ISSN 2348-1218 (print)
International Journal of Interdisciplinary Research and Innovations ISSN 2348-1226 (online)
Vol. 7, Issue 3, pp: (1-6), Month: July - September 2019, Available at: www.researchpublish.com

Virtualization of Traditional Networks using SDN

N.Sabitha[1], H.Jayasree[2], A.V. Krishna Prasad[3]

[1] Assistant Professor, [2, 3] Associate Professor

[1,2,3] Dept. of CSED, MVSR Engineering College, Nadergul, Hyderabad

Abstract: **The advancement in technology has been exponentially rapid. Computer networks have lead new challenges for the future internet usage in terms of accessibility, bandwidth and dynamic management of networks. Available techniques are not sufficient to tackle the increasing challenges. There is a need for efficient network technology that supports dynamic nature of future Internet applications and network devices. Software Defined Network is the solution to the limitations of the current networks. SDN is a perception which has the potential to change the networks and the way they are designed, built and operated. The goal of SDN is to improve network control by enabling enterprises and service providers to respond quickly to changing business requirements. In this paper we discuss the requirements, architecture and applications of the SDN.**

Keywords: **SDN, Open Flow Protocols, NBI, CDPI.**

1. INTRODUCTION

Computer networks are the backbone of our everyday activity. The network connectivity has been widely expanded to integrate with wide variety of electronic devices. There is a huge drift in data traffic in large volumes carrying audio, video, user generated data and big data. The network architecture has evolved from few computers and local networks to World Wide Web which could be reaching globally and independently. Network devices like routers, switches, gateways, multiplexers etc. are counted in millions. Hence maintenance of such a vast network which is dynamically changing according to the business needs and user demands is very difficult [1].

1.1 Traditional networks

Network devices like routers, switches, firewalls etc. are used for specified tasks in networking. These networks are bottlenecked by their static nature of the architecture. The traditional network has some limitations like scalability, complexity, devices from multiple vendors [1]. The fundamental components of traditional network are Control plane and Data plane shown in Fig 1.

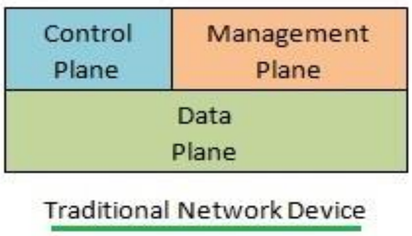

Fig 1: Traditional Network Device

The data plane is an abstract layer which moves data from one link to another link, using an end to end communication between client and server. The control plane defines how connections are to be made, how the data has to be forwarded, what rules to apply and which route to follow. These layers are embedded inside the devices. Network administrators configure the control plane using command line interface and vendor specific interface to define rules how data has to be moved. The problem in this model is; as control flows and data flows are already defined any modification requisite can be carried out only through reconfiguration of the device. As the increase in data traffic and network changes, it has become necessary for the network to adapt quickly and efficiently to these changes.

Recently, Service Oriented Architecture [2] (SOA) has changed the concept of how computer exchange the information via services without human interventions, by virtualizing both the application and the resources. The same thoughts are now applied successfully to the world of computer networks for network resource virtualization.

1.2 Software Defined Networks

Software Defined Network is the solution to the limitations of the current networks. SDN is a perception which has the potential to change the networks and the way they are designed, build and operated. It is a technology which works on managing network devices. SDN gives a complete view of the network where configuring, monitoring, trouble shooting of network devices can be done with ease.

SDN can be defined by two characteristics namely decoupling of control and data planes and programmability of the control plane [2]. By separating the network form the hardware, policies no longer have to be executed on the hardware itself. So, the use of a centralized software application function as the control plane makes network virtualization possible. A detailed SDN architecture is discussed in the III section.

2. EVOLUTION OF SDN ARCHITECTURE

SDN supports both centralized and distributed controller models. Both the models have different infrastructure elements and requirements to consider. This section describes the SDN models along with their advantages and disadvantages. Also, the hybrid SDN model is described which combines the benefits of both the approaches.

2.1 The Centralized SDN Model

In centralized SDN [9] architecture a single centralized controller manages and supervises the entire network with a single decision point. Since a single centralized controller is used to program the entire network, so it must have a global view of the loads on each switch across the routing path. Also, it must keep a track of which flow inside which router is presenting a bottleneck on which link.

The controller communicates with OpenFlow [4], [9] switches to collect network statistics from the network devices, and sends this data to the management plane. The management plane is software that consists of a database module and analytic algorithms that detects the switch overloads and predicts the future loads that may occur in the network.

