Overview and Analysis of Software Defined Networks (SDN): The Next Big Thing in Digital Networks

Amogh Santosh Pai Raiturkar
Department of Computer Engineering
Padre Conceicao College of Engineering, Verna, Goa, India

Andre Exequiel Anthony Pacheco
Department of Computer Engineering
Padre Conceicao College of Engineering, Verna, Goa, India

Abstract- Software-Defined Networking (SDN) has received a great deal of attention in industry since 2008. Researchers, Network operators are trying to establish new standards and provide guidelines for proper implementation of SDN. In this paper, we will discuss about architecture and its models along with various specific SDN Protocols, security challenges, state of art in programming trends and challenges that are faced in SDN environments. We will also discuss about research challenges and future developments on techniques, specifications and other methodologies.

Keywords – Software-Defined Networking, SDN, Implementation, Protocols, Security, Challenges

I. INTRODUCTION

Software-Defined Networking (SDN) is one of the rapidly emerging next-generation networking at the forefront. With the function of decoupling of the control and data planes in network switches and routers, SDN enables the rapid innovation and optimization of routing and switching equipment functions. SDN greatly simplifies network management by offering administrators network-wide visibility and direct control over the underlying switches from a centralized controller. SDN also integrates the concept of programmability in network architecture in order to offer better network management. In such case, Open Flow has been considered as widely accepted solution to implement SDN. Open Flow is a protocol that defines an open standard interface for SDN and uses a programmable counter to communicate with the plane, manage the network and possibly receive instructions from a network application. However, Full development and deployment of SDN software applications in staging and production environments remains a challenge for network operators. The Open Flow protocol also provides foundational element for building SDN solutions. In this Paper, we will discuss in brief about Software Defined Networking and its features with analysis.

II. SDN ARCHITECTURE

Figure below shows architecture of SDN and its three components.

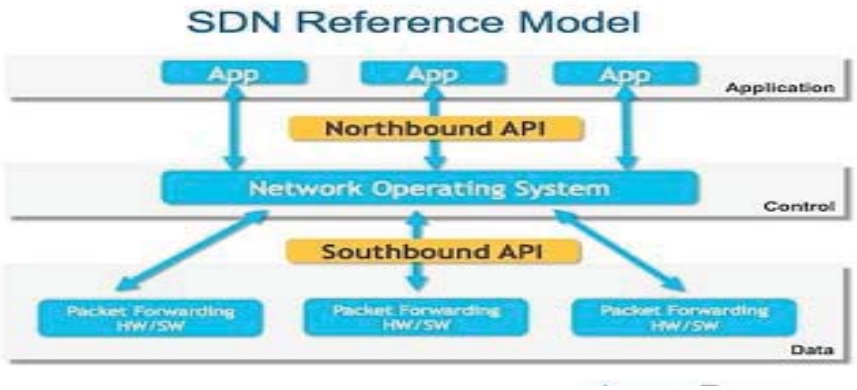

Figure 1. Architecture of SDN

The SDN architecture is divided into 3 layers: - Application Layer, Control Layer and Data Layer

- Application Layer: It is the topmost layer in the architecture. Their main responsibility is to communicate with the SDN controllers that are present in the Control Layer in order to communicate the application behaviors and needed resources via Application Programming Interface (API). They then collect the data and other needed information from this control layer viz. Controller for decision making purposes. This layer typically includes business applications that are required in large data centers.
- Control Layer: It is situated between the Application Layer and Data Layer. It receives the resources and other instructions from the Application Layer thereby taking necessary actions and returning the result back to the Application Layer. This layer typically provides the abstract view of network and also informs the events of what is happening.
- Data Layer: It is the bottommost layer of SDN architecture. Its main job is to control the forwarding and processing of the data path thus sending or forwarding packet hardware/software to the network.

To communicate between these 3 layers, two interfaces are defined mainly Northbound Interface and Southbound Interface.

- Northbound Interface: It is defined as the connection between Application Layer and Control Layer for higher layer protocols. They are mainly involving in describing the areas of protocol supported communication between control layer and application layer. It is generally considered as going upward.
- Southbound Interface: It is defined as the connection between the Control Layer and Data Layer. It enables the communication between Control Layer and network nodes at Data Layer in order to define network flow and implement the requests given by Northbound API. In SDN, Southbound Interface is Open Flow Protocol specification and generally considered as going downward.

