Software Defined Networking: Basic Architecture & Its Uses In Enterprises

Yashi Aggarwal[1], Uma Kumari[2]
[1, 2] Dept of Computer Science & Engineering
[1, 2] Mody University, Lakshmangarh

Abstract- *Networks are controlled by software applications which are called Software Defined Networking (SDN) controllers. Traditional network controllers are tedious and it is hard to manage old, rigid commands and consoles. A key requirement of SDN is speed and automation which is not possible in traditional network controller. In a traditional network controller, we require a lot of administrative work. So it requires a lot of administrative work. SDN is an open technology and it is more flexible to the previous technology and network system. It has more interoperability functions and it has greater innovation in it. It gives a cost effective solution which also gives most important advantage. This paper provides SDN Architecture as well as Enterprise networks architecture. It also provides applications of SDN and advantages and disadvantages of SDN. This paper compares SDN with traditional systems.*

Keywords- Software Defined Network (SDN), Open Flow, Network Function Virtualization

I. INTRODUCTION

Software defined networking can be said to a new paradigm which separates network control logic from data (traffic) forwarding [1,2]. It centralizes all traffic management decisions in the SDN controller.

The primary benefit of this approach is that it allows real time and dynamic updating of traffic forwarding policies in network fabric infrastructure and due to real time programmability availability of SDN, we can also change the role of individual network devices. For e.g., a forwarding switch to a forwarding firewall or just a firewall by remotely changing the traffic management primitives implemented in respective switches. This scheme is well suited for modern network infrastructures comprising specially cloud based multi-tenanted user environments because these requires frequent changes in business and user requirement [2,3]. However, cloud based system or data centers are not the only beneficiary of the SDN technology. The scope of SDN is also expanded in network environment including 5G mobile communications to campus based network due to the notion of real time network programmability which is offered by software defined networking. SDN attempts to build a computer network by separating into two systems. The first system is the control plane which provides performance and fault management via net flow IP fix SNMP (Simple Network Management Protocol) and other standard protocols[4,5]. It typically handles configuration management of the SDN compliant devices and understands the network topology loaded with these details. The controller can process connection request based on desired requirements such as QoS (Quality of Service) levels. The controller can also perform link management between devices. The second system is the data plane which is responsible for forwarding traffic to the selected destination. Switches can either be reliant on the controller to make forwarding decisions or make the decisions on their own. The control plane configures connection paths or flows into the data plane through the use of a control protocol. The control protocol is used by the controller in a software defined network to perform important functions such as connection set-up[4,5].

1.2. Software-Defined Networking

Today's networks have two functions. The first is the data plane whose task is to forward packets to their ultimate destination. But in order for the data plane to work, we also need a way of computing the state that each of these routers has that allows the routers to make the right decision in forwarding traffic to the destination[2,3]. The state that lives in each of these routers that allows the routers to make these decisions about how to forward packets which are called routing tables. It is the job of the network's control plane to compute these routing tables. In conventional networks, the control and data plane both run on the routers that are distributed across the network.

In SDN, the control plane runs in a logically centralized controller. Additionally, the controller typically controls multiple routers across the network and often, the control program exerts control over all the routers in the network [2,3]. Thus facilitating network-wide control. These two characteristics are the defining features of SDN. The separation of data and control allows a network operator to build a network with commodity devices where the control

resides in a separate control program. This refactoring allows us to move from a network where devices are vertically integrated and making it very tough to innovate to a network where the devices have open interfaces that can be controlled by software.

SDN is a new technology designed to make the network more agile[1]. Traditional networks are often quite static, slow to change and dedicated to single services. With SDN, we can create a network that handles many-many different services in a dynamic fashion allowing us to consolidate multiple services onto one common infrastructure for both service providers and carriers. For e.g., a service provide may want to allocate most of their metro network bandwidth to business services during the day, to internet services during the evening and to new high capacity on demand batch jobs and virtual machine movements as well as backup solutions overnight. This allows the service provider to consolidate three different networks into one powerful network that can be allocated on demand[4,5].

