
An introduction to software-defined networking

Babak Darabinejad¹, Seyed Rasoul Mousavi Fayyeh²

¹Department of Computer Engineering, Mehrarvand International Institute of Technology, Abadan, Iran

²Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

Email address:

b.darabi@gmail.com (B. Darabinejad), sr.mousavi@srbiau.ac.ir (S. R. M. Fayyeh)

To cite this article:

Babak Darabinejad, Seyed Rasoul Mousavi Fayyeh. An Introduction to Software-Defined Networking. *International Journal of Intelligent Information Systems*. Special Issue: Research and Practices in Information Systems and Technologies in Developing Countries. Vol. 3, No. 6-1, 2014, pp. 71-74. doi: 10.11648/j.ijis.s.2014030601.23

Abstract: Although computer networks have spread worldwide, organizations that use networking technologies, have complained about the fact that no new feature has been added. Due to their newly emerged needs, they also prefer to automate many tasks. Moreover, they prefer their networks to be developed using software, rather than expensive and new hardware. Software-defined networking and Open Flow protocol separate data level from control level which makes the network smarter and more manageable. Network's main infrastructure is also separate from applications. This makes organizations to program, automate, and control networks more efficiently. In this paper we will introduce this technology by reviewing the literature.

Keywords: Software Defined Networking, Protocol, Open Flow, Overlay Networks

1. Introduction

By the development of portable devices and accessories, server virtualization, and the emergence of cloud systems, the revision of network architecture is made necessary. Many network architectures are traditional and hierarchical which are comprised of nodes in Ethernet switches in a tree structure. In client-server communications architecture, this is more evident. This static architecture, however, does not meet the requirements of dynamic communications and organizations' needs regarding data center and media servers [1].

It is almost impossible to meet the emergent needs of market using common network architectures. Information technology (IT) companies make use of management tools at machine and manual-level processing to deal with financial crisis or budget cuts.

Telecommunication service providers are also faced by similar issues. It is due to the fact that there is an increasing demand for dynamic networks bandwidths. At the same time, however, their profits decrease due to the costs of central equipment and a decrease in their income. Network architectures are not designed in a way to meet the current requirements of companies, telecommunication service providers, and users. In other words, network designers mislead IT technology due to limitations, such as complexity, contradictory policies, lack of scalability, dependence on the seller [1], and lack of coordination between market needs and

network capabilities. In order to prevent these, software-defined networking was introduced along with its standards.

2. The Idea of Software Defined Networking

The idea of Software Defined Networking (SDN) is not a new one and it was introduced more than a decade ago. An efficient definition of SDN describes it as the separation of routers' data and controlling operations as well as other infrastructures of the second layer using a programming interface [2]. The foundations of the current SDN model which includes projects, such as active networking, form the basis of many network infrastructures that could be programmed by custom services [3, 4]. Open signaling (OPENSIG) in 1995 started to be used for making ATM and cell phone networks and the Internet as an extendable and programmable SDN [5]. Network configuration (NETCONF) was proposed as a management and adjustment modification protocol in network devices in 2006 [6].

The foundations of new SDNs were laid by two scholars in computer sciences. Nick McKeown of Stanford, Scott Shenker of Berkley, and a group of students redefined SDN in 2008 [2, 7]. Their project, named Ethane, had started 15 years earlier. The aim of Ethane was to improve network security using a series of flow-based protocols [2, 8].

3. SDN Architecture

SDN is a newly emerged network architecture in which network control is separate from traffic exchange and is directly programmed. This immigration which used to be limited to hardware, makes virtual machines and network infrastructure to define and provide new services. It also makes them capable of communicating with a new range of applications for more network flexibility and widespread access to exchanged data [1].

Figure 1 illustrates an overview of SDN architecture. Logically, the smart layer of the network, which maintains the overall network structure, lies in the center of SDN software controllers. Therefore, applications consider the network as a unified logical switch. Using SDN, companies and telecommunication operators can control the network using a unified central controller, regardless of its hardware and its manufacturer. It makes network designing and its exploitation much easier. Moreover, SDN simplifies devices and machines used in a network, since there is no need for identifying and processing thousands of protocol standards. Orders are only issued by the SDN controller [1].

