SDN – Networking Of The Future

Vijayakumar R H
M.Tech (Computer Network Engineering),
Department of Computer Science and Engineering,
New Horizon College of Engineering, Bangalore
vijaykrh@gmail.com

Beena B M,
Senior Assistant Professor,
Department of Computer Science and Engineering,
New Horizon College of Engineering, Bangalore
beena.nh@gmail.com

***Abstract*—High bandwidth in information and communication technologies are commanding new challenges to future network. The management of these networks has also become difficult. Traditional approaches for manual configuration of devices are cumbersome and error-prone. The emergence of software defined networking has become one of the most promising solutions for future network. SDN offers centralized control, reduced complexity and decrease in capital and operational costs. In this paper we provide an overview of SDN, architecture underlying SDN, and also explore SDN with an emphasis on benefits, future trends and challenges that must be overcome to move forward.**

I. INTRODUCTION

According to open networking foundation (ONF) [1], SDN is defined as the physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices. A key element in SDN is the introduction of an abstraction between the (traditional) forwarding and control planes in order to separate them and provide applications with the means necessary to programmatically control the network.

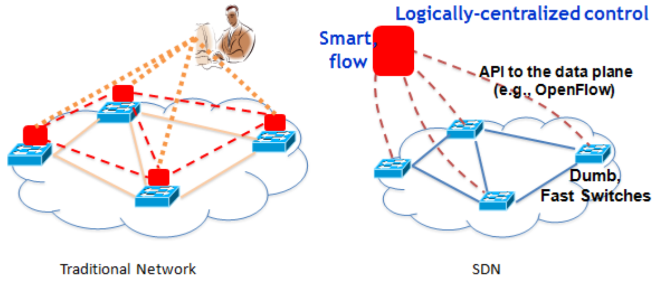
Logically-centralized control
Smart flow
API to the data plane (e.g., OpenFlow)
Dumb, Fast Switches
Traditional Network
SDN

Figure 1: Traditional and SDN style of networking

Software-Defined Networking (SDN) is an emerging architecture that is dynamic, manageable, cost-effective, and adaptable, making it ideal for the high-bandwidth, dynamic nature of today's applications. This architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. The OpenFlow® protocol is a foundational element for building SDN solutions

According to IETF's RFC7426 [2] Software-Defined Networking (SDN) refers to a new approach for network programmability, that is, the capacity to initialize, control, change, and manage network behaviour dynamically via open interfaces.

II. SDN LAYERS AND ARCHITECTURE

According to ONF, SDN consists of three layers, Application Layer, Control Layer and Infrastructure Layer.

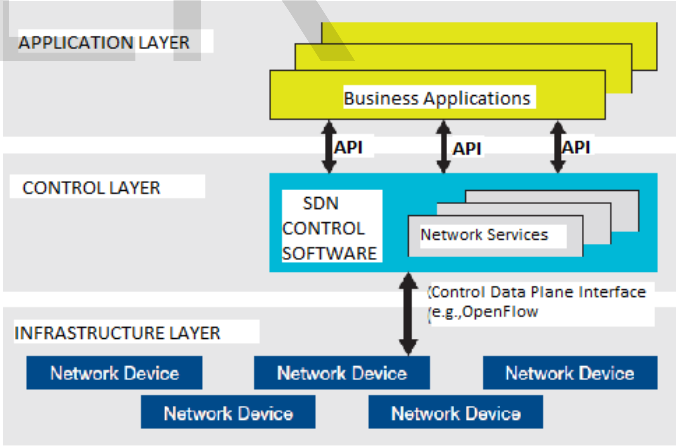

Figure 2: SDN Layer Architecture

The SDN architecture is [6] [7]:

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

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

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

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

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

According to RFC 7426, SDN Layers and Architecture is as shown below. The different planes in the architecture is defined as follows

Forwarding Plane (FP) - The collection of resources across all network devices responsible for forwarding traffic.

Operational Plane (OP) - The collection of resources responsible for managing the overall operation of individual network devices.

Control Plane (CP) - The collection of functions responsible for controlling one or more network devices. CP instructs network devices with respect to how to process and forward packets. The control plane interacts primarily with the forwarding plane and, to a lesser extent, with the operational plane.

Management Plane (MP) - The collection of functions responsible for monitoring, configuring, and maintaining one or more network devices or parts of network devices. The management plane is mostly related to the operational plane (it is related less to the forwarding plane).

Application Plane (AP) - The collection of applications and services that program network behaviour.

The detailed SDN Layer Architecture as per RFC 7426 is as shown below.

