ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

The Exposition of Enigma Software Defined Networking

Mansimar Singh[1], Gurpreet Singh[2]

Student, Computer Science and Engineering, GNDU Regional Campus, Sultanpur Lodhi, India[1]

Assistant Professor, Computer Science and Engineering, GNDU Regional Campus, Sultanpur Lodhi, India[2]

Abstract: SDN helps networks keep up with the speed of change, made possible by virtualization, convergence and cloud, by centralizing and simplifying control thereby making them more agile and programmable. The basic idea is unriddled and whys, whats, hows and whens of software defined networking(SDN) are to be discussed along with its pros and cons. The basic idea behind SDN and the OpenFlow protocol will be explained in detail. Some light would ne shone on the traditional data network along with illuminating the shift to software and why is it needed.

Keywords: OpenFlow, traditional data networks, centralisation of control plane, separation of control, data plane.

I. INTRODUCTION

Over the last few years SDN has become one of the most ballyhooed topics in IT. But despite the hype it is a highly misunderstood technology shift. According to Network World, only 10% of 450 IT practitioners who attended a recent event raised their hands when asked if they understood it. It would be easy to ignore SDN as yet another passing fad, but according to the new research by Plexxi, SDNCentral and Lightspeed Venture Partners, the SDN market is expected to surpass $35 billion in the next 5 years [1].

Traditional network architectures are ill-suited to meet the requirements of today's enterprises, careers and end users. Many conventional networks are hierarchical, built with tiers of Ethernet switches arranged in a tree structure. This design made sense when the client-server computing was dominant, but such a static architecture is ill-suited to the dynamic computing and storage needs of today's enterprise data centers, campuses and carrier environments [2]. This led to the advent of a more dynamic, flexible and programmable network architecture, SDN.

To understand the promise of SDN we need to start with a short history lesson, explaining the traditional and the soon to be traditional data network.

II. BACKGROUND

A. Traditional Data Network

In the traditional approach to networking, most networking functionality is implemented in a dedicated appliance i.e. switch, router, application delivery controller. Also, in these dedicated appliances, most of the functionality is implemented in dedicated hardware such as an ASIC (Application Specific Integrated Chip).

Some of the key characteristics of this approach to developing network appliances are:

- The ASICs evolve slowly,
- The evolution of the ASIC functionality is under the control of the provider of the appliance,
- The appliances are proprietary,
- Each appliance is configured individually,
- Tasks such as provisioning, change management and de-provisioning are very time consuming and error prone.

Networking organizations are under increasing pressure to be more efficient and agile than is possible with the traditional approach to networking. One source of that pressure is due to the widespread adoption of server virtualisation, in which virtual machines(VM) are dynamically moved from one server to another within a matter of seconds or minutes. Despite all its briskness, if the movement of a VM crosses layer 3 boundary, it can take days or weeks to reconfigure the network to support the VM in its new location. If a network takes weeks to reconfigure then that network certainly isn't agile.

The crux being that a traditional network evolves slowly, is limited in functionality, has a relatively high level of OPEX and is relatively static in nature.

B. The Shift to Software

After largely relying on hardware-centric data networks the tide of software based network functionality came with the adoption of virtualised network appliances and the skyrocketing flourish of SDDC (Software Defined Data Centres). Network appliances such as WAN Optimization Controllers and Application Delivery Controllers have been transformed from being hardware appliances to just being softwares running on general purpose servers or on a VM because of the need for increased agility [3].

An SDDC is a vision for IT infrastructure that extends virtualisation concepts such as abstraction, pooling and automation to all the data centre's resources and services to achieve IT as a service [4]. It is the complete opposite of a traditional data centre network as the entire data centre infrastructure is virtualised and delivered as a service. SDN is the centre piece of the artwork that SDDC is.

DOI 10.17148/IJARCCE.2015.4498

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

Why did this need for a shift to software arise? Why do we need SDN?