The two essential concepts behind SDN are abstraction and automation. By abstracting the network service required for an application into a network policy, applications are freed from underlying low level complexity and empowered to program their network connectivity. SDN automates the manual time intensive request based provisioning model. The network policy instantiates new application configuration tasks automatically replacing a laborious expensive process possibly hundreds of times a day. Nuage networks virtualized services platform unleashes the full power of SDN.

In 1999, we all were really cared about the size of the pipe. It was 1.5 megabits per second. At that time we were transferring files, we were transferring pictures, we were transferring entire video files, and we were transferring emails[5].

Basically, everything was a file. We only cared speed. How fast can we transfer entire files from point A to point B. Since we were transferring files now let's move forward to about 2005. YouTube comes out, Skype is now around, and Voice over IP becomes a bigger deal. Digital surveillance becomes an issue. So now we started dealing with YouTube, Skype, and voice over IP, we are now doing real time communication. Real time communication (RTC) needs something different than simple transfer of files. So what real time communication cares about is something called latency.

So before it was how fast we move a chunk of data from point A to point B. With latency, we cared about how fast we can get individual packets from point A to point B. The more the latency is, the longer it takes for these individual packets and the worse this real time communication will work. If there is more latency then jitter, then it results in timing drift, network congestion and route changes[3,4,5]. Jitter is the packet delay variation from sender to receiver. So in speed, we were saying this is a 10 megabit per set connection. In latency, we generally talk about like 21 millisecond connection. So that is what we were worried about in 2005. We were dealing with file stuff and then we were dealing with RTC which were 2 separate things we dealt with. So with RTC, we have dealt with latency and that is when we added to our networking equipment called QoS. What quality of service allowed is? It is allowed for the ability to prioritize packets[6]. So we could say that a VoIP (voice over IP) packet was more important than an FTP (File Transfer Protocol) packet. Since the void packet needs less latency so it moved down the pipe faster than the FTP packet. When we were dealing with RTC in 2005 time frame that is when they really started pushing quality of service (the prioritization of packets). So first, we had speed then we had QoS. Well, now we get to the 2011-2012-2013 timeframe and when we are dealing with our networking systems, it is becoming more and more complicated. So we used to say that FTP traffic is of low priority and sip traffic is of high priority. It was really easy to say. Now as more and more devices are being connected to the network, FTP traffic is more important than sip traffic. Sometimes video traffic is more important than voice traffic and sometimes it's less important. The problem is with the systems that we have with QoS, We can't dynamically configure this information. We can say basically that one packet is more important than another. We cannot programmatically state that at certain times FTP traffic is more important than its sip traffic and other times sip traffic is more important than FTP traffic[4]. Well, that is where this SDN or Software defined networking comes into play. Now we can dynamically model and shape our traffic depending on what we need to do. So essentially the big thing is that we separate networking into the control plane and to the data plane. What this means, the data plane is what actually has all the switches and all the routers that allow packet to go from point A to point B. So the data plane is the switches, the routers, the bridges all of them. The control plane is a set of management servers that communicate with all of the different networking equipment on the data plane and it tells all sip traffic should have priority over FTP traffic and then we can say, all SMTP traffic should have more priority over sip traffic. From a control console, we are able to control the entire network at 1 place[4,5,6]. So the big thing with SDN is that we are separating out the different components of the networking infrastructure so that we can deal with them separately.

