Magic of SDN in Networking

Author: Santha Rami Reddy
<table>
<thead>
<tr>
<th>Contents</th>
<th>P.no</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executive Summary</td>
<td>1</td>
</tr>
<tr>
<td>Why SDN?</td>
<td>2</td>
</tr>
<tr>
<td>Limitations with existing Network Devices for Emerging Network needs</td>
<td>3</td>
</tr>
<tr>
<td>SDN as a Framework</td>
<td>4</td>
</tr>
<tr>
<td>Open Flow as Technology</td>
<td>6</td>
</tr>
<tr>
<td>Applications of SDN and Opportunities</td>
<td>9</td>
</tr>
<tr>
<td>Conclusion</td>
<td>11</td>
</tr>
<tr>
<td>About Calsoft Labs</td>
<td>11</td>
</tr>
</tbody>
</table>
Executive Summary

Software Defined Networking (SDN) brings a transformational paradigm shift from traditional vendor-locked networking to vendor-independent networking, offering greater flexibility and significant reduction in Capex and Opex to network operators. SDN recommends split architecture by separating the device functionality into control plane and data plane, allowing them to operate independently. It aggregates the distributed intelligence spread across in different network elements of a large network into a centralized Control layer that helps to improve the efficiency of networks and optimum utilization of resources. SDN is flexible in its design to operate, in co-existence with legacy networking devices and the new genre of low cost switching infrastructure with tremendous advantage of programmability, thereby ensuring competitive benefits to Service Providers, Data Centres and Enterprises with respect to Scalability, Performance, and High Availability of services. Through its well defined split architecture, current device functionality is divided into layered architecture, viz. Infrastructure Layer, Control Layer and Application Layer.

Software Defined Networking (SDN) Architecture

Each of the layers can work independently and communicate over the network interface to deliver the layered functionality from different physical devices. This aspect of the architecture allows network administrators to address some of the pressing challenges in live networks.
Why SDN?

Infrastructure Advancements
Every 6 to 7 years, advancements in silicon and availability of processors with higher computational capabilities at lower costs make the existing infrastructure obsolete. Legacy hardware is unable to handle the Packets Per Second (PPS) and Throughput needed to meet the ever increasing bandwidth requirements, compared to new hardware. At present, replacing the entire box is the only option for network operators. Advancements at this pace do not allow Service Providers and large Enterprises to recover their costs (Capex and Opex). With SDN coming in to existence, network operators could rely on low cost and server class commodity hardware to build networks, and replace the hardware when new silicon becomes available. Software components will be designed to accommodate the changes in infrastructure layer from time to time without significant changes to the entire system and configuration. Most equipment vendors now offer Virtual Appliances that run in the control plane, to support dynamism in the entire ecosystem.

Scalability
With the switching infrastructure based on forwarding hardware, networks can be scaled linearly. SDN allows network administrators to scale the network up or down by adding or removing forwarding devices attached to the SDN controller. The SDN framework supports this architecturally, and virtualized software components with a suitable orchestration layer enables seamless scaling within the network topology. Each SDN controller is capable of supporting thousands of forwarding devices in the switching infrastructure.

Innovation and adaptation of new Network services
Advancements in Hardware, Controller-Network Services and Applications, independently or combined, open up new possibilities to support several Value Added Services (VAS). The VAS applications can be easily developed and deployed with support for programmability inherent in SDN. Else with legacy network architectures invention and innovation is either not feasible, or a big time taking exercise as computing resources often become a bottleneck since the devices were never designed to run custom specific VAS applications.

Modular functionality of Software Applications and allowing new applications to run
Equipment vendors offering Virtual Appliances that can run on any standard hardware gives rise to modularity in Network and Business applications. Each of these applications communicate with other applications using physical or virtual network interface thus allowing the new applications to be deployed and offer the services to end customers in a systematic fashion. Continuous improvements happen at each of these SDN layers, with backward compatibility.
Limitations with existing Network Devices for Emerging Network needs

Vendor Locked
Over the decades networking products have achieved great stability and performance to meet specific needs of various users and operators. Devices such as routers and switches are compliant to industry standards to a certain level, and beyond that these products have been enhanced by different OEMs to support niche requirements in a non-standard way. This makes the network devices from few top vendors tightly coupled leading to vendor locked networks.

Protection of the vendor interests influence the continuous research and innovation to make better products at a lower cost. SDN helps this objective by reducing the complexity of devices in achieving the desired functionality and fostering innovation.

