4.1 WHY SDN?

Software defined network is the next generation innovation in network architecture where the data and control plane are separated, moving the control plane to a centralized controller. The controller is called the “Network Operating System” and has a complete view of the network topology and full control of the network resources. The devices such as routers and switches in the data plane acts only as forwarding device.

In conventional IP network, the hardware and software are bought together from a vendor and configured as per the requirement. Any additional features to be added or to fix any software bug the customer has to rely on the vendor. Vendor softwares are proprietary in nature and this has led to slower innovation and deployment of new service. SDN eliminates the vendor dependence [49]. The hardware and software are separately sold and users are free to load a network operating system of their own requirements. The data flows that occur in the data plane are defined by the controller. The advantages of SDN are

- SDN Facilitates Network Innovation
- Layered architecture with Standard Interfaces which are open source
- At each layer there is independent innovation.
- Research and experiment can be conducted using non–expensive equipments.
- Software can be developed easily using open-flow protocol

- Customization is very easy, fast upgrade and can be integrated with other software application.

### 4.2 THE ARCHITECTURE OF SDN

SDN is an emerging technology in the networking field. The main aim of SDN architecture is to program network easily and control network hardware remotely through software [50] The components of SDN framework are

1. OpenFlow switches (Data plane)

2. Controller (Control plane)

3. SDN applications

#### 4.2.1 OpenFlow switches

The data plane consists of forwarding devices such as switches and routers which are interconnected through wired cables and wireless radio channels. These devices have flow rules that helps to take actions on the incoming packets like drop or forward packet, send to a specific port. Figure 4.1 shows the structure of an open flow switch [51-52], The switch has three tables

- Flow table

- Meter table

- Group table
4.1 Flow Table

Flow table matches an incoming packet to a particular flow and specifies the function to be performed. Flow table directs a flow to a group table to trigger different actions. Meter table triggers performance related function on a flow.

4.2.2 Controller

The Controller connects with the OpenFlow switches and the Flow forwarding decisions are made by the controller [53]. The network intelligence is centralized in software defined network controller, which maintains a global view of the network. The different types of open source controller are NOX, POX, floodlight, Open daylight and ONOS. The SDN Controller platform contains a collection of “pluggable” modules that perform different network tasks. Cisco, HPE, Juniper, Lumina, NEC are few examples of SDN proprietary controller.
4.2.3 SDN application

Software defined network application is a service to introduce new functionality or to replace the functionality of a traditional network device. The different services are security, load balancing, traffic engineering, firewall, Mobility and wireless, cloud integration and distributed application control. A simple view of SDN architecture is shown in figure 4.2

The southbound interfaces connect the forwarding devices with the control plane. Open flow is the standard southbound API. The other API’s are ForCES, OVSDB, OpFlex and POF. The northbound interface connects the controller with the application layer.

Figure 4.2 A simple view of SDN Architecture

The heart of the software defined network is the controller. The software application is used to program the network and runs on the top of the network
operating system such as POX, NOX and ONOS that interacts with the data plane. The three-fundamental abstractions of SDN are

(i) forwarding

(ii) specification and

(iii) distribution.

SDN architecture is more flexible and can operate at different protocol layers with different types of switches.

4.3 OPENFLOW PROTOCOL

The standard southbound interface between the data plane and control plane is the OpenFlow protocol [54] of a Software defined architecture. The network devices such as switches and routers in the forwarding plane can be directly accessed and flow tables manipulated by the OpenFlow protocol. It is a protocol and API. A flow table acts as the building block of the logical switch architecture. Three information sources are provided by the open flow protocol to the network operating system, is mentioned below.

(a) When a port or link change is triggered the forwarding devices send event-based messages to the controller.

(b) The forwarding devices generate flow statistics and send to the network operating system (i.e.) controller

(c) When a new flow arrives and do not know what to do, packets are forwarded to the controller for further action.

The OpenFlow protocol is defined by the Open networking foundation and it is layered on top of the transport protocol TCP (Transmission control protocol) and
uses transport layer security. The controllers listen to the standard TCP port number 6653 for switches to setup a connection.

The protocol has the following components

- Message Layer
- State Machine
- Configuration
- Data model
- System Interface

**Message layer:** The message layer defines the semantic and valid structure of all messages. It is the core of the protocol stack.

**State Machine:** It describes various actions such as flow control, negotiation, discover, delivery, capability etc and defines the “core low-level behaviour “of the protocol.

