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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

Software Defined Networking - A Networking Paradigm to meet the emerging trends

Ms.Kiruthika M.[1], Mr.Vivek Kaarthek[2], Mr.Srikant Shetty[3], Mr.Roshan Kadam[4]

Associate Professor, Department of Computer Engineering, Fr. C. Rodrigues Institute of Technology,

Navi Mumbai,India[1]

Department of Computer Engineering, Fr. C. Rodrigues Institute of Technology, Vashi, Navi Mumbai, India[2,3,4]

Abstract: The past decade has seen the emergence of new computing trends namely Big Data Analysis, Server Virtualization, Mobile computing, Cluster computing and Cloud Computing. While IT sectors have improved in compute and storage, the networking industry is yet to meet the performance needs of these emerging trends. In this paper, Software Defined Networking (SDN), an emerging network paradigm that promises to revolutionize networking and meet the performance needs of today's enterprises and data centers is discussed. SDN is the physical separation of the network control plane from the forwarding plane, where the control plane controls several devices. In this paper we have described the SDN architecture and its implementation with the open flow protocol. Some of its benefits and how it can meet the current limitations of traditional network architectures is also discussed.

Keywords: SDN**,** Openflow, Controller, Architecture

I. INTRODUCTION

For over the past two decades, networking architectures have used devices that generally have proprietary management systems and protocols, and lack reliable interfaces (APIs) for external programmability or automation. These networking architectures are inherently non-scalable, inflexible, and administratively cumbersome and are unable to meet the requirements of computer trends that have emerged over the recent years. Some of the key trends are as follows:

- Dynamic and unpredictable traffic patterns
- BYOD (Bring your own device)
- Cloud computing
- The rise of "big data" and cluster computing

While the enterprise compute sector has experienced improved flexibility and scalability to meet these trends, there are certain issues to be addressed [1] with respect to the networking industry. They are:

- Inability to experiment new ideas [2]: Due to an enormous installed base of equipment and protocols, Researchers have been unable to test new ideas in realistic settings with production traffic. This has resulted in a belief that the networking industry has stagnated and inhibited innovation in this field. In fact, this is one of the reasons that researchers have cited [2] as the motivation for creating open flow.
- Complexity and inconsistent policies: As mentioned before, networking has seen the development of a multitude of protocols that have been designed in isolation. Network managers have to work with multiple protocols and interfaces which are cumbersome to manage and troubleshoot. As a result of this complexity, to even configure a network-wide policy, Network managers may have to configure thousands of devices and mechanisms. This makes it very difficult for implementing consistent policies for access control, security, QoS etc.
- Inability to scale: With emerging trends like server virtualization and cluster computing, the demands on data center networks have grown exponentially. Thousands of network devices and millions of virtual servers may need to be configured dynamically which simply cannot be done by manual configuration.
- We believe that the SDN architecture can overcome these challenges and enable much needed research in networking. By decoupling forwarding and control functions, centralizing network intelligence and state information, and providing an abstraction layer with open APIs (e.g. Openflow), SDN allows network managers to build more scalable, agile and easily-manageable networks.

II. SOFTWARE DEFINED NETWORKS

In a traditional network, data flow is controlled by switches and routers. Each switch and router contains the following basic elements:

- Data plane: It physically carries data packets from one port to another by following rules that are programmed into the device hardware (e.g. TCAMs). The data forwarding plane operates at the speed of the network (line rate).
- Control plane: It contains the logic that the device uses to program the data plane, so packets are forwarded correctly throughout the network. E.g.: Network Topology, L2 Switching Decisions, Routing, ACLs, Link Management etc.
- Management plane: It lets an administrator log in to the device and configure it for basic activities. Most devices can be configured locally or through a network management tool. E.g.: Switch Management:

Telnet SSH, SNMP, SYSLOG, NTP, HTTP, FTP/TFTP etc.

Traditionally, Control plane and Management planes are implemented in proprietary firmware that is tightly coupled to the device.

