Software Defined Networking: A Survey

Shreyas Prabhakar
Computer Science Department
Clemson University
Clemson, United States of America
sprabha@clemson.edu

***Abstract*— Emerging mega-trends in information and communication technologies are commanding new challenges for the future of Internet. Ubiquitous accessibility, efficient resource utilization, higher bandwidth and dynamic management become very crucial. The manual configuration of the network devices proves to be very cumbersome and provides least flexibility. The traditional approaches prove to be error-prone and fail to utilize the capabilities of the network infrastructure to its maximum potential. Software Defined Networking (SDN) is an emerging network paradigm in which all the network traffic may be managed dynamically based on the user requirements and demands, making it more feasible for all networking domains. SDN breaks the vertical paradigm of the traditional networking and provides the flexibility to program the network through (logical) centralized network control. SDN is characterized by its two distinct features, which include – 1. The decoupling of the control plane from the data plane by moving the control plane out of the network hardware and providing centralized network control. 2. Providing programmability of the control plane for any network application. As a result, SDN provides better resource management, high performance, efficient configuration and flexibility to innovate and build network designs and applications. In this paper, we present a survey of SDN. We start by first presenting the definition of the SDN, explain its key concepts, its characteristics and how it differs from the traditional networking architecture. Next, we present the advantages of SDN compared to the traditional networking approaches. Later, we will talk about the issues and problems that arise from the use of SDN. Finally, we conclude the paper with some future trends and other research opportunities in SDN.**

Keywords—Software Defined Networking; SDN; Network Virtualization; Network Programmability

I. INTRODUCTION

Emerging mega trends in Information and communication technologies domain, like mobile, social, cloud, big data and Internet of Things, put a lot of stress on the computer networks asking for higher bandwidth, ubiquitous accessibility and dynamic management. The growing popularity of the rich multimedia content on the Internet like streaming HD videos and audio, ever increasing applications of Internet of Things and the increasing demand for analytics on Big Data require faster connection speeds than ever before. The widespread consumption of mobile technologies and social networking has put a lot of demand on ubiquitous communications to fulfil the social need of the general population. With the emergence of big data and cloud computing, more computing, storage and other resources and moved to remote data centres. Higher bandwidth has become critical to access these resources via a network. As such, computer networks are becoming crucial in enabling the emerging technologies to move forward.

In the traditional networking systems, the distributed control and transport network protocols run inside the network hardware like the routers and switches, to allow information to travel around the world. Despite their widespread adoption, traditional networks are complex and very hard to manage. A lot of these hardware are proprietary devices and reconfiguration becomes very hard. Network administrator needs to configure each device individually. Configuring the devices dynamically to adapt to load changes, to endure device faults and automating tasks therefore has become highly challenging. To make things even more complicated, the traditional networks are vertically integrated. In that, the control plane and the data plane both reside on the networking devices, thereby reducing the flexibility. This hinders the innovation and evolution of networking infrastructure.

Software Defined Networking is an emerging networking paradigm that was developed to facilitate the innovation and enable simple programmable control over the network data-path. Figure 1 visualizes the high level comparison between the traditional networks and SDN. The separation of the data forwarding data plane from the control logic plane in SDN, makes it easier for the deployment of new protocols and applications, and straightforward network visualization and management. Instead of enforcing and running policies on the hardware devices which are distributed across the network, the network is simplified as in the simple data forwarding devices and the decision making central controller.

II. SOFTWARE DEFINED NETWORKING

A. *Definition of SDN*

The Open Network Foundation(ONF) is a non-profit consortium dedicated to development, standardization and commercialization of SDN. ONF has provided the most explicit definition of SDN as follows:

> SDN is an emerging network architecture where network control is decoupled from forwarding and is directly programmable.