1.3 SDN Architecture

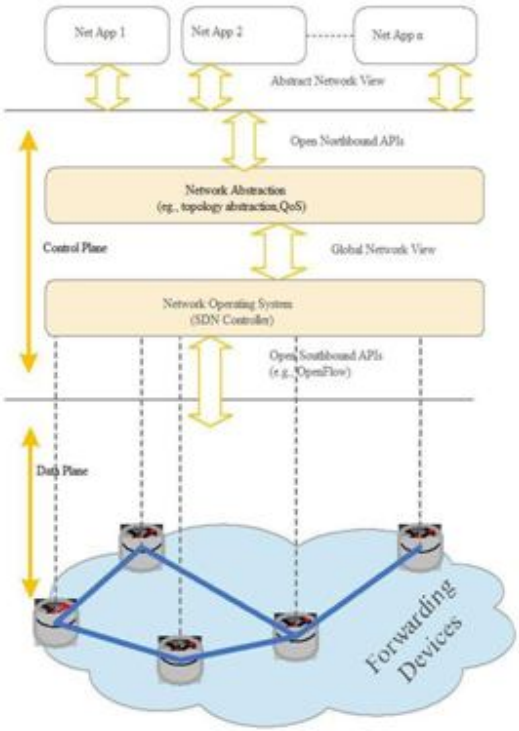

Fig. 1 Basic SDN Architecture

The SDN architecture could be divided in 3 distinct plane. At the very bottom is the data plane which comprises of hardware such as network switches. Above the data plane is the network control plane. This comprises of a centralized controller which computes and communicates network policies to individual devices. The centralized controller could be as simple as a server machine attached to the network running on controller software. Residing above the control plane is the application plane[4,5,6]. This plane comprises of individual applications which could be network monitoring utilities, voice over IP applications which has a particular set of requirement such as delay, latency etc. These applications communicate their requirements to the controller on the control plane. Communication between the application and the control plane is by means of northbound application programming interface (API) such as the restful protocols. While the controller communicates with the data plane devices such as network switches using the southbound application programming interface commonly using the open flow protocol. The application residing at the top of the SDN architectural hierarchy, therefore this is an

abstract view of the available network resources and it is the job of the network controller to translate these requirements into policies or rules in individual network devices.

II. SDN FOR ENTERPRISE NETWORKS

Today, a wide range of business applications inside enterprises are being used over a network. In addition to ordinary companies, IT infrastructure is now utilized across various fields such as education and hospital facilities.

Secure Enterprise SDN (or SES) from AT (Allied Telesis) reduces the cost associated with network operation and management and enhances security by relating the applications and networks utilized in enterprises and various other industries.

SES, based on Openflow, dynamically determines the access policy of enterprise network by getting various information from enterprise applications. This architecture will provide high security and usability for enterprise networks. Specifically the solution comprises three enterprise application and cloud services. The three applications are: Comprehensive Threat detection engine from Trend Micro, IT Asset Management software from Quality Soft, and a Human resource management service from Lacrasio. SDN controller from Allied Telesis will automatically maintain enterprise networks related to these applications.

Secure Enterprise SDN has three key features, the first is Application related provisioning. When personnel are transferred within a company. An administrator has to gather information such as user section, device information and working hours from separate databases under a legacy network, and then assign and configure the information manually one by one.

Today since it has become so normal for a single user to own multiple devices, an administrator has to consume a lot of time for the maintenance and configuration which is very complicated. The application related provisioning which consists of a human resource management workflow, ITS management software and the SDN controller automatically configures ENTERPRISE LAN.

This removes the need to configure individual settings and enables simple operation and management[8]. For eg., An Administrator registers personnel information using a Human resource management service in advance. Then, only the user device and such information needs to be updated through the IT asset management software which will be distributed to the SDN controller. The users will be able to use the network from a point of joining their new sections. This enables automatic network access control based on people's

place and time. Dramatically reducing complicated operational management.

The second feature is security control by venerability countermeasure. After devices are granted through the provisioning feature, the devices will deal with various information and that will include personal and confidential information. In such cases, unless there is a high level security in the device itself, there is a risk of information security incidents, such as personal information leakage. Even when bad software is installed on a device and security measure alerts the Administrator or a user following a security policy, the risk of security incident in enterprise will not resolve until the infected device is blocked.

However, SES guarantees much stronger security compared to legacy measures. When SES detects and identifies the infected device which installed banned software that violates the security policy, the device is automatically isolated in a quarantine or recovery network. This feature prevents the spread of damage and enables the operation of a higher level secure network.