III. MODELS OF SDN

There are mainly 3 architecture models of SDN that we will discuss now. They are as follows:-

A. Network Virtualization Model –

Network Virtualization Model developed in 2012 has a major objective of eliminating the restrictions on Local Area Network (LAN) that is present in Ethernet and other cable standards. This model also address various issues such as reliability, availability, scalability that are present in various Ethernet based network architectures. To accomplish both these functions, cloud building software like Open stack are used where they can be modified which can be further used on various virtual LAN (VLAN) and then are made to run on top of Ethernet. In this way many networks or rather many virtual networks could be created.

The major advantages of this model are that integration of network with cloud service is better and secondly it supports multi-tenant cloud without any changes in the network.

The major disadvantages of this model is that it appears as traffic to many network devices and due to this it becomes very difficult to prioritize or generate reports for individual virtual networks.

B. Evolutionary Model –

This model was introduced to enhance software control and its operations of the network within the limits of current networking arena. This is carried out by partitioning the network into virtual communities and then managing it thereby combining the solutions through various tools or virtual interface. Some Networking companies still use this model approach but not very feasible and available in all devices.
The major disadvantage is that it requires special integration between the management system and cloud networking interfaces.

C. Open Flow Model –

The final model which is being currently highlighted and rapidly emerging is Open Flow model. It defined as the communication protocol that enables the SDN controllers to interact directly with switches and routers in order to adapt to changing business requirements. It also enables controllers to determine the path of packets across network of switches or routers. The routing decision are either done periodically or adhoc basis which are then deployed to switch flow table thereby leaving actual forwarding of packets. However, different packets identified by switch will be given to the controller.
The advantage of this model is that it improves reliability, network availability and reducing costs and other expenses.
The major disadvantage is lack of proper functional details of the components. Also researchers have found that in some cases they tend to be very less secure; in the sense attacks such as man in middle attack and single point of attack and failure can occur. However instead of these disadvantages, most of networking companies have been attracted to using Open Flow model particularly when deploying new equipment. But as always careful operations and conditions should be considered when choosing the appropriate model.

IV. SDN BENEFITS AND CHALLENGES

A. SDN BENEFITS

The various benefits of SDN are as follows:

- It is highly efficient in optimizing services and existing applications and infrastructures
- It provides increased productivity and enhanced quality as compared to other digital networks.
- It is highly scalable and involved in rapidly growing of existing and new applications and services
- It has simplified network operations thus saving large amount of cost thereby benefitting business oriented people
- It acts as central management where SDN tries to deliver all the required needs in one go, in order and in time thus enabling to control every piece of network.
- It helps in increasing bandwidth utilization in WAN, LAN, VLAN thus reducing any kind of bottleneck caused
- It helps in forwarding virtual packets to various devices irrespective of hardware or software oriented thereby reducing downtime and large amount of overhead.

B. SDN CHALLENGES

Given above the benefits of enhanced architecture, improved performance, and encouraged innovation, there are various issues or challenges SDN has faced. First and foremost, the challenge that SDN faces is with respect to debugging; in the sense that debugging is difficult in SDN as compared to other networking devices. Secondly, many networking companies have desire of using various SDN services but in return they want assurances that the services running on current network shouldn't be disrupted or affected as whole. However, one partial solution for above is maintaining or using one physical network and two logical networks where first logical network will support the existing services and second logical network will be supporting new services. Thirdly, another SDN challenge can be related with business adoption challenges which are encountered in various networking sectors. The main adoption challenges arising in mindset of business are with respect to new service opportunities, integration and testing before bringing it live, security vulnerabilities and SDN help development. Lastly, the standardization issues of SDN such as Open Flow driver and standard northbound API are missing in SDN Controllers. Also lacks of technical experts, along with privacy of centralized control are the issues or challenges dealt in level of standardization in networking arena.

