic. OpenFlow switches can be categorized as OpenFlow Only switches and OpenFlow Hybrid switches. OpenFlow Only switches exclusively use OpenFlow operations, while OpenFlow Hybrid switches utilize both OpenFlow and traditional Ethernet operations.

OpenFlow was designed to enable researchers to experiment with new SDN concepts and devices in diverse environments, facilitating the abstraction of physical networks through an intermediate layer. The controller acts as an intermediary between OpenFlow switches and other standard OpenFlow controllers, managing data transfer capacity, CPU usage, and flow tables.

Traditional network systems often struggle to express clients' requirements dynamically, leading to increased costs and potential misclassification. SDN empowers clients to clearly articulate their requirements for adaptable and customized network services. Control decisions are made from a global perspective of the network state, rather than being distributed across separate modules for each network operation. In SDN, the control plane acts as a centralized, reasonably unified system for both planning and resolving resource conflicts, even at the level of low-level device components.

Networks can be managed through programming interfaces called Application Programming Interfaces (APIs), which can achieve hardware-independent technology. In software-defined networks, devices such as switches and routers solely forward packets, with all decisions made by the controller through network applications within the controller. In SDN, network devices are configured centrally, eliminating the need to access each device individually. The OpenFlow Protocol, as an ascending protocol, enables network devices' accessibility through APIs, providing comprehensive information on network services such as routing and Quality of Service (QoS). This contrasts with traditional networks where network devices require individual provisioning.

SDN controllers feature two separate interfaces with distinct functions. The North Bound Interface, also known as the application layer-connected interface, communicates with the upper layer to obtain updates on communication rules. The South Bound Interface facilitates downward communication with the network and provides status updates on forwarding policies pushed to downstream devices, typically switches.

IV. Selection of SDN Controllers

Selecting an SDN controller represents a crucial challenge for network administrators. The controller functions as the universal hub for the entire network, supporting applications and services. The effectiveness of the entire network is influenced by various properties of SDN controllers. Consequently, evaluating controllers based on a single property is impractical. Instead, a thorough analysis of multiple properties using the Analytical Hierarchy Process (AHP) is employed. The AHP aids in reaching decisions by considering the impact of multiple properties.

SDN has evolved from various research paradigms, notably SANE and Ethane. NOX, an SDN controller, was initially developed by Nicira Networks and was one of the primary OpenFlow Controllers. Subsequently, NOX was transferred to the research community in 2008. NOX provides asynchronous and high-speed Input and Output (I/O), a C++ OpenFlow 1.0 API, and focuses on the latest Linux technologies. It incorporates various experimental components for topology discovery, switch learning, and network-wide switches.

The FloodLight controller defines the open communication standard in an SDN environment, allowing the SDN controller to communicate with the forwarding plane, including switches and routers.

4.1 Modernization Using SDN-based Networks

Network management devices and services can be seamlessly integrated into SDN system applications, offering solutions in various domains such as SDN Network Management, Load Balancing, SDN Security, and Virtualization. Applications added to SDN-based systems typically require a reasonable number of control components to be updated.

4.2 Load Balancing for Application Servers

Several SDN-based applications have been proposed for large commercial networks, including load balancing for application servers. An OpenFlow switch distributes traffic across various servers under the control of a centralized device (load balancer) that manages packets destined for specific servers.

4.3 Security and Network Access Control

SDN plays a pivotal role in controlling the flow of virtualized resources within a network, particularly with Service Function Chaining (SFC) and Software-Defined Perimeter (SDP) systems. These systems ensure secure, low-latency data transfer between users in the SDN network and services accessed through Software as a Service (SaaS) and Internet applications.

4.4 Network Function Virtualization (NFV)

In collaboration with SDN, Network Function Virtualization (NFV) offers scalable network services without requiring dedicated appliances for individual network functions. This approach enables network functions to be treated as separate, logical entities that can be orchestrated and automated as per specific requirements.

V. Integration with Artificial Intelligence (AI)

The realm of Artificial Intelligence offers the integration of AI in SDN, giving rise to the concept of Artificial Intelligence in Software-Defined Networking (AISDN). AISDN employs various AI techniques, including Machine Learning (ML), to optimize network performance, predict outages, enhance security, and adapt to changing traffic patterns. Key areas of AI integration include:

5.1 Network Analytics

AI systems can analyze massive volumes of network data to uncover patterns, anomalies, and potential security threats in real-time. This empowers network administrators to make informed decisions quickly.

5.2 Predictive Maintenance

AI-driven predictive maintenance techniques can predict network equipment failures, enabling proactive maintenance to reduce downtime.

5.3 Traffic Optimization

AI can optimize network traffic to reduce latency and improve overall network performance. This is particularly valuable in SDN, where traffic can be dynamically controlled.

