Software Defined Networking - A Survey

Shivam Pandit
Clemson University
pandit@clemson.edu

***Abstract*—Software defined Networking has totally changed the networking world. It has not just given more power and control in the hands of network administrators but also resulted in maximum network utilization. The best thing about SDN is that it decouples data plane and network controller. Earlier network implementations were vertically integrated and thus lacked control of the forwarding components of the network. SDN simplified this by centralizing network with a global view of network at the network controller. Centralized controller helps in abstracting the network and also ensures all network components utilization is good thus improving efficiency of the network. Unlike earlier network implementations were vendor programming at forwarding components like at CISCO routers resulted in heterogeneous system that created problems in compatibility. SDN ensures a more robust and flexible network that can even scale well without worrying about underlying structure. SDN helps network administrators to have better control of network while solving problems of heterogeneous networking components. SDN also helps in network virtualization which is fundamental to cloud services. In this survey paper, we will discuss overall topics around SDN like main techniques that are used, issues and problems in their implementation along with the future trends in this field.**

I. Introduction

Software-defined networking (SDN) is key towards a robust and scalable network implementation. It enables flexible delivery and reducing network maintenance cost in the long run. SDN is relatively new concept but gaining pace due to nature of its efficiency and control. Not only it ensures better network control but also ensure heterogeneous network components connect without facing compatibility issues. The benefits of SDN doesn't stop at network control and efficiency but it also serves as basis for network virtualization [1] that can serve cloud services which requires centralized intelligence and a clear view of the network as a whole which ensures maximum network component usability while easing maintenance activities.

Conventional Networks had the biggest limitation of limited view of network and heterogeneous components communicating without having common controlling unit [2]. This resulted in difficult maintenance activities due to multivendor programming at the networking components that determined forwarding logic. This fact resulted in managing network activities expensive as well as complex. During network failures these networks would take too much time to get online if backup systems are not affected. Due to decoupling of network components like routers, switches from the data plane, the management of network becomes fairly easy and also gives a more robust network. Network controller dictates the routing decisions and determines the flow of traffic unlike in conventional networks where routers decided the same. Also, since all layers in conventional networks used to be vertically integrated in a way that control plane and data plane were bundled in same networking devices. This was not only affecting efficiency but also innovation and scalablity in the networks. One most recent examples is transition of IPv4 to IPv6 which is started almost a decade ago but still mostly incomplete shows how inflexible conventional networks are towards any innovation despite this transition was mere protocol update. One can think to start building new networking protocols from scratch but that will be daunting task provided it will have to reflect on all networking components worldwide with universal consensus.

SDN aims to overcome the limitations of conventional networks and also provide a path towards a robust network that is future ready. Having a robust and scalable network is very essential since number of devices on the internet are increasing on exponential rate and also with the progress in IOT and smart home devices, networks are bound to be more powerful than ever before to handle enormous traffic with minimal downtime. A typical 3 layer SDN architechture is shown in Fig.1 which shows three important layers in SDN namely Application Layer, Control Layer and Infrastructure Layer which will be discussed in next section. OpenFlow protocol is the key towards SDN implementaton which describes how networking components will communicate.

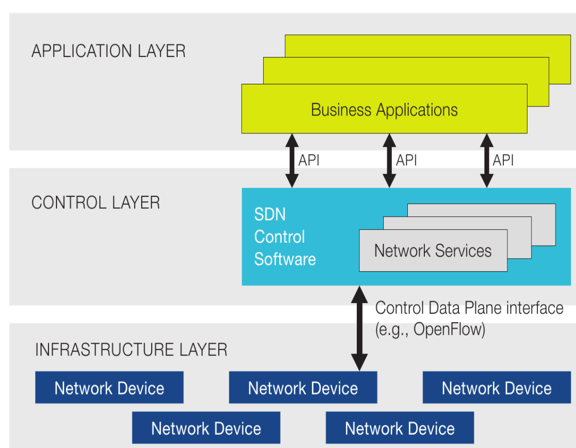

Fig. 1. A three-layer SDN architecture Source: sdxcentral.com

OpenFlow fixed the problem of heterogeneous devices communication within the network by providing standardized protocols that will help in managing traffic and also communication between network components like routers and switches.