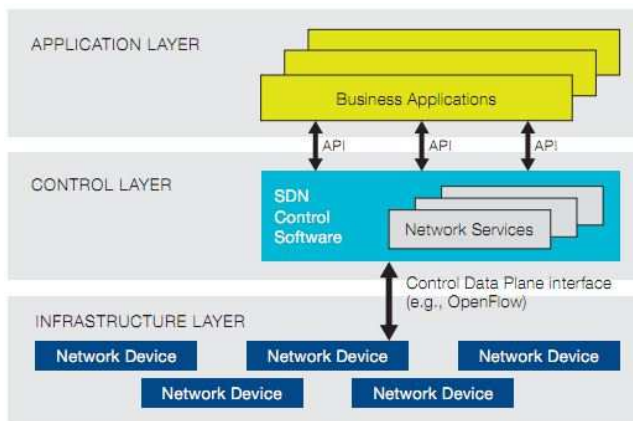


Figure 1. SDN architecture[1]

3.1. SDN Network Components

3.1.1. Controllers

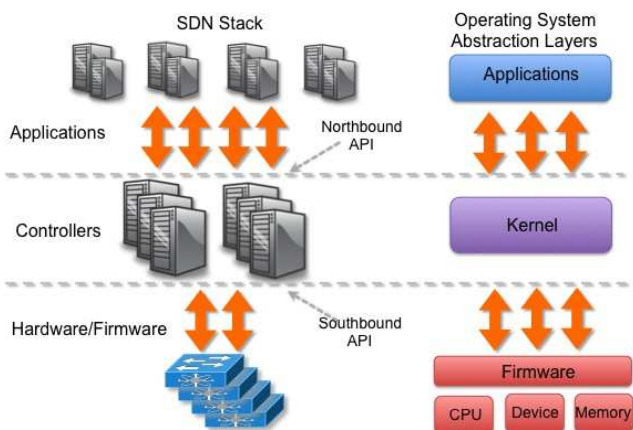


Figure 2. Controllers in SDN[10]

One of the major ideas behind SDNs is that a device, called controller, directly communicates with all devices on a network domain, is aware of network's topology, and programs the network from a central point. An SDN controller changes network programming from a distributed state to a centralized one. Figure 2 illustrates an SDN network as firmware. It also shows a connection through a controller. A connection with lower rank devices which is called a southbound connection and a connection with higher ranks which is called a northbound connection are also depicted.

3.1.2. Virtual Switches

The emergence of server virtualization technologies, that are exploited using hyper wires, highlighted the role of virtual switches in connecting virtual servers using virtual network adapters, traffic congestion, and in sending the traffic outside hyper wires in physical networks. Hardware and software switches play a role in SDN, since they are directly responsible for sending scheduled tables using controllers [11, 12].

3.1.3. Overlay Networks

Overlay networks are virtual networks that jointly use the same physical network platform, but are logically independent. Some SDN controllers use overlay networks for their communications in dispersed data centers and selecting from various virtual hosts.

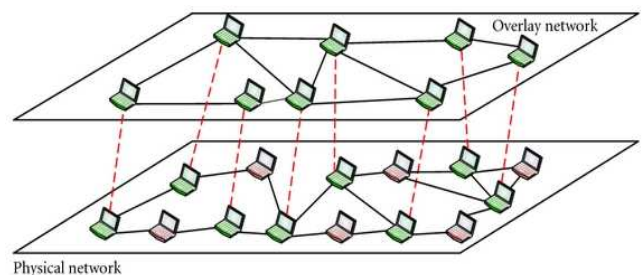


Figure 3. Overlay networks structure

4. Open Flow Protocol

Open Flow protocol is the first standard communication interface that lies between control layer and sending layer in an SDN [1]. Open Networking Foundation (ONF) started in 2011 with the aim of promoting a new form of SDN networks compatible with Open Flow protocol [13]. To this end, ONF took the responsibility of standardizing Open Flow protocol [1]. Unlike most other groups or industrial consortia for IT standardization, ONF was not founded by infrastructure suppliers. It was, in fact, founded by companies that were willing to use this technology, such as Google, FaceBook, Microsoft, Yahoo!, and 19 other companies [14].

Open Flow protocol facilitates direct access and changing network accessory sending programs, such as switches and routers, both physically and virtually based on hyper wires. The lack of an open interface in sending data programs led

today's networks into becoming integrated, closed, and semi-centralized processing. No other standard protocol can carry out the tasks of Open Flow protocol. A similar protocol is needed to get the network out of switch's control and directs it to the central controller software [1].

Open Flow protocol first defines the central controller. Then, it determines how it can safely connect to network devices and control them. It then determines how incoming packages are manipulated, processed, and forwarded. Before Open Flow protocol, there was no standard for manipulating and forwarding network routing table. Therefore, SDN was bound to run independently which faced it with shortcomings in network performance [15].

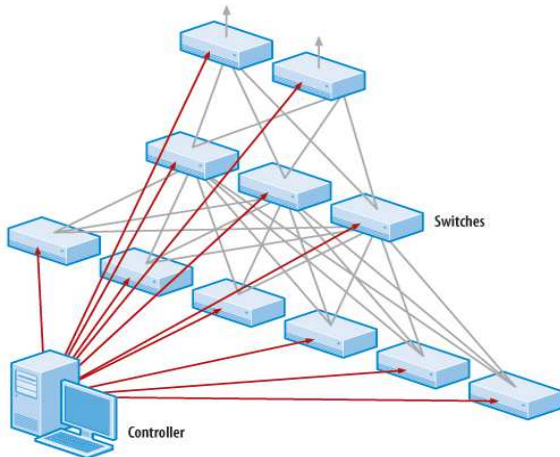


Figure 4. A network with a centralized controller[15]

As Figure 4 shows, SDN focuses on network controlling unit which makes it possible for users and IT managers to control routing and data traffic in switches through defining rules and policies for the central controller. In those networks, sending units and data units are maintained distributively between Open Flow-based switches and routers. The smartness of network lies in the central controller [15].

