SOFTWARE DEFINED NETWORK– AN EMERGING NETWORK PARADIGM

Bonyl Santhmayor, Jayson Ruzario, Jenil Joseph, Siddhant Sawant
Department of Electronics and Tele-Communication,
St. John College of Engineering and Management, Palghar, India

Abstract- The internet connects almost everything and is accessible by almost anywhere in the world. However, traditional IP networks are complex and very hard to manage. It is difficult to reconfigure it to respond to changes made. Also, managing the network infrastructure has become a complicated and time-consuming task. Recently, Software Defined Networking (SDN) has gained the attention of researchers due to the ease of centralized control and programmability of the network. SDN is characterized by its two distinguished features, including decoupling the control plane from the data plane and providing programmability for network application development. This paper first presents a generally accepted definition of SDN with the potential need for SDN. This paper provides a snapshot of the related work done by various researchers. Finally, we conclude this survey paper with some open research challenges.

Indexterms- **Software defined network, SDN controller, openflow.**

I. Introduction:

The demand for networked computer systems is increasing rapidly day by day. Today, organizations are using more advanced computing environments to meet the demand, like cloud-based networks, virtualized desktops and servers, and remote data-storage devices [1]. Today's internet applications and services require the networks to be fast, carry large amounts of data and to deploy a number of dynamic applications and services along with a least chance of errors and network crashes. In SDN, the controller enables adaptable management and centralized control with programmable interfaces by keeping in view the global network [2]. This technology is introduced as initial implementation to manage large IP/Ethernet networks by using centralized control over forwarding, and to manage communications in virtual networks, is used today in different aspects for examples data centre LANs and the Google's IP core network. In networking techniques, one of the major tasks is to provide optimal solution to allocate resources i.e., bandwidth, hardware, memory sharing, program sharing etc. Proper resource allocation increases speed of network and decreases cost of overall network. SDN is one of the significant techniques used to allocate resources. SDN helps to flexibly allocate resources when many users connect to the network. Different parameters are defined to allocate resources in SDN. The feature of centralized control help to gather global resource information in the controller and resource allocation can be easily executed by OpenFlow [3].

II. Definition, architecture and need for SDN

A. Definition of SDN:

SDN stands for Software defined network. SDN is equipped with logically centralized controller with a global view of Network. SDN brings the centralized approach of network configuration as opposed to distributed management. The controller takes decision on forwarding the traffic in a logically way. SDN is a networking architecture which decoupled the data plane and control plane. SDN uses centralized controller to manage networks [4].

B. Architecture of SDN

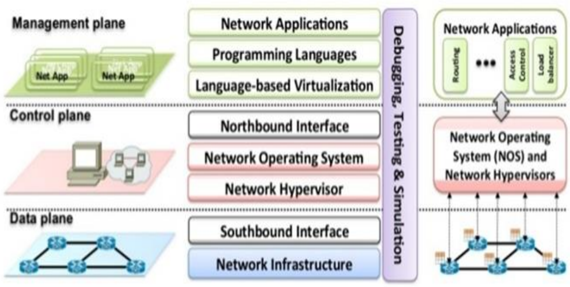

Fig. 1. SDN planes summarized [6]

SDN is a layered architecture, consisting of three basic layers; application/services layer, controller layer (control plane), and data plane layer called forwarding layer consisting of forwarding devices. These SDN layers communicate with each other via open APIs called Northbound Interface (NI) API and Southbound Interface (SI) API. In traditional networks, the network operator had to do configuration of the individual network device as per service layer agreements (SLAs) and could not be programmed otherwise. Unlike this traditional network architecture where each device has a separate control plane, in SDN architecture it is separated and centralized on an isolated process (called controller) running at control layer. This isolated process (controller) provides universal view of the network. The controller acts as the main brain or main controller which is responsible for the decision routing which is placed at switches using protocols named as "OpenFlow". This technology is being considered one of the favourable technologies for isolation of control plane & data plane and logical placement of centralized control from SDN controller.

