July – August 2020
ISSN: 0193-4120 Page No. 561 - 564

Software-Defined Network: Importance and Issues

Nikul Jayswal*1, Purvi N. Ramanuj2,
*[1]Computer Engineering Department, Shri K. J. Polytechnic Bharuch, Gujarat Technological University, Ahmedabad, Gujarat, INDIA
phd.nikul@gmail.com
[2]Information Technology Department, L. D. College of Engineering, Gujarat Technological University, Ahmedabad, Gujarat, INDIA
purviramanuj@yahoo.com

Article Info
Volume 83
Page Number: 561 - 564
Publication Issue:
July - August 2020

Article History
Article Received*: 06 June 202*
Revised*: 29 June 2020*
Accepted*: 14 July 2020*
Publication*: 25 July 2020*

Abstract

With the evolution of internet of things (IOT) almost every device is connected to internet. We are living in a digital society where huge amount of data is being generated due to technological innovations. Technologies like cloud computing generates large amount of data every second. Traditional IP networks were not designed to handle these types of rapid technological change. Managing traditional IP networks are becoming more complex due to the variety of data being generated by new technologies. Software-defined networking (SDN) is emerging as a promising technology that can change this complexity by separating the cores of traditional IP network: data planes and control planes. The separation will help to remove major barriers to new protocols and services. In this paper, we present a survey on Software-defined network and its major issues. We start with introducing motivation for SDN. Next, we present the basic architecture of SDN. We discuss major issues related to SDN and conclude discussion.

Keywords: Software-defined network, SDN, reliability, security.

I. INTRODUCTION

Traditional computer network consists of forwarding devices called as Routers that transfers the packet from source to destination. Normally routers or we can say forwarding switches consist of 2 planes, (1) Control Plane and (2) Data Plane. Traditionally both the planes are highly integrated in the forwarding devices. This architecture made a great success in commercial network and from 90's no major change in this technology has been made. Now traditional networks are becoming more complex as a richer set of data are generated by applications. And it is becoming more difficult to manage network and becoming major barrier to new protocol and services. As mentioned in [1], transition of IPv4 to IPv6 is the best example to understand the scenario. It has been started before a decade ago, but still it is not completed. Current network architecture had significant deficiency and modification in architecture based on application requirement is complex [2].

Software Defined Network (SDN) is emerging as a promising technology that can change network behavior dynamically as per application requirement that make it highly programmable. In traditional network both the control plane and data plane are tightly coupled in a box(Router), creating vendor specific dependency. As shown in Figure 1, SDN separates control plane and data plane in a forwarding device, making the switches to deal with only packet forwarding.

Authors in [3] describe IP network scenario in terms of architecture and infrastructure. They refer architecture as an IP protocol and infrastructure as a physical environment that runs on IP protocol. Now if we want to change architecture (IP protocol) than we need to replace infrastructure (routers and switches) also and that is not feasible at all. Thus, to introduce a new protocols and services architecture and infrastructure must be separated, and here SDN comes into picture. SDN do what we need, decoupling data plane and control plane.

Published by: The Mattingley Publishing Co., Inc.

July – August 2020
ISSN: 0193-4120 Page No. 561 - 564

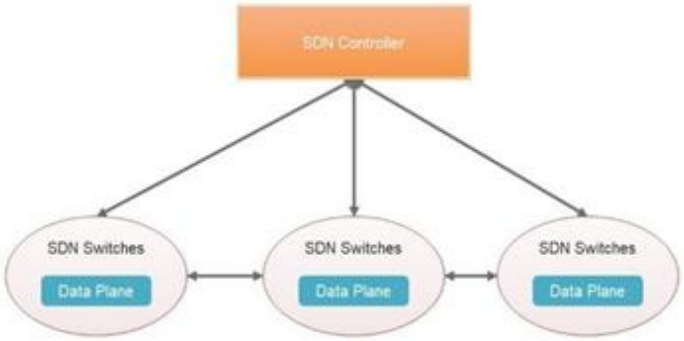

Figure 1: SDN architecture

II. WHAT IS SDN?

Software-defined network is emerging network approach that enables the network to be more dynamic in nature. It enables network to become cost-effective as it removes vendor dependency in network infrastructure. The basic idea in SDN is to separate control plane and data plane. That make network more programmable and centrally manageable. SDN allows new protocols to define and readily experimented in real condition on production network [4].