Limited scope for development of custom specific Network Services
Traditional networking devices are designed by the equipment vendor with limited computing resources to perform a set of intended functions. With such appliances it is not possible to open up the wiring closets to leverage the underlying platform to add new network services. As these devices are devised to be generic, varying needs of different customers cannot be accommodated.

Global view of resources and capacity utilization
With conventional network devices, the possibility of orchestration within the network is limited except when using devices from same vendor, where it is possible to some extent. In a heterogeneous environment getting a single view of the network for resource optimization is an uphill task. Additionally, Backup and Restore during failovers is a big challenge in a multi-vendor network. With SDN, it is possible to have the standard defined interfaces to the core control functionality and orchestration can be totally achieved among these heterogeneous devices.

Leveraging commodity Hardware for Switching infrastructure
With existing network devices that are tightly coupled between Control Plane and Data Plane, it is not possible to leverage the infrastructure advancements at an affordable cost. It requires replacement of the entire device. On the contrary, SDN makes it easy to upgrade the infrastructure and permits interworking with legacy devices for new deployments.

For the last few years, the communication industry has started using commoditized Chassis-Blade servers and Appliances. These Network Servers with routing and switching capabilities are being used as the new Network Infrastructure, permitting replacement of the hardware alone, and protecting capital investments.
SDN as a Framework

SDN Framework comprises of Data Plane (fast path) and Control Plane; further sub-dividing in to 3 major components in a layered approach.

1. SDN compliant Forwarding Hardware – Fast Path
2. SDN Controller – Control Plane (Middleware)
3. SDN Applications

Controller will establish the communication channel, with one or more SDN compliant switches to monitor and control the traffic forwarding. Flow Entries are formulated based on the inputs from protocols running as applications of control plane, and these entries are sent as instructions to the switching infrastructure through the established channel. Controller applications running the necessary protocol stacks would have the simulation of network topology and would run a virtual instance for each switch to come up with forwarding paths. Controller applications are shown as Virtual Appliances in the above diagram.

As shown in the picture, Controller is capable of communicating with OF compliant and other Vendor switches with device specific adaptors as plugin components. These specific adaptors would facilitate the communication as desired by the equipment vendors to enable SDN on their devices, so that Network Operators and Enterprises can orchestrate the resources in a heterogeneous network. This is one of the many potential benefits of SDN.
Multiple Controllers can co-exist in the network to control multiple SDN enabled switches forming a many to many relationship. Controllers can also be configured to operate as Master-Master and Master-Slave. Communication between controllers ensures information synchronization. Controllers can operate in distributed fashion to share the traffic load among switches and to ensure high-availability in fail over scenarios.

**Some of the many benefits those are possible with SDN**

- **Bring Your Own Network (BYON):** SDN through its distinguished and simple framework allows Enterprises and Network Operators to build custom specific networks with ease. Traditional device functionality in an appliance is mapped in to SDN layers and interfaces to the layers are provided to enable direct monitoring and control. This not only allows network administrators to design the networks but also to change according to the varying needs of the organization in due course.

- **Pay as You Grow:** SDN allows network administrators to dynamically increase and/or decrease the devices that are up and running depending on traffic flow statistics, latencies and drops; thus orchestrating the capacity serving needs within the network.

Hardware capacity can be scaled up seamlessly without changes to software components - "Pay as You Grow" approach. The SDN framework also allows SDN enabled devices to operate along with the existing equipment, thus reducing the need for heavy capital expenditure.

Similarly upgrades to the control plane software components for improvements and advancements in technology need not necessitate any change to the Switching Infrastructure. SDN also accomplishes global Load Balancing among devices that are spread across different geographical locations, by creating a larger pool of available resources and effective utilization of time zones.
Orchestration of Network Devices according to traffic patterns

SDN has the operational view of devices in a network and can monitor traffic routing periodically. This monitoring capability enables SDN to manipulate Port Status (Operational), Aggregation of links, and Forwarding Entries in Flow Table for dynamic adaptation of optimum alternative paths under network severe conditions, to best serve the SLAs.

Services Roll-Out and Revenues from Value Added Services

VAS roll-out on SDN platform is easier, as platforms remain constant with little or no change to add new VAS Applications; thus avoiding the entire system to undergo changes for launching/obsoleting VAS services.

Open Flow as Technology

Open Flow Protocol is a standard and one of the ways of achieving communication between controller and switching infrastructure in SDN framework. Open Flow is standardized through Open Networking Foundation (ONF) to achieve the objectives of Increased Network Functionality, while lowering operational expenses through simplified Hardware, Software and Management. ONF through its working groups responsible for the development of protocol, configuration and interoperability testing. Open Flow had evolved rapidly through several versions and we would refer to the latest standard version 1.3.1 in this whitepaper. Switches can operate in OpenFlow only mode or Hybrid mode (OpenFlow+Normal) as shown below by sharing the switch resources.