Although the centralized control plane has an advantage of a single point of management and better control over the network, it incurs several limitations:

(a) The controller needs to update OpenFlow switches more frequently than traditional ones

(b) The controller determines the future path for the flow hop-by-hop. When a new flow is to be programmed, the controller needs to contact all the switches in the path, which is a scalability challenge for large networks

(c) The centralized controller represents a single point of failure which makes the network highly vulnerable to intrusions and attacks

(d) The services like intrusion detection, firewall, network virtualization, and load balancing, need to coordinate their activities in the control plane to achieve complicated control objectives and maintain a global view of the entire network.

2.2 The Distributed SDN Model

The distributed SDN [9] model focuses on eliminating the single point of failure and enabling scale up by sharing the load among distributed controllers. Distributed SDN control planes are designed to be more responsive to handle local network events in data centers in particular, for multi-domain SDNs. A distributed controller is more responsive and can react faster and efficiently while handling global events [3].

The two-layers of hierarchical distributed controllers: (i) bottom-layer consists of a group of locally non-connected distributed controllers each managing one or more switches without any global knowledge of the network and (ii) the top-layer consists of a logically centralized root controller that manages the network-wide state. A master controller is selected based on the load in the network so that if the loads increases, the master node can be switched to a less loaded one.There are several key challenges that must be addressed to improve scalability and robustness of networks, it incurs several limitations:

(a) The above approaches require a consistent global view in all controllers. The mapping between control planes and forwarding planes must be programmed instead of the present static configuration, which can result in uneven distribution of load among the controllers.

(b) Finding an optimal number of distributed controllers that ensure linear scale up of the SDN network is hard.

(c) There is a need to synchronize the local algorithms and the distributed events to provide a global view of the network.

2.3 The Hybrid SDN Architecture

To tackle the limitations in above approaches, hybrid SDN [9] architectures are being taken into consideration. However, a critical challenge arises when determining how much of network abstraction modules can be centralized and efficiently designed to support logically centralized control tasks, and at the same time provide physically distributed protocols. Consequently, to take the advantages of both the centralized and the distributed architectures, a hybrid control plane is required to achieve such coordination.

The hybrid SDN model is influenced from the benefits of the simple control of managing specific data flows as in the centralized model with the scalability and flexibility of the distributed model. It requires various components to coordinate the communication between SDN controllers. The network administrators will require standard interfaces, and policies to manipulate and interact with the control planes in distributed environments.

3. SDN ARCHITECTURE

SDN can control and manage network behavior dynamically through software via open interfaces. SDN architecture [8] follows Open Flow Standards. Open Flow architecture [4], [8] includes three important components.

1) Switches: It defines an open source protocol to monitor and change the flow tables in switches/ routers. Each switch contains three major components: **a) Flow tables:** it consists of action field associated with every flow entry communication **b) Channels:** provides a link for the transmission of commands and packets between controllers and switches **c) Open Flow protocols:** enables the controller to communicate with any router or switch.

2) Flow-entries: Each flow-entry includes an action for that flow item. The Open Flow switches support the following actions: (a) sending the packets to the respective ports (b) encapsulating the packets and sending to a controller and (c) dropping the packets.

3) Controllers: A controller can update, add, or delete flow entries from the flow table on behalf of the user's testing. A static controller is usually a simple software unit running on a system to statically establish a path between groups of network devices during a scientific experiment.

The Open Flow protocol[4] can be used in SDN technologies. The SDN architecture is: Directly programmable, agile, centrally managed, programmatically configured, and Open standards-based and vendor-neutral.

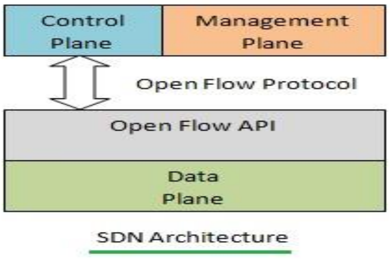

Fig 2: SDN Architecture

The SDN architecture [8] consists of three major parts: application, control plane, and data plane shown in Fig 2. The application label uses the decoupled nature of SDN to achieve specific goals, such as a security mechanism, a network measurement solution etc. The Applications communicate with a SDN controller at the control plane via the northbound interface to enforce their policies in the data plane without directly interacting with the data plane. The interface between the control and data plane is supported by southbound APIs, there the SDN controller uses these APIs to communicate with the network devices in the data plane. The control plane manipulates forwarding devices using the SDN controller to achieve the specific goal of the target application. The controller uses the southbound interface to connect to the data plane. The data plane handles the actual packets based on the configurations that are manipulated by the controller.

3.1 Architectural Components

The architectural components [2] of SDN are

SDN Application: SDN Applications are programs that explicitly, directly, and programmatically communicate their network requirements and desired network behavior to the SDN Controller via a northbound interface (NBI). An SDN Application consists of one SDN Application Logic and one or more NBI Drivers.