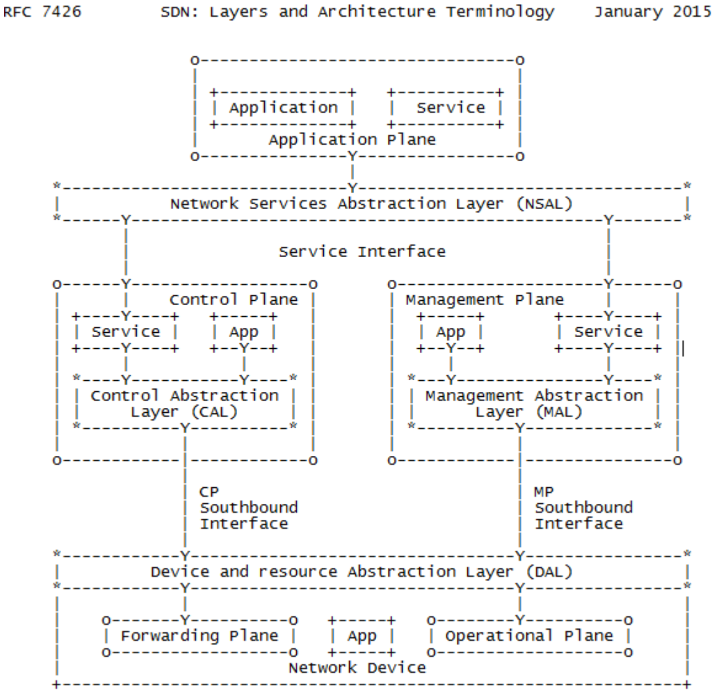
RFC 7426 SDN: Layers and Architecture Terminology January 2015
Application
Service
Application Plane
Network Services Abstraction Layer (NSAL)
Service Interface
Control Plane
Service
App
Control Abstraction Layer (CAL)
Management Plane
App
Service
Management Abstraction Layer (MAL)
CP Southbound Interface
MP Southbound Interface
Device and resource Abstraction Layer (DAL)
Forwarding Plane
App
Operational Plane
Network Device

Figure 3: Detailed SDN Layer Architecture as per RFC7426

Device and resource Abstraction Layer (DAL) - The device's resource abstraction layer based on one or more models. If it is a physical device, it may be referred to as the Hardware Abstraction Layer (HAL). DAL provides a uniform point of reference for the device's forwarding- and operational-plane resources.

Control Abstraction Layer (CAL) - The control plane's abstraction layer. CAL provides access to the Control-Plane Southbound Interface.

Management Abstraction Layer (MAL) - The management plane's abstraction layer. MAL provides access to the Management-Plane Southbound Interface.

Network Services Abstraction Layer (NSAL) - Provides service abstractions that can be used by applications and services.

Some of the key concepts that are part of the SDN system architecture are

Business Applications: This refers to applications that are directly consumable by end users. Possibilities include video conferencing, supply chain management and customer relationship management.

Network & Security Services: This refers to functionality that enables business applications to perform efficiently and securely

Pure SDN Switch: In a pure SDN switch, all of the control functions of a traditional switch (i.e., routing protocols that are used to build forwarding information bases) are run in the central controller. The functionality in the switch is restricted entirely to the data plane.

Hybrid Switch: In a hybrid switch, SDN technologies and traditional switching protocols run simultaneously. A network manager can configure the SDN controller to discover and control certain traffic flows while traditional, distributed networking protocols continue to direct the rest of the traffic on the network.

Hybrid Network: A hybrid network is a network in which traditional switches and SDN switches, whether they are pure SDN switches or hybrid switches, operate in the same environment.

Northbound API: The northbound API is the API that enables communications between the control layer and the business application layer. There is currently not a standards-based northbound API.

Southbound API: The southbound API is the API that enables communications between the control layer and the infrastructure layer. Protocols that can enable this communications include OpenFlow, the extensible messaging and presence protocol (XMPP) and the network configuration protocol.

III. SDN BENEFITS

The SDN flexible deployment models provides several advantages

Applications can be deployed on any network node in any location that needs services without requiring reachability between the node and a central controller.

Distributed application deployment facilitates scalability and minimizes control-plane latency.

Network devices can more swiftly adjust service policies in response to fluctuating flow demands when distributed device control planes are closely coupled with applications

The key benefits are

Unified view of the network fabric: Provides unified view of the network, simplifying configuration, management and provisioning.

High utilization: Centralized traffic engineering 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 the links.

Faster failure handling: Failures whether it is link, node or otherwise are handled much faster.

Faster time to market/deployment: With SDN, better and more rigorous testing is done ahead of rollout accelerating deployment. The development is also expedited as only the features needed are developed.

Hitless upgrades: The decoupling of the control plane from the forwarding/data plane enables to perform hitless software upgrades without packet loss or capacity degradation.