5.4 Security Enhancements

AI-based intrusion detection systems can identify and respond to security threats faster and more accurately than traditional rule-based systems.

VI. Conclusion

Software-Defined Networking represents a pivotal shift in the field of network management. It enables centralized control, network virtualization, and efficient resource allocation, improving network flexibility and scalability. The selection of the right SDN controller is vital for optimizing network performance and management. With ongoing advancements in SDN, its integration with AI is expected to bring significant enhancements to network analytics, security, and overall network performance.

SDN controllers like FloodLight and NOX have paved the way for software-defined networks. While challenges remain, such as scalability and security, these can be addressed with ongoing research and development. The integration of AI into SDN offers promising solutions to these challenges, further revolutionizing network management.

References:

[1]. Russell, S., Norvig, P.: ‘Artificial Intelligence (A Modern Approach)’. 3rd ed. New Jersey: Prentice Hall, 1995. 1152 p.

[2]. Negnevitsky, M.: ‘Artificial Intelligence A Guide to Intelligent Systems’. 2nd ed. Essex: Addison Wesley, 2005. 415 p.

[3]. Zhang, G.P.: ‘Neural networks for classification: a survey’, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2000,30,(4), pp. 451 462

[4]. Nguyen, T.T., Armitage, G.: ‘A survey of techniques for internet traffic classification using machine learning’. IEEE Communications Surveys & Tutorials. 2008,10,(4), pp. 56 76.

[5]. LeCun, Y., Bengio, Y., Hinton, G.: ‘Deep learning’, Nature, 2016, 521,(7553), pp.436 444.

[6]. Deng, L.: ‘A tutorial survey of architectures, algorithms, and applications for deep learning’, APSIPA Transactions on Signal and Information Processing, 2014, 3,(e2), pp. 1 19.

[7]. Mohammadi, M., Al Fuqaha, A., Sorour, S. Guizani, M.: ‘Deep Learning for IoT Big Data and Streaming Analytics: A Survey’, IEEE Communications Surveys & Tutorials.

[8]. Tang, T., Zaidi, S.A.R., McLernon, D., Mhamdi, L. Ghogho, M.: ‘Deep Recurrent Neural Network for Intrusion Detection in SDN based Networks’. In 2018 IEEE International Conference on Network Softwarization (NetSoft 2018), Montreal, Canada, Jun 2018.

[9]. Fan, X., Guo, Z.: ‘A semi supervised Text Classification Method based on Incremental EM Algorithm’. WASE International Conference on Information Engineering, Beidaihe, China, August, 2010, pp. 211 214.

[10]. Gui, T., Ma, C., Wang, F., Wilkins, D.E.: ‘Survey on swarm intelligence based routing protocols for wireless sensor networks: An extensive study. In: Proc. of IEEE International Conference on Industrial Technology (ICIT), Taipei, Taiwan, March 2016, pp. 1944 1949.

[11]. Soysal, M., Schmidt, E.G.: ‘Machine learning algorithms for accurate flow based network traffic classification: Evaluation and comparison’, Performance Evaluation, 2010,67,(6), pp. 451 467.

[12]. McGregor, A., Hall, M., Lorier, P., Brunskill, J.: ‘Flow clustering using machine learning techniques. In: Proc. of International Workshop on Passive and Active Network Measurement (PAM), Antibes Juan les Pins, France, April 2014, pp. 205 214.

[13]. Xu, X., Wang, X.: ‘An adaptive network intrusion detection method based on PCA and support vector machines’. In: Proc. of Advanced Data Mining and Applications, Wuhan, China, July 2005, pp. 696 703.

[14]. Kim, H.Y., Kim, J.M.: ‘A load balancing scheme based on deep learning in IoT’, Cluster Computing, 2017,20,(1), pp. 873 878

[15]. Moustapha, A.I., Selmic, R.R.: ‘Wireless sensor network modeling using modified recurrent neural networks: Application to fault detection, IEEE Transactions on Instrumentation and Measurement, 2008,57,(5), pp. 981 988

[16]. Mushtaq, M.S., Augustin, B., Mellouk, A.: ‘Empirical study based on machine learning approach to assess the QoS/QoE correlation. In: Proc. of 17th European Conference on Networks and Optical Communications (NOC), Vilanova i la Geltru, Spain, June 2012, pp. 1 7

[17]. Testolin, A., Zanforlin, M., De Grazia, M.D.F., et al.: ‘A machine learning approach to QoE based video admission control and resource allocation in wireless systems. In: Proc. of 13th Annual Mediterranean Ad Hoc Networking Workshop (MED HOC NET), Piran, Slovenia, June 2014, pp. 31 38