1) Virtualisation: Use network resources without worrying about where it is physically located, how much it is, how it is organized, etc.
2) Orchestration: Should be able to control and manage thousands of devices with one command.
3) Programmable: Should be able to change behaviour in real time, dynamically.
4) Dynamic scaling: Should be able to change size, quantity.
5) Automation: To lower OpEx minimize manual involvement.
6) Transparency: Monitor resources and connectivity.
7) Performance: Optimize network device utilization such as capacity optimization, load balancing, fast failure handling, high utilization.
8) Multi-tenancy: Tenants need complete control over their addresses, topology and routing and security.

Service integration: Load balancers, firewalls, Intrusion Detection Systems (IDS), provisioned on demand and placed appropriately on the traffic path.

III. OPENFLOW

Open Flow is considered one of the first software-defined networking standards. It originally defined the communication protocol in SDN environments that enables the SDN/OperFlow Controller to interact with the forwarding plane and make adjustments to the network. Devices wanting to communicate with an SDN/Open Flow Controller must support the standard Open Flow protocol. It is because of this interface that the SDN controller can push down changes to the switch/router flow-table allowing network administrators to partition traffic, control flows for optimal performance and start testing new configurations and applications.[5]

The OpenFlow is a communication protocol that gives access to the forwarding plane of a network switch or router over the network. OpenFlow enables the SDN Controller to determine the path of the network packets through the network of switches. The separation of control from the forwarding allows for more sophisticated traffic management than is feasible using access control lists and routing protocols.[6]
OpenFlow is largely responsible for the origin of SDN.

IV. SOFTWARE DEFINED NETWORKING

OpenFlow laid the foundation for the development of Software Defined Networking. The Open Network Foundation was formed, on March 21, 2011, to promote and adopt software-defined networking. So what is SDN? is the question that arises.

SDN is a framework to allow administrators to automatically and dynamically manage and control a large number of network devices, services, topology, traffic paths and packet handling policies using high-level languages and APIs.

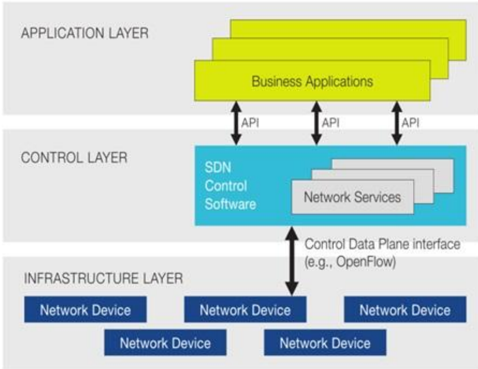

Fig. 1 Software-Defined Network Architecture.

Management includes provisioning, operating, optimizing, monitoring and managing FCAPS (faults, configuration, accounting, performance and security) in a multi-tenant environment [7].

According to the ONF, the SDN architecture is:

A. Directly programmable
Network control is directly programmable because it is decoupled from forwarding functions.

B. Agile
Abstracting control from forwarding lets administrators dynamically adjust network traffic flow to meet changing needs.

C. Centrally managed
Network intelligence is centralized in software-based SDN controllers that maintain a global view of the network, which appears to applications and engines as a single, logical switch.

D. Programmatically configured
SDN lets network managers configure, manage, secure and optimize network resources very quickly via dynamic, automated SDN programs, which then write themselves because the programs do not depend on proprietary software.

E. Open standards-based and vendor-neutral
When implemented through open standards, SDN simplifies network design and operation because instructions are provided by SDN controllers instead of multiple, vendor-specific devices and protocols [8].

Figure 1 depicts a logical view of the SDN architecture. Network intelligence is (logically) centralized in software-based SDN controllers, which maintain a global view of the network. As a result, the network appears to the applications and policy engines as a single, logical switch. With SDN, enterprises and carriers gain vendor-independent control over the entire network from a single logical point, which greatly simplifies the network design and operation. SDN also greatly simplifies the network devices themselves, since they no longer need to understand and process thousands of protocol standards

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

but merely accept instructions from the SDN controllers. The network operators and administrators can programmatically configure this simplified network abstraction rather than having to write tons of lines of code. SDN's centralized intelligence also means that network behavior can be altered in real-time and new applications can be deployed in a matter of hours or days instead of the weeks needed today.