Routing flow tables are comprised of flow gateways. As Figure 5 shows, each gateway is comprised of matching fields, counter fields, and instruction fields [17].

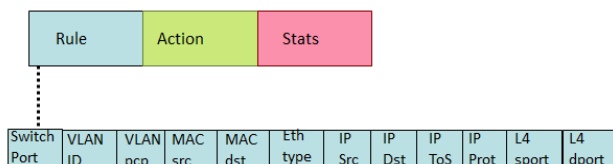


Figure 5. Flow routing table in Open Flow [17]

There may be multiple flow gateways for determining the mode of each packet. The gateway with the highest number of modes for a packet is finally chosen for matching. Then, the input instruction is applied on gateway's instructions. A packet may be assigned an address by a physical or virtual port. Packets may be sent for the controller, flood the controller, or get referred to the switch. Moreover, packets may be sent to virtual ports in flood or normal mode [16].

5. Conclusion

SDN is not a product or hardware/software concept, but rather a new architecture and approach for enhancing networks flexibility and controllability, facilitating the use of various applications, services, and software services on current networks. It reduces the tasks and lowers the highlighted role of hardware and network devices, while simplifies networking management and control and highlights the roles and tasks of software layers of a network. The way in which SDN can help improve network highly depends on the problems that are to be solved. Exploiting appropriate SDN solutions can simplify operational processes, decrease human error, and control traffic based on uncommon ways that are defined by the unique criteria of a particular organization. In short, it enhances performance and flexibility.

References

- [1] White paper, Software-Defined Networking: The New Norm for Networks, Open Networking Foundation, April 13, 2012. Retrieved August 22, 2013.
- [2] David Strom, 100Gbps and beyond: What lies ahead in the world of networking, www.arstechnica.com, Feb 19 2013.
- [3] D. Tennenhouse, J. Smith, W. Sincoskie, D. Wetherall, and G. Minden. A survey of active network research. *Communications Magazine*, IEEE, 35(1):80-86, 1997.
- [4] D. Tennenhouse and D. Wetherall. Towards an active network architecture. In *DARPA Active Networks Conference and Exposition*, 2002. Proceedings, pages 2-15. IEEE, 2002.
- [5] A. Campbell, I. Katzela, K. Miki, and J. Vicente. Open signaling for atm, internet and mobile networks (opensig'98). *ACM SIGCOMM Computer Communication Review*, 29(1):97-108, 1999.
- [6] R. Enns. NETCONF Configuration Protocol. RFC 4741 (Proposed Standard), Dec. 2006. Obsoleted by RFC 6241.
- [7] Scott Shenker, Gentle Introduction to Software-Defined Networking, Technion lecture, The 2nd Annual International TCE Conference Networking, Cloud and Beyond, June 6-7 2012
- [8] M. Casado, M. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker. Ethane: Taking control of the enterprise. *ACM SIGCOMM Computer Communication Review*, 37(4):1-12, 2007.
- [9] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. Nox: towards an operating system for networks. *ACM SIGCOMM Computer Communication Review*, 38(3):105-110, 2008.
- [10] Brent Salisbury, The Northbound API- A Big Little Problem, www.networkstatic.net, June 12 2012
- [11] G. Lu, R. Miao, Y. Xiong, and C. Guo. Using cpu as a traffic co-processing unit in commodity switches. In *Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12*, pages 31-36, New York, NY, USA, 2012. ACM.

- [12] J. C. Mogul and P. Congdon. Hey, you darned counters!: get off my asic! In Proceedings of the first workshop on Hot topics in software defined networks, HotSDN '12, pages 25-30, New York, NY, USA, 2012. ACM.
- [13] John Markoff, Open Networking Foundation Pursues New Standards, The New York Times, www.nytimes.com/2011/03/22/technology, March 22, 2011
- [14] Katherine Noyes, Google and other titans form Open Networking Foundation, www.computerworld.com.au, 23 March 2011
- [15] Open Flow Tutorial: Next-Gen Networking Has Much To Prove, www.techweb.com, Oct 17, 2011
- [16] Jeff Doyle, Strategy: Inside Open Flow, informationweek, October, 2011, <http://reports.informationweek.com/abstract/19/8351/Network-Infrastructure/Strategy:-Inside-OpenFlow.html>
- [17] Srini Seetharaman , Open Flow/SDN tutorial , March, 2012, <http://www.yuba.stanford.edu/~sd2/> Open Flow Tutorial_O FC2012.ppt