SDN Controller: The SDN Controller is a logically centralized entity in charge of (i) translating the requirements from the SDN Application layer down to the SDN Data paths and (ii) providing the SDN Applications with an abstract view of the network. An SDN Controller consists of one or more NBI Agents, the SDN Control Logic, and the Control to Data-Plane Interface (CDPI) driver.

SDN Data path: The SDN Data path is a logical network device that exposes visibility and uncontested control over its advertised forwarding and data processing capabilities. The logical representation may encompass all or a subset of the physical substrate resources. An SDN Datapath comprises a CDPI agent and a set of one or more traffic forwarding engines and zero or more traffic processing functions. These engines and functions may include simple forwarding between the data path's external interfaces or internal traffic processing or termination functions.

SDN Control to Data-Plane Interface (CDPI): The SDN CDPI is the interface defined between an SDN Controller [6] and an SDN Data path, which provides at least (i) programmatic control of all forwarding operations, (ii) capabilities advertisement, (iii) statistics reporting, and (iv) event notification. One value of SDN lies in the expectation that the CDPI is implemented in an open, vendor-neutral and interoperable way.

SDN Northbound Interfaces (NBI): SDN NBIs are interfaces between SDN Applications and SDN Controllers and typically provide abstract network views and enable direct expression of network behavior and requirements. This may occur at any level of abstraction (latitude) and across different sets of functionality (longitude).

SDN Control Plane:

Centralized - Hierarchical – Distributed: The implementation of the SDN control plane can follow a centralized, hierarchical, or decentralized design. Initial SDN control plane proposals focused on a centralized solution, where a single control entity has a global view of the network. While this simplifies the implementation of the control logic, it has scalability limitations as the size and dynamics of the network increase.

Controller Placement: A key issue when designing a distributed SDN control plane is to decide on the number and placement of control entities. An important parameter to consider while doing so is the propagation delay between the controllers and the network devices, especially in the context of large networks. Other objectives that have been considered involve control path reliability fault tolerance and application requirements.

4. CHARACTERISTICS OF SDN

Software Defined Networking is characterized by five fundamental traits: plane separation, a simplified device, centralized control, openness, and network virtualization.

A. Plane Separation: The very key characteristic of SDN is the decoupling of the forwarding plane and the control plane. The Forwarding plane contains the forwarding tables and the logic for dealing with incoming packets based on MAC address and IP address. The main logic that is used to control the forwarding plane resides in the control plane. The control plane determines how the forwarding tables in the data plane be configured. In SDN, the control planes of all the switching devices are moved onto a centralized controller [10].

B. Simplified device and Centralized Controller: In SDN, instead of running thousand lines of code of complicated control plane software, that software is removed from the device and placed in a centralized controller [7]. The device is allowed to behave autonomously by a centralized system on which management and control software run. The controller provides the instructions to these simplified devices, when needed, in order to allow them make faster decisions about how to deal with the arriving packets [10].

C. Openness: A basic characteristic of SDN is openness that its interface should remain well documented, standard, and not proprietary [10]. Individuals can take advantage of this capability in order to test new ideas, resulting in better and faster technological advancement in the functioning of networks.

D. Network Virtualization: The idea of virtualization is to create a higher-level abstraction that runs on top of the actual physical instance being abstracted. With the help of network virtualization, the network administrator is able to create, expand and contract a network anytime and anywhere as per the requirements. Network Virtualization enables coexistence of multiple network instances on a shared physical infrastructure, thus, NV can be used to run an SDN solution.

5. ADVANTAGES OF SDN

Major advantages of SDNs include:

1) Intelligence with Speed: SDN is intelligent enough to efficiently distribute the workload via powerful control plane resulting in high speed transmissions and making more efficient use of the resources [3].

2) Network Management Made Easy: The administrators have a centralized control over the network and can change the network characteristics as per the demand of environment. This enables administrators to modify the network configurations with ease.

3) Virtual Application: Virtual application networks make use the virtualization of network resources to hide the low level physical details from the user applications and allow the users to reconfigure the network tasks easily.

4) Content Delivery: The ability to direct and automate data traffic makes implementing quality of services (QoS) for voice over IP (VOIP), video and audio transmissions much easier. Software defined networking provides a seamless experience for end-users streaming high quality audio and video.

5) Lessen Capital Expenditure: By implementing software defined networking, enterprise businesses can easily optimize existing network devices. Existing hardware can be repurposed to follow the instructions of a SDN controller, and more cost efficient hardware can be deployed with greater effect.

6) Centralization: SDNs can speed service delivery and provide more agility for both virtual and physical network provisioning, all from a central location.

7) Management: By implementing SDN, IT teams are able to change network configurations with no effect to the network. SDN supports the management of physical and virtual switches and network equipment from a single centralized controller, something that cannot be done using simple network management protocol (SNMP).