As per this definition, there are two major characteristics which define the architecture of SDN –

1. Network as decoupled control and data planes.
2. The control plane of SDN architecture is programmable.

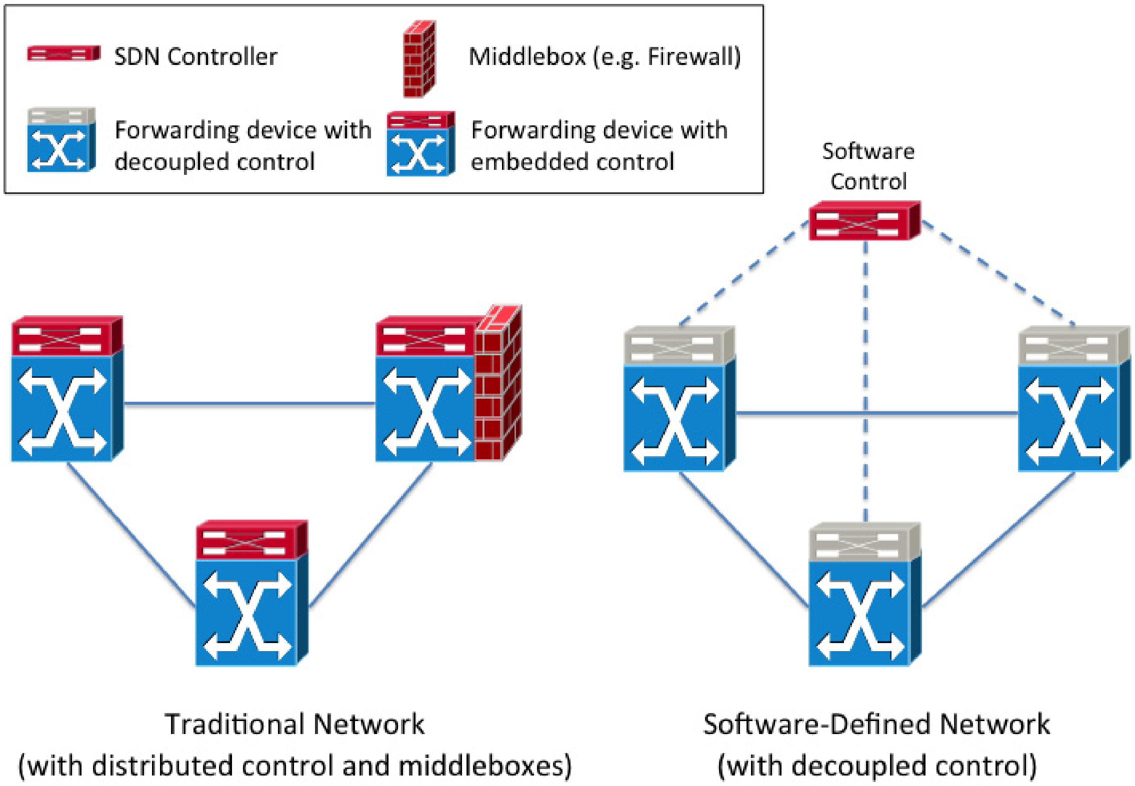

Fig. 1. The SDN architecture decouples control logic from the forwarding hardware, and enables the consolidation of middle boxes, simpler policy management, and new functionalities. The solid lines define the data-plane links and the dashed lines the control-plane links.

The uniqueness of SDN resides on the fact that it decouples the control and data planes and provides programmability of the central control layer. With the growing complexity of the networks, SDN makes it easier to manage and control the network by separating the control plane and the data plane. This helps in abstracting the network design and virtualization of the network. In SDN, the intelligence is moved from the network hardware equipment, such as routers and switches, to the central control system. With this design, the switching systems can take care of forwarding the data and can be controlled externally by the software. The network control can be done independently on the control plane by using simple programming environments without affecting the data plane giving it more flexibility.

B. *SDN Reference Model*

ONF has also suggested a reference model for SDN, as illustrated in Figure 2. This model consists of three layers, namely an infrastructure layer, a control layer, and an application layer, stacking over each other.

The infrastructure layer consists of switching devices such as routers and switches, and other network hardware resources in the data plane. The main functionality of this layer is data forwarding based on the instructions provided by the control layer. This layer is also responsible for collecting network status, storing them temporarily and forwarding them to the controller. This status may include network traffic statistics, network usage and network topology.

