Research Journal of Computer and Information Technology Sciences _____ISSN 2320 – 6527
Vol. **5(7),** 15-18, September **(2017)** *Res. J. Computer and IT Sci.*

Short Review Paper

Software defined network

Manju Lahare*, Kauleshwar Prasad and Sudip Bhattacharya
Bhilai Institute of Technology, Durg, CG, India
manjulahare001@gmail.com

Available online at: www.isca.in
Received 10th April 2017, revised 6th September 2017, accepted 18th September 2017

Abstract

The traditional network needs to change the entire affected devices in order to maintain the flow. It was time consuming and many questions have been raised on the scalability and flexibility. It's quite difficult to manage the complex network and maintain pace with the increasing demands. Software Defined Network provides the architecture to manage the large complex networks. Open-flow protocol is used for separation of control plane and data plane. Here the centralized controller is used to decide the forwarding of packets. This paper represents the brief review of SDN. The advantages and limitations of SDN have been discussed. Various simulation tools with supporting experimental analysis is being enlightened.

Keywords: SDN, Open Flow, Software, Defined network.

Introduction

The large number of devices like routers and switches with the supporting protocol together constitutes the network. The network operators design policy to meet the demands of the network. They struggle very hard to convert high level policy according to the changing conditions of the network with the limited tools. SDN brings solution to this problem by providing the network architecture that supports the partition plane. The implementation of SDN is done through a protocol named Open-Flow, which decouples the plane and helps in the selection of path for the transmission of data[1]. It provides an open interface between the two planes. Because of the centralized control in the Open-Flow network, it becomes quite easy to deploy routing strategies to the switch[1]. The devices like switch and routers comes under data plane and control plane consists of controller. The forwarding plane i.e. data plane is responsible for the transmission of data and control plane formulates policy for the forwarding of data[2]. The intelligence of the whole network is shifted to the centralized software based SDN controller, which acts like manager[3]. The modification, control, and checking of transmitted data is being done by the SDN control components.

Architecture of SDN

SDN architecture supports the centralized approach for the flow of traffic. It can operate with various types of switches and different protocol layers. The SDN architecture differs from traditional architecture in terms of the carrier grade network.

The separation of the plane makes it quite easy to implement new protocols and applications. There are basically three main components of the SDN architecture i.e. the switch, the controller, and the interface present in the controller in order to communicate with the forwarding devices[3].

Switch: Switches are taken as forwarding hardware that can be operated via open interface. An Open-Flow switch has namely three entities, flow table, set of command and secure channel. The flow table stores the flow entry for packet lookup and forwarding. These flow entries consist of match fields, counters and set of instructions that handles the matching of packets[3]. On arrival of the packets at the switch, the packet header is being extracted and matched with the matching fields. When matching entry is found, appropriate set of instructions is applied and in case of failed match the action will be taken according to the table miss flow entry. For example: Dropping of packets, continue the match process on the next flow table.

Controller: The control plane is regulatory body of the SDN architecture, so it is very important to give proper concern towards the design parameter of controller. Controller provides a programming interface to the network. It has been explained that multiple controllers must be connected to the switch in order to maintain the back up of the data in case of failure of the controller as it provide single point of connection to the entire network[3]. According to the experiments done controller ONOS has good performance on clusters, linkup and throughput whereas Open-Daylight works well with topology discovery and stability. A controller is designed in such a way that it can handle upto 6.000.000 flows/sec[3].

Comparative study of several controller have been done based on the multiple criteria[4]. They found out that, the controllers coded by c language like MUL, LIBFLUID_MSG gave highest performance whereas the controllers coded by java language like BEACON, IRIS and MAESTRO remains below. Finally

after going through several test, they found that OPEN DAY LIGHT is full featured controller[4].

Interface: The interaction between control plane and data plane is done with the help of protocol. Southbound API's made the controller capable of changing the forwarding rules dynamically. While the application layer uses Northbound API's to relate with the controller.

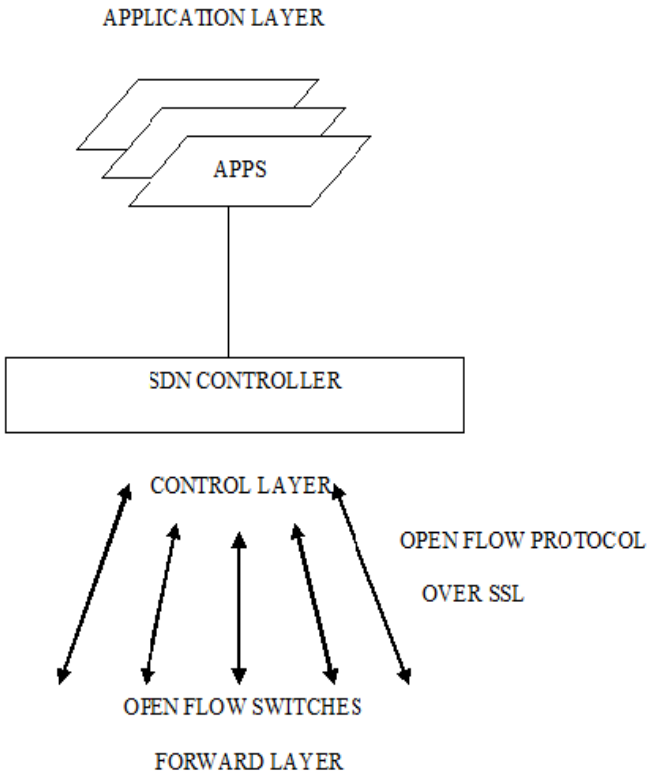

Figure-1: Architecture of SDN.