With Open Flow, the required match fields of a flow at a Networking device (Small, Medium and Large scale) are well defined. All the Open Flow parameters can be selectively applied for incoming traffic to match and execute respective instructions and actions. OpenFlow permits the Networking Hardware to be a Commodity item providing control hooks to Network Analysts to design and instruct the forwarding engine intelligently and dynamically.
OpenFlow capable switch will provide hooks to make flow entries for forwarding, Access Control lists for defining Firewall Policies, Meter Tables to support QOS Rate Limiting. Traffic that could not be forwarded through the flow entries can be selectively forwarded to controller for Network applications to process and take appropriate decisions. These decisions are converted to Flow Entries and instructions are given to switch to add/modify flow tables/entries for the subsequent handling at switch.

Every possible functional aspect of the current Networking device, which is based on following fields, can be met through SDN.

<table>
<thead>
<tr>
<th>Ingress Port</th>
<th>ETHERNET</th>
<th>IP LAYER</th>
<th>TCP</th>
<th>UDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA</td>
<td>DA</td>
<td>ETHER TYPE</td>
<td>IPV4</td>
<td>IPV6</td>
</tr>
<tr>
<td>Source Port</td>
<td>Destination Port</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**OpenFlow Packet Match Fields**

Along with above match fields for a flow, many additional optionally specifiable match fields are defined to match a flow. For the matched flows, defined instructions and actions will be executed along with updates to counters. Counter updates can be optionally linked to the Meter Table entries, which contain specific actions for defined traffic metering thresholds. Traffic Thresholds are defined using Meter Bands and the actions such as DROP and DSCP Re-Marking would be applied for traffic on satisfying the Meter Band criteria.

**OpenFlow Controller and Switch**

OpenFlow switch consists of Flow Tables, Group Table and a Metering table to perform packet lookups and forwarding. OpenFlow channel, part of this standard is for the communication between the controller and the OpenFlow switch.
Using OpenFlow protocol, Controller can add, update, and delete Flow entries in to Flow tables. Each flow table in switch contains a set of flow entries; each flow entry consists of match fields, counters, and a set of instructions to apply to matching packets. OpenFlow match fields can be selectively applied for incoming traffic to match and execute the respective instructions.

**Pipeline Processing**

Matching starts at first flow table and may continue to additional flow tables. Flow entries match packets in priority order, with the first matching entry in each table being used. If a matching entry is found, the instructions associated with the specific flow entry are executed. If no match is found in a flow table, the outcome depends on configuration of the table-miss flow entry of the respective table; table-miss flow entry can forward the packets to controller over OpenFlow channel or drop or may continue with finding a match in next flow table.

Pipeline processing instructions allow packets to be sent to cascaded tables for further processing and pass information, in the form of metadata, to be communicated between tables. Table pipeline processing stops when the instruction set associated with a matching flow entry does not specify a next table; at this point the packet is modified according to action set and forwarded. Instructions associated with each flow entry either contain actions or modify pipeline processing. Actions included in instructions describe packet forwarding, packet modification and group table processing.

**Group Table**

The group table contains group entries; each group entry contains a list of action buckets with specific semantics dependent on group type. The actions in one or more action buckets are applied to packets sent to the respective group-id. Actions associated with flow entries may direct packets to a group entry, which specifies additional processing. This action summarization allows common output actions across flow entries to be managed efficiently.

**Meter Table**

A Meter Table consists of meter entries, defining per-flow meters. Per-flow meters enable OpenFlow to implement various simple QoS operations, such as rate-limiting, and can be combined with per-port queues to implement complex QoS frameworks, such as DiffServ. A meter measures the rate of packets assigned to it and enables controlling the rate of those packets. Meters are attached directly to flow entries (as opposed to queues which are attached to ports). Any flow entry can specify a meter in its instruction-set, the meter measures aggregate of all flow entries to which it is attached. Multiple meters can be used in the same table, but in an exclusive way (disjoint set of flow entries). Multiple meters can be used on the same set of packets by using them in successive flow tables.

<table>
<thead>
<tr>
<th>Meter Identifier</th>
<th>Meter Bands</th>
<th>Counters</th>
</tr>
</thead>
</table>

**Main components of a meter entry in the meter**
**Meter Bands**

Each meter entry can have one or more meter bands. Each band specifies the rate at which the band applies and the way packets should be processed. Packets are processed by a single meter band based on the current measured meter rate, meter applies meter band with the highest configured rate that is lower than the current measured rate. If the current rate is lower than 'any' specified meter band rate, no meter band is applied.