The key drivers for deploying SDN [11] are

- Better utilization of network resources
- Automate provisioning and management
- Improved network security and reliability
- Implementing network wide policies
- Lower capital and operational costs (CAPEX and OPEX)
- Supporting the dynamic movement, replication and allocation of virtual resources.
- Easing the administrative burden of configuration and provisioning
- Perform traffic engineering with an end-to-end view of the network
- More easily scale network functionality
- Lead to faster introduction of new capabilities into the network

The key to truly reduce enterprise network OPEX, is through simplifying network management [10]. The tasks include

- Adding new nodes to the network
- Detecting and replacing failed links and failed units
- Investigating network slow-downs
- Maintaining security
- Resolving users' connectivity issues
- Cha+nging switch configurations.
- Upgrading switch software.

IV. SDN AND OPENFLOW

OpenFlow is the first standard communications interface defined between the control and forwarding layers of an SDN architecture. OpenFlow allows direct access to and manipulation of the forwarding plane of network devices such as switches and routers, both physical and virtual (hypervisor-based). It is the absence of an open interface to the forwarding plane that has led to the characterization of today's networking devices as monolithic, closed, and mainframe-like. No other standard protocol does what OpenFlow does, and a protocol like OpenFlow is needed to move network control out of the networking switches to logically centralized control software.

The OpenFlow protocol is implemented on both sides of the interface between network infrastructure devices and the SDN control software. OpenFlow uses the concept of flows to identify network traffic based on pre-defined match rules that can be statically or dynamically programmed by the SDN control software. It also allows IT to define how traffic should flow through network devices based on parameters such as usage patterns, applications, and cloud resources. Since OpenFlow allows the network to be programmed on a per-flow basis, an OpenFlow-based SDN architecture provides extremely granular control, enabling the network to respond to real-time changes at the application, user, and session levels. Current IP-based routing does not provide this level of control, as all flows between two endpoints must follow the same path through the network, regardless of their different requirements.

The OpenFlow protocol is a key enabler for software-defined networks and currently is the only standardized SDN protocol that allows direct manipulation of the forwarding plane of network devices. While initially applied to Ethernet-based networks, OpenFlow switching can extend to a much broader set of use cases. OpenFlow-based SDNs can be deployed on existing networks, both physical and virtual. Network devices can support OpenFlow-based forwarding as well as traditional forwarding, which makes it very easy for enterprises and carriers to progressively introduce OpenFlow-based SDN technologies, even in multi-vendor network environments.

OpenFlow is designed to support policy-based flow management within a network. OpenFlow is particularly well suited to use cases satisfied by pushing predefined policies to implement network segmentation. In addition to simple flowmatching and forwarding capabilities, later releases of the OpenFlow specification have introduced ways to implement simple quality of service (QoS) and flow metering.

Benefits of OpenFlow-Based Software-Defined Networks

OpenFlow-based SDN is currently being rolled out in a variety of networking devices and software, delivering substantial benefits to both enterprises and carriers, including

- Centralized management and control of networking devices from multiple vendors;

- Improved automation and management by using common APIs to abstract the underlying networking details from the orchestration and provisioning systems and applications;

- Rapid innovation through the ability to deliver new network capabilities and services without the need to configure individual devices or wait for vendor releases;

- Programmability by operators, enterprises, independent software vendors, and users (not just equipment manufacturers) using common programming environments, which gives all parties new opportunities to drive revenue and differentiation;

- Increased network reliability and security as a result of centralized and automated management of network devices, uniform policy enforcement, and fewer configuration errors;

- More granular network control with the ability to apply comprehensive and wide-ranging policies at the session, user, device, and application levels; and

- Better end-user experience as applications exploit centralized network state information to seamlessly adapt network behaviour to user needs.

V. ISSUES IN SDN

Network service providers and operators are facing increased network complexity, cost of deployments and variety of service offerings but declining revenues. SDN is attractive in terms of network simplification and reduction of the capital and operational expenditure. A number of implementation issues that need to be resolved to make large-scale deployment of SDN a reality.

Performance

The performance of SDN has to be ensured in terms of throughput and latency to commensurate with the traditional networks. These networks should be flexible enough so that new features and capabilities can be added. Introduction of new protocols, applications and security features are examples of the changes that needs to be supported without affecting performance.

Scalability

The question of scalability of the controller arises in large scale deployments. Operators fear increased latency with increased number of network nodes. The issue of inter-controller communication needs to be resolved if the number of controllers is increased.

Interoperability

Transition to SDN requires coexistence of SDN with the existing legacy infrastructure. Hybrid SDN can be used in which SDN enabled and traditional nodes inter-operate. Such interoperability requires the support of an appropriate protocol that provides backward compatibility with existing IP protocol.

Security

A greater focus on security is required if SDN is going to be accepted widely. At the controller-application level, authentication and authorization mechanisms needs to be addressed that would allow protection of resources of multiple organizations accessing the network.

