INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

A SURVEY ON ORCHESTRATION OF NETWORK USING SOFTWARE DEFINED NETWORKING

Raghuram .P[1] , G.Praveen babu[2]

[1]M.Tech , Computer Networks Information Security , School of IT, JNTU.

[2]Associate Professor, School of IT, JNTU.

[1]raghuram369@gmail.com; 2 pravbob@jntuh.ac.in

Abstract

Software defined networking; a new paradigm in networking industry has led to the stir up of the idea of programmable networks. Throughout the years it is discussed that network management can be simplified and innovation in networking can be achieved through network programmability. SDN, which is most often referred as a "radical new idea in networking", hits the right cord in this aspect. In this survey paper the application delivery networks which can be achieved using software defined networking is emphasized. We discuss the traditional networking environment and the ways SDN approach would solve the existing problems and also the advantages it would offer to various personnel working in the networking field. We discuss the OpenFlow architecture, standards and implementation on which the SDN is built on. Orchestration is the key in software defined networking applications, as it would drive the entire network automatically and fault free.

Keywords: Software-Defined Networking, programmable networks, survey, data plane, control plane, virtualization.

i. Introduction

Several networking equipment which includes routers, switches and devices like firewalls, load balancers etc. which manipulate the traffic flowing them for purposes other than just steering the packets together form the traditional computer networks. The computer networks apart from just physical equipment always have complex algorithms and protocols running on them. Network engineers are responsible for changing/configuring these policies as per the requirement, applications and network environment. Network engineers and architects receive the implementation and changes in the form of policies. Hence it is the work of these professionals to covert these high-level policy documents into the low level networking commands and configurations. Implementation of complex tasks has to be accomplished with the help of limited tools available to them.

Hence, the task of managing the networks and implementation of quality of service over networks is often error prone and also quite challenging. The fact that network devices are usually tightly coupled, vertically integrated black boxes exacerbates the challenge network administrators and engineers face.

The Internet is developing at a rate that outpaces all existing data networks in both speed and scale – except for perhaps the rate at which Internet service innovation is occurring. As the variety of real-time services continue to develop, including video and audio, cloud data center, and mobile services, bottlenecks can form that prevent traditional networks from delivering expected service quality. In this environment, traditional networks face three problems:

• Lack of user experience guarantees: Most IP networks are connectionless, providing only minimum bandwidth service quality. The lack of quality controls for delivered services results in situations that thwart user expectations for a quality experience and impacts customer relationships.

• Inefficient service deployment: In traditional networking, services and networks are deployed separately. Most networks are configured using commands or network management systems. These networks are essentially static and inefficient at deploying dynamic services that require timely adjustments. In extreme cases, these networks may even fail to support such services.

• Slow adaptation to new services: It may take years for traditional networks to upgrade features, adjust architectures, or introduce new devices to meet new service requirements. For example, the traditional Layer 2 VLAN mechanism of a cloud data center with virtual machines and virtual networks is required to run new bearer protocols on switches to meet scalability requirements; however, the physical devices involved cannot adapt to these requirements quickly enough. In this situation, software-defined virtual switches and Virtual Extensible Local Area Network/Network Virtualization Generic Routing Encapsulation (VXLAN/NVGRE) overlay networking can bypass the limitations of physical switches and still satisfy the scalability requirements of the network.

Over the years, many solutions have been proposed to solve the problems facing traditional networks. As is often the case, when one problem is solved, another one crops up. The nature of traditional networking determines the decoupling between networks and services (represented by IP and Ethernet services). Networks and services are transparent to each other. This decoupling mode enables rapid Internet service innovations, but creates a barrier between networks and services. The problems facing traditional networking cannot be solved without focusing on the fundamental design of network infrastructure.

Software Defined Networking (SDN) is changing the way we design and manage networks. SDN has two defining characteristics. First, an SDN separates the control plane (which decides how to handle the traffic) from the data plane (which forwards traffic according to decisions that the control plane makes). Second, an SDN consolidates the control plane, so that a single software control program controls multiple data-plane elements.