**IPV4 and IPV6 support**

Internet Protocol versions of V4 and V6 are totally supported under the category of essential match fields and desired optional fields extensively. All with the essential match fields of IPv4 and IPv6 fields such as Flow Label, ICMPV6 Type & Code, Neighbour Discovery fields (Target, Source Link Layer and Destination Link Layer addresses) and Extension Header for IPV6 are supported as match fields in flow entries to match specific packets.

With these Data Plane and Middleware capabilities, Network Applications plays a major role in providing the control information. Layer-2, Layer-3 control protocols would run as Controller Applications and make the converged flow entries in to data path through controller. Along with essential control protocols, niche applications will be developed to customize network behaviour. Applications scope will be of the entire network in an autonomous system, to measure performance, optimize resource availability and ensuring greater scalability and high availability.

**Applications of SDN**

- New Generation of Layer-2 and Layer-3 Switches and Routers – Network devices that are Scalable, Fault Tolerant with High Availability, Performance on par with wiring closets along with investment protection against technological advancements.

  Scalability: SDN Switches and Controllers would communicate through either a standard defined protocol (Open Flow) or a Proprietary interface protocol where some of the equipment vendors are complying to. Thus providing flexibility to Network Operator or Administrator to orchestrate network resources and extend to several switches from a wide choice, leveraging Pay as you Grow. In a Network of SDN compliant switches, controller running on a large server can manage up to few thousands of switches. However on hitting the controller's maximum limit of connected switches, network can be extended with multiple controllers sharing the switches forming a cluster. It is possible for multiple controllers to share a single switch, with switch virtualization.
High Availability (HA): HA through N+1 redundancy can be achieved among the controllers and switches in a network, to ensure the Physical and Operational Redundancy. SDN allows forming a cluster of devices assigned with designated responsibilities through configuration and/or automatically by allowing the respective protocols to elect.

- Layer-4 – Layer-7 Switching with DPI – Content aware switching and Load Balancing among controllers and application servers with global view of health status of the application servers.

- Dynamic Policy Based Routing and QOS – With the availability of Device Statistics at continuous intervals for all the switches in the SDN framework, Network parameters such as Latencies and Congestion can be monitored and corrective actions can be initiated through dynamic changes to routing policies. Meter table containing meter bands would allow specifying the actions on meeting upper thresholds. In a situation of network congestion, Meter band actions can trigger events to controller; respective applications at controller can process this data and initiate flow modification messages to switches in SDN framework to ensure smooth flow of traffic.

- Orchestration of Network Devices in a Heterogeneous Environment with varying device capabilities under the varying traffic conditions

- Data Centre Load Balancing and Power Saving Applications - With the significant view of the network elements and the respective capabilities, applications running on the controllers will be able to off-load the traffic forwarding capabilities to the devices across the global pool of resources. This act can happen dynamically at a consistency, achieving the Global Load Balancing.

- Development of SDN Applications running above the control plane for various requirements. These applications can be new or built around the existing applications on virtualized or non-virtualized resources.

- Existing NMS systems need to be modified to support the SDN framework. As the device functionality is split into layers, NMS architecture has to be modified to deal with this. There are multiple ways to support NMS in SDN framework, the most preferred can be chosen depending on the distribution of functionality.
Conclusion

SDN as a Networking Technology is poised to meet the demands of Operators and Enterprises with tremendous improvement in cost, flexibility and maintenance. OpenFlow as a technology is rapidly evolving and has reached a state of maturity, where major vendors are offering competitive platforms and solutions. With promising benefits from SDN as a framework and OpenFlow as technology, every Network operator and SME establishment must align the network requirements with technological advantages SDN is bringing forth. SDN provides tremendous opportunities for Server Platform Vendors, Network Operating System developers, Independent Software Vendors (Network and/or Business applications) to meet the desires of Cloud Operators, Data Centres and Enterprises.

About Calsoft Labs

Calsoft Labs provides specialized concept to market Product Engineering services to product and technology companies in select market segments. Our target markets include Automotive, Consumer Electronics, Media, Networking, Storage and Independent Software Vendors (ISVs). Calsoft Labs delivers unmatched business value to its customers through a combination of process excellence, reusable frameworks and technology innovation.

Calsoft Labs is a wholly owned subsidiary of ALTEN. Set up in 1988, ALTEN is a European leader in Engineering and Technology Consulting (ETC) with 15,950 employees in over 16 countries worldwide.