VI. SDN TRENDS

The general familiarity with SDN has increased significantly over the last couple of years. The percentage of IT organizations that have SDN in production will likely increase in the coming days. Most IT professionals regard overlay network virtualization solution as being a form of SDN. The Open Networking Foundation (ONF) established the Northbound Interface (NBI) working group with the goal of eventually standardizing SDN's northbound interface. A number of vendors have announced their intention to use the OpenDaylight solution as the basis of their SDN controller. This creates the potential for SDN solution based on OpenDaylight solutions to reach critical mass and hence accelerate the adoption of SDN. The wide range of factors that were driving the interest for adoption of SDN include better utilization of network resources and to perform traffic engineering with an end-to-end view of the network. SDN also helps to reduce OPEX and CAPEX costs and also to reduce complexity.

Over the next couple of years the primary focus of SDN deployment will likely to be in the data centre. SDN can also be deployed in the WAN and in campus network.

SDN is moving towards next level of standardization and various companies are bringing out their products in the market. The success of the open-source controller platforms like Opendaylight, with well-defined North-bound APIs, has fuelled the trend of controlling SDN networks smoothly through applications. These applications are developed targeting various domains and network services.

Trends such as user mobility, server virtualization, IT-as-a-Service, and the need to rapidly respond to changing business conditions place significant demands on the network—demands that today's conventional network architectures can't handle. 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 to become as programmable and manageable at scale as the computer infrastructure that it increasingly resembles. An SDN approach fosters network virtualization, enabling IT staff to manage their servers, applications, storage, and networks with a common approach and tool set. Whether in a carrier environment or enterprise data center and campus, SDN adoption can improve network manageability, scalability, and agility. The Open Networking Foundation has fostered a vibrant ecosystem around SDN that spans infrastructure vendors large and small, including application developers, software companies, systems and semiconductor manufacturers, and computer companies, plus various kinds of end users. OpenFlow switching is already being incorporated into a number of infrastructure designs, both physical and virtual, as well as SDN controller software. Network services and business applications already interface with SDN controllers, providing better integration and coordination between them.

VII. SDN CHALLENGES

The inhibition to SDN adoption includes the immaturity of current products and also of enabling technologies. Some of the key inhibitors, such as the lack of compelling business case, need to be addressed or they will continue to impede SDN adoption. Another significant inhibitor to SDN deployment is security vulnerabilities. One of the ways that SDN can enhance security is by implementing security services on OpenFlow based access switches that can filter packets as they enter the network. Another example is role based access that can be implemented by deploying a role-based resource allocation that leverages the control information and capability of the SDN controller.

Some of the security challenges [3] are

- The centralized controller emerges as a potential single point of attack and failure that must be protected from threats.
- The southbound interface between the controller and underlying network devices is vulnerable to threats that could degrade the availability, performance and integrity of the network.

Although SDN deployment can ease the administrative burden of management tasks such as configuration and provisioning while some network organizations have concerns that to manage SDN can become a significant inhibitor to SDN deployment.

In SDN environment the challenges associated with end-to-end service performance management are more demanding than they are in traditional network environments.

CONCLUSION

Software Defined Networking (SDN) has become one of the hottest topics in the industry, and for a good reason. SDN is about separating the control plane from the data plane. As a result, SDN disaggregates proprietary closed boxes into the SDN control plane, the data plane and the applications and services. The SDN control plane behaves like the brain and uses appropriate abstractions, APIs and protocols to support a diversity of applications to control, configure and manage the network.SDN gives users centralized control, greater visibility, and greater choice because they can mix and match technologies at different layers. It enables greater innovation because each layer can evolve at its own pace. The end user result is new services and lower costs, which is the real promise of SDN.

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. 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 astonishing industry momentum, SDN is on the way to becoming the new norm for networks.

REFERENCES

[1] Open Networking Foundation web site: https://www.opennetworking.org

[2] Internet Engineering Task Force (IETF) : https://tools.ietf.org/html/rfc7426

[3] The 2015 Guide to SDN, Executive Summary: https://www.cisco.com

[4] Practical Implementation of SDN and NFV in the WAN: www.heavyreading.com

[5] SDN Performance – Raising the bar on SDN control plane performance, scalability and high availability, © 2015 ONOS

[6] An introduction to Software Defined Networking – www.citrix.com

[7] Software-Defined Networking: The New Norm for Networks – ONF WHITE PAPER

[8] Inter-Datacenter WAN with centralized TE using SDN and OpenFlow -© 2012 Google Inc

[9] Software Defined Networking – Why we like it and how we are building on it – www.cisco.com

[10] Demystifying Software-Defined Networking, Allied Telesis – www.alliedtelesis.com

[11] How SDN interest has evolved, Jim Metzlet – www.networkcomputing.com