Software Defined Networking (SDN) is an emerging network architecture where network control (Control & Management plane) in the switch is decoupled from forwarding (Data plane) and is directly programmable. The Management & Control planes in the traditional switch are now replaced by an openflow API, while the Control & Management planes are transferred to the Software controller, which then uses the openflow protocol to control and manage the devices. This migration of control (as shown in Fig. 1), formerly tightly bound in individual network devices, into accessible computing devices enables the underlying infrastructure to be abstracted for applications and network services, which can treat the network as a logical or virtual entity (Fig. 2).

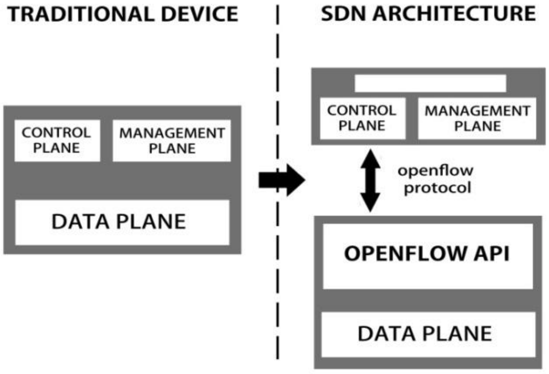

Fig. 1. Migration of control in SDN

A. SDN Architecture
As shown in figure, the SDN architecture consists of 3 layers. They are as follows:

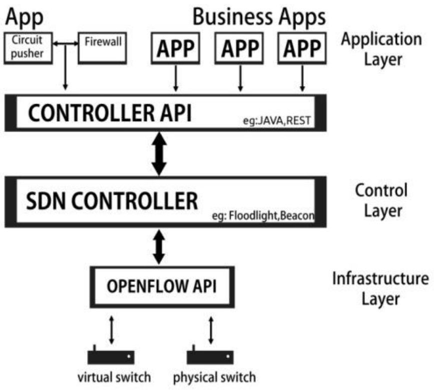

Fig. 2. SDN Architecture

1) Infrastructure layer:
This layer consists of openflow enabled physical switches or hypervisor switches. As seen in Fig. 2., the device consists of an openflow API that allows the switch's flow tables (typically built from TCAMs [3]) to be programmed. A Secure Channel that connects the switch to a remote control process (called the controller), allowing commands and packets to be sent between a controller and the switch. The openflow protocol defines the packet format for enabling the communication between the switch and controller. Companies like IBM [4], HP [5] etc. are providing firmware for enabling openflow in their physical switches. Similarly, Openvswitch [6] is an example of an openflow enabled hypervisor switch.

2) Control layer:
This layer consists of the remote control process called the controller. The controller is basically software, residing on a computer or server, that implements the control and management plane functions of traditional networking devices. The controller uses the openflow protocol to program the flowtables in networking devices. By providing a simplified network abstraction and an open API to the upper layer, the controller enables network managers to alter network behavior in real-time and deploy new applications and network services in a matter of hours or days, rather than the weeks or months needed today. Beacon [7] and floodlight [8] are examples of controllers (java based) that provide a Java and REST API respectively.

3) Application layer:
This layer consists of networking softwares and business applications that are built over the APIs provided by the controller. Moreover, with the controller providing a centralized network state, these applications provide network managers with the flexibility to configure, manage, secure, and optimize network resources dynamically.

B. Openflow
OpenFlow is the first standard communications interface defined between the control and infrastructure layers of SDN architecture. OpenFlow allows direct access to and manipulation of the data plane of network devices such as switches and routers. OpenFlow uses a well defined set of matching rules to classify network traffic into flows. Network managers can use OpenFlow's match rule attributes (table I), to take forwarding actions. It also defines a set of actions that the network architect can use to instruct OpenFlow-enabled network devices to manage these flows. These actions could be forwarding packets to switch ports or the controller, flooding along Spanning Tree, dropping packets, or pushing the packets through the device's normal packet pipeline.

Table I
OpenFlow's match rule attributes

In Port	VLAN	Ethernet			IP			TCP	
		SA	DA	Type	SA	DA	Type	SA	DA

III. ISSUE ADDRESSED BY SDN

Following issues of traditional networking architectures can be addressed by SDN.

