© 2018 IJSRSET | Volume 4 | Issue 1 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology

A Study of Software Defined Networking

Chahat Monga
Assistant Professor, Guru Nanak College, Department of Computer Science, Ferozepur, Punjab, India

ABSTRACT

The Internet, a collection of networks allows transmission of information from one place to another. Today's internet is growing fast, devices to the network for accessing and providing the services. The Traditional network makes it a difficult task to add and remove the network devices. The Static Nature of the conventional network makes it difficult to accomplish the dynamic computing and storage desires of big data centers and campuses. Software-Defined Networking is the new approach to networking. Software Defined Network model promises to simplify the network configuration and resource management. Software Defined Networking will replace the current network and fulfill the business needs via software rather than hardware. This paper introduces the concepts of Software Defined Network which will help to configure and manage the network needs.

Keywords : Software Defined Network, OpenFlow, Open Network Foundation, Northbound Interface, Southbound

I. INTRODUCTION

Today's Internet applications require the underlying networks to be fast, carry large amounts of traffic, and to deploy a number of distinct, dynamic applications and services. Adoption of the concepts of "inter-connected data centers" and "server virtualization" has increased network demand tremendously. In addition to various proprietary network hardware, distributed protocols, and software components, legacy networks are inundated with switching devices that decide on the route taken by each packet individually; moreover, the data paths and the decision-making processes for switching or routing are collocated on the same device. This situation is elucidated in Fig. 1. 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.

In addition, to implement network-wide policies and to support any new services, managers today have to con-figure thousands of network devices and protocols, which makes it difficult to apply a consistent set of QoS, security, and other policies. Networks become vastly more complex with the addition of thousands of network devices that must be configured and managed. These devices have their control and forwarding logic parts both integrated in monolithic, closed, and mainframe-like boxes. Consequently, only a small number of external interfaces are standardized (e.g., packet forwarding) but all of their internal flexibility is hidden.

II. SDN ARCHITECTURE

SDN architecture is dynamic, controllable, commercial, and adaptable; a designing model for the high bandwidth and dynamic nature of today's applications. This architecture decouples the network control (network intelligence) and data plane (perform forwarding functions) which enables the network control to be directly programmable, and the underlying infrastructure layer abstracted from applications. The SDN architecture consists three layers, Application Layer (Application Plane), Control Layer(Control Plane) and Infrastructure Layer(Data Plane).

The applications (Security, Bandwidth Management, and Load Management) exist in the application layer and send their network requirements send to control plan through a Northbound Interface (NBI). SDN uses NBI to communicate with the applications and business logic present in the application layer to help network executives to programmatically manage traffic and deploy the services.

In SDN, control plan is detached from the data plane. This control plane is a centralized software-based controller that maintains the centralized view of the overall network and enables the network administrator to direct the underlying system about how to forward the traffic.

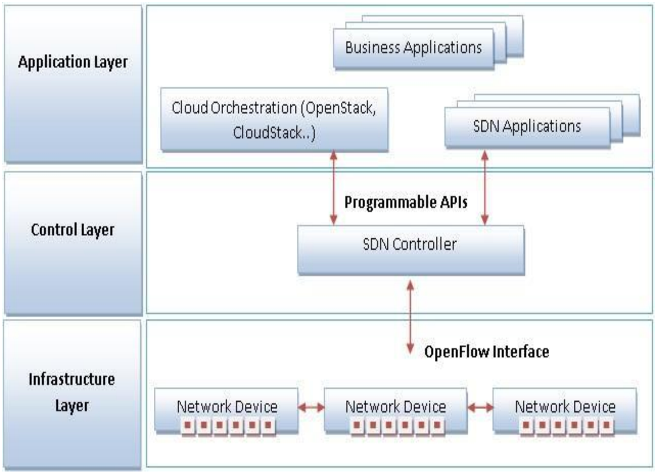
Software-Defined Networking (SDN) has emerged as the network architecture where the control plane logic is decoupled from the forwarding plane. SDN is a new approach for network programmability, which refers to the ability to control, change, and manage net-work behavior dynamically through software via open interfaces in contrast to relying on closed boxes and proprietary defined interfaces. The SDN framework enables centralized control of data path elements independently of the network technology used to connect these devices that can originate from different vendors. The centralized control embeds all the intelligence and maintains a network-wide view of the data path elements and links that connect them.

1) Control Plane: The control plane/controller presents an abstract view of the complete network infrastructure, enabling the administrator to apply custom policies/protocols across the network hardware. The network operating system (NOX) controller is the most widely deployed controller.

2) Northbound Application Interfaces: The "northbound" application programming interfaces (APIs) represent the software interfaces between the software modules of the controller platform and the SDN applications running atop the network platform. These APIs expose universal network abstraction data models and functionality for use by network applications. The "northbound APIs" are open source-based.

3) East-West Protocols: In the case of a multi-controller-based architecture, the East-West interface protocol manages interactions between the various controllers.

4) Data Plane and Southbound Protocols: The data plane represents the forwarding hardware in the SDN network architecture. Because the controller needs to communicate with the network infrastructure, it requires certain protocols to control and manage the interface between various pieces of network equipment. The most popular "southbound protocol" is the OpenFlow protocol. The following section explains OpenFlow and its architecture.



III. SDN CHALLENGES

SDN originally employed in data centers. It has helped companies adjust the network structure and requirements as per needs. Centralized Control of open flow architecture allows the network administrator to program the network behavior. When we try to implement SDN network we require to focus challenges of SDN.