C. Need for SDN

The decision-making capability or network intelligence is distributed across the various network hardware components. This makes the introduction of any new network device or service a tedious job because it requires reconfiguration of each of the numerous network nodes. Networks today depend on IP addresses to identify and locate servers and applications. This approach works fine for small networks but is extremely tedious for large virtual networks. In traditional networking architecture all switches use separate control planes and data planes. So, each device needs to be configured separately. Whereas, SDN centralized controller needs centralized configuration for all switches[1]. The control provided by SDN may include not only packet forwarding at a switching level but also link tuning at a data link level, breaking the barrier of layering. Moreover, it has the ability to acquire instantaneous network status. SDN permits a real-time centralized control of a network based on both instantaneous network status and user defined policies[5]. SDN encourages innovation by providing a programmable network platform to implement, experiment, and deploy new ideas, new applications, and new revenue earning services conveniently and flexibly.

III. Related work

SDNs are a hot topic and grabbing the attention of industry and market. Many comprehensive studies are done by the research communities to look into the detail perspective; implications, use cases, and technological demand.

Kreutz et al. [6] presented a comprehensive study on SDN. The authors provide a detail and all-inclusive on SDN, SDN evolution from programmable networks to SDN architecture, protocols, application, use case scenarios and future research trends. It discusses the three tier architecture along with different implementation details. The paper further discusses the use cases of SDN by providing details of applications at each layer of SDN namely Data plane, control plane and the application layer.

Zhaogang Shu [7], presented a paper on the traffic management for SDN. The paper firstly talks about how conventional TE takes place (ip-based, mpls-based) and what are its drawbacks and why TE in SDN would be better option enabling us to achieve fine-grained control of traffic management. It also propose a reference framework for TE in the SDN, which consists of two parts, traffic measurement and traffic management. Traffic measurement is responsible for monitoring and analysing real time network traffic and management uses various applications and algorithms to manage the traffic efficiently.

Ying Dar Lin [8], talks about fast failover in case of link failures. In his paper, he presented a fast failover mechanism and a fast switchover mechanism to deal with link failure and congestion problems. In the fast failover mechanism, the controller pre-establishes multiple paths for each source-destination pair in the related OpenFlow-enabled (OF) switches. When a link becomes faulty, OF switches are able to failover the affected flows to another path. Based on the pre-established paths, in the fast switchover mechanism, the controller periodically monitors the status of each port of each OF switch. They also emulated their mechanism this on using RYU and mininet emulator and showed that the average recovery time of the fast failover mechanism is less than 40ms.

IV. OpenFlow

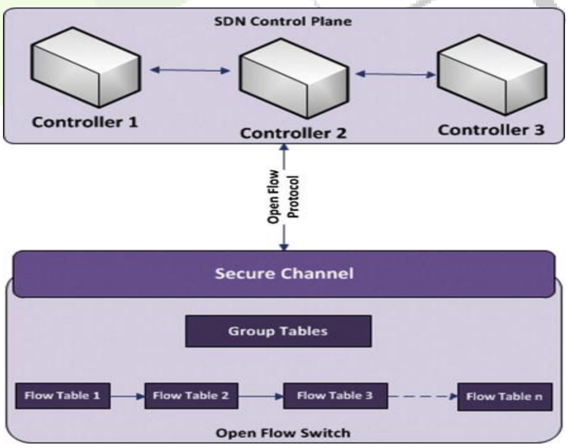

Fig. 2. Basic architecture of OpenFlow [9]

OpenFlow system was initially created at Stanford University but now it is under the Open Networking Foundation (ONF). Open Networking foundation(ONF) defined that OpenFlow protocol is based on SDN layered architecture. OpenFlow is the protocol used for managing the southbound interface of the generalized SDN architecture. It is the first standard interface defined to facilitate interaction between the control and data planes of the SDN architecture[9]. OpenFlow provides software-based access to the flow tables that instruct switches and routers how to direct network traffic. Using these flow tables, administrators can quickly change network layout and traffic flow. In addition, the OpenFlow protocol provides a basic set of management tools which can be used to control features such as topology changes and packet filtering. OpenFlow is usually implemented between SDN controller and OpenFlow enabled switches and it uses flow tables for matching the traffic or flow going through the network. There are two types of OpenFlow switches, OpenFlow only switches, it uses only OpenFlow Operations and OpenFlow Hybrid Switch it uses both OpenFlow and normal Ethernet operations. An OpenFlow switch consists of one or more flow tables and a group table. It performs packet look-ups and forwarding. The controller manages the OpenFlow-enabled switch using the OpenFlow protocol over a secure channel. Each flow table in the switch is made up of a set of flow entries in which each flow entry consists of match header fields, counters, and a set of instructions to apply to matching packets[10]. The OpenFlow channel is the interface that connects each OpenFlow switch to a controller. Using this interface, the controller configures and manages the switch.