Employment of SDN

Internet of things: The integration of SDN and IOT bring exciting opportunities. SDN has the capacity to smartly route the traffic and handle the zillions of data emerging from the device that is being linked to the IOT[5]. SDN can divide the complete IOT network into small segments and each part can be controlled by different controller so as to make the network function run smoothly. SDN linked IOT network serves better for security concerns as it has the global view of the network.

Home networking devices: They proposed an architecture that is based on the concept of SDN to resolve the fragmentation challenges of the home network by adapting the centralized approach[6]. It proposed a new type of device that realizes the home networking devices based on the preference for multimedia applications. This proposed technology provides high flexibility in configuring devices and controlling, it helps the users to rely on the software applications rather than depending on the manual configuring of multiple users.

Cloud computing: The number of data released by the network is too big to handle. It gives rise to the concept of cloud, networks create cloud. Due to the large space needed for storing applications, these applications may need to modify before storing in cloud. SDN makes it possible by the help of centralized controller, that is configured by software related protocols.

Mobile and wireless network: The contribution of SDN in wireless network is known as SDWN i.e. software defined wireless network. The researchers have studied about the Open-flow proposed by whitepaper. Open-flow is based on the Ethernet switch; consist of flow table with the capacity of adding or removing flow entries.

Deployment of SDN in wireless network provides seamless handover between different wireless technologies by the embedding of Open-road. SDN supports the flow centric model that helps in resolving the problem of node migration by implementing functions and making it configurable at higher layer[7]. The SDN architecture is open and sharable between different service providers and employs test-bed using Open-flow such as Wifi and WiMAX.

Enterprise network and business: SDN architecture supports large no of enterprises that work in the field of wireless technology[7]. SDN manages the network by middle-boxes and deploy functions with centralized controller. NAT, firewall and load balancer are some example of SDN middle-boxes. With the motive of better management rather than following the traditional network, the researchers created prototype network in which SDN provides users centralized view of the whole network.

Simulation tools

There are large number of simulators and emulators for SDN, for e.g. fs-sdn, NS-3, EstiNet and Mininet. Among them Mininet is the most popular and tested platform[8]. Mininet has the capability to emulate different types of network like host, layer 2 switches, layer 3 routers. The researchers have created and optimize SDN network using Mininet[9]. It is open source software that emulates OpenFlow devices and SDN network. Mininet does not replace the physical switches but virtually displays the network[9]. Mininet has the capability to emulate the entire network on a single machine.

The Table drawn describes some of the common SDN simulation tools like W3, FatTire and Fs-sdn. W3 was introduced to troubleshoot bugs in SDN environment. Performance of Mininet tool is being studied for emulation of SDN[10]. Many parameters like change in topology, increase in the number of nodes, controlling the behavior of switches are being taken into consideration to study the emulation tool and it has been concluded that simulation environment plays very vital role in performance of Mininet[10].

It is being observed that Mininet uses more RAM for same topology if available. SDN testing and debugging tools have been studied in tabular manner along with the pros, cons, supported versions and license[11]. The tools stated here, also helps to detect security vulnerabilities like configuration attack or data integrity attack.

Benefits and limitations

In the architecture of SDN, the whole control logic is transferred to the centralized server called controller. The controller manages the network via API making it easier to configure new functions[12].

The centralized approach in SDN reduces the equipment cost by eliminating support of multiple standards and protocol on data plane[13].

It has been presented that scalability of SDN network can be improved by the reduction of energy consumption by the use of energy efficient algorithm[14].

In order to achieve high link utilization, high frequency requests are being sent to the controller resulting in high controller load[15].

The SDN architecture beautifully balances the load by distributing the traffic among the possible paths and thus makes it possible to respond to large number of data flows in less time[16]. The cost of deployment is reduced by the complete utilization of distributed system.

The flow table size is limited in SDN, as the result it cannot handle large number of requests making the switch overloaded. This results in dropping of packets or knocking down of switch.

It is observed that lot of energy and power is wasted by the use of redundant links and idle devices. SDN makes it possible to reduce the carbon footprint by the intelligent approach.

The software needs to be continuously upgraded in the controller in order to improve the execution of control plane and reduction of harmful effect on network.

SDN has a faster response to the dynamic traffic scenario. It provide better balancing of load and dynamic provisioning.

The simulation tool called Mininet is hungry for RAM, generally RAM usage in SDN is very small but also it uses the more RAM available for the same topology.