A. *Controller Scalability:*

SDN controller is the brain of a network. SDN network implementation requires defining a number of controllers needed for network and their location. The Network may require a single controller or hierarchy of controller in control plan."when the network scales up in the number of switches and the number of end hosts, the SDN controller can become a key bottleneck." The latency increases during transmission of information between multiple nodes and the single controller. The number of switches flows and bandwidth increaser's number of requests will be pending to the controller, which may not able to process. The SDN controller [NOX] studies said that it can handle 30 k requests. For SDN network, a main challenge is a number of controller want and their localization.

B. *Convergence and Management:*

SDN OpeFlow architecture was originally developed for enterprise campuses networks that help researchers to experiment their protocols. In other way, SDN designed for the small network like the private network. However, to try and extend this architecture to large networks requires attend some issues, for example, the problem of Inter-domain Routing (Routing between two networks).

C. *Security:*

Security is a main concern in networking to detect and prevent anomalies. Networks mostly consist of host based and network based security mechanism which helps to network to detect intrusion be a part of their network or outside network. Current security solutions are difficult to manage, expensive, complex, inflexible. Programmable SDN requires intelligent security models because SDN systems handle by the network administrator who is configured the network as per requirements through software. Security needs to be developed in architecture to protect the controller securely.

D. *Controller Flexibility:*

OpenFlow proposed centralized controller is venerable to the network. SDN allow network administrator effectively program a network with software running on a central controller. A malfunction of the controller can negatively compromise a flexibility of the whole network. SDN network needs to focus on to define the way to handle the controller failure. Currently emerging SDN technique focus on separate the control plan from the data plan and provide programmable interfaces to fulfill the business needs.

IV. PROS AND CONS OF SDN MODEL

The centralized SDN model is based on a single cen-tralized controller that manages and supervises the en-tire network. This model is supported by the Open Net-working Foundation (ONF). The network intelligence and states are logically centralized inside a single decision point. OpenFlow is the o cial protocol for use by the cen-tralized controller to make global management and con-trol operations.

Since only one centralized controller is used to pro-gram the entire network, it must have a global vision about the load on each switch across the routing path. It must also keep track of which flow inside which router presents a bottleneck on certain links between the remote SDN nodes. Additionally, the controller communicates with OpenFlow switches to collect statistics, errors and faults from each network device, and sends these data to the management plane. The latter is often a software composed of a database module and analytic algorithms that can detect the switch overloads and predict the future loads that may occur in the network.

Although the centralized control plane promises a single point of management and better control over the consistency of the network state, it incurs several key limitations. First, the controller needs to update OpenFlow switches more frequently than traditional routers. Thus, the topology discovery produces higher overload because all ports must be scanned linearly, which increases the response time and may impose a higher overload. For ex-ample, the controller may classify flows with different priorities into multiple classes, where each class requires a specific QoS setting that should be approved individually at setup time for every new flow received by OpenFlow switches. Such an approach can incur substantial flexibility and robustness challenges for large-scale net-works.

V. CONCLUSIONS

Most researchers argue that the current Internet architecture is inadequate and has reached a tipping point where most of the time and e ort is spent addressing existing flaws rather than developing new ideas. To address the challenge of ossification of the Internet, researchers have started to focus on redesigning the overall architecture by breaking the tight integration of the forwarding and control plane implementations. As a result, major research projects focus on the SDN architecture to construct and present a logically centralized map of the net-work. The next-generation of SDN networks will benefit not only from the simplicity of the implementation, but also from the fact that maintaining and updating applications will be easier as well.

We argue that while these research e orts are important, they need to occur in the context of overall net-work programmability and scalability goals. Although OpenFlow, which is a prominent SDN implementation, promises a flexible, open, and dynamic flow transmission mechanism, it also poses a set of challenges in terms of network virtualization, mobility management, operation, which will require coordinated attention from the research community for its success and wide acceptance.

VI. REFERENCES

[1]. A. Metzger, C. C. Marquezan, Future internet apps: The next wave of adaptive service-oriented systems, in: Ser-viceWave, 2011, pp. 230-241.

[2]. C. C. Marquezan, A. Metzger, K. Pohl, V. Engen, M. Boniface, S. Phillips, Z. Zlatev, Adaptive Web Ser-vices for Modular and Reusable Software Development: Tactics and Solution, IGI, 2012, Ch. Adaptive Future In-ternet Applications: Opportunities and Challenges for Adaptive Web Services Technology.

[3]. N. McKeown, Software-defined networking, INFOCOM keynote talk, Apr.

[4]. H. Kim, N. Feamster, Improving network manage-ment with software defined networking, Communications Magazine, IEEE 51 (2) (2013) 114-119.

[5]. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, Openflow: enabling innovation in campus networks, ACM SIGCOMM Computer Com-munication Review (2008) 69-74.

[6]. ONF, The openflow 1.3.1 specification, Tech. rep.

[7]. C. Lam, H. Liu, B. Koley, X. Zhao, V. Kamalov, V. Gill, Fiber optic communication technologies: What's needed for datacenter network operations, IEEE Communica-tions Magazine.

[8]. L. Suresh, J. Schulz-Zander, R. Merz, A. Feldmann, T. Vazao, Towards programmable enterprise wlans with odin, HotSDN '12, 2012, pp. 115-120.

[9]. M. Bansal, J. Mehlman, S. Katti, P. Levis, Openradio: a programmable wireless dataplane, HotSDN '12, 2012, pp. 109-114.