The SDN control plane exercises direct control over the state in the network's data-plane elements (i.e., routers, switches, and other middle boxes) via a well-defined Application Programming Interface (API). OpenFlow is a prominent example of such an API.

ii. Building blocks of SDN

Building blocks of SDN

Software-defined networking (SDN) is a modern approach to networking that eliminates the complex and static nature of legacy distributed network architectures through the use of a standards-based software abstraction between the network control plane and underlying data forwarding plane, including both physical and virtual devices. This standards-based data plane abstraction, called OpenFlow, provides a novel approach to dynamically provision the network fabric from centralized software based controller.

An open SDN platform with centralized software provisioning delivers dramatic improvements in your network agility via programmability and automation, while substantially reducing the cost of your network operations. And using an industry standard data plane abstraction protocol like OpenFlow, you are now free to use any type and brand of data plane devices, since all the underlying network hardware is addressable through a common abstraction protocol. Importantly, OpenFlow facilitates the use of "bare metal switches" and eliminates traditional vendor lock-in, giving you freedom of choice in networking like you have in other areas of your IT infrastructure, such as servers. SDN controllers also expose APIs northbound, which allow you to deploy a wide range of off-the-shelf and custom-built network applications – many of which were fundamentally not feasible prior to the advent of SDN.
Compared to legacy networking architectures that are vertically integrated from the underlying hardware, through the operating system, and into the network application functionality, the inherently open nature of an SDN represents a fundamental shift in power giving you choice, agility and automation of network operations not previously possible.

III. SDN Architecture

Figure 1 illustrates the logical structure of an SDN. A central controller performs all complex functions, including routing, naming, policy declaration, and security checks. This plane constitutes the *SDN Control Plane*, and consists of one or more SDN servers.