1. As mentioned before openflow uses a well defined set of matching rules to classify network traffic into flows. This ability can enable the network administrator to classify traffic as experimental traffic and production traffic in an openflow network (Fig. 3.). For example, consider a researcher experimenting with a new routing protocol in the openflow network. Traffic belonging to users other than the researcher could be routed using a standard and tested routing protocol running in the switch or routers, while those belonging to the researcher could be forwarded to the controller. When her packets reach the controller, her new protocol (running as an application over the controller) chooses a route and adds a new flow-entry (for the application flow) to every switch along the chosen path. Additionally, the administrator could also provide her the rights to add new flow entries over production traffic to study and analyze her protocol. Thus openflow networks can enable researchers to experiment in realistic scenarios with actual production traffic.

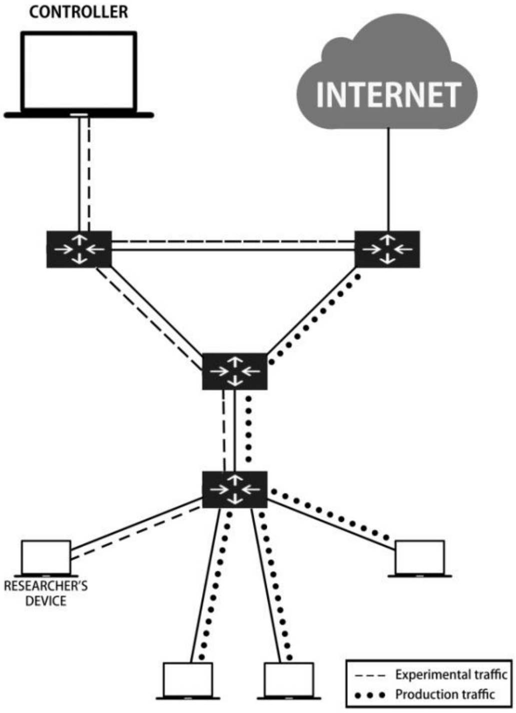

Fig. 3: Classification of traffic as production and experimental traffic

2. With a multitude of protocols and proprietary software, traditional networks have become complex and difficult to configure. To even configure a network-wide policy, Network managers may have to configure thousands of devices and mechanisms, which might result in inconsistent policies. SDN promises an easier, more dynamic interaction with the network by providing a simplified network abstraction (shown in Fig. 4). This reduces the complexity of managing, provisioning, and changing the network and enables network managers to deploy network -wide policies in a matter of hours or days, rather than the weeks or months needed today.

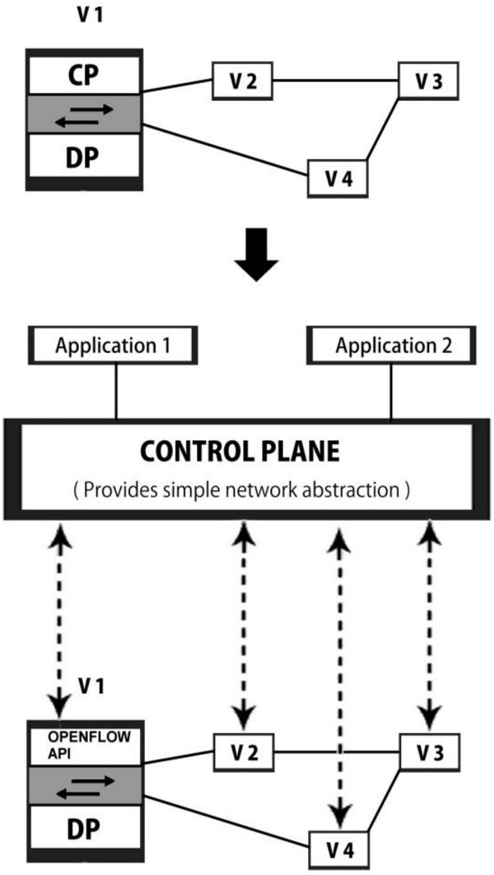

Fig. 4. Simplified network abstraction by SDN,
V- vendor ;CP-Control plane; DP- Data plane

3. With emerging trends like server virtualization and cluster computing, traditional data center networks simply cannot scale to meet the performance demands of these trends. SDN has the ability to provide network virtualization and automation of configuration across the entire network/fabric so new services and end systems can be deployed rapidly and operational cost can be minimized. This automation can be used to configure thousands of networking devices and virtual servers on the fly thus enabling scalability.