V. SDN SECURITY CHALLENGES

As SDN is known as new design paradigm in networking field and in digital networking technologies, the security challenges in SDN also arises and they thus needed to be addressed from scratch.
First of all, as seen in above SDN architecture, control plane and data plane are separated. This is the first clearing centralized point of attack in SDN. Secondly, in SDN unlike other networks, entire network can be compromised by compromising the controller. Thus, attacker need not follow any attacking strategies in order to attack the network. Another main security issue with respect to SDN architecture is the placement of the Southbound Interfaces and Northbound Interfaces. In Southbound Interface, there is high possibility that device could degrade the availability, performance and integrity of the network thus making it largely insecure. In case of Northbound Interface, attackers can install special application programs/ software that will reprogram the network entirely and thus attackers can make network do something unexpected by sending crafted packet stream to the router via network. Thus, SDN security should be managed and controlled in SDN environment. They can be either placed in servers, storages, computing devices or placed within the network.
Researchers however due to this security issue have introduced software security software called Software Defined Security (SDSec). It separates the control plane and data plane and try to manage network services by decoupling the networking functions such as Intrusion detection, Firewalls. This software security is also follows in example of Network Function Virtualization (NFV).

VI. SDN PROTOCOLS

The various specific SDN protocols which are of common interest are:

A. Open Flow Protocol –

Open Flow Protocol is defined as open standard protocol that allow researchers to run the experimental protocols without need to expose the internal workings of network devices. They are used in various applications such as IP based cellular networks, high-security based networks and virtual machine mobility. Open Flow Protocol are broken down into 4 components mainly; A) Message Layer that defines the structure and semantics for all kind of messages in various networking devices. They have ability to construct, copy, compare and print messages. B) State Machine Layer that describes flow control, reliability, capability and delivery of packets. C) System Interface Layer that defines the interaction of protocols with each other and with the outside world. They mainly use TCP, TLS as transport channels. D) Configuration Layer that mainly includes or rather provides default buffer sizes and intervals to X509 Certificates. This protocol is most widely used and preferred protocols by researchers around the world and in networking arena.

B. Border Gateway Protocol (BGP)-

Border Gateway Protocol (BGP) is defined as protocol that exchanges all routing information between two or more gateway hosts in a network. They can be classified on basis of either Path Based Vector Protocol or Distance Vector Protocol. In BGP, each gateway has its own router .Each router will maintain routing information table where each table contain known routers, addresses of routers and cost of path to each router. This information is needed to allow or calculate the best and shortest route to the destination.

C. NETCONF –

NETCONF is one that provides an administrator or network engineer with a secure way to configure a firewall, switch, router, or other network devices based on remote procedure call (RPC) method. They are built on four-layer approach. They are Secure Transport Layer that provides authenticity and integrity, Message Layer that contains exchange of RPC messages, Operations Layer that contains various operations performed by RPC Call and Content Layer that provides structure and semantics of the data. NETCONF is and will be the next emerging networking standard that has been considered by the researchers since 2014.

D. Extensible Messaging and Presence Protocol (XMPP) –

XMPP is a protocol that is based on Extensible Markup Language used for instant messaging and online presence detection. XMPP nowadays have powered various emerging technologies like Internet of Things (IOT), social etc. They were originally called Jabber. They provide high flexibility and security. However because of limited data transfer, non-support of end to end encryption and text based Protocol they were not taken much into account by engineers.

E. Open vSwitch Database Management Protocol (OVSDB)-

OVSDB is an Open Flow Protocol that manipulates configuration of Open vSwitch present in Ethernet switches. They store both provisioning and operational state in a network. It makes use of Java Script Object Notation (JSON) that will provide with network automation and supports distribution across multiple servers.

VII. PROGRAMMING LANGUAGES FOR SDN

A. Frenetic –

Frenetic is high level language used in programming the distributed collections of the network routers and switches. It is in form of declarative query language where operators can classify and aggregate network traffic and use a functional reactive special library to describe high level packet forward policies and conditions. They have 2 levels of abstractions namely : i) source level operators for stream network traffic and ii) run time system handling all the details of removing low-level rules on routers and switches.

B. Flog –

Flog uses logic programming as its paradigm that result in a declarative language. The relation of Flog with Frenetic involves the idea that SDN applications have various components to be developed.

C. FatTire –

FatTire is a high level programming language providing constructs in writing programs in terms of paths through the network. The main features of FatTire are that they are expressive, efficient and correct.

D. Pyretic –

Pyretic was created with major objective to provide abstractions involving the building of SDN applications with independent modules that jointly manage network traffic. Pyretic is an imperative, domain specific language coded with Python language. It enables the programmers to specify the network policies at high level of abstractions.

E. NetCore –

NetCore is high level declarative language used in expressing packet forwarding on SDN. It aims at enabling programmers to construct and reason about policies in natural way. It analyses the programs and divides into two parts wherein one part runs on the switches while other part runs on the controller.