A software called as SDN controller can update flow-entries from the flow table on behalf of the user's experiments. A static controller can be a simple software running on a computer to establish packet path between a group of test computers during a scientific experiment [5].

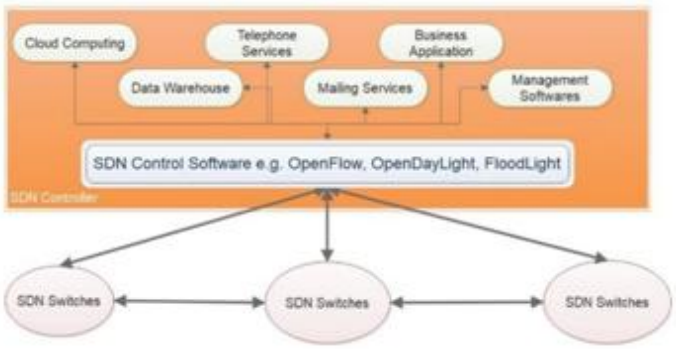

Figure 2: SDN controller and switches

As shown in figure 2, in SDN architecture, all the decision will be taken by central controller called as SDN controller and forwarding devices will forward the packets based on the decision that has been taken by SDN controller. Typically, the SDN controller will be a server that will run controller software like OpenFlow. Management and control of the network can be done by central location.

A variety of protocols are available on the use of SDN in real implementation. Among all other protocols, OpenFlow is the most popular protocol. OpenFlow protocol enables SDN concepts to implement in hardware as well as in software. The OpenFlow controller communicate with OpenFlow switches through protocols [4].

III. IMPORTANCE OF SDN

Traditional IP networks have created vendor specific dependencies in network infrastructure. As per described in [5], big cloud service providers need to install high end routers and switches from same vendors, as it will be easy to re-configure the router parameters. So, as per cloud computing perspective, SDN will be more helpful as it removes vendor dependency as well as it is also mainly open source. SDN uses universal data forwarding switches and provides more control over network as compared to traditional networks [5].

Traditional IP networks do not provide direct programmability to network. That impacts new ideas and innovations, as it cannot be tested and verified on production network. SDN provides better programmability that eliminates the barriers to new innovations in network technology [6].

SDN can improve network management and data handlings as per application requirements. SDN can be a solution that is faced by traditional networks. Major advantages of SDN can be:

- Highly and direct programmable.
- Centrally Managed.
- Open Standard.
- Agile.
- No Vendor Dependency.

IV. MAJOR ISSUES IN SDN

Even though, SDN makes network highly programmable which is not very easy for traditional

Published by: The Mattingley Publishing Co., Inc.

July – August 2020
ISSN: 0193-4120 Page No. 561 - 564

architecture, it also holds the risk of single point of failure. That means, in traditional network, any node failure will affect the traffic that flows from it. But in SDN, the controller failure will shut down the working of whole network.

There are several major issues in SDN. In this paper we discuss 2 major issues:

- Reliability.
- Security.
-

A. Reliability

Reliability is the major issue in Software-defined network. SDN controller is responsible for the traffic management. If any switch fails, traffic is rerouted in new path, but if controller fails, network disaster is likely to happen. As SDN switches forward the traffic based on the decision taken by SDN controllers, reliability of control links between SDN switches and SDN controllers is also very important aspect.

A single point failure risks the entire reliability of the system. A centralized controller may fail due to any software, hardware or connectivity reason, that will stop working of entire system.

Redundancy is the key to improve reliability. One way to deal with this situation is multi controller SDN. In the event of controller failure, redundant controller may take charge immediately. Authors in [7] classifies multi- controller reliability in two aspects: (1) control path and
(2) control node reliability. Control path reliability take care of reliability of network links whereas control node reliability takes care of reliability of network nodes (mainly SDN controllers).