Typical SDN realization is three layered as shown in Fig.1 which is constituted of:

1) Application Layer: Upper layer of SDN and contains SDN applications that communicate to controller via APIs (northbound communication). This layer is also responsible to set flow rules.
2) Control Plane: Middle layer of the SDN and contains controllers that communicate via southbound protocol. OpenFlow is most widely used here to route packets using different flow rules.
3) Data Plane: Lower layer of the SDN also called infrastructure layer. This layer contains all the components that make the network like switches, router,gateways, etc.

The surge in the demand of the cloud services like (e.g., SaaS, PaaS, etc) has posed new challenges as client expectations are really high since most of these services are paid and user wants to have best experience. This puts burden on network providers to put more servers and ensure high quality of service with very rare downtime.

II. Main Techniques

Since SDN aims at replacing conventional networks that were vulnerable to threats as well as not scalable and manageable. SDN implementation aims at addressing those issues.

A. SDN Concepts and implementation

SDN helped in better control of the network by taking control from individual nodes which earlier used to have routing decision powers and gave more controls to centralized controllers. This helped in global view of the network and also decoupled the network [3] which includes routers, bridges, etc as shown in the Fig 2 below which is reference architecture of SDN along with the underlying components.

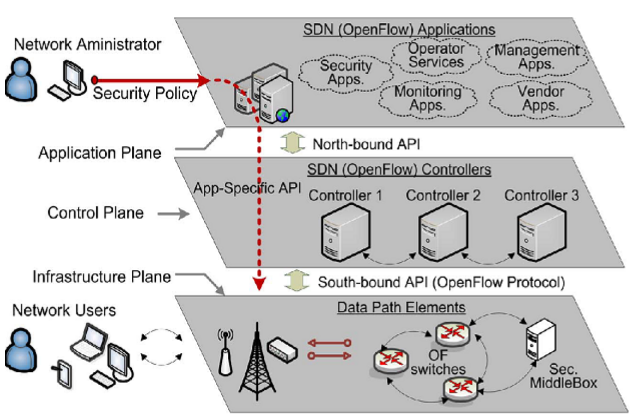

Fig. 2. SDN reference architecture showing network constituents

So, we have a data plane here that is also called infrastructure layer and includes all networking components like router, switches, etc are now controlled by centralized controller that has global view of entire network. This global view has many benefits most important one being easy to enforce policy rules, flow rules and detect malicious traffic before it affects client machines. Before SDN, network lacked global view and even small security or protocol update would need to reconfigure all routers in the network thus wasting time and money. Application plane enforces security and has connection to data plane through control plane.

SDN solved the issue of conventional networks that used to be vertically integrated by decoupling network and data planes. This in turn helped in implementing network abstractions. All networking logic can now easily be pushed down by control plane down to data plane using southbound API OpenFlow. So, this decoupling actually gave control of a network in the hand of network controller that can control full network even though data plane has heterogeneous network components. All applications connect at application layer but communicate to network components through SDN layers using southbound APIs. All network components security patches with SDN now just required one push of instructions from network controller thus reducing operational costs largely. SDN also enabled us to use network virtualization that serves cloud services. Since OpenFlow is most popular standard among SDN, we will discuss that along with brief details of different SDN layers.

B. Application Plane

Application Plane is top layer in SDN and provides platform for application development that will communicate with the networking components down the SDN layers using southbound API. This layer also houses network security solutions which and enforced in the network using push of instructions. Since attacks have turned into more sophisticated and stakes getting higher, security is among the top most priority while building any system that too when it will be on internet which is accessible by anyone. Application plane also manages DNS , firewall, virtual network services, etc that are very crtitical for clients.