**Configuration:** Configuration covers default buffer size and reply intervals to X.509 certificates.

**Data Model:** A relational data model containing attributes for each and every flow abstraction is maintained by the switch. The attributes describe about the configuration state, abstractions and set of current statistics

**System interface:** It identifies the optional and necessary interfaces along with their use such as TCP and TLS.
4.4 COMPARISON OF SDN CONTROLLERS

Network operating system or controller is the brain of Software defined architecture. The controller contains a collection of “pluggable modules “that can perform various networking tasks. The basic services of the controller are

(i) Communications

(ii) I/O operations control

(iii) Program execution and

(iv) Protection

The various opensource SDN controllers are ONOS, NOX, POX, Open daylight, Floodlight, Beacon and Ryu controller [55]. The controller sends messages to the devices in the data plane through southbound API and communicates with the application layer through northbound interface. Controllers are classified into two types based on the architectural view.

(i) Centralized controller

(ii) Distributed controller.

Centralized controller manages all the forwarding devices such as switch in the data plane, It is a single entity with scaling limitations. It represents a single point of failure. These controllers have multithread design to explore the properties of parallelism. One controller Beacon can handle 12 million flows per second and used by companies like Amazon. e.g. Trema, Ryu and Meridian. Distributed controller is a centralised cluster of nodes. It is fault tolerant and can improve scalability, e.g. ONOS, Hyper flow, Onix. Types of SDN controller
4.4.1 NOX Controller

NOX is the first controller that was designed from the Nicira Networks based on C++ for development of network control application. It supports the open flow protocol. It has components for topology discovery and Learning switch. It supports Linux and Ubuntu operating system.

4.4.2 POX controller

POX is a development platform for SDN. It is an open flow controller based on python language. It communicates with the switches in the data plane through the standard south bound API or OVSDB. It supports Linux, Mac and Windows operating system. POX can be used as a basic controller by using the stock components bundled together. It supports a few command line options.

4.4.3 Open daylight

It is an open source project based on Java programming language supported by the Linux foundation. It supports the standard OpenFlow protocol, a southbound API and it is focussed on data center. The latest version is called “Oxygen” and many southbound API’s such as COAP, PCEP, LACP, SNMP and HTTP has been added. It supports integration with IOT controller.
Table 4.1 Feature Comparison of Open Flow based SDN controllers

<table>
<thead>
<tr>
<th>Controller</th>
<th>Supported Language</th>
<th>Interface</th>
<th>Supported Platforms</th>
<th>Southbound API</th>
<th>Northbound API</th>
<th>Modular</th>
<th>Openstack support</th>
<th>Multithreading</th>
</tr>
</thead>
<tbody>
<tr>
<td>POX</td>
<td>Python</td>
<td>Python</td>
<td>Mac, Linux and Win</td>
<td>OF 1.0</td>
<td>REST</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>NOX</td>
<td>C++</td>
<td>Python</td>
<td>Linux</td>
<td>OF 1.0</td>
<td>REST</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Floodlight</td>
<td>Java</td>
<td>Web UI</td>
<td>Linux</td>
<td>OF1.0, 1.3</td>
<td>REST</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Beacon</td>
<td>Java</td>
<td>Web UI</td>
<td>Linux, Mac, Win</td>
<td>OF 1.0</td>
<td>REST</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Ryu</td>
<td>Python</td>
<td>Web UI</td>
<td>Linux</td>
<td>OpenFlow 1.0,1.2,1.3,1.4 And NETCONF</td>
<td>REST</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MuL</td>
<td>C</td>
<td>Web UI</td>
<td>Linux</td>
<td>OpenFlow 1.0,1.3,1.4, OVSDB, OFCONFIG</td>
<td>REST</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>OpenDaylight</td>
<td>Java</td>
<td>Web UI</td>
<td>Win Linux Mac</td>
<td>OF 1.0,1.3,1.4, NETCONF</td>
<td>REST</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Maestro</td>
<td>Java</td>
<td>Web UI</td>
<td>Win Linux Mac</td>
<td>OF 1.0</td>
<td>REST</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
4.4.4 Flood light

Flood light is an open flow controller which is Java based and works similar to other controllers that supports the standard southbound API i.e. the open flow protocol. The Floodlight controller supports Open stack using a Neutron plugin, a network service model with a REST API and runs as the network backend. Port bandwidth statistics can be measured by the newly incorporated statistic module.

4.4.5 ONOS

It is an SDN controller to manage switches and other forwarding devices in the data plane and provide communication service to end host. It supports distributed system across multiple servers and has GUI to view multi-layer topologies. The applications communicate with the controller through the Northbound API i.e. REST API. ONOS software is written in Java. It supports CLI commands used for debugging.