The third feature is behavioral detection. Most of the time, security control itself has to check vulnerability which is not enough as a protection. Because the users network access behavior can face a threat of virus infection which could lead to an IT incident. Malware and other threats are becoming increasingly more effective each year and cannot be completely prevented through antivirus software alone. The latest comprehensive threat detection engine is capable of detecting threats such as Malware, but it can only alert. The administrator has to rush and isolate the infected device manually, but this leaves a huge security risk of spreading the damage[11,12].

SES has the answer to change the situation. The behavioral detection monitors network traffic in real time. For e.g., when a user accesses a harmful website and is suspected to be infected with Malware, SCS indicates infected device and automatically isolates it in a quarantine or recovery network in order to prevent an IT incident. To activate SES just add in SDN controller and update firmware of the existing network by Allied Telesis. SES is an ideal solution for enterprises, whether it is a wired or wireless network. Secure Enterprises SDN can improve usability, provide high level security and reduce operational costs by integrating various applications.

III. ADVANTAGES AND DRAWBACKS OF SDN MODEL

SDN doesn't require expensive high end switches and can run on inexpensive hardware or switches. It doesn't have a lot of processing lead because of in expensive high end switches. It helps to lower the organization operating cost control. SDN also provide central control method which helps in deciding network data flow since the controller can be managed easily by administrators. Basically, they can manage the entire network data flow centrally. SDN provides dynamic response since the SDN response actively to request from the clients in the real time. It's basically managing the network dynamically. This is how the network is managed in real time. SDN provides results optimization[6]. SDN optimally uses all the available bandwidth since the controller knows about what traffic is flowing through which switches. It also knows if any switches are idle and the controller can redirect traffic through those idle switches in this way. It's basically load balancing the network packet filtering.

SDN can filter the packets as they enter in the network says that all the switches now act like a firewall. This helps to control all the malicious activities and as well as secure the data. SDN can follow the certain suspicious traffic for thread detection and data loss prevention fault tolerance handling so we can also replicate the controller. By replicating the controller and the cluster we can create a fault tolerance controller. We could provide high availability of the network.

Every system have some disadvantages and drawbacks. By analyzing the drawbacks, it helps us to understand the limitation of the system[7].

SDN has a drawback of scalability performance issue. It's tough for a single controller to produce all the connection requests and forwarding decision. This is managed by replicating the controller but then this controller must be synchronized which brings more challenges that can be achieved. However, this is difficult. Basically, this controllers need to be synced and synchronization can be real challenge for the network administrator. The second drawback is technology change. Any change is scary. SDN is a new technology that requires training and buying a new software and hardware for the network. It might also require some effort to clean out all settings and policies in the network and definitely needs some effort from the company which may be scary for the already working production system. Enterprise are great being in new standards. Different people have somewhat varying definition of SDN and that all the protocols and devices might be working with SDN especially because of a lot of legacy system present in all network. Software of all

devices may not be compatible, so a lot of large companies are waiting for some time and conducting research and feasibility analysis and are waiting to wearing a year or so before adopting it.

The another drawback is central point of failure. If central controller gets hacked then the whole network is hacked or if something goes wrong with this controller or the replicated few controller and the network is in trouble.

SDN gives us more granular service oriented system to manage the security especially with the thousands of devices in the network where more devices are added and removed ironically. Whenever we transfer the data from one point to another point, it has a set of rules defined and guided by the controller which protects it's from hacking and doing analysis on the network.

SDN provides efficient networking, computing and storage resources management and planning. Since there is a central controller, it gives us a flexibility in setting up the rules, changing the route from one location that is controller. We don't have to go through all the devices and servers to set up rules, ACLs (Access Control List) etc. This also gives us a central view of the network which further helps in planning network topology and network traffic management. Flexibility in network operations, in this again centralizing helps in changing the rules and operating the policy from one location controller[9,10]. Even if we have to hand over and adopt a new policy in one location then we don't have to debug or run through hundreds of servers. In corporate network protocols, in this, we can update and incorporate new protocols easily by managing the controller[15,16].