The control layer bridges the infrastructure layer and the application layer. For downward interaction (also called as south-bound interface) with the infrastructure layer, it specifies the functionalities to access and control the functionalities provided by the routers and switches. These functionalities may include communicating instructions for data forwarding and status reporting of the infrastructure layer. For upward interaction (also called as north-bound interface) with the application layer, it provides service access points in various forms, such as Application Programming Interface (API). SDN applications can use these APIs to access the network status reported by the switching devices to make routing decisions and other system tunings and send the packet forwarding instructions to the infrastructure layer using these API.

The SDN application layer consists of applications that are used to fulfil the user requirements. Through the programmable platform provided by the control layer of the SDN architecture, users are able to access and control the switching devices in the infrastructure layer using the APIs provided by the control layer.

Identify applicable sponsor/s here. If no sponsors, delete this text box (*sponsors*).

III. Benefits and Challenges of SDN

A. Benefits of SDN

The inherent features of SDN like the decoupling of the control and data planes, and programmability of the control layer provides great control of the network. These features combined together delivers a lot of benefits over the traditional networks.

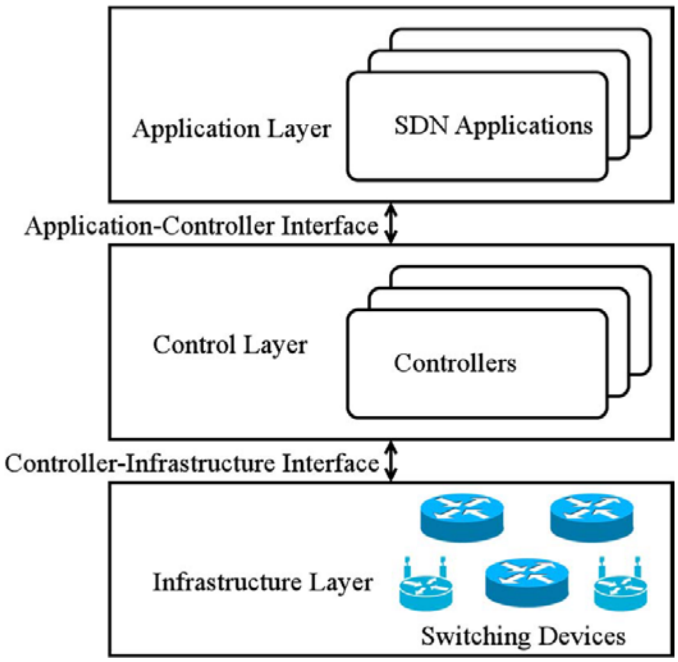

Fig. 2. SDN Reference Model: a three-layer model, ranging from an infrastructure layer to a control layer to an application layer, in a bottom-up manner.

- *Centralized Network Provisioning*: SDN provides central view of the network, making it easier to centralize the management and provisioning of resources.
- *Holistic Management and Network Flexibility:* SDN provides the network administrators to experiment with different network designs without impacting the network. SDN provides single set of APIs to create the management console for both the virtual and physical devices, as opposed to using different APIs for different vendor-specific proprietary devices.
- *Lower Operating Costs: Administrative efficiency*: improvement in server utilization, better control of virtualization and other benefits of SDN result in operational savings. Many of the routine administration issues can be centralized and automated. Many existing hardware can be repurposed using the instructions from SDN controller. Lesser expensive hardware can be deployed as the intelligence is moved to the SDN controller.
- *Cloud Abstraction*: By abstracting cloud resources using SDN it is easier to unify cloud resources. The networking components that make up massive data storage centre platforms can be managed from SDN controller.
- *Isolation and Traffic Control*: SDN provides isolation mechanisms such as configuring network access control lists and firewalls at the infrastructure level. We can also define the traffic rules using the application layer, which helps in providing full control over the traffic.
- *Guaranteed Content Delivery*: Being able to direct and automate data traffic makes it easier to implement Quality of Service (QoS) for Voice over IP (VoIP) and multimedia transmission.
- *Reduced Downtime*: Since SDN helps in virtualizing most of the physical networking devices, it becomes easy to perform an upgrade for one piece rather than needing to do it for several devices. SDN also supports snapshotting the configuration, which helps you quickly recover from any failures caused by the upgrades.