F. Nlog –

Nlog is a declarative language that computes forwarding state separating the logic specification from the controllers that implements such logic.

G. Flowlog -

Flowlog resembles SQL in its design that provides an unified abstraction for both control-plane and data-plane. It has limited expressivity in order to facilitate the reasoning about correctness and rules proactively installed into switches. They also discover bugs in SDN applications.

H. Merlin –

Merlin is declarative language that has been created to address problems related to bandwidth and other packet-processing functions. They have high level components for classifying packets, controlling forward packets, defining bandwidth properties.

I. Kinetic –

Kinetic provides abstractions for automating changes in network policy in response to dynamic network conditions. This makes it possible and verifies that if such changes will satisfy network operator requirements and how it should react to network conditions.

VIII. RESEARCH CHALLENGES AND FUTURE DIRECTIONS

Software Defined Network and OF have confronted with some challenges. Security needs to be everywhere within SDN: A) Architecture and its controller, applications, devices, channels (TLS with plain text) and flow table, B) services (to Protect availability), C) connected resources and D) information. Also a robust policy framework in order to check and balance controllers, a recovery, report policy and security deployment is still very much up for grabs. The solutions should be simple to deploy and maintain, cost effective and assuredly secure. A new category called software defined security, an example of network functions virtualization (NFV) delivers network security enforcement by separating the security control plane from processing and forwarding plane.

Besides security, availability (controllers' existence), flexibility, controllers and applications compatibility, link and controller reliability are considerable issues. A centralized controller could recover itself in some processes by using backup flows in a way that is not as faster as it is expected.

Capital and operational expenses called CAPEX and OPEX is another challenge debated by OF adapters. Availability and reducing system bottleneck would increase CAPEX, however adapters believe that using software defined network and OF reduce the CAPEX. OPEX in SDN would be decreased by diminishing the number of human based configuration, time and error prone fields.

Besides these challenges there are some implementation issues for example having 40+ matching fields in a flow, several tables and their large number of flow entries, instructions and actions, flow level programming and controllers' own way programming that must be considered. Also lack of standard APIs in case of overlapping domain among controllers, necessity of encryption APIs for data plane packets, injection APIs for packet and instantaneous APIs for services like IDS and firewall on a switch, absence of operations in case of absence of controller, existence of other packet format are too responsible.

IX. CONCLUSIONS

In this paper, we provided an overview of software defined networks (SDN) along with the architecture of SDN. We also discussed about various architectural models along with their benefits and disadvantages. Then we discussed about various benefits and challenges encountered in SDN. We also described about various SDN security challenges and counterproductive ways for it. Finally, we discussed specific SDN protocols and various Programming languages in SDN along with, we pointed out various research challenges and Future directions in this paper.

REFERENCES

[1] Elby, Stuart. "Software Defined Networks: A Carrier Perspective." Proc. of Open Networking Summit (2011).J. U. Duncombe, "Infrared navigation—Part I: An assessment of feasibility," IEEE Trans. Electron Devices, vol. ED-11, pp. 34-39, Jan. 1959.

[2] Wang, Anjing, et al. "Network Virtualization: Technologies, Perspectives, and Frontiers." Journal of Lightwave Technology 31.4 (2013): 523-537.

[3] Open Networking Foundation , "Software-defined Networking: The New norm for Networks.", ONF White Paper, (2012) , available online: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf, last visit:18.10.2014

[4] S. Dotcenko, A. Vladyko, and I. Letenko, "A fuzzy logic-based information security management for software-defined networks," Advanced Communication Technology (ICACT), 2014 16th International Conference on, (2014), pp. 167-171

[5] Gurbani, Vijay K., et al. "Abstracting network state in Software Defined Networks (SDN) for rendezvous services." Communications (ICC), 2012 IEEE International Conference on. IEEE, 2012.

[6] B. N. Astuto, M. Mendonça, X. N. Nguyen, K. Obraczka and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks", hal-00825087,(2013), available online: http://hal.inria.fr/hal-00825087, last visit:18.10.2014

[7] M.K. Shin, K.H Nam and et al., "Formal specification for software defined networks (SDN),"

[8] S. Vissicchio, D.Lebrun and O.Bonaventure, "Towards test-driven software defined networking" Proceedings of IEEE norms, May 2014.

[9]] Handigol, Nikhil, et al. "Where is the debugger for my software-defined network?." Proceedings of the first workshop on Hot topics in software defined networks. ACM, 2012.