IV. APPLICATIONS OF SDN

Software defined networking which is network virtualization, so we will first know about what network virtualization is and then we will know about how it is implemented and then we will know about some examples and applications such as mini net[8].

Network virtualization is simply an abstraction of the physical network where multiple logical networks can be run on the same underlying shared physical substrate[1,8]. For e.g., a logical network might map a particular network topology on to the underlying physical topology and there might be multiple logical networks that map onto the same physical topology and these logical networks might actually share nodes and links in the underlying physical topology but each logical network has its own view as if it were running its own private version of the network. The nodes in the physical network need to be shared or sliced. So the nodes in the physical topology might be virtual machines. Similarly a single link in the logical topology might map to multiple links in the physical topology. The mechanism to achieve these virtual links is typically through tunneling. So a packet that's destined from A to B in the logical topology might be encapsulated in a packet that's destined for node X first before the packet is encapsulated and ultimately sent to be. It may also be easy to understand virtual networking as an analogy to virtual machines which we may be familiar with already. So in a virtual machine environment, we have virtual machines where a hypervisor arbitrates access to the underlying physical resources providing to each virtual machine the illusion that its operating on its own dedicated version of the hardware[17,18].

Similarly with virtual networking a network hypervisor of sorts arbitrates access to the underlying physical network to multiple virtual networks providing the illusion that each virtual network actually has its own dedicated physical network[7,8].

SDN controller that is centralized and it is using the Open Flow protocol to manage the data plane on the access layer devices. So in this case, we have got 80 specific SDN flow created that takes any DNS request that is sent through. So it is just the DNS request not the entire flow and sends it up to the SDN controller for interrogation and the SDN controller in this case has a security application running on it which is going to interrogate that DNS packet against its vulnerability database to see if there is s problem with that particular packet and if there is then, it will actually go through and block it but in this case, we don't have any problem with the packet goes through the infrastructure. In the case that we present a request to a malware site, the specific DNS packet is rerouted over to the SDN controller not the entire flow and the SDN controller looks at the DNS request against the reputation database identifies it as a vulnerability or a probable[10]. The nice thing about this approach versus the traditional ids approach is because we are only looking at specific packets. We are not looking at the entire flow. This doesn't require have the same scale issues that you would have with a traditional IDs system. So it's much easier to add more access devices to the environment because we are only pulling out a portion of the flow instead of trying to look at the entire flow. The other nice thing about this particular solution over traditional IDs is that we are actually able to manage the security at the perimeter of the network. So instead of doing that management inside the core, inside the aggregation points of the environment. We have actually turned these off-the-shelf switches that are running open flow on them into security collectors. So we are actually able to manage the environment on the ingress[19].

V. SDN VS. TRADITIONAL NETWORK INFRASTRUCTURE

In traditional networking, the switch doesn't have programmability. The rules cannot be changed dynamically as per our wish. For e.g., The Skype call gets cut when the path is down.

In Software Defined Networking, the switch is connected to a controller, which controls the actions of the switch. The controller can be programmed dynamically to control the switch, thus giving programmability to the switch. Hence the switch contacts the controller to perform any action and will not do anything on its own[19].

For e.g., When the Skype call gets cut, the SDN switch receives appropriate command from the controller. Controller tells switch to take alternate path and reroute the packets. Then the SDN switch takes an alternate path as per the controller's instruction.

VI. CONCLUSION

SDN increases flexibility, visibility and efficiency while reducing complexity and costs. SDN are more dynamic networks, easy to manage, quicker to react and more efficient at delivering services with or without elastic bands. In this paper we have discussed SDN Architecture. Different Enterprise networks architectures. This also provides applications of SDN like virtualization and advantages and disadvantages of SDN. This paper compares SDN with traditional network infrastructure and problems that are solved with SDN.