How many controllers are needed and how to place them to optimize predefined reliability? This is the question that may help to improve reliability of the SDN. It has been observed by [8], that the number of controllers and the location of controller is very important for reliability in controller. This problem is called as reliability-aware controller placement (RCP). According to [8], there is a major impact of controller number on reliability of SDN. If the number of controllers is too small, some switches have to use long paths to connect to controllers, which increase the possibility of control path loss. If the number of controllers is too many, the control paths between controllers become the main determinant of the control network reliability.

There have been number of reliability studies in SDN. Authors in [9] categories the SDN reliability solutions in
(1) Data Plane Reliability, (2) Control Path Reliability and, (3) Control Plane Reliability. Data Plane Reliability further categorized in application server reliability, fast failure detection and fast recovery. Control planereliability issues have been studied from various aspects like state synchronization and controller placement scenario. According to [9] SDN not only separates a control plane and a data plane, but also create a new network called as control path network. As a first step for control path reliability, a redundant virtual control connection between data plane and control plane is established.

B. Security

In commercial networks, huge amount of data is being transmitted every second. Data contains some personal and private information which are very sensitive in nature. That's why the security of data is one of the top concerns of IT environment. As per the industry experts, security is the major barrier for commercial SDN implementation.

Authors in [10] identified seven potential threat vectors in Software-defined network: (1) forged and fake traffic flows to break availability of network devices (DOS), (2) attacks on forwarding devices (switches), (3) attacks on control plan communication to take control of controller,
(4) attack on controller to gain command on whole network, (5) installation of malicious applications on controller, (6) attack on administrative stations, (7) no fast recovery due to improper behaviour. Among these seven vectors, vector 3, 4 and 5 is specific to SDN.

Published by: The Mattingley Publishing Co., Inc.

TEST Engineering & Management

July – August 2020
ISSN: 0193-4120 Page No. 561 - 564

V. CONCLUSION

In traditional network control planes and data planes are tightly coupled, and that is the reason for complexity of network. It is also very hard to manage. It is also a barrier for new protocols and innovations. SDN has capability to overcome the problems faced by traditional networks. It is highly programmable and dynamic in nature. Data flow in SDN can be based on application requirement, and it also eliminates vendor dependencies in production networks. Although it is emerging as a promising technology that can change the network technology, it has also some major issues like reliability and security that need to be addressed to commercially deployment of SDN.

VI. REFERENCES

[1] Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2014). Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1), 14-76.

[2] Raghavan, B., Casado, M., Koponen, T., Ratnasamy, S., Ghodsi, A., & Shenker, S. (2012, October). Software-defined internet architecture: decoupling architecture from infrastructure. In Proceedings of the 11th ACM Workshop on Hot Topics in Networks (pp. 43-48).

[3] Raghavan, B., Casado, M., Koponen, T., Ratnasamy, S., Ghodsi, A., & Shenker, S. (2012, October). Software-defined internet architecture: decoupling architecture from infrastructure. In Proceedings of the 11th ACM Workshop on Hot Topics in Networks (pp. 43-48).

[4] de Oliveira Silva, F., de Souza Pereira, J. H., Rosa, P. F., & Kofuji, S. T. (2012, October). Enabling future internet architecture research and experimentation by using software defined networking. In 2012 European Workshop on Software Defined Networking (pp. 73-78). IEEE.

[5] Hu, F., Hao, Q., & Bao, K. (2014). A survey on software-defined network and openflow: From concept to implementation. IEEE Communications Surveys & Tutorials, 16(4), 2181-2206.

[6] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., ... & Turner, J. (2008). OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 38(2), 69-74.

[7] Hu, T., Guo, Z., Yi, P., Baker, T., & Lan, J. (2018). Multi-controller-based software-defined networking: A survey. IEEE Access, 6, 15980-15996.

[8] Hu, Y., Wang, W., Gong, X., Que, X., & Cheng, S. (2014). On reliability-optimized controller placement for software-defined networks. China Communications, 11(2), 38-54.

[9] Song, S., Park, H., Choi, B. Y., Choi, T., & Zhu, H. (2017). Control path management framework for enhancing software-defined network (SDN) reliability. IEEE Transactions on Network and Service Management, 14(2), 302-316.

[10] Kreutz, D., Ramos, F. M., & Verissimo, P. (2013, August). Towards secure and dependable software-defined networks. In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking (pp. 55-60).