V. SDN challenges

SDN is a promising solution for IT and cloud providers and companies but it faces certain challenges that could hinder its performance and implementation in cloud and wireless networks. Many fundamental issues still remain not fully solved. One of them are standardization and adoption of policies. The SDN controller must intelligently configure and validate network topologies to prevent manual errors and increase network availability. However, this is difficult to achieve because the controller is liable to a single point of failure. In traditional networks, when one or more network devices fail, network traffic is routed through alternative or nearby nodes or devices to maintain flow of data. However, in centralized controller architecture (SDN), only one central controller is in charge of the whole network. If this controller fails, the whole network may collapse. To address this challenge, IT organizations should ensure that the SDN controller should have the ability to support multiple-path solutions.[11] Also measuring Quality of Service (QoS) specifications like bandwidth utilization, packet loss, and delay require different techniques as compare to traditional network environments. This measurement will help for resource allocation in SDN.[1] This area also needs to be further research. SDN offers a platform for innovative networking techniques, however the shift from traditional networking to SDN can be a difficult process. Common concerns also include SDN interoperability with legacy network devices. There is also lack of experts for technical support. Existing deployments of SDN are often limited to small testbed for research prototypes.[4]

VI. Conclusion and Future work

This paper presents the basic architecture of SDN. This paper is a survey on the need and related work and the difficulties in implementation of a software defined network. After conducting a detailed survey on SDN, the concept of SDN along with its definitions this paper has been presented along with the benefits over the traditional architecture. In future, implementation of a Raspberry based firewall using on SDN will be done.

REFERENCES

1. A Survey on Resource Allocation in Software Defined Networks (SDN)- Urooj Zehra, Munam Ali Shah Department of Computer Science. 23rd International Conference on Automation & Computing, University of Huddersfield, Huddersfield, UK, 7-8 September 2017.
2. T. Sato, S. Ata, I. Oka, and Y. Sato, "Abstract model of SDN architectures enabling comprehensive performance comparisons," Proc. 11th Int. Conf. Netw. Serv. Manag. CNSM 2015, pp. 99–107, 2015.
3. F. Hu, B. Chen, X. Zhu, and H. Shen, "SDN-based Efficient Bandwidth Allocation for Caching Enabled Cognitive Radio Networks," pp. 1383–1390, 2016.
4. H. Tahaei, R. Salleh, S. Khan, R. Izard, K. K. R. Choo, and N. B. Anuar, "A multi-objective software defined network traffic measurement," Meas. J. Int. Meas. Confed., vol. 95, pp. 317–327, 2017.
5. A Survey on Software-Defined Networking- Wenfeng Xia, Yonggang Wen, Senior Member, IEEE, Chuan Heng Foh, Senior Member, IEEE, Dusit Niyato, Member, IEEE, and Haiyong Xie, Member, IEEE
6. D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, ―Software-defined networking: A comprehensive survey,‖ Proc. IEEE, vol. 103, no. 1, pp. 14–76, 2015.
7. Shu, Z., Wan, J., Lin, J., Wang, S., Li, D., Rho, S. and Yang, C., 2016. Traffic engineering in software-defined networking: Measurement and management. *IEEE Access*, *4*, pp.3246-3256.
8. Fast Failover and Switchover for Link Failures and Congestion in Software Defined Networks Ying-Dar Lin1 , Hung-Yi Teng1 , Chia-Rong Hsu1 , Chun-Chieh Liao1 and Yuan-Cheng Lai, IEEE 2016.
9. Software defined networking: State of the art and research challenges Manar Jammal a, , Taranpreet Singh a , Abdallah Shami a , Rasool Asal b , Yiming Li c. Computer networks journal, July 2014.
10. Software Defined Networking: Research Issues, Challenges and Opportunities, by Shailendra Mishra and Mohammed Abdul Rahman Al Shehri. Indian Journal of Science and Technology, August 2017.
11. M.A. Sharkh, M. Jammal, A. Shami, A. Ouda, Resource allocation in a network-based cloud computing environment: design challenges, Commun. Mag., IEEE 51 (11) (2013) 46, 52.