IV. ADVANTAGES OF SDN

Apart from addressing the limitations of current networking architectures SDN has the following benefits.

1. Centralized control of multi-vendor environments:
In SDN, the switch's proprietary software is replaced by the openflow API .This enables a centralized software to control and manage devices from multiple vendors. With this centralized control, IT can quickly deploy, configure, and update devices across the entire network.

2. Elastic compute :

With control and management residing on external servers, large scale computation can be done using the latest generation of servers. Thus the Compute capability of network devices may no longer a limitation in large data center networks.

3. Increased network reliability and security:

With a centralized control, SDN makes it possible for IT to define high-level configuration and policy statements, which are then translated down to the infrastructure via OpenFlow. This 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.

Because SDN controllers provide complete visibility and control over the network, they can ensure that security policies are enforced consistently across the wired and wireless network infrastructures, including branch offices, campuses, and data centers.

4. More granular network control:

The OpenFlow switch flow table is used to give network managers both coarse and fine-grained control over data flows. This control enables cloud operators to support multi-tenancy. Multi-tenancy logically divides a hardware resource so that multiple users can securely share the same physical device, such as a server. SDN lets administrators expand this concept to the network, so traffic flows within multiple groups of users can safely share network resources. Users gain greater access to available bandwidth, and increased network utilization reduces costs.

5. 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. For instance, a carrier could introduce a video service that offers premium subscribers the highest possible resolution in an automated and transparent manner instead of users explicitly selecting a resolution setting, which the network may or may not be able to support. With OpenFlow-based SDN, the video application would be able to detect the bandwidth available in the network in real time and automatically adjust the video resolution accordingly.

6. Unified view of the underlying network infrastructure :

With SDN administrators get a unified view of the network, simplifying configuration, management and provisioning. Failures whether it is link, node or otherwise are handled much faster. Furthermore, the systems converge more rapidly to target optimum and the behavior is predictable. Unified view of the network fabric provides a global view of the supply and demand of network resources. Managing end-to-end paths with this global view results in high utilization of network resources.

V. SUMMARY

Software-Defined Networking provides a new, dynamic network architecture that transforms traditional network backbones into rich service-delivery platforms. By decoupling the network control and data planes, OpenFlow-based SDN architecture abstracts the underlying infrastructure from the applications that use it, allowing the network behavior to be determined by business logic. SDN promises to make high-capacity networks cheaper to build and especially to re-configure on the fly - as well as potentially faster and more efficient. As cloud computing grows, those network improvements will be critical to keeping everything affordable and available. SDN could enable corporate networks to be reconfigured on-the-fly. With a centralized control, SDN can enable networks to adapt effortlessly to handle sudden new loads or changes in the network traffic patterns.

The future of networking will rely more and more on software, which will accelerate the pace of innovation for networks as it has in the computing and storage domains. Although much research is yet required , SDN promises to transform today’s static networks into flexible, programmable platforms with the intelligence to allocate resources dynamically, the scale to support enormous data centres and the virtualization needed to support dynamic, highly automated, and secure cloud environments. With its many advantages and a groundswell of industry support, we conclude that SDN is on the way to becoming the new norm for networks.

REFERENCES

[1] ONF whitepaper, ”Software Defined networking: A new norm for networks” ,April 12 ,2013

[2] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar et al. “ OpenFlow: Enabling Innovation in Campus Networks” , March 14, 2008

[3] HP,”Realizing the power of SDN with HP virtual application networks”, October 2012

[4] IBM, “Software defined networking”, October 2012

[5] TCAM- A deeper look and the impact of IPV6 ,website – http://etherealmind.com/tcam-detail-review

[6] Openvswitch, website – http://openvswitch.org

[7] Beacon,website- https://openflow.stanford.edu/display/Beacon/Home

[8] Floodlight , website –http://www.projectfloodlight.org