The programs can be written by the network managers and the features can be added dynamically rather than waiting for the vendor's to do so in closed software environments in the middle of the network. Because SDN architecture supports a huge set of APIs it makes it possible to implement common network services, including routing, multicast, security, access control, bandwidth management, traffic engineering, quality of service, processor and storage optimization, energy usage, and all forms of policy management, custom tailored to meet business objectives. For example, SDN architecture makes it easy to define and enforce consistent policies across both wired and wireless connections on a campus. With open APIs between the SDN control and applications layers, business applications can operate on an abstraction of the network, leveraging network services and capabilities without being tied to the details of their implementation.

Because of the decoupling of the control and application layer SDN is not 'application-aware', rather it is 'application-customized' as in you could use a switch to act as a router as well as a firewall or a load balancer with just a few tweaks in the application layer code.

A few of the advantages of SDN are listed below:
1) Centralized control of multi-vendor environments: Software defined networks provide a centralized view of the entire network, making it easier to centralize enterprise management and provisioning. SDN control software can control any OpenFlow-enabled network device from any vendor, including switches, routers, and virtual switches. By abstracting the control and data planes, SDN can accelerate service delivery and provide more agility in provisioning both virtual and physical network devices from a central location.
2) Reduced complexity through automation: Because of automation many management tasks that are done manually today can be done using tools. These automated tools will reduce operational overhead, decrease network instability introduced by operator error and support emerging ITAAS.
3) Holistic enterprise management: SDN allows IT managers to experiment with network configuration without impacting the network. Both physical and virtual switches and network devices can be managed by a central controller. A single management console can be created for physical and virtual devices.
4) More granular network control: Because of a centralized control It policies can be applied at a very granular level, including session, user, device and application levels in a highly abstracted and automated fashion.
5) Increased network reliability and security: The SDN Controller provides a central point of control to distribute security and policy information consistently throughout the enterprise. Centralizing security control into one entity, like the SDN Controller, has the disadvantage of creating a central point of attack, but SDN can effectively be used to manage security throughout the enterprise if it is implemented securely and properly. SDN architecture eliminates the need to individually configure network devices each time an end point, service, or application is added or moved, or a policy changes, which reduces the likelihood of network failures due to configuration or policy inconsistencies.
6) Better user experience: By centralizing network control and making state information available to higher-level applications, an SDN infrastructure can better adapt to dynamic user needs. Today, users must explicitly select a resolution setting, which the network may or may not be able to support, resulting in delays and interruptions that degrade the user experience.
7) Cloud abstraction: By abstracting cloud resources using software defined networking, it's easier to unify cloud resources. The networking components that make up massive data center platforms can all be managed from the SDN controller.
8) Hardware savings and reduced capital expenditure: SDN can give new life to existing network devices. An old hardware can be optimized and customized to work even better than a new one. Existing hardware can be repurposed using instructions from the SDN controller and less expensive hardware can be deployed to greater effect since new devices essentially become "white box" switches with all the intelligence centered at the SDN controller.
9) Lower operating costs: Administrative efficiency, improvements in server utilization, better control of virtualization, and other benefits should result in operational savings. Although it is still early to show real proof of savings, SDN should lower overall operating costs and result in administrative savings since many of the routine network administration issues can be centralized and automated [9].

SDN is not just a bed of roses it also has some thorns that might be erratic king its blissful aura. Some of its disadvantages are:
1) SDN is easy if control plane is centralised but not necessary. Distributed solutions may be required for legacy equipment and for fail-safe operation.
2) Complete removal of control plane may be harmful. Exact division of control plane between centralized controller and distributed forwarders is yet to be worked out
3) SDN is easy with a standard southbound protocol like OpenFlow but one protocol may not work in all cases.

Diversity of protocols is a fact of life. There are no standard operating systems, processors, routers or Ethernet switches.

4) If industry finds an easier way to solve the same problems by another method, that method may win. e.g., ATM vs. MPLS.