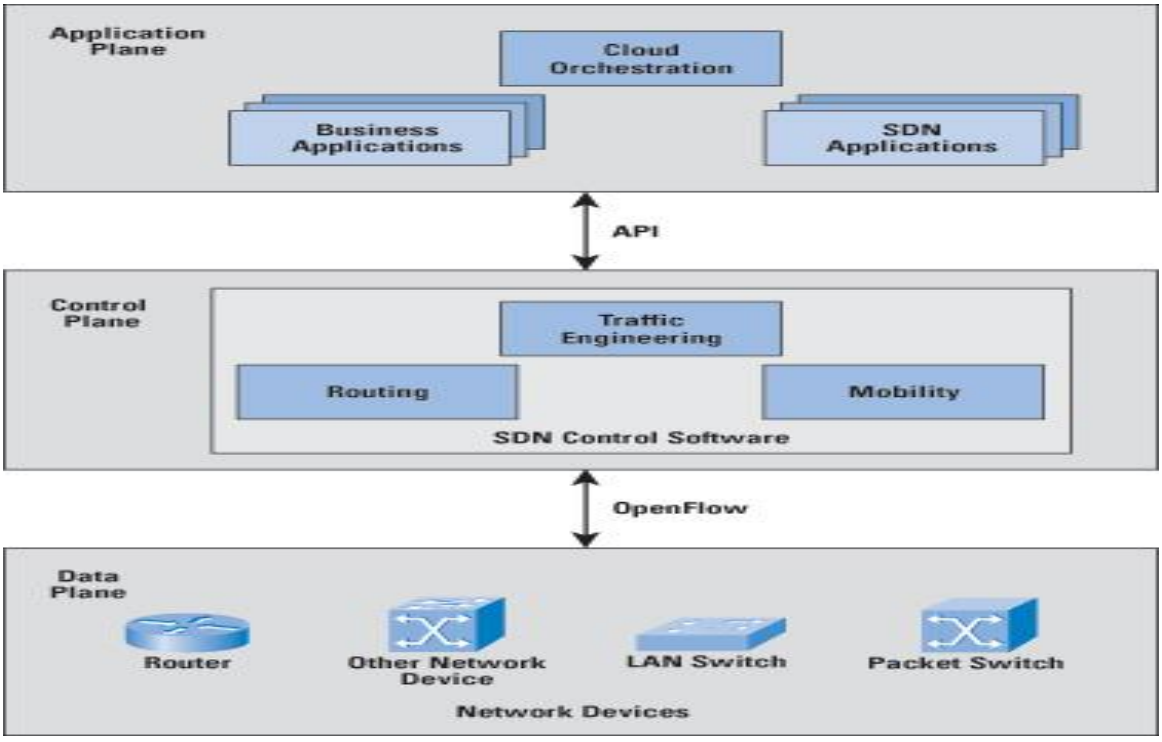

Fig .1 SDN Architecture

The *SDN Controller* defines the data flows that occur in the *SDN Data Plane*. Each flow through the network must first get permission from the controller, which verifies that the communication is permissible by the network policy. If the controller allows a flow, it computes a route for the flow to take, and adds an entry for that flow in each of the switches along the path. With all complex functions subsumed by the controller, switches simply manage flow tables whose entries can be populated only by the controller. Communication between the controller and the switches uses a standardized protocol and API. Most commonly this interface is the OpenFlow specification, discussed subsequently.

The SDN architecture is remarkably flexible; it can operate with different types of switches and at different protocol layers. SDN controllers and switches can be implemented for Ethernet switches (Layer 2), Internet routers (Layer 3), transport (Layer 4) switching, or application layer switching and routing. SDN relies on the common functions found on networking devices, which essentially involve forwarding packets based on some form of flow definition.

In an SDN architecture, a switch performs the following functions:

1. The switch encapsulates and forwards the first packet of a flow to an SDN controller, enabling the controller to decide whether the flow should be added to the switch flow table.
2. The switch forwards incoming packets out the appropriate port based on the flow table. The flow table may include priority information dictated by the controller.
3. The switch can drop packets on a particular flow, temporarily or permanently, as dictated by the controller. Packet dropping can be used for security purposes, curbing Denial-of-Service (DoS) attacks or traffic management requirements.

In simple terms, the SDN controller manages the forwarding state of the switches in the SDN. This management is done through a vendor-neutral API that allows the controller to address a wide variety of operator requirements without changing any of the lower-level aspects of the network, including topology.

With the decoupling of the control and data planes, SDN enables applications to deal with a single abstracted network device without concern for the details of how the device operates. Network applications see a single API to the controller. Thus it is possible to quickly create and deploy new applications to orchestrate network traffic flow to meet specific enterprise requirements for performance or security.

IV. Openflow protocol architecture

OpenFlow is an open standard that enables researchers to run experimental protocols in the campus networks we use every day. OpenFlow is added as a feature to commercial Ethernet switches, routers and wireless access points – and provides a standardized hook to allow researchers to run experiments, without requiring vendors to expose the internal workings of their network devices. OpenFlow is currently being implemented by major vendors, with OpenFlow-enabled switches now commercially available.

In a classical router or switch, the fast packet forwarding (data path) and the high level routing decisions (control path) occur on the same device. An OpenFlow Switch separates these two functions. The data path portion still resides on the switch, while high-level routing decisions are moved to a separate controller, typically a standard server. The OpenFlow Switch and Controller communicate via the OpenFlow protocol, which defines messages, such as packet-received, send-packet-out, modify-forwarding-table, and get-stats. The data path of an OpenFlow Switch presents a clean flow table abstraction; each flow table entry contains a set of packet fields to match, and an action (such as send-out-port, modify-field, or drop). When an OpenFlow Switch receives a packet it has never seen before, for which it has no matching flow entries, it sends this packet to the controller. The controller then makes a

decision on how to handle this packet. It can drop the packet, or it can add a flow entry directing the switch on how to forward similar packets in the future.