B. Challenges of SDN

Given the promises of enhanced configuration, improved performance, and encouraged innovation, SDN is still in its infancy. Many fundamental issues still remain not fully solved, among which standardization and adoption are the most urgent ones.

Though the definition of SDN provided by ONF is the well received by most, OpenFlow, which is sponsored by ONF is in no way the only standard available and is not mature enough for large scale deployment. A standard API for the interfaces or a higher level programming language for the application development is still missing. The SDN API programmers, network device vendors and the consumers is yet to be come together and create a healthy environment to innovate, create and deploy SDN.

The SDN incurs huge over heads as all the intelligence is moved from the distributed switching systems to a central control unit. This could potentially add a few micro-seconds delay to the data and thereby increasing the latency of the data propagation in the network. This could become a huge challenge in financial applications and other latency-sensitive applications.

The shift from traditional network to SDN can be disruptive and painful. Common concerns include security, performance, privacy and interoperability with legacy networks and devices. Lack of technical experts in for managing and support of SDN could also be a major concern for immediate deployment.

IV. Conclusion and Future Work

The success of SDN requires improvements and developments at all the three layers, including the infrastructure layer, the control layer, and the application layer. It needs collaboration of different organizations including vendors, academia, and communities, and interdisciplinary knowledge covering both hardware and software. A few design guidelines for further development in SDN are:

- SDN switching are relatively simple as the intelligence has been moved to the control layer. This makes the production costs cheaper. However, SDN switching devices need higher processing speed and bigger memory. Integration or other new hardware technologies also become necessary.
- An SDN controller has to handle a massive amount of interaction events with its associated switching devices. To guarantee efficiency of network operations, methods in software optimization and algorithm analysis can be used to improve controller's performance, and properly designed architecture can help decrease request frequency.

Many challenges in SDN still need further research attention. A few open issues covering the whole lifecycle of SDN from standardization, implementation, to deployment are:

- *Standardization of SDN*: OpenFlow, which is sponsored by ONF, is in no way the only standard of implementation that is available today. Upon the maturity of various different standards of SDN, a comprehensive comparison among all SDN standards should be done.
- *Implementation of SDN*: The current SDN approach of decoupling the data plane completely from the control plane may be too idealistic, and could cause a serious problem for deployment and wide acceptance of SDN. In transition phase, moving from the traditional networks to fully decoupled SDN, semi-coupled switching devices which run the routing protocols and also control the routing protocols remotely could be helpful in smooth transitioning to achieving the idealistic SDN.
- *Deployment of SDN*: Further study of SDN in carrier networks with carrier-grade requirements, wireless mesh networks with fast client mobility, and wireless sensor networks which require high reliability and reachability is also needed for wide deployment of SDN.

References

[1] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A Survey on Software-Defined Networking," IEEE Communications Surveys and Tutorials, vol. 17, pp. 27-51, 2015.

[2] D. Rawat, and S. Reddy, "Software Defined Networking Architecture, Security and Energy Efficiency: A Survey," IEEE Communications Surveys and Tutorials, vol. pp, issue. 99, pp.1-22, 2016.

[3] Kreutz et al, "Software-Defined Networking: A Comprehensive Survey," Proceedings of the IEEE, vol. 103, pp. 14-76, 2015.

[4] Astuto et al, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," IEEE Communications Surveys and Tutorials, vol. 16, pp. 1617-1634, 2014.

[5] Nishtha, M. Sood, "A Survey on Issues of Concern in Software Defined Networks," Third International Conference on Image Infonnation Processing, pp. 295-300, 2015.

[6] N. Feamster, J. Rexford, E, Zegura, "The road to SDN: an intellectual history of programmable networks," ACM SIGCOMM Computer Communication Review, vol. 44, pp. 87-98, 2014.