V. CONCLUSION

Although the origins of SDN began around 1995 it was relatively unknown for about 2 decades. But after countless researches and modifications and the foundation of Open Network Foundation in 2011 SDN exploded on the scene and is now one of the most coveted technologies that every major and minor, IT and even non-IT enterprise wants to capture and be a part of. The SDN market is growing faster than anyone anticipated with the market expected to grow up to $35bn by 2018. The number of companies registered as SDN companies grew from 0 to 225 just in a miniscule time period of 3 years, 2000-2013 [10].

With the generation next's need for mobile computing, BYODs, server virtualization and IT-as-a-service's rise to fame the traditional network architectures are not going to survive for long and the need to respond to changing business conditions places SDN in a prime position to be the next big thing. SDN's dynamic architecture, on the go customization will transform traditional network backbones into rich service-delivery platforms. Because of decoupling the network control and data plane OpenFlow-based SDN architecture is programmable, flexible, scalable, resilient, reliable, secure and cost efficient, kind of like a dream architecture. With a million to one pros to cons ratio SDN is not going to be the one that got away.
The future of networking would rely more and more on software, which would accelerate the pace of innovation for network as it has had in the computing and storage domains. SDN promises to transform today's static networks into flexible platforms with the intelligence to allocate resources dynamically and the scale support enormous data centers and the virtualization needed to support dynamic, highly automated and secure cloud environments, thereby making it the new norm for networks of the time soon to come [11].

ACKNOWLEDGMENT

I earnestly wish to extend my heartiest gratitude to my venerable mentor **Er. Gurpreet Singh**. His exemplary guidance has made a colossal contribution to my venture. I stand beholden to the faculty members of the Dept. of Computer Science and Engineering, Guru Nanak Dev University for being my beacon light and source of constant support and motivation.

REFERENCES

[1] "Is the future of networking software defined?", CS / DDMS-1366 / 06/13, June, 2013.
[2] Open Network Foundation, "Software-Defined Networking: The New Norm for Networks", April 13, 2012.
[3] Citrix, "SDN 101: An Introduction to Software Defined Networking", 2014.
[4] Software-defined data centre on Wikipedia, Mar. 19, 2015 .[Online]. Available:http://en.wikipedia.org /wiki/ Software-defined_data_center
[5] Open Flow [Online]. Available: http://archive .openfow.org/wp/ leranmore/
[6] OpenFlow on Wikipedia, Dec. 15, 2014. [Online]. Available: http://en.wikipedia.org/wiki/OpenFlow
[7] Raj Jain, Washington University, "Introduction to Software Defined Networking(SDN), 2013.
[8] Citrix, " SDN 101: An Introduction to Software Defined Networking", 2014.
[9] Open Network Foundation, "Software-Defined Networking: The New Norm for Networks", April 13, 2012.
[10] SDN Market Size Forecast: April, 2013 on sdxcentral, Apr., 2013. [Online]. Available: https://www.sdxcentral.com/sdn-market-sizing-2013-april/
[11] Open Network Foundation, "Software-Defined Networking: The New Norm for Networks", April 13, 2012.

BIOGRAPHIES

Gurpreet Singh received his B.Tech. degree in Information Technology from DAV Institute of Engineering & Technology, Jalandhar, Punjab, India in 2009 and M.Tech. degree in Computer Engineering from University College of Engineering, Punjabi University, Patiala, India in 2011. He was teaching as a Lecturer at Rayat Bahra College of Engineering and Bio-Technology for Women, Mohali, Punjab, India. Presently, he is teaching as Assistant Professor at Guru Nanak Dev University, Regional Campus Sultanpur Lodhi. He has more than 20 research publications in International, National Conferences and Journals. His research interest includes Wireless Communication, Network Security, Computer Networks and Data Communication.

Mansimar Singh is pursuing his B.Tech in Computer Science and Engineering from Guru Nanak Dev University, Sultanpur Lodhi, Punjab. He is a proud recipient of a merit certificate by C.B.S.E. for reasons of outstanding academic performances and being among the top .01 percent of successful candidates that appeared in 10th board. He likes to indulge himself in various research oriented technologies the likes of which include Cloud Computing, Openstack, SDN, Light Weight Virtualisation.