6. APPLICATIONS

Software-defined networking has applications in a wide variety of networked environments.

Cellular Networks:

Applying the SDN principles to cellular networks solve some of the deficiencies. First of all, decoupling the control from the data plane and introducing a centralized controller that has a complete view of the whole network allows network equipment to become simpler and therefore reduces the overall infrastructural cost[5]. Also, operations like routing, real-time monitoring, mobility management, access control and policy enforcement can be assigned to different cooperating controllers making the network more flexible and easier to manage. In addition, using a centralized controller acting as an abstract base station simplifies the operations of load and interference management, no longer requiring the direct communication and coordination of base stations. Instead, the controller makes the decisions for the whole network and simply instructs the data plane (i.e. the base stations) on how to operate. One final advantage is that the use of SDN eases the introduction of virtual operators to the telecommunications market, leading to increased competitiveness.

Data Center Networks

One of the most important requirements for data center networks is to find ways to scale in order to support hundreds of thousands of servers and millions of virtual machines. However, achieving such scalability can be a challenging task from a network perspective. Initially the size of forwarding tables increases along with the number of servers, leading to a requirement for more sophisticated and expensive forwarding devices. Moreover, traffic management and policy enforcement can become very important and critical issues, since datacenters are expected to continuously achieve high levels of performance. Additionally, it becomes increasingly difficult to make the data center operate at its full capacity, since it cannot dynamically adapt to the application requirements. The advantages that SDN offers to network management come to fill these gaps. By decoupling the control from the data plane, forwarding devices become much simpler and therefore cheaper. At the same time all control logic is delegated to one logically centralized entity. This allows the dynamic management of flows, the load balancing of traffic and the allocation of resources in a manner that best adjusts the operation of the data center to the needs of running applications, which in turn leads to increased performance. Finally, placing middle boxes in the network is no longer required, since policy enforcement can now be achieved through the controller entity.

Enterprise Networks:

Enterprises often run large networks, while also having strict security and performance requirements. Furthermore, different enterprise environments can have very different requirements, characteristics, and user population. Adequate management is critically important in Enterprise environments, and SDN can be used to programmatically enforce and adjust network policies as well as help monitor network activity and tune network performance. Additionally, SDN can be used to simplify the network.

7. CONCLUSION

Recent developments in ICT domain such as mobile, multimedia, cloud and big data are demanding for more convenient Internet access, more bandwidth from users, as well as more dynamic management from service providers. SDN is considered as a promising solution to meet these demands. In this paper, we have presented the concept of SDN and highlighted benefits of SDN and its applications.

REFERENCES

[1] SOFTWARE DEFINED NETWORKS - A SURVEY. Dr.B.Sujatha, Mr.K.Ashokkumar2, First International Conference on NexGen Technologies@2018.ISBN:978-93-86171-90-0

[2] Software-Defined Networking: A Survey Sakshini Hangloo1, Sudesh Kumar National Conference on Recent Advances in Computer Science and IT (NCRACIT) International Journal of Scientific Research in Computer Science, Engineering and Information Technology © 2018 IJSRCSEIT | Volume 4 | Issue 1 | ISSN: 2456-3307

[3] Marc Mendonca and Xuan-Nam Nguyen et al., “A Survey of Software-Defined Networking: Past, Present,and Future of Programmable Networks”, Communications Surveys and Tutorials, IEEECommunications Society, Institute of Electrical and Electronics Engineers, 2014, 16 (3), pp.1617-1634.

[4] Open Networking Foundation. Software-defined networking: The new norm for networks. Available from: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf Software-Defined Networking: Challenges and research opportunities for Future Internet

[5] Software Defined Networks A Comprehensive Approach M.O.S. VolkanYazici, A.O. Ercan, Controlling a software-defined network via distributed controllers, in: 2012 NEM Summit Proceedings.

[6] Cao, Z. Li, Analysis of sdn controller cluster in large-scale production networks, Tech. rep. 00, 2013.

[7] B. Heller, R. Sherwood, N. McKeown, The controller placement problem, in: Proceedings of the First Workshop on Hot Topics in Software Defined Networks, HotSDN ’12, 2012

[8] T. Nadeau, P. Pan, Framework for software defined networks, Internet-Draft draft-nadeau-sdn-framework-01, Internet-Draft, October 2011.

[9] Akram Hakiri [a,b], Aniruddha Gokhale[c] , Pascal Berthou[a,b], Douglas C. Schmidt[c] , Gayraud Thierry[a,b] , “Software-defined Networking: Challenges and Research Opportunities for Future Internet”, Preprint submitted to Elsevier October 12, 2014

[10] Gary Lee, “Software Defined Networking” Cloud Computing – Science Direct 2014