REFERENCES

[1] "The future of network virtualization and SDN controllers", Oct. 2016, [online] Available: https://www.temperednetworks.com/sites/default/files/pdfs/fe atured_content/2016-SDN-NV-Report_Tempered_Networks.pdf.

[2] S. K. N. Rao, SDN and its use-case–NV and NFV, Oct. 2014, [online] Available: http://www.nectechnologies.in/en_TI/pdf/NTI_whitepaper_SDN_NFV.pdf.

[3] Software-defined networking: The new norm for networks, Oct. 2012, [online] Available: https://www.opennetworking.org/images/stories/downloads/s dnresources/white-papers/wp-sdn-newnorm.pdf.

[4] J. Vestin, A. Kassler, J. Akerberg, "Resilient software defined networking for industrial control networks", 2015 10th International Conference on Information Communications and Signal Processing (ICICS), pp. 1-5, 2015.

[5] D. Levin, A. Wundsam et al., "Logically centralized?: state distribution trade-offs in software defined networks", Proc. ACM HotSDN12, pp. 1-6, 2012.

[6] Brocade Communications Systems, "Network Transformation with Software-Defined Networking and Ethernet Fabrics," California, USA, http://www.brocade.com/downloads/documents/positioning-papers/network-transformation-sdn-wp.pdf, 2012

[7] NICIRA, "It's Time to Virtualize the Network," http://www.netfos.com.tw/PDF/Nicira/It%20is%20Time%20To%20V irt ualize%20the%20Network%20White%20Paper.pdf, 2012

[8] Lippis, N. J., "Network Virtualization: The New Building Blocks of Network Design," https://www.cisco.com/en/US/solutions/collateral/ns340/ns517/ns431/ ns 725/net_implementation_white_paper0900aecd80707cb6.pdf, 2007.

[9] Oltsik, J., Laliberte, B., IBM and NEC Bring SDN/OpenFlow to Enterprise Data Center Networks, Enterprise Strategy Group Tech Brief, 2012.

[10] Ashton, Metzler, and Associates, Ten Things to Look for in an SDN Controller, Technical Report, 2013.

[11] Ferro, G., OpenFlow and Software-Defined Networking, http://etherealmind.com/software-defined-networking-openflow-so-far-and-so-future/, November 2012.

[12] Yazýcý, V.,Sunay, O., Ercan, A.O., "Controlling a Software-Defined Network via Distributed Controllers", NEM Summit, Istanbul, Turkey, http://faculty.ozyegin.edu.tr/aliercan/files/2012/10/YaziciNEM12.pdf, October 2012.

[13] Macapuna, C.A.B., Rothenberg, C.E., Magalhaes, M.F., "In-Packet Bloom Filter-Based Data-Center Networking with Distributed OpenFlow Controllers, "IEEE 2010GLOBECOM Workshops, pp.584– 588, 6–10 December 2010.

[14] Big Switch Networks, The Open SDN Architecture, http://www.bigswitch.com/sites/default/files/sdn_overview.pd f, 2012.

[15] OpenFlow Switch Consortium, OpenFlow Spec, v1.3.0 https://www.opennetworking.org/images/stories/downloads/s dn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf, 2012.

[16] Fang, S.,Yu, Y., Foh, C.H., Aung, K.M.M., "A Loss-Free Multipathing Solution for Data Center Network Using Software-Defined Networking Approach," IEEE Transactions on Magnetics, vol.49, no.6, pp.2723– 2730, June 2013.

[17] Yeganeh, S.H., Tootoonchian, A., Ganjali, Y., "On Scalability of Software-Defined Networking," IEEE Communications Magazine, vol.51, no.2, pp.136–141, February 2013.

[18] Voellmy, A.,Wang, J.C., "Scalable Software-Defined Network Controllers," Proceedings, ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, pp. 289–290, 2012.

[19] Tavakoli, A., Casado, M.,Koponen, T.,Shenker, S., "Applying NOX to the Data Center," Proceedings, Ninth ACM SIGCOMM Workshop on Topics in Networks (Hotnets-IX), October 2009