C. Control Plane

Control Plane in SDN is decoupled and made as a seperate plane in SDN unlike conventional networks. SDN controller has Networking operating system(NOS) that implements all necessary functions to control network while keeping a global view of all the network resources downstream. API's serves as information carrier from network components to the network controller. OpenFlow protocol standardizes all communications across the network. The switch doesn't take decision on its own, it checks its flow table for any matching entry. If it is unable to find any flow rule for the packet, it routes that packet to the network controller which updates the flow table for next flows.

D. Data Plane

Data plane which is also referred as infrastructure plane has all the networking components that make the network like routers, switches, gateways and all necessary routing hardwares. All components on this layer are controlled by the control plane through Network operating system through a secure communication channel. Any updates/security patches to any components can be easily performed without need to physically access any component using southbound API's. All these calls are mostly remote but their functions may vary. OpenFlow makes all switches in the network easily reconfigurable thus making network robust and reliable. Flow tables on this layer networking components are also initially set by the network controllers which are then updated constantly.

E. OpenFlow

OpenFlow is fundamental in SDN implementation. It served as fundamental towards making SDN so effective. OpenFlow comprises of protocols and all the logic, how devices in network should communicate despite having different architectures [9]. Fig.3 shows schematic overview of SDN implemented with OpenFlow. Global view assisted in quick network issue resolutions and also enforcing security across components. OpenFlow consist of two logical components, one being flow tables and other is OpenFlow API's. Flow tables determine the forwarding logic of packets within the network and APIs handles switches between switch/router and the network controller. OpenFlow comprises of two modes of operation:

- Equal interaction: In this mode, all controllers have read/write acces to control switch.
- Master/Slave interaction: In this mode, their will be just one master but multiple slaves.

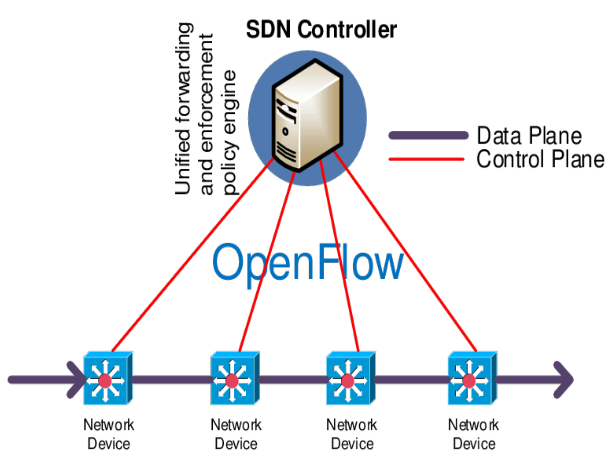

Fig. 3. A schematic overview of SDN implemented with OpenFlow.

III. Issues and Problems

Even though SDN has immense uses and advantages than conventional networks. Still, there are some issues and challenges that need to be addressed [5]. Biggest one is the implemntation of SDN itself since conventional network is massively large in size and transferring all networking components to SDN will take a substantial amount of time. Even mere protocol change of IPv4 to IPv6 has started over a decade ago and still not complete. Some important issues and challenges we face in SDN enivornment are described below:

A. Scalability:

This defines if SDN will be able to handle the increased workload efficiently [10]. So, this is basically increasing capacity using various methods like clustering and parallel computing systems to achieve desired results.

B. Reliability:

Reliability is key factor in determining if a system will have a failure free operation. Since some systems process critical data where reliability is a top priority, there SDN controllers must meet real-time requirements to ensure timely delivery of important data.

C. Availability:

Availability ensures system is up and running to provide services. System outages can be catastrophic for critical data and proper backup systems needs to be deployed to handle any system crash like events which is normally done by network providers.

D. Elasticity:

Elasticity refers to scalability of the SDN system if it manages to handle variations in workload. It normally is related to control plane that is in control of the infrastructure plane.