Conclusion

This paper presents the drawback of traditional network and how it is overcome by the emerging architecture of SDN. The SDN architecture is studied with the application of SDN in various other fields. Different type of simulation tools is presented but in particular, simulation tool called Mininet is highlighted. The paper is concluded by the pros and cons of SDN and its emulation tool.

References

1. Yang Jun, Dai Bin, Lv Lu and Xu Guan (2015). Coding Open-flow: Enable network coding in SDN networks. *International journal of Computer Networks and Communications*, 7(5), 29-38.

2. Yamei F., Qing L. and Qi H. (2016). Research and Comparative analysis of performance test on SDN Controllers. 1st IEEE Conference on Computer Communication and the internet, 207-210.

3. Nunes Bruno Astuto A., Mendonca Marc, Xuan-Nam Nguyen, Obraczka Katia and Turletti Thierry (2014). A Survey of Software Defined Networking: Past, Present, and future of Programmable Networks. hal-00825087,version 1, may 2013, IEEE Communications Surveys & Tutorials, 16(3), 1617-1634.

4. Salman Ola, Elhajj Imad H., Kayssi Ayman and Chehab Ali (2016). SDN Controllers: A Comparative Study. Electrotechnical Conference (MELECON), 18th Mediterranean, IEEE.

5. Sood K., Yu S. and Xiang Y. (2016). Software Defined Wireless Networking Opportunities and Challenges for Internet-Of-hings: A Review. *IEEE Internet Of Things Journal*, 3(4), 453-463.

6. Jo J., Lee S. and Kim J. (2014). Software Defined Home Networking Devices for Multi-home Visual Sharing. *IEEE Transactions on Consumer Electronics*, 60(3), 534-539,

7. Gupta L. (2013). SDN: Development, Adoption and Research Trends.

8. Isata P. and Guan L. (2016). Performance Benchmarking of SDN Experimental Platforms. IEEE, 116-120.

9. Rawat D. and Bajracharya C. (2016). Software Defined Networking for reducing energy consumption and carbon emission. IEEE.

10. Keti F. and Askar S. (2015). Emulation of Software Defined Networks using Mininet in Different Simulation Environments. 6th International Conference on Intelligent Systems, Modelling and Simulation, 205-210.

11. Nde G. and Khondoker R. (2016). SDN Testing and Debugging Tools: A Survey. 5th International Conference on Information Electronics and Vision, 631-635.

12. Wang Fei-Yue, Yang Liuqing, Cheng Xiang, Han Shuangshuang and Yang Jian (2016). Network Softwarization and Parallel Networks: Beyond Software Defined Networks. *IEEE Network*, 30(4), 60-65.

13. Abu-Jassar A. (2015). Mathematical Tools for SDN Formalisation and Verification. 2015 second International Scientific-practical Conference , PIC S&T, 35-38

14. Staessens Dimitri, Sharma Sachin, Colle Didier, Pickavet Mario and Demeester Piet (2011). Software Defined Networking: Meeting Carrier Grade Requirements. Local & Metropolitan Area Networks (LANMAN), 2011 18th IEEE Workshop on, 1-6.

15. Metter Christopher, Gebert Steffen, Lange Stanislav, Zinner Thomas, Tran-Gia Phuoc and Jarschel Michael (2015). Investigating The Impact of Network Topology on the Processing Times of SDN Controllers IFIP 2015, 1214-1219.

16. Cammarata Giovanni, Stefano Antonella Di, Morana Giovanni and Zito Daniele (2016). Evaluating the Performance of A4SDN on various network topologies. 2016 IEEE International Parallel and Distributed Processing Symposium Workshops, 801-808.

17. Sabeeh Ann, Al-Dunainawi Yousif, Maysam Abbod F. and Al-Raweshidy H.S. (2016). A Hybrid Intelligent Approach for Optimising Software Defined Networks Performance. International conference on Information Communication and Management, 47-51.

18. McKeown Nick, Anderson Tom, Balakrishnan Hari, Parulkar Guru, Peterson Larry, Rexford Jennifer, Shenker Scott and Turner Jonathan (2008). Open-Flow: Enabling Innovation in Campus Network. *ACM SIGCOMM Computer Communication Review*, 38(2), 69-74.

19. Luo S., Wu J., Li Jianhua and Guo Longhua (2016). A Multistage Attack Mitigation Mechanism for Software Defined Home Networks. *IEEE Transactions on consumer electronics*, 62(2), 200-207.

20. Yao Xibo, Wang Hua, Gao Chuangen and Yi Shanwen (2016). Maximizing Network Utilization for SDN Based on Particle Swarm Optimization. IEEE 28th International Conference on Tools with Artificial Intelligence, 921-925.

21. Yi Shanwen, Wang Hua, Yao Xibo, Gao Chuangen and Zhai Linbo (2016). Maximizing Network Utilization for SDN Based on Wise Ant Colony Optimization. IEEE, 18th conference on High Performance Computing and Communications, IEEE 14th International Conference on Smart City, IEEE 2nd International Conference on Data Science and Systems, 959-966.