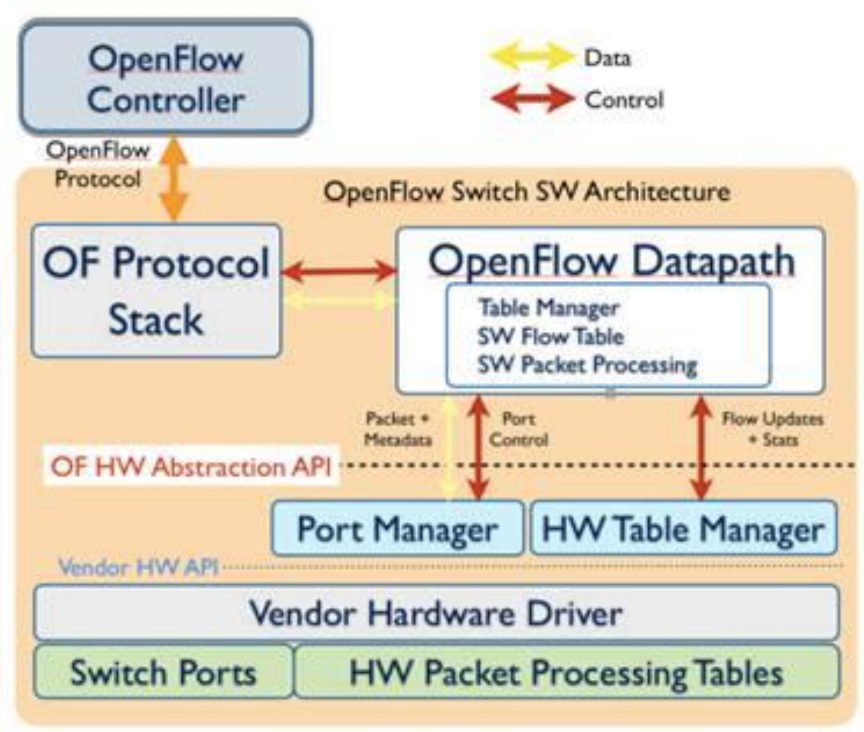

Fig. 2 Openflow protocol

V. Benefits of Software Defined Networking

The concept of the software-defined network (SDN) initially caused some confusion in IT, but the picture gets clearer the more we learn. Similar to network virtualization, SDN allows for the direct network-related abstraction of services. This abstraction of services can be accomplished on both logical as well as physical infrastructures, but is not actually defined by specific physical devices or logical components.

1. Complete cloud abstraction

The cloud model of computing is here to stay. But, like all technologies, the cloud is evolving. Driven by both the enterprise and the end-user, cloud computing is becoming a more unified infrastructure. Users are able to access content that spans numerous environments via hybrid platforms that extend the datacenter infrastructure out to them.

2. Intelligent global connections

SDN can create very intelligent and globally connected environments. It can also help with load-balancing cloud and datacenter infrastructures. SDN already provides global traffic management by sending traffic to appropriate datacenters based on network logic. Moving forward, SDN will al-low architects to create even more fluid automation for datacenter traffic flow. Efforts like this of will help reduce downtime, increase data resiliency, and make disaster recovery planning more effective.

3. Near-flawless content delivery

Users are demanding more content to an ever increasing number of devices. Furthermore, they are asking for this content in high definition. Organizations like Netflix have achieved success building intelligent edge networks that cache rich content closer to the user. Advancements in streaming technologies also allow for rich media to be transmitted live via the cloud. By implementing SDN, network operators will be able to increase network responsiveness, providing a near-flawless experience for the user.

VI. Conclusion

SDN may sound abstract, but the technology is on target to provide concrete advantages to network infrastructure from increased global connectivity to better content delivery.

The concept of the software-defined network (SDN) initially caused some confusion in IT, but the mere outstanding advantages it offers are really placing it amongst the top networking technologies Similar to network virtualization, SDN allows for the direct network-related abstraction of services. This abstraction of services can be accomplished on both logical as well as physical infrastructures, but is not actually defined by specific physical devices or logical components. Software-defined technologies are already creating new ways to abstract resources at the enterprise level, and products are coming onto the market that allows IT departments to benefit from SDN. SDN orchestration techniques discussed above leverage the power of programmable networks to facilitate all the networking functionalities at the fingertips of networking administrators and architects.

REFERENCES

[1] http://tools.ietf.org/html/rfc7149

[2] https://www.cs.uaf.edu/2011/spring/cs641/proj1/dpkline/

[3] http://h20195.www2.hp.com/v2/GetPDF.aspx/4AA3-8562ENW.pdf

[4] http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_16-1/161_sdn.html

[5] http://pronetworkingh17007.external.hp.com/nz/en/solutions/technology/openflow/index.aspx