E. Security:

SDN security is very important as it will ensure device connected with infrastructure layer doesn't get any threat from southbound protocol control layer. It also ensure physically protecting hardware components while ensuring disruption free operation.

F. Performance and Resilience:

Performance measures how resources are effectively utilized that includes computing components like RAM and CPU. In SDN we can ensure performance in terms of bandwidth, latency ,etc. Resilience is capability of system to handle failure and ensuring minimal recovery time.

Dependability, high availability and reliability are among the most important things in terms of any system. Also, their is ongoing research going on in this topic to have ideal realization of SDN system while addressing common switch designs, ensuring communication in heterogenous components and standardize flow table capacity.

IV. Future Trends

SDN is growing at a good pace and also has immense benefits but still we need to address some challenges. As we discussed in issues and problems where SDN requires to address issues like scalability, performance, reliability, availability, elasticity, etc. Without those issues being addressed we are still very far from ideal realization of SDN. Moreover, implementation of SDN across the internet still requires many stakeholders to come to consensus since network is not owned by any particular individual. With conventional networks facing poor network utilization rate of only 30-50%, SDN has proved that it can give around 95% utilization ratio which was shown by big companies like google and facebook when they migrated their systems to SDN. But still just about 30% popular organizations have migrated to SDN which shows that their is still some time when it will become new networking standard. SDN's effective network utilization can help companies to increase pace of their software delivery by reducing operational costs and speeding up processes. The rise in the adaptability of SDN can be predicted from the facts like:

- Surge in number of private clouds.
- Automating network to reduce people that had to earlier control the same.
- More people and corporate switching to virtual environments.
- Efficiency and reliabilty becoming new norm.
- Importance of centralized control.

Even some companies that were earlier hesitant to SDN have slowly started to migrate to SDN. One of the latest and popular entrants is AT&T which is migrating to SDN code and also the Microsoft that successfully migrated to SDN on its Azure environment. Passionate contributors and researchers will further contribute towards making SDN a success.

References

[1] Mijumbi, Rashid, et al. "Network function virtualization: State-of-the-art and research challenges." IEEE Communications Surveys Tutorials 18.1 (2016): 236-262.

[2] Benzekki, Kamal, Abdeslam El Fergougui, and Abdelbaki Elbelrhiti Elalaoui. "Softwaredefined networking (SDN): a survey." Security and communication networks 9.18 (2016): 5803-5833.

[3] Sezer, Sakir, et al. "Are we ready for SDN? Implementation challenges for software-defined networks." IEEE Communications Magazine 51.7 (2013): 36-43.

[4] Flauzac, Olivier, et al. "SDN based architecture for IoT and improvement of the security." Advanced Information Networking and Applications Workshops (WAINA), 2015 IEEE 29th International Conference on. IEEE, 2015.

[5] Cabaj, K., Wytrebowicz, J., Kuklinski, S., Radziszewski, P., & Dinh, K. T. (2014, September). SDN Architecture Impact on Network Security. In FedCSIS position papers (pp. 143-148).

[6] Networkworld.com, Software defined Networking, https://www.networkworld.com/category/software-defined-networking

[7] Searchsdn.techtarget.com, The future of SDN in 2018 moves toward increased automation, https://searchsdn.techtarget.com/tip/The-future-of-SDN-in-2018-moves-toward-increased-automation

[8] Naudts B, Kind M, Westphal FJ, Verbrugge S, Colle D, Pickavet M. Techno-economic analysis of software defined networking as architecture for the virtualization of a mobile network. Software Defined Networking (EWSDN), European Workshop on. IEEE, 2012, pp. 6772.

[9] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, OpenFlow: enabling innovation in campus networks, SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 6974, Mar. 2008.

[10] S. Yeganeh, A. Tootoonchian, and Y. Ganjali, On scalability of software-defined networking, Communications Magazine, IEEE, vol. 51